PIRASE-STRUCTURE PARSING:
A METHOD FOR TAKING ADVANTAGE OF
ALLOPHONIC CONSTRAINTS

by

Kenneth Ward Church

SB Massachusetts Institute of Technology
(1978)

SM Massachusetts Institute of Technology
(1980)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE
DEGREE OF
DOCTOR OF PHILOSOPHY
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
January 13, 1983
© Kenneth W. Church and Massachusetts Institute of Technology

Signature of the Author

Department of Electrical Engineering and Computer Science, January 13, 1983

Certified by

Prof. Jonathan Allen, Thesis Supervisor

Accepted by

Eng.

Chairman, Department of

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY
SEP 1 1983
LIBRARIES
Extended Abstract

It is well-known that phonemes have different acoustic realizations depending on the context. Thus, for example, the phoneme /t/ is typically realized with a heavily aspirated strong burst at the beginning of a syllable as in the word Tom, but without a burst at the end of a syllable in a word like cat. Variation such as this is often considered to be problematic for speech recognition:

(1) "In most systems for sentence recognition, such modifications must be viewed as a kind of ‘noise’ that makes it more difficult to hypothesize lexical candidates given an input phonetic transcription. To see that this must be the case, we note that each phonological rule [in a certain example] results in irreversible ambiguity — the phonological rule does not have a unique inverse that could be used to recover the underlying phonemic representation for a lexical item. For example, ... schwa vowels could be the first vowel in a word like 'about' or the surface realization of almost any English vowel appearing in a sufficiently stressed word. The tongue flap [t] could have come from a /t/ or a /d/.” [65, pp. 548-549]

This view of allophonic variation is representative of much of the speech recognition literature, especially during the late 1970's. One can find similar statements by Cole and Jakimik [22] and by Jelinek [50].

We prefer an alternative view of phonological processes, similar to the one currently being advocated by Nakatani and his co-workers [87, 88, 89, 90, 91]. The speech utterance is modeled in terms of two types of acoustic/phonetic cues: those that vary a great deal with context (e.g., aspiration, flapping) and those that are relatively invariant to context (e.g., place, manner, voicing). Both variant and invariant cues are helpful: variant cues reveal properties of the suprasegmental context and invariant cues reveal properties of the local segmental identity. This much has been observed elsewhere. However, we are not aware of a concrete proposal describing how these cues are to be integrated in a practical recognition system. The recognizer to be discussed here is designed to exploit variant cues by parsing the input utterance into syllables and other suprasegmental constituents using phrase-structure parsing techniques. (This application of parsing is novel.) Invariant constraints are applied in the usual way to match portions of the utterance (constituents in my case) with entries from the lexicon.

---
1. The distinctiveness of invariant cues (e.g., place, manner, voicing) is uncontroversial. The distinctiveness of variant cues (e.g., aspiration, flapping) is somewhat more debatable. We find examples such as a tease/at ease and night rate/nitrate to be completely convincing. The fact that these contrasts can be perceived seems to demonstrate that people can make use of the context depended realizations of /t/.
Only part of the proposed recognizer has been implemented. We have written a program to parse lattices of segments\textsuperscript{2} into lattices of syllables and other phonological constituents. Phonological constraints are expressed in terms of phrase-structure rules. For example, we could restrict aspiration to syllable initial position (a reasonable first approximation) with a set of rules of the form:

(2a) \( \text{utterance} \rightarrow \text{syllable}\)\textsuperscript{*}  
(2b) \( \text{syllable} \rightarrow \text{onset rhyme} \)  
(2c) \( \text{onset} \rightarrow \text{aspirated-t} \mid \text{aspirated-k} \mid \text{aspirated-p} \mid \ldots \)  
(2d) \( \text{rhyme} \rightarrow \text{peak coda} \)  
(2e) \( \text{coda} \rightarrow \text{unreleased-t} \mid \text{unreleased-k} \mid \text{unreleased-p} \mid \ldots \)

This sort of context-free phrase-structure grammar can be processed with well-known parsers like Earley's Algorithm [30]. Thus, if we completed this grammar in a pure context-free formalism, we could employ a straightforward Earley parser to find syllables, onset, rhymes and so forth.

However, it has been our experience that the context-free formalism is not exactly what we want for this task. Context-free rules are generally considered to be awkward for manipulating features (e.g., \textit{place}, \textit{manner}, \textit{voicing} in phonology and \textit{person}, \textit{number}, \textit{gender} in syntax). This observation has led many researchers to introduce augmentations of various sorts (e.g., ATN registers [125], LFG constraint equations [55], GPSG meta-rules [36], REL bit vectors [29, 82]). To serve this purpose, we have implemented a simple language of matrix (lattice) operations. This implementation appears to be easier to work with than many others.

In summary, we hope to make two points. First, we want to suggest the use of parsing techniques at the segmental level in speech applications. Secondly, we want to advocate the matrix view of parsing on engineering grounds. In our experience, the matrix view allows us to define a number of constructions in relatively natural ways. Excerpts of the grammar, lexicon, and output are presented in the appendices.

\footnote{2. A phonetic transcription is a trivial case of a segmentation lattice: a transcription generally doesn't have any branching whereas a segmentation lattice might be quite bushy.}
Acknowledgments

I have had a lot of help with this thesis. To all of those who I have had the pleasure of working with, thank you! I especially want to mention:

Jon Allen, as my thesis advisor

Victor Zue and Peter Szolovits, as my readers

Victor Zue, Janet Pierrehumbert, Meg Withgott, Dave Shipman, Dan Hunttenlocher, Lori Lamel, Francine Chen, Scott Cypher and the Speech Group

Peter Szolovits, Ramesh Patil, Bill Martin, Glenn S. Burke (alias Grandiose/Grumpy System Builder) and the Medg Group/Third Floor

Janet Pierrehumbert, Meg Withgott, Margaret Fleck for reading very very early drafts

Joan Bresnan, Ron Kaplan, Lori Levin, Kris Halvorsen and the LFG Group

Mitch Marcus, Bob Berwick and the AI Lab

Jay Keyser and Cognitive Science Center

MIT-ML (alias Medical Liability) and MIT-SPEECH

Sarah for some happy times

Mom, Dad and the family

and finally Bill Martin, who we all miss very much
CONTENTS

1. Introduction ................................................................................................................. 11
  1.1 Historical Background and Problem Statement ......................................................... 11
      1.1.1 Simplifications to the General Speech Understanding Problem .................... 12
      1.1.2 Major Components of a CSR System .............................................................. 14
  1.2 Allophonic Constraints are Useful ............................................................................. 16
      1.2.1 “Did you hit it to Tom?” .................................................................................. 17
          1.2.1.1 Phonological Rules: A Source of Noise? .................................................. 17
          1.2.1.2 Phonological Rules: A Source of Constraint ............................................ 20
          1.2.1.3 Redundant Constraints ............................................................................. 21
      1.2.2 “Plant some more tulips” .................................................................................. 23
      1.2.3 “We make all of our children” .......................................................................... 24
  1.3 Problems with Rewrite-Rules .................................................................................... 28
  1.4 Trends Toward Larger Constituents .......................................................................... 29
      1.4.1 Trends in Speech Research .............................................................................. 29
          1.4.1.1 Advantage 1: Improved Performance due to Sharing .................................. 30
          1.4.1.2 Advantage 2: Lexicon is Free of Allophones .............................................. 31
      1.4.2 Trends in Linguistic Theory .............................................................................. 33
          1.4.2.1 The ‘Classical’ Position .............................................................................. 33
          1.4.2.2 Metrical Foot Structure ............................................................................ 35
          1.4.2.2.1 Metrical Foot Structure and Stress ......................................................... 38
          1.4.2.2.2 Is Foot Structure Just a Notational Variant? ........................................ 39
  1.5 Parsing and Matching ............................................................................................... 40
      1.5.1 Parsing ............................................................................................................. 43
      1.5.2 Matching ......................................................................................................... 43
          1.5.2.1 Canonicalization ...................................................................................... 44
          1.5.2.2 The Final Result ....................................................................................... 46
  1.6 Summary ................................................................................................................... 47
  1.7 Outline of What’s To Come ......................................................................................... 47

2. Representation of Segments ......................................................................................... 51

   2.1 Stevens’ Theory of Invariant Features .................................................................... 51
   2.2 Our Position .......................................................................................................... 54
   2.3 What’s New ............................................................................................................ 55
       2.3.1 Use of Parsing Constraints ............................................................................. 55
       2.3.2 Decompositional View of Segments ............................................................... 55
   2.4 Motivations for Representing Phonetic Distinctions ............................................ 56
       2.4.1 Capture Phonetic Constraints on Suprasegmental Constituents ..................... 56
       2.4.2 Challenge to the Theory of Invariant Features .............................................. 62
           2.4.2.1 Example I: /t, d/ ...................................................................................... 63
           2.4.2.2 Example II: Nasals .................................................................................. 63
2.4.2.3 Example III: /t₁ k/ ................................................................. 64

2.5 Capturing Generalizations ................................................................. 67
   2.5.1 Allophonic Proliferation ......................................................... 67
   2.5.2 Consistency in Transcriptions .................................................. 59
   2.5.3 Consistency Among Dictionaries ............................................. 71

2.6 Summary ......................................................................................... 73

3. Allophonic Rules ............................................................................. 74
   3.1 Flapping and Syllable Level Generalizations .................................... 74
      3.1.1 Modification 1: Optional Consonant ........................................ 75
      3.1.2 Modification 2: Word Boundaries .......................................... 77
      3.1.3 Modification 3: Stress .......................................................... 78
   3.2 Non-Linear Formulations of Flapping ........................................... 79
      3.2.1 Kahn's Ambi-Syllabicity ....................................................... 79
      3.2.2 Metrical Foot Structure ....................................................... 80
      3.2.3 Differences between Syllable and Foot Structure .................... 81
         3.2.3.1 Word Boundaries ......................................................... 81
         3.2.3.2 Tri-Syllabic Feet .......................................................... 81
      3.2.4 The Optional Consonant (Revisited) ....................................... 82
      3.2.5 Interaction with Morphology ................................................. 84
         3.2.5.1 #-Prefixes ................................................................... 84
         3.2.5.2 #-Suffixes ................................................................... 86
      3.2.6 Summary of Argument for Non-Linear Analysis ....................... 86
   3.3 Implementation Difficulties and the Lexical Expansion Solution ........ 87
      3.3.1 Generative in Nature ........................................................... 88
      3.3.2 Restrictions on Allophonic Grammars .................................... 89

   4.1 PS Trees Bear More Fruit Than You Would Have Thought ................ 92
   4.2 The Constituency Hypothesis ....................................................... 93
      4.2.1 Foot-Internal Rules ............................................................ 93
      4.2.2 Non-Foot-Internal Rules ..................................................... 94
      4.2.3 Foot-Initial Rules .............................................................. 95
      4.2.4 Re-organization of the Grammar .......................................... 96
   4.3 Advantages of Phrase-Structure Formulation ................................ 96
      4.3.1 Robustness ........................................................................ 96
      4.3.2 Efficiency ........................................................................ 100
      4.3.3 The Chicken or Egg Paradox .............................................. 101
   4.4 Summary ................................................................................... 103

5. Parser Implementation .................................................................... 105
   5.1 An Introduction to Chart Parsing ................................................. 106
   5.2 Representation Issues .................................................................. 109
      5.2.1 The Chart Tends to be Sparse ............................................ 110
      5.2.2 Taking Advantage of the Sparseness .................................... 111
   5.3 A Parser Based on Matrix Operations ........................................... 112
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.1 Concatenation and Union</td>
<td>112</td>
</tr>
<tr>
<td>5.3.2 Optionality</td>
<td>113</td>
</tr>
<tr>
<td>5.3.3 Transitive Closure</td>
<td>115</td>
</tr>
<tr>
<td>5.4 No Recursion</td>
<td>116</td>
</tr>
<tr>
<td>5.5 Order of Evaluation</td>
<td>117</td>
</tr>
<tr>
<td>5.6 Feature Manipulation</td>
<td>118</td>
</tr>
<tr>
<td>5.7 Additional Lattice Operations</td>
<td>120</td>
</tr>
<tr>
<td>5.7.1 Over-generate and Filter</td>
<td>120</td>
</tr>
<tr>
<td>5.7.2 Context Primitives</td>
<td>122</td>
</tr>
<tr>
<td>5.7.3 Garbage Collection</td>
<td>123</td>
</tr>
<tr>
<td>5.8 Debugging Capabilities</td>
<td>124</td>
</tr>
<tr>
<td>5.9 Summary</td>
<td>126</td>
</tr>
<tr>
<td>6. Phonotactic Constraints</td>
<td>127</td>
</tr>
<tr>
<td>6.1 The Affix Position</td>
<td>128</td>
</tr>
<tr>
<td>6.2 The Length Restriction</td>
<td>130</td>
</tr>
<tr>
<td>6.3 The Sonority Hierarchy</td>
<td>131</td>
</tr>
<tr>
<td>6.3.1 Exceptions to the Sonority Hierarchy</td>
<td>131</td>
</tr>
<tr>
<td>6.4 Practical Applications of Phonotactic Constraints</td>
<td>132</td>
</tr>
<tr>
<td>6.4.1 The Sonority Bound on Syllable Ambiguity</td>
<td>133</td>
</tr>
<tr>
<td>6.4.2 Removing Redundancy From the Lexicon</td>
<td>135</td>
</tr>
<tr>
<td>6.4.2.1 A Note on Liquids and Glides</td>
<td>137</td>
</tr>
<tr>
<td>6.5 Summary</td>
<td>138</td>
</tr>
<tr>
<td>7. When Phonotactic Constraints are Not Enough</td>
<td>140</td>
</tr>
<tr>
<td>7.1 Basic Principles</td>
<td>140</td>
</tr>
<tr>
<td>7.1.1 Maximize Onsets and Stress Resyllabification</td>
<td>140</td>
</tr>
<tr>
<td>7.1.2 Morphology</td>
<td>141</td>
</tr>
<tr>
<td>7.1.3 Maximize Onsets</td>
<td>141</td>
</tr>
<tr>
<td>7.1.4 Stress Resyllabification</td>
<td>143</td>
</tr>
<tr>
<td>7.2 Against Stress Resyllabification</td>
<td>145</td>
</tr>
<tr>
<td>7.2.1 Alternative 1: No Resyllabification</td>
<td>147</td>
</tr>
<tr>
<td>7.2.2 Alternative 2: Limited Stress Resyllabification</td>
<td>149</td>
</tr>
<tr>
<td>7.2.3 Alternative 3: Vowel Resyllabification</td>
<td>150</td>
</tr>
<tr>
<td>7.2.4 Summary of Alternatives to Stress Resyllabification</td>
<td>151</td>
</tr>
<tr>
<td>7.3 Practical Applications of Vowel Resyllabification</td>
<td>151</td>
</tr>
<tr>
<td>7.3.1 Alternative Points of View</td>
<td>151</td>
</tr>
<tr>
<td>7.3.2 Vowel Class and Allophonic Cues are Often Redundant</td>
<td>153</td>
</tr>
<tr>
<td>7.3.3 Reduction of Branching Factor</td>
<td>154</td>
</tr>
<tr>
<td>7.3.4 Schw Enrich the Speech Signal</td>
<td>155</td>
</tr>
<tr>
<td>7.3.5 Stronger Form of Vowel Resyllabification</td>
<td>157</td>
</tr>
<tr>
<td>7.4 Automatic Syllabification of Lexicons</td>
<td>158</td>
</tr>
<tr>
<td>7.5 Summary</td>
<td>160</td>
</tr>
</tbody>
</table>
8. Robustness Issues ........................................................................................................ 162
  8.1 Alternatives in the Input Lattice ............................................................................ 162
  8.2 Problems for Parsing ............................................................................................ 163
    8.2.1 Allophonic Constraints on an Impoverished Lattice ................................... 164
    8.2.2 Phonotactic Constraints Aren't Helping Either ........................................... 165
  8.3 Relaxing Phonological Distinctions ...................................................................... 166
  8.4 Conservation of Distinctive Features .................................................................... 168
    8.4.1 Deletion in Homorganic Nasal Clusters ....................................................... 169
    8.4.2 Deletion in Fricative Clusters ....................................................................... 169
    8.4.3 Tunafish Sandwich ....................................................................................... 170
  8.5 Probabilistic Methods ............................................................................................ 172
    8.5.1 Problems with Relaxing Distinctions ......................................................... 172
    8.5.2 The Similarity of Probabilistic and Categorical Approaches ....................... 173
  8.6 Distinctive Features .............................................................................................. 174
    8.6.1 Modeling Confusions at the Segmental Level ............................................. 175
    8.6.2 A Comparison of Seneff's Performance with BBN's Front End .................... 176
    8.6.3 Practical Application of Confusion Matrices ............................................. 177
    8.6.4 Modelling Confusions at the Distinctive Feature Level .............................. 177
    8.6.5 Feature Integration ..................................................................................... 179
    8.6.6 Practical Applications ................................................................................ 183
  8.7 Summary ................................................................................................................ 184

9. Conclusion .................................................................................................................. 186
  9.1 Review of the Standard Position .......................................................................... 186
  9.2 Review of Nakatani's Position ............................................................................. 188
  9.3 Review of the Constituency Hypothesis ............................................................... 189
  9.4 Review of Phonotactic Constraints ...................................................................... 191
  9.5 Comparison with Syntactic Notions of Constituency .......................................... 192
  9.6 Contributions ........................................................................................................ 194

References .................................................................................................................... 196

Appendix 1. The Organization of the Lexicon ............................................................... 206
  1.1. Linear Representation and Linear Search ......................................................... 206
  1.2. Non-Recursive Discrimination Networks ......................................................... 207
  1.3. Recursive Discrimination Networks .................................................................. 208
  1.4. Hash Tables Based on Equivalence Class Abstractions ................................... 209
  1.5. Shipman and Zue .............................................................................................. 210
  1.6. Morse Code ...................................................................................................... 210
  1.7. Selecting the Appropriate Gross Classification .............................................. 212
  1.8. Summary ........................................................................................................... 213
Table of Contents

Appendix II. Don’t Depend Upon Syntax and Semantics ......................................................... 214

II.1. Higher Level vs. Lower Level Constraints ................................................................. 214
II.2. Too Much Dependence in the Past ............................................................................. 215
II.3. How Much Can Higher Constraints Help? ................................................................. 216
II.4. Detraction from the Important Low-Level Issues ....................................................... 217
II.5. New Directions: Recognition without Understanding ............................................... 218
II.6. Lower-Level Constraints bear More Fruit ................................................................. 219
II.7. Summary ..................................................................................................................... 220

Appendix III. Lexical Phonology ....................................................................................... 222

III.1. Difference Between + and # ..................................................................................... 222
III.2. Pipeline Design ........................................................................................................ 224
III.3. Distinctions Between Lexical and Postlexical Rules ................................................ 226
III.4. Which Rules are Lexical and Which are Postlexical? .............................................. 228
III.5. The Implementation of Lexical and Postlexical Rules .............................................. 231

Appendix IV. A Sample Grammar .................................................................................... 235

Appendix V. Sample Lexicon ............................................................................................ 245

Appendix VI. Sample Output ............................................................................................. 252

Appendix VII. Glossary ..................................................................................................... 261
# FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 1</td>
<td>Major Components of a CSR System</td>
<td>14</td>
</tr>
<tr>
<td>Fig. 2</td>
<td>“Did you hit it to Tom?”</td>
<td>18</td>
</tr>
<tr>
<td>Fig. 3</td>
<td>“Say 'conservative' again”</td>
<td>36</td>
</tr>
<tr>
<td>Fig. 4</td>
<td>“Say 'conservation' again”</td>
<td>37</td>
</tr>
<tr>
<td>Fig. 5</td>
<td>Parsing and Matching</td>
<td>41</td>
</tr>
<tr>
<td>Fig. 6</td>
<td>Klattese</td>
<td>42</td>
</tr>
<tr>
<td>Fig. 7</td>
<td>Examples of Features for Some Consonant Sequences</td>
<td>51</td>
</tr>
<tr>
<td>Fig. 8</td>
<td>Some Features for /t/ and /ð/</td>
<td>52</td>
</tr>
<tr>
<td>Fig. 9</td>
<td>“night rate” vs. “nitrate”</td>
<td>57</td>
</tr>
<tr>
<td>Fig. 10</td>
<td>“I said ‘grape lane’ not ‘gray plane’”</td>
<td>59</td>
</tr>
<tr>
<td>Fig. 11</td>
<td>“I walked along grape lane yesterday”</td>
<td>60</td>
</tr>
<tr>
<td>Fig. 12</td>
<td>“The gray plane is about to take off”</td>
<td>61</td>
</tr>
<tr>
<td>Fig. 13</td>
<td>“Attribute” Examples</td>
<td>95</td>
</tr>
<tr>
<td>Fig. 14</td>
<td>A Taxonomy Organized by Syllable Structure</td>
<td>97</td>
</tr>
<tr>
<td>Fig. 15</td>
<td>A Taxonomy Organized by Phonological Class</td>
<td>98</td>
</tr>
<tr>
<td>Fig. 16</td>
<td>“récord” vs. “recórd”</td>
<td>144</td>
</tr>
<tr>
<td>Fig. 17</td>
<td>“áttribute” vs. “áttribute”</td>
<td>152</td>
</tr>
<tr>
<td>Fig. 18</td>
<td>A Hard Segmentation Lattice</td>
<td>164</td>
</tr>
<tr>
<td>Fig. 19</td>
<td>Two Confusion Matrices for /p t k/</td>
<td>177</td>
</tr>
<tr>
<td>Fig. 20</td>
<td>Summary of Lexical/Postlexical Distinctions</td>
<td>232</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Historical Background and Problem Statement

The main goal of speech recognition/understanding research is to develop techniques and systems for speech input to machines. Ideally, we would like to construct an intelligent machine that could listen to a speech utterance and respond appropriately. This task is called speech understanding. Speech recognition is a different problem. A recognizer is an automatic dictation machine; it transforms an acoustic speech signal (utterance) into a text file (e.g., a sequence of ascii characters).

In earlier attempts, it was hoped that learning how to build simple recognition systems would lead in a natural way to more sophisticated systems. Systems were built in the 1950's for vowel recognition and digit recognition, producing creditable performance. But these techniques and results could not be extended and extrapolated toward larger and more sophisticated systems. This has led to the appreciation that linguistic and non-linguistic contextual cues must be brought to bear on the recognition strategy if we are to achieve significant progress. In understanding an utterance, a native speaker (subconsciously) invokes his knowledge of the language, the environment, and the context. These sources of knowledge (KS's) include parameterization of speech signals (signal processing and electrical engineering), the physiology of the vocal tract and the ear (articulatory phonetics), descriptive accounts of the acoustic characteristics of speech sounds (phonetics), positional (tactic) constraints of phonemes within syllables (phonotactics), theoretical explanations of the variability in pronunciations (phonology), the stress and intonation patterns of words and phrases (prosodies), lexical constraints, the composition of words (morphology), the grammatical structure of language (syntax), the meaning of words and sentences (semantics), and the context of conversation (pragmatics). An ideal speech understanding system would make use of all of these knowledge sources (or constraints) in order to decode an utterance.

---

3. Much of this discussion is borrowed from Reddy's review of the ARPA speech project [97].
4. Ascii is the standard representation of letters (graphemes) in many (but not all) computer systems.
5. Knowledge Source (KS) is a term from Artificial Intelligence (especially common at CMU and Stanford). It corresponds to the linguistic term: level of representation.
1.1.1 Simplifications to the General Speech Understanding Problem

In practice, it is necessary to make some compromises. In most (if not all) existing systems, the implementations of (many of) these knowledge sources are incomplete, or non-existent. For instance, syntax and semantics are artificially constrained for a particular application in most understanding systems (e.g., the four ARPA Speech Understanding projects: CMU Harpy, CMU Hearsay, BBN HWIM, SDC).\textsuperscript{6}

<table>
<thead>
<tr>
<th>Group</th>
<th>Task</th>
<th>Sample Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDC</td>
<td>Facts about Ships</td>
<td>How fast is the Theodore Roosevelt?</td>
</tr>
<tr>
<td>BBN WHIM</td>
<td>Travel Budget Management</td>
<td>What is the plane fare to Ottawa?</td>
</tr>
<tr>
<td>CMU Harpy</td>
<td>Document Retrieval</td>
<td>How many articles on psychology are there?</td>
</tr>
<tr>
<td>Hearsay-II</td>
<td>Document Retrieval</td>
<td>How many articles on psychology are there?</td>
</tr>
</tbody>
</table>

Some researchers (most notably those at IBM) object to these sorts of domain restrictions as \textit{ad hoc}. At IBM, they prefer to approximate higher level constraints (e.g., syntax, semantics and pragmatics) with statistical methods. Despite these differences in engineering orientation, there is general agreement that approximations must be made to the ideal goals because of practical limitations in our current understanding in (almost) every area of speech understanding. In this thesis, we will push higher level constraints aside, focusing more on speech recognition than on speech understanding. See appendix II for some discussion of this decision.

At the lexical retrieval level, it is common practice to restrict the size of the lexicon. Most of the ARPA systems and IBM limit the lexicon to approximately 1000 words. Our implementation is typically run with about 3000 words. Smith [113, 114] reports on a 20,000-word system. It should be noted, however, that the size of the lexicon can be deceptive. Certain tasks with large vocabularies can be easier than other tasks with smaller vocabularies.\textsuperscript{7}

\textsuperscript{6} This figure appears in Klatt's Review of the ARPA Speech Projects [64, p. 1347].

\textsuperscript{7} For example, there has been a considerable amount of effort devoted toward the construction of a machine that could recognize \textit{spoken} alpha characters (i.e., letters of the alphabet) and/or digits. It turns out that the 26 alpha characters are far more confusable than the 10 digits. Alpha characters tend to have fewer segments (phonemes), so there isn't very much speech to base the discrimination on. In addition, the vowel segment has lower information since almost half of the alpha characters share the same vowel: /i/ (e.g., bcdegptz). Thus, most of the discrimination will have to be made on the basis of the first segment; there is little room for error at the front end. For these reasons, the alpha-digit task (vocabulary size = 36) seems to be easier (in terms of recognition accuracy) than the alpha task (vocabulary size = 26), even though the alpha-digit task has a larger vocabulary.
At the phonetic level, one can limit the scope of the problem by insisting that the speaker delimit words with short pauses. These isolated word systems avoid two difficulties of continuous/connected speech. First, in continuous speech (as the name implies), it is difficult to determine where one word ends and another begins because there are no (easily perceived) delimiters between words.\(^8\) Secondly, the characteristic acoustic patterns of words exhibit much greater variability depending on the phonetic/acoustic context.\(^9\) Isolated word systems will not be discussed in this thesis.

Additional simplifications are possible at the signal processing level. Speaker dependence avoids some of the difficult labeling problems. Every speaker has slightly different acoustic characteristics (because of his/her unique vocal tract). Speaker independent systems have to filter out these irrelevant non-distinctive details, and yet preserve the very subtle distinctive characteristics.

Examples of practically all of these simplifying assumptions can be found in the literature, as well as many others. Some systems insist on a cooperative speaker, low background noise, high bandwidth (as opposed to a telephone), and so on. In this thesis, we will strive toward continuous, speaker-independent, large-vocabulary recognition from high bandwidth clean signals produced by a cooperative speaker.

We have only obtained a small step toward this very ambitious goal.

(3) "Why is Connected Speech Recognition Difficult? When isolated word recognition systems are getting over 99-percent accuracies, why is it that CSR [Connected Speech Recognition] systems are straining to get similar accuracy? The answers are not difficult to find. In connected speech it is difficult to determine where one word ends and another begins. In addition, acoustic characteristics of sounds and words exhibit much greater variability in connected speech, depending on the context, compared with words spoken in isolation.

Any attempt to extend the design philosophy of isolated word recognition systems and recognize the utterance as a whole becomes an exercise in futility. Note that even a 10-word vocabulary of digits requires the storage of 10-million reference patterns if one wanted to recognize all the possible 7-digit [telephone numbers]. Some way must be found for the recognition of the whole by analysis of the parts. The technique needed becomes one of analysis and description rather than classification (moving away from pattern recognition paradigms toward hierarchical systems, i.e., systems in which component subparts are recognized and grouped together to form larger and larger units)...

---

8. There is a common misconception among naïve speakers of the language that words are delimited by pauses.
9. This second problem is generally known as The Invariance Problem.
Finally, error and uncertainty in segmentation [the decomposition of the utterance (along the time axis) into segments that correspond roughly to phonemes], labeling [the assignment of phonemic-like labels to segments], and matching [the comparison of portions of the utterance with entries from the lexicon] make it necessary that several word matches be considered as alternative paths. If there were 5 words in an utterance and we considered 5 alternative paths after each word, we would have $3125 (5^5)$ word sequences, out of which we we have to pick the one that is most plausible. Selection of the best word sequence requires a tree search algorithm and a carefully constructed similarity measure.” [97, p. 509]

1.1.2 Major Components of a CSR System

Most of the ARPA speech systems were organized in a pipeline fashion. We envision a slightly longer pipeline than was actually implemented in those systems. First the speech waveform would be digitized from an analog signal into a digital one. Then it would be filtered and transformed into a spectrogram-like representation, a two dimensional plot displaying acoustic energy (or amplitude) as a function of time (horizontal axis) and frequency (vertical axis). From this representation we can extract parameters such as

---

**Fig. 1.** Major Components of a CSR System

```
Digitization
  ↓
Signal Processing
  ↓
Parameter Extraction
  ↓
Feature Extraction
  ↓
Segmentation and Phonetic Labeling
  ↓
Suprasegmental Parsing
  ↓
Lexical Retrieval
  ↓
Higher-Level Constraints
```

*Focus of this Thesis*

---

10. In our laboratory, we sample speech at 16 kHz. Each sample has 16 bits of precision.
zero-crossings, energy in various frequency bands, formant tracks, derivatives (differences) of the above, and so forth. In our laboratory, digitization, signal processing and parameter extraction are performed in SPIRF, written by David Shipman.

From these parameters, the system extracts distinctive features (e.g., place of articulation, voicing, manner) and phonetic diacritics (e.g., flapped, aspirated, glottalized, retroflex, rounded, devoiced). With the aid of these features, the utterance is transformed into a labeled segmentation lattice. In our laboratory, feature extraction, segmentation and labeling are currently being investigated by

11. Strong fricatives such as /s, z/ are correlated with a very high rate of zero-crossings. Weaker fricatives such as /f, v/ have a somewhat lower rate zero-crossings. Pauses and the silence portion of a voiceless stop (e.g., /t/) have even fewer zero-crossings. Voiced regions such as vowels also tend to have relatively few zero-crossings.

12. Voiced regions (e.g., vowels) tend to have relatively more energy at low frequencies whereas voiceless segments tend to have relatively more energy at high frequencies.

13. Formants are natural resonances (poles) of the vocal tract. On a wideband spectrogram, they generally look like dense black lines running horizontally along the page. They are black because frequency poles have high energy.) Formants are important for vowel recognition because they reveal the configuration of the articulators. A high front vowel such as /i/ in the word See is associated with a low first formant F₁ and a high second formant F₂. A low back vowel such as /a/ in Bgb is associated with a high first formant and a low second formant.

14. Some sounds (e.g., the release of a /t/) are associated with sharp onsets in energy (especially at high frequencies); other sounds (e.g., glides) tend to have more gradual onsets.

15. We assume a distinctive feature framework such as the one established in Sound Patterns of English SPE [15]. Phonemes such as /t/ are considered to be notational abbreviations for bundles of features such as [+ coronal], [− voice], [− stop]) in the case of /t/.

16. The place of articulation (sometimes abbreviated simply as place) describes the location of the constriction in the articulatory tract. Some common examples of place are: velar (at the velum), labial (at the lips) and coronal (at the tip of the tongue) dental (at the teeth). An example of a velar phoneme is /k/ and an example of a labial phoneme is /p/, and an example of a coronal/dental phoneme is /t/.

17. There are two sources of acoustic energy in the vocal tract: voicing and frication. /b, d, g/ are examples of voiced stops and /p, t, k/ are voiceless. Voicing is a periodic source located at the larynx (informally known as "the Adam’s Apple"). The reader can feel this vibration by holding his hand against his larynx as he utters a voiced sound. Frication is an aperiodic source created by turbulent air at a constriction. (See appendix VII).

18. There are several different values of manner: stop (e.g., /p, t, k, b, d, g/), fricatives (e.g., /s, z/), vowels, nasals (e.g., /n, m/), liquids and glides (e.g., /l, r, y, w/).

19. A flap (also called tap) is formed when the tongue quickly brushed against the roof of the mouth in a billiard fashion forming a very short constriction (10-40 ms in duration). Flapping is very common in American English (as opposed to British Received Pronunciation (RP)), especially in words like butter.

20. Aspiration is a term in phonetics for the audible breath which may accompany a sound’s articulation, as when plosive consonants are released. It is usually symbolized by a small raised [ʰ] following the main symbol. In examples such as English pin [pʰɪn], the aspiration may be felt by holding the back of the hand close to the mouth while saying the word; the contrast with bin, where there is no aspiration is noticeable. Some languages, such as Hindi, have contrast of aspiration applying to both voiceless and voiced stops. In English, aspiration is generally controlled by the phonetic environment: voiceless stops such as /p/ are aspirated in initial position (e.g., pin), but not in final position (e.g., stop, spin). Aspiration devoices a following /l, r, w, y/ in please, twice, queue, but only when the aspiration and the semi-vowel are found in the same constituent. (We will refine the characterization of these phenomena shortly.)

21. Glottalization is often found in utterances beginning with a vowel. (Intuitively, the vocal cords are a bit like automobile engines. When you first start them up on a cold day, they jerk forward at irregular intervals until they warm up.) Glottalization can also be found in certain stops such as the /t/ in atlas [aqˈtə s] and in Cockney in bottle (bɒtˈl]).

22. Retroflexion is formed by curling the tip of the tongue backward, a trademark of /l/ sounds.

23. Rounding is formed by "rounding" the lips as in the word oh.

24. Some voiced sonorants (e.g., /l/) are devoiced in the environment of a voiceless obstruent (e.g., /p/). For example, the /l/ is place is often said to be (partially) devoiced.
Huttenlocher, Chen and Seneff.

From the segmentation lattice, the utterance is parsed into syllable-like chunks with the aid of phonotactic and allophonic constraints. These syllable sized chunks are then input to the lexical retrieval process and then onto higher level constraints (e.g., syntax, semantics, pragmatics). In our laboratory, there is currently no one working on higher level constraints. Appendix II will explain some of our motivations for postponing efforts in these areas until the other areas have had a chance to develop further.

Our contribution to the over-all design of a speech recognition device is the suprasegmental parsing step. In most previous systems, there is no level of processing between segmentation and lexical retrieval. We hope to demonstrate that the intermediate parsing step will be helpful for exploiting allophonic and phonotactic constraints on syllable structure and word stress domains.

1.2 Allophonic Constraints are Useful

This thesis will argue that allophonic constraints provide an important source of information which should be exploited to the fullest possible extent by a speech recognition device. Despite the fact that allophonic variation can occasionally obscure certain cues, allophonic variation should not be viewed as a source of random noise. Allophonic variation is the result of very predictable processes. These processes provide important cues for the determination of syllable boundaries and stress assignment. This information will (often) compensate for whatever segmental cues may be occasionally obscured.

Lloyd Nakatani and his co-workers have been advocating a similar position for many years [87, 88, 89, 90, 91]. They argue that allophonic and prosodic cues should be applied to constrain the possible locations of word boundaries.

(4) "We see the fission and fusion processes fitting into a comprehensive model of speech perception in the following way. The prosodic stress and rhythm cues indicate when a word boundary is likely to occur, but they do not indicate precisely where the word boundary is. For example, a word boundary must occur somewhere between the syllabic nuclei of 'gray twine' since two consecutive stressed syllables cannot belong to a single English word. The precise location of the word boundary is indicated by the allophonic cues. In this example, the word boundary must

---

25. Phonotactics is the study of positional constraints (tactics) on phonemes.
26. Allophones are variants of phonemes. The variation is dependent on syllable and stress context. Thus, it is possible to exploit the distribution of allophones in order to parse the possible syllable and stress environment.
fall before the /t/ because its strong aspiration is characteristic of word-initial allophones of voiceless stops. [In contrast, in ‘great wine’, the word boundary must fall after the /t/ because its realization is characteristic of word-final allophones.] One should not assume that a word boundary must be located roughly before it can be located precisely. Probably both prosodic and allophonic cues are simultaneously and continuously monitored, and a word boundary can be detected from allophonic cues alone without prior prosodic indications that a word boundary was likely.” [91, p. 3]

Nakatani’s observations were motivated by studies in human perception. This thesis will explore the feasibility of Nakatani’s approach from a computational point of view.

1.2.1 “Did you hit it to Tom?”

Most computational attempts in the past have viewed allophonic variation as a source of “noise”, not as a source of information. Let us consider three arguments for the standard position, and show how the arguments can be reformulated from our point of view.

1.2.1.1 Phonological Rules: A Source of Noise?

First, let us turn to an example which appeared in Klatt’s review paper of the ARPA Speech projects [64, p. 1346] and also in Klatt’s description of LAFS, a new model of lexical access [65, pp. 548-549]. Klatt presents the phonetic transcription:

(5) [dɪˈhɪlɪˈtəm]

for an utterance of the sentence

(6) Did you hit it to Tom?

This example is intended to show that lexical retrieval is difficult because, among other things, “there are extensive rule-governed changes to the way that words are pronounced in different sentence contexts.” Indeed, this transcription is the result of five allophonic rules.\(^{27}\)

---

\(^{27}\) This transcription is written in IPA (International Phonetic Alphabet). Unfortunately, these symbols are difficult to produce correctly on our printer. We have had to make do with a crude approximation. The \(\ddagger\)wa symbol is particularly badly rendered as \(\ddagger\).
"did you hit it to Tom?"
(7a) Palatalization of /d/ before /y/ in *did you*
(7b) Reduction of unstressed /u/ to schwa in *you*
(7c) Flapping of intervocalic /t/ in *hit it*
(7d) Reduction of schwa and devoicing of /u/ in *tq*
(7e) Reduction of geminate /t/ in *it to*

These rules are employed by the (lazy) speaker, Klatt concludes, “to modify and simplify the pronunciation of individual words in some sentence environments.” These simplifications lose valuable distinctions, so the standard position argues, since it is no longer possible to distinguish a palatalized /d/ from a /f/, an unstressed /u/ from a schwa, and so on. Hence, according to the standard position, the recognition device will have to be especially clever in order to undo the effects of these allophonic processes.

(8) “In most systems for sentence recognition, such modifications must be viewed as a kind of ‘noise’ that makes it more difficult to hypothesize lexical candidates given an input phonetic transcription. To see that this must be the case, we note that each phonological rule [in the example above] results in irreversible ambiguity — the phonological rule does not have a unique inverse that could be used to recover the underlying phonemic representation for a lexical item. For example, the [I] observed in the sample phonetic transcription [above] could be the first or last segment of a word like ‘judge’, or it could be the surface manifestation of an underlying /d/ — # — /y/. The schwa vowels [in the example above] could be the first vowel in a word like ‘about’ or the surface realization of almost any English vowel appearing in a sufficiently stressed word. The tongue flap [t] could have come from a /t/ or a /d/. The [t] of ‘it to’ could be the single /t/ of ‘is’ or the surface form of two identical segments separated by a word boundary. The voiceless vowel in ‘to’ could represent the release state of a word-final /t/ or the devoicing of a short vowel between two voiceless consonants. Deletions of segments (e.g. the /u/ in ‘list some’) complicate the picture even more, for now one must entertain the possibility that any moderately long-duration [s] seen in the input is really an underlying /st/ — # — /s/.”

[65, pp. 548-549]

The recognition device must be able to cope with the fact that allophonic processes may insert, delete, and modify certain segments under certain contextual conditions. Klatt suggests the application of syntactic and semantic constraints in order to make up for the information (supposedly) lost through the allophonic

---

28. In fairness, it would be pointed out that Klatt (personal communication) currently agrees that allophonic processes should be viewed as useful constraints.
rules.

(9) “All of these phonological phenomena result in lexical ambiguity so that even the best lexical hypothesis routines will propose many words that are not in the original sentence, simply due to fortuitous matches. The third step in the process [speech understanding] would therefore be to use syntactic - semantic modules to weed out the false lexical hypotheses and put together a word string that represents what was spoken.” [64, p. 1346]

It was generally felt, during the ARPA Speech Project, that syntax and semantics were required in order to compensate for the inadequacies of phonetic level processing, which was extremely poor at that time. Arguments like this were more or less accepted as proof that the speech signal, without syntax and semantics, was too impoverished to permit direct recognition. However, in more recent years, more and more researchers are beginning to question the usefulness and necessity of syntax and semantic constraints in speech recognition. This thesis will explicitly avoid the application of higher level constraints in an attempt to explore the capabilities of constraints at the phonetic, segmental, and suprasegmental levels.

1.2.1.2 Phonological Rules: A Source of Constraint

Returning to Klatt's example (repeated as (10a)), let us show how allophonic constraints can be of use. Suppose that we enhance the transcription to include glottalization and aspiration cues:

(10a) [dɪˈθɛlɪˈtʌm]  
(10b) [dɪˈθɛlɪˈtʰʌm]  
argins

Then it will be much easier to hypothesize words. First, we can introduce the following syllable boundaries:

(11) [dɪˈθɛ # hɪˈ # ɪˈ # thˈɪ # thˈam]

because of the following phonetic constraints on syllable structure:29

29. Admittedly there are many problems with these constraints. They should be treated as first approximations, not hard and fast rules.
(12a) /h/ is always syllable initial,
(12b) [f] is always syllable final,
(12c) [tʰ] is always syllable final, and
(12d) [tʰ] is always syllable initial.30

We are assuming that phonological processes depend on syllable syllable structure, not on word boundaries. Perhaps many of the so-called "word boundary phenomena" are really "syllable boundary phenomena". (See Appendix III for some theoretical motivation for this position.) This assumption will turn out to be an important observation because we can find many of the syllable boundaries in a bottom-up way from allophonic and phonetic cues without consulting the lexicon.

At this point, with the transcription parsed into manageable syllable sized chunks, it is relatively easy to decode the utterance.

<table>
<thead>
<tr>
<th>parsed transcription</th>
<th>decoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>dLF∅</td>
<td>→ did you</td>
</tr>
<tr>
<td>hLF</td>
<td>→ hit</td>
</tr>
<tr>
<td>iʔ</td>
<td>→ it</td>
</tr>
<tr>
<td>tʰt</td>
<td>→ to</td>
</tr>
<tr>
<td>tʰam</td>
<td>→ Tom</td>
</tr>
</tbody>
</table>

Hence if we enhance the phonetic representation and we exploit the allophonic constraints on syllable structure, we are in a vastly superior position to hypothesize word candidates.

1.2.1.3 Redundant Constraints

In this example, we mostly took advantage of the highly constrained distribution of /t/ allophones. In addition, though, there are a large number of other cues that we could have employed instead in order to parse the utterance into syllables. Suppose, for example, that we failed to notice the glottalized /t/ in the word it. We could still deduce the fact that there must be a second consonant between the /l/ and the aspirated /t/ because there is some redundant information encoded in the vowel.

30. Aspiration can also be found in word final position, but that kind of aspiration is of a very different sort.
Redundant Constraints

(13) ḏəəɬɪtʰəəhəm

↑

Note, in particular, that /I/ is a lax vowel. This is a useful cue because the distribution of lax vowels is very limited. We hypothesize (as suggested in Barnwell’s thesis [8]) that lax vowels (except schwa) do not appear in open syllables.\(^{31}\) If this is correct, then we know that the /I/ vowel in it is not syllable final. This observation conflicts with the fact that the following consonant must be syllable initial because it is an aspirated /t/, an allophone that, by hypothesis, is found only in syllable initial position. Thus, we have a contradiction. On the one hand, the lax /I/ vowel cannot end the syllable; but on the other hand, it must end the syllable, because the following consonant starts its own syllable. Thus, we know that this input transcription is internally inconsistent.

The system could resolve this conflict by guessing what the correct transcription might have been. Gemination\(^{32}\) is sufficiently likely that the system would perform reasonably well by guessing that the transcription ought to contain a second /t/. However, it might not be advisable to attempt to correct the input transcription in this way since there are quite a few errors that the front end acoustic processor might have made. For example, there might have been an unreleased /k/, instead of an unreleased /t/. Actually, there might have been practically any kind of unreleased stop or weak fricative. Without looking closer at the acoustic signal, it really isn’t possible to tell what sort of error the front end processor might have made. Perhaps, it would be reasonable to go back to the front end and find out what went wrong. (In this case, the front end acoustic processor (i.e., Klatt) overlooked (on purpose) the unreleased /t/.)

Thus we can use multiple redundant cues (e.g., Barnwell’s vowel constraint and /t/ aspiration) in order to check the transcription for internal consistency. In this way, we can often take advantage of redundant phonetic and phonemic cues in order to locate and possibly correct errors in the front end acoustic processor. This thesis will expand this basic approach of exploiting low level phonetic and phonemic cues in order to parse the utterance into smaller constituents such as clusters, syllables, metrical feet, morphemes, and so forth.

---

\(^{31}\) Roughly speaking, a closed syllable ends with a consonant; an open syllable ends with a vowel. It is not clear, though, how to classify certain syllables ending with glides and other heavily vocalized consonants (e.g., /I, r, n/).

\(^{32}\) Gemination is a term from phonetics for the combination of two short adjacent consonants into one long one. For example, the /ʃʃ/ in gas shortage can be geminated (in some analyses) into a single long /ʃʃ/.
Let us examine two more examples like “Did you hit it to Tom?”

1.2.2 “Plant some more tulips”

Consider another example taken from [22] which is also intended to illustrate the necessity to introduce syntax and semantics, though again we see that this conclusion is not necessarily warranted.

“Consider the sentence ‘Tell the gardener to plant some more tulips’ spoken with the following phonetic structure:

/teɪlɜːdɡɑːrdnərtɔːplænsʌmɔːrtʊlɪps/

The spoken version of the sentence contains several alternative word-level segmentations. ‘Plant some’ can be perceived as ‘plan some’ (due to stop deletion of the word final /t/ in ‘plant’), ‘some more’ may be perceived as ‘some ore and ‘tulips’ may be parsed as ‘two lips.’ As we will see, the intended segmentation is provided by the listener’s use of context, and not by acoustic information.” [22, p. 134]

Again their conclusion may be based upon an inadequate representation at the allophonic level. There are acoustic differences in each of the minimal pairs mentioned that can probably be exploited in a bottom-up way:

(14a) plant some / plan some
(14b) some more / some ore
(14c) tulips / two lips

(duration of nasal)

(possible glottalization; duration of nasal)

(color of /l/; duration of the second vowel)

The nasals are different in plant some and plan some. It is well-known that mono-morphemic words like plant ending in /nt/ have a very short (or non-existent) nasal murmur, that mono-morphemic words like plan ending in /n/ have a slightly longer murmur, and that mono-morphemic words like planned ending in /nd/ have a very long murmur.33

(15) murmur of /nt/ < murmur of /n/ < murmur of /nd/

33. More generally, the duration of a nasal murmur shortened when followed by a voiceless obstructant (as in plant, fence, sink, bump) and lengthened when followed by a voiced obstructant (as in planned, send, men's, longer). The effect is probably limited by syllable and/or stress domains.
These differences are very large; Zue and Sia [131], for instance, found a 55 ms. difference in the duration of the nasal murmur in /nt/ and in /nd/. The magnitude of this difference gives us reason to suspect plant some can be distinguished from plan some on phonetic grounds alone.

The other two proposed confusions, some more/some ore and two lips/tulips, are somewhat less well studied. We suspect that the first pair would have different length /m/s and different degrees of nasality in the following vowel. We also expect to find differences in two lips and tulips. First, we would hope to find a light /l/ in lips and a dark /l/ in tulips. Secondly, there ought to be a longer /l/ in the two lips case than tulips case. We would also hope to notice differences in both vowels and in F0 as a consequence of the different stress patterns. All of these predictions need to be tested.

If we are correct, and phonetic cues are sufficient to distinguish these pairs, then we need not accept the standard position that recognition requires the application of higher-level knowledge. Rather we suggest that recognition proceeds by dividing the utterance into syllable size units through the aid of phonotactic and allophonic constraints. The utterance can be syllabified as follows

(16) \[ \text{[te\# # \varepsilon \# gard # n\varepsilon # t\varepsilon \# p\varepsilon \# \varepsilon \# A\varepsilon m\varepsilon \# r\# t\varepsilon u\# # lps]} \]

because

(17a) A dark /l/ (i.e., [\l]) is syllable final in English,

(17b) a released stop (e.g., [g], [t\varepsilon], [p\varepsilon]) is syllable initial, and

(17c) the phoneme sequence /dn/ must contain a syllable boundary because of English phonotactics.

As mentioned above, we could probably syllabify the middle region plant some more based upon a deeper phonetic understanding. Nevertheless, although it would be desirable to constrain further the syllabification possibilities, it would probably be sufficient simply to look up what we have in a dictionary of English syllables and words. This will result in a slight amount of ambiguity, but one that we can probably live with.

1.2.3 “We make all of our children”

Let us consider one more example that has been given to demonstrate the difficulties of decoding linguistic transcriptions into English words. Jelinek [lecture notes, 1982] gave the transcription

(18) \[ \text{[wime\varepsilon k\varepsilon l\varepsilon v\varepsilon d\varepsilon l\varepsilon drIn]} \]
and asked the class to decode the message. After no one came up with an answer, he suggested not one, but two equally correct solutions:

(19a) We make all of our children.
(19b) We may call of our children.

From the fact that this transcription is ambiguous, Jelinek concluded that it would be difficult to build a linguistic decoder that did not take into account some higher level constraints. He then proposed his probabilistic model of word tri-grams.

We prefer to draw a different conclusion, as it should be clear by now. Suppose, for the sake of discussion, the /k/ in (19a) is a syllable final unreleased /k/ whereas the /k/ in (19b) is a syllable initial aspirated /k/.

(20a) We ma[k⁷] all of our children.
(20b) We may [kʰ] all of our children.

With this enriched representation of the transcription, we can find the syllable/word boundary (in this case), without higher level knowledge. (Other low level cues such as F₀ might also be useful.)

Even with an inadequate phonetic representation, it is relatively easy to find many of the syllable boundaries. Suppose, as a first approximation, we inserted all possible syllable boundaries.\(^{36}\)

---

34. It is possible for the /k/ to be aspirated in both cases since a word final voiceless stop can be aspirated if it is followed by a vowel or a pause. However, the unreleased /k/ will only be found in the syllable final position, and not in the syllable initial case. Thus, we can rule out the syllable initial case if we know that the /k/ is unreleased, but if the /k/ is aspirated, neither case can be ruled out.

35. F₀ is an abbreviation for fundamental frequency (i.e., the direct current DC component of the signal). Fundamental frequency is an acoustic correlate of prosody (rate).

36. The following notation will be used to represent syllabification lattices. A proposed syllable is represented by a string of dots delimited by a vertical bar on each side. So for example we have the syllable /wi/ denoted by a string of dots spanning from the beginning of the /w/ to the end of the /i/.
It is possible to look up all of these possible syllables in a dictionary of syllables since the number of possible syllables is quite limited. The list of possible syllables, though, can be shortened considerably, if we include just a little more phonetic information. Notice that four arcs can be removed by replacing the phoneme /l/ with the allophone [l] and /k/ with [k']. "x"es are used to mark the four arcs which are deleted as a result of enhancing the representation in this way.

And, now, we can remove seven more arcs on the grounds that they are not part of a complete path through the lattice.

At this point, we are left with seventeen arcs, seven of which are part of the correct decoding (marked with "-"s) and eleven of which are spurious.
Even with eleven spurious arcs, this lattice is sparse enough for our purposes. If we simply looked every syllable up in a lexicon of syllables, we would correctly decode the transcription. Thus, for example, we could correctly reject the spurious path

\[(25a) \ [wim \ # \ e^y k^T] \]  

in favor of the correct path

\[(25b) \ [wi \ # \ me^y k^T] \]  

because /wim/ is not in our lexicon of syllables. This is, of course, a risky and time consuming method of filtering out spurious paths. It is risky, because there is always a chance of a random undesired match, and time consuming, because dictionary lookups are expensive (generally inducing a page fault). It would be safer and more efficient to rule out the undesired arcs on phonological and/or phonetic grounds, thus avoiding the need to consult the dictionary.

Since our coverage of phonological and phonetic constraints is currently incomplete, and will certainly remain so for some time to come, the program will have to depend on gaps in the lexicon in order to rule out certain spurious paths. We wish to downplay this dependence on accidental gaps in the lexicon, because it is both risky and expensive. In this way, our approach differs radically from previous lexical retrieval programs, which have no runtime models of phonological and/or phonetic constraints. Instead, they depend solely on gaps in the lexicon to rule out ill-formed candidates.

In summary, we have seen that allophonic constraints are extremely useful. If we take full advantage of these constraints utterances like “Tell the gardener to plant some more tulips” and “Did you hit it to Tom” and “We make all of our children” become much easier to recognize, without resorting to syntactic and semantic constraints. This observation provides extremely strong motivation for a recognizer that can exploit detailed allophonic cues by parsing the utterance into syllable-like constituents.
1.3 Problems with Rewrite-Rules

Whether or not one adopts the standard position that allophonic processes obscure the speech signal, or our own position that these processes enhance the signal, one is nonetheless faced with the fact that these processes induce regular alternations. A model is required to account for these alternations in a linguistically attractive and computationally efficient manner. Over the years, linguists (e.g., [15]) and speech researchers (e.g., [6, 21, 92]) have developed a system of allophonic rules which are designed to capture the relevant generalizations. A typical rule might look something like:

$$t \rightarrow f / v - v$$

which says that an intervocalic /t/ can be flapped. Of course this rule will need to be enhanced considerably if it is to obtain descriptive adequacy. It will, for example, be necessary to account for the fact that /d/ and /n/ can also flap, that certain consonants such as /r/ can appear before a flap, that stress can affect flapping, and so on. We will discuss these issues and more in chapter 3.

However, somewhat more importantly, we wish to question the rewrite rule formalism on two points. First, we object to the transformational nature of rewrite rules. Transformations work by modifying (side effecting) the string of phrase-markers (e.g., segments), in violation of the basic principles of structured programming (e.g., Dijkstra [27]). It is extremely difficult to undo these sorts of transformational rules, because it is very difficult to hypothesize what the string of phrase-markers might have been before it was transformed. In natural language applications, most researchers have been convinced that "inverse transformational parsing" is unlikely to succeed by experience obtained over the last twenty years.\(^{37}\) We believe the transformational view is also unlikely to succeed in speech applications for similar reasons. Most speech researchers are well aware of the difficulties of inverse transformational parsing. Instead of confronting this problem, they choose to avoid the problem by resorting to methods such as lexical expansion, where every rule is literally applied to every lexical item at compilation time. Lexical expansion is not very attractive, and may not scale up as we learn more and more about phonetics and phonology. However, it does allow the researcher to avoid inverse transformational parsing, while maintaining the transformational rewrite rule formalism. This argument is elaborated in chapter 3.

---

\(^{37}\) This remains a controversial point. There are still a few advocates of inverse transformational parsing, most notably Stan Petrick at IBM, though they are very much in the minority.
We suggest a different solution. If the grammar were reformulated to be free of transformations (side effects), then we would no longer need to deal with inverse transformational parsing. The resulting problem would be a standard context-free parsing problem, for which there are many well-known efficient parsing methods (e.g., Earley’s Algorithm [30]).

Secondly, we are unhappy with the representation employed by the rewrite rule framework, where it is usually assumed that the utterance is represented as a flat linear sequence of segments. We are inclined toward hierarchical representation schemes based on suprasegmental constituents (e.g., onsets, codas, syllables, and metrical feet) and subsegmental phonemic features (e.g., place, voice, manner) and phonetic features (e.g., silence, frication, aspiration).38 Our move away from linear formulations of allophonic processes is part of a much larger movement that has been evolving in phonological theory and speech research over the past decade.

1.4 Trends Toward Larger Constituents

1.4.1 Trends in Speech Research

In the mid 1970’s, linear representations were predominant in speech research. Consider, for instance, the POMOW word hypothesizer used in CMU’s Hearsay-II speech understanding system, which was demonstrated on September 8, 1976. This word hypothesizer modeled each lexical item as a flat Markov chain of labels. In the late 70’s, Smith replaced POMOW with NOAH which modeled each lexical item with a 3-tiered hierarchy of markov chains, one at the word level, one at the syllable level, and one at the cluster level.

(27) “Noah uses two levels between the input level of segment-label hypotheses and the output level of word hypotheses, where POMOW uses one. There are 1) the sylpart level, consisting of parts of syllables — onsets (the initial non-nucleus part of syllables), vowels, and codas (the final non-nucleus part of syllables) — and 2) the syllable level, consisting of complete syllables (not

38. One could push this multiple level of representation to its logical conclusion and speculate that the front end signal processing, too, should employ a hierarchical view of low level events. Indeed, it is probably the case that different events take place over asynchronous overlapping time scales. Low frequency events (e.g., nasality) tend to take longer than high frequency events (e.g., aspiration) and tend to overlap (co-articulate) with adjacent segments. Thus, it may not be appropriate to force asynchronous events such as nasality and aspiration into a linear sequence of time aligned segments. In contrast, with a hierarchical representation, we can assign a long event like nasality to a larger unit (e.g., syllable rhyme) and a short event like aspiration to a smaller unit (e.g., a daughter of syllable onset).

(The human ear appears to be well-suited for dealing with the fact that low frequency events take longer than high frequency events. It seems that the ear has better time resolution at high frequencies. This is one of the basic motivations behind critical band analysis, e.g., [67].)
syltypes as found in POMOW). Knowledge is stored in a hierarchy-tree representation. That is, between each pair of adjacent levels (segment-sylpart, sylpart-sylpart, and syllable-word level pairs) is a tree structure storing a sequence of lower level units to define a higher level unit. The last node of the sequence of lower level units points to the defined higher level unit. For example, the syllable-word tree stores sequences of syllables defining each word in the vocabulary. The tree between each pair of levels permits merging common initial parts of sequences to reduce storage costs and recognition time. Thus, the words *confide* and *confuse* share the first syllable node *con*, in the tree, which then points to subnodes *fide, fuse*, etc. [114, p. 13]

### 1.4.1.1 Advantage 1: Improved Performance due to Sharing

Smith achieved two bottom-line engineering advantages by adopting a hierarchical representation. First, there are reduced storage costs and recognition time due to sharing. For example, the tree stores the syllable *con* just once, not once for each word containing the prefix *con* (e.g., *confide, confuse*, and so forth). Furthermore, the hierarchical system will recognize the prefix *con* just once, not once for each word containing the prefix. Smith argues that his representation will reduce recognition costs by 41% over the HWIM design for the same 1000 word dictionary, at a 22% increase in storage due to pointer overhead [114, p. 40]. In addition, Smith found that savings increase approximately logarithmically with larger lexicons. Smith also reported that accuracy improves with larger lexicons, at least under his searching heuristics and his definition of accuracy.

One might object to Smith's argument that increased sharing is a consequence of his hierarchical organization on the grounds that most lexicons are stored in a clever way so as to take advantage of the distributional properties of the lexicon. So, for example, many of the earlier systems like POMOW which represented words as markov chains, would not store the chains in a linear list, as Smith's argument implies, but rather, they would store the chains in a discrimination network that would share the common left parts of the chains. These flat systems, it follows, would in fact share the *con* prefix in *confide* and *confuse* because the left parts of the markov chains for these two words would be identical up through the portion representing the *con* prefix.

---

39. However, one must include certain overhead due to increased pointer structure. This overhead turns out to be significant, especially if pointers are stored in the obvious way.
Nevertheless, although it is possible to introduce some sharing in a flat system, it is possible to introduce even more sharing in a hierarchical system. As Smith shows in his thesis, a hierarchical system can share any subsequence (constituent) that is common to two or more lexical entries. Thus, for example, it is possible to share the \textit{con-} prefix in words such as \textit{confide} and \textit{confuse} as well as the \textit{-con-} “infix” in words like \textit{unconfusable}. On the other hand, a flat discrimination network, can share only the the left (or right) common subparts. Thus, it is possible to share the \textit{con-} prefix, but not the \textit{-con-} “infix”.

Smith rested most of his argument for hierarchical representation on the force of these engineering advantages. There are, though, some equally important linguistic advantages, which we will focus on in this work.

1.4.1.2 Advantage 2: Lexicon is Free of Allophones

A hierarchical representation allows us to represent the lexicon in ways that reflect natural linguistic generalizations. That is, it is desirable to organize the lexicon so that the common shared sequences correspond to natural linguistic constituents (e.g., onsets, peaks, codas, rhymes, syllables, metrical feet, morphemes), not just arbitrary sequences of segments. Thus, for example, the prefix /ri/ ought to be shared in words like \textit{reduce} and \textit{retry} where it is a linguistically motivated constituent, but it should not be shared in words like \textit{read} and \textit{real} where it is merely a common subsequence of segments.

Organizing the lexicon along linguistically motivated lines allows us to factor many allophonic processes out of the lexicon. So for example, we need not introduce rules which rewrite a phoneme such as \textipa{/t/} into one set of allophones in pre-vocalic position (e.g., \{[t^b]\}), and another set of rules which rewrite the same phoneme into another set of allophones in post-vocalic position (e.g., \{[t^l], [t], [c]\}). Rather, our system can incorporate the fact that \textipa{/t/} is realized differently in pre- and post-vocalic position by computing different spectral templates for the /t/ label in the onset position than for the /t/ label in coda position. In essence, then, we are encoding many of the allophonic rules into the structure of the lexicon.

In Smith's system, there is no explicit statement of the allophonic rules. It is expected that they can be accounted for by the training procedure which distinguishes between the /t/’s in each structural position. Thus, the fact that \textipa{/t/} (generally) aspirates in onset position, and (generally) doesn’t aspirate in coda position will be accounted for by training the /t/ in onset position as a distinct symbol from the /t/ in coda position. In training, it will turn out that many of the onset /t/’s will be aspirated and few of the coda /t/’s will be aspirated. From this evidence, the system will acquire an onset template for \textipa{/t/} that favors aspirated /t/’s and a coda template for \textipa{/t/} that penalizes aspirated /t/. With this approach, the system will acquire the aspiration
rule by itself, given sufficient training data and sufficient structural distinctions.  

For this approach, though, to be truly successful, it will be necessary to make a large number of structural positions. (Smith has only three structural positions: onsets, peaks and coda.) How many structural positions are required? It might be necessary to introduce a position for each allophonic feature (e.g., aspiration, glottalization, retroflex). We might also want to factor for the stress contour, the height of the following vowel, and so on. Unfortunately, Smith’s approach requires an enormous amount of training data. In the worst case, we might have to train every phoneme separately for each position that it might appear in. This was not such a serious problem for Smith, since he had only two consonant positions (i.e., onset and offset), so he only had to train consonants twice, once for onset position and once for coda position. But if we want to make a large number of structural distinctions, then we will have to train all the consonants many more times. Perhaps we can’t afford to let the system learn the distribution of allophones in this way. It may be necessary to provide the system with a more explicit model of allophonic variation.

We are extremely sympathetic with Smith’s hierarchic approach, and have incorporated many of the basic ideas in our own work. The model of the syllable to be employed here is somewhat more detailed than Smith’s. He decomposed a syllable into just three sylparts: an onset, a peak and a coda. We wish to enrich the model with a morpheme final appendix position [35], in order to constrain morpheme internal codas much more tightly than Smith was able to. Furthermore, we wish to introduce an additional level between a syllable and a morph, called metrical foot structure. This level is intended to capture the constraints on stress assignment and its implications on allophonic realization. This modification, we believe, leads to an even more elegant formulation of allophonic processes. Moreover, we will attempt to develop explicit models of allophonic processes, rather than acquiring the rules through training. This has a number of practical advantages (e.g., speaker independence, reduced training, scales up with the number of allophonic distinctions) as well as the theoretic advantage of providing falsifiable models of grammar.

It should be noted, however, that despite the elegance of Smith’s design, the performance is most disappointing.

"The word hypothesizer was trained on 174 utterances and tested on 105 new utterances (705 words) for 7 different vocabulary sizes ranging from 500 words to 19,000 words. The

---

40. Smith did not handle schwa deletion this way. Rather, he used the lexical expansion technique. As we will discuss, when we introduce lexical phonology, there are some very interesting differences between schwa deletion, which we analyze as a lexical process, and /i/ realization, which we analyze as a post-lexical allophonic rule.
performance for these vocabularies ranged from a word accuracy of 73% at an average rank of 2.6 for the 500-word vocabulary to a word accuracy of 58% at an average rank of 5.8 for the 19,000-word vocabulary. The rank of a word hypothesis was limited to be less than or equal to 20 (i.e., a maximum of 20 hypothesis per word were allowed). According to the average efficiency measure, which combines accuracy and rank measures, this performance degrades at approximately a logarithmic rate over the range of the vocabulary sizes tested.” [114, p.88]

The performance may be good enough, perhaps, to enable use of the system as a front end to a more accurate, but more expensive word verification system. However, the performance is not nearly as high as one would wish.

It is not clear to us why Smith did not achieve better results. It may be the case that his basic design is appropriate, and that the low performance is the result of a number of relatively minor faults in the implementation. We know, for example, that his lexicon was derived from the computer readable version of the Merriam-Webster Pocket Dictionary [84] with little or no modifications. This dictionary is known to contain many serious flaws that make it unsuitable for such use without considerable modifications. The dictionary suffers from numerous outright errors, poor syllabification, poor secondary stress, no distinction between [θ] and [t], and so on. Furthermore, there may have been problems resulting from Smith’s limited model of allophonic processes. Except for schwa deletion, all other allophonic processes must be captured within the matching procedures alluded to above. Since his model of the syllable is rather limited, lacking a slot for morpheme final appendices and a separate representation of stress, it is quite possible that his system was simple unable to capture allophonic processes adequately. We maintain, though, that his basic approach toward allophonic variation is correct, even though the implementation may have suffered in certain fatal ways.

1.4.2 Trends in Linguistic Theory

1.4.2.1 The ‘Classical’ Position

Linguistic theory has been evolving in a similar fashion. From the time SPE [15] was written until the completion of the ARPA Speech project, rewrite rules were essentially the only tools available to the linguist. It was widely believed that syllables and other larger constituents were unnecessary, and therefore should not be included in the formal statement of the theory.
(28) "The 'classical' model of generative phonology (as presented in, e.g., Chomsky & Halle, Schane) recognized only one sort of structural unit in phonological and phonetic representations: the segment. There was thus no explicit provision for syllables (or other units, such as prosodic feet and the like) as significant elements contributing to the organization of speech. This was not, as some have suggested, simple oversight or failure of imagination, but rather a matter of principle: while traditional phonetic descriptions of course frequently refer to syllabic structure, and many informal statements of processes in Chomsky & Halle do so as well, the convenience of this unit for ordinary language description does not ipso facto establish its linguistic significance. If it were to turn out that all the statements we might want to make in terms of syllables were, when expressed formally, representable simply in terms of (strings of) segments, without important loss of generality, this would suggest that the more parsimonious and restrictive theory which only allowed reference to such units was in fact essentially correct, and thus to be preferred. It was the attempt to establish this program that lay behind the exclusion of syllable structure from the formalism of early generative phonology." [5, p. 546]

Text books in phonology give a slightly different rationale for the "classical" position:

(29) "It is generally accepted that at the phonetic level of representation the sounds of any utterance are organized into larger units called syllables. However, the syllable is probably the most elusive of all phonological/phonemic notions. There is a vast and inconclusive literature on the subject... The primary reason for the great confusion surrounding the syllable is a lack of any adequate phonetic definition. At this point the only thing that can be said with any confidence is that the syllable is an abstract programming unit in terms of which speech is articulated. The chest pulse (units of musculature activity controlling the flow of air from the lungs) is the only articulatory correlate for the syllable that has been discovered. Furthermore, studies by Ladefoged (1967) indicate that the chest pulse is not always an accurate diagnostic for the syllable. ...

Until very recently the syllable has been largely ignored in generative phonology on the assumption that, though it may make sense to talk of sounds being organized into larger units phonetically, all phonological generalizations could be satisfactorily stated in terms of the individual sounds themselves without invoking the notion of the syllable. In this section we discuss a few examples that show this assumption to be unwarranted.

Kahn [54] has argued that a number of phonological processes in English can be adequately explained only if it is assumed that sounds are organized into syllables before the operation of these rules...." [56 pp. 255-256]
Although linear representations have been widely accepted for many years, they have been fading from the limelight in more recent treatments. Kahn [54] was perhaps the first MIT linguist to break from the SPE tradition, by arguing that the syllable was an essential constituent for describing certain allophonic behavior, especially flapping, glottalization, aspiration, and /r/ insertion and deletion. We have adopted many key aspects of his analysis.

1.4.2.2 Metrical Foot Structure

After Kahn came metrical-phonology [78] which was introduced in order to explain the tendency in English and many other languages for stressed\(^{41}\) and unstressed syllables to alternate with each other. So, for example, we have words like *Mississippi* with a stressed first and third syllable and a unstressed second and fourth syllable. This stress alternation could be accounted for by the following set of phrase structure rules:\(^{42}\)

\[(30a)\] morpheme \(\rightarrow\) foot*  
\[(30b)\] foot \(\rightarrow\) stressed-syllable unstressed-syllable

In addition to accounting for stress alternations, foot structure also encodes the stress conditions which are crucial for many allophonic processes. Thus, for example, consider the pair

\[(31a)\] consér\-vative  
\[(31b)\] cônserv\-ação

which have differences in stress, vowel durations, and allophonic realizations of /n/, /s/ and /v/ (see figure 4). Traditionally, the phonetic differences are considered to be consequences of the stress differences. One would say that the /v/ in *conservative* is a post-stress allophone whereas the /v/ in *cônservação* is a pre-stress allophone. The post-stress allophone of /v/ is sometimes called “flap-like” because it is similar to a flap in its spectral characteristics and its post-stress distribution.

\[(32a)\] Post-stress allophones: consér\-vative, bùtter  
\[(32b)\] Pre-stress allophones: conserv\-ação, Tom

---

\(^{41}\) We use a number of different notations for representing stress/accents. See the entry for stress in the glossary (Appendix VII).

\(^{42}\) These rules will have to be complicated shortly to account for words like *addict* which start with an unstressed syllable and for words like *América* where the stress is followed by two unstressed syllables.
"conservative"
Similar comments account for the differences in the /s/s. That is, the /s/ in consérvative is pre-stress and hence stronger and longer than in conservațion where it is post-stress. In general, consonants are stronger and longer in pre-stress position than in post-stress position.

1.4.2.2.1 Metrical Foot Structure and Stress

Metrical foot structure attempts to capture all of these facts and more in terms of constituent structure. According to the grammar of foot structure (e.g., Hayes [40]), the two words would be parsed as:

(33a) con [servative]
(33b) [conser] [vation]

There is a straightforward correspondence between the bracketing structure above and the traditional notion of word stress. That is, as a general rule of thumb, the first syllable of a foot is stressed, and all other syllables in a foot are unstressed.43 The pre-/post-stress distinction can also be recovered from the foot structure. Consonants in foot-initial position are pre-stress, and all other consonants are post-stress. Thus, we say that the /v/ in cons[ervative] is post-stress because it is not foot-initial, and hence it will be realized with a flap-like allophone of /v/. In contrast, the /v/ in [conser][vation] will be realized with a stop-like allophone of /v/ because it is foot-initial.

Thus, foot structure provides us with a very natural way to express stress dependencies in terms of phrase structure rules. This approach may be extremely promising for application in speech recognition, especially when combined with Smith’s ideas about the organization of the lexicon. That is, we might be able to avoid storing allophonic features (e.g., aspiration, flapping, glottalization) in the lexicon. Instead, foot (stress) structure might be represented in the lexicon, and there might be different templates for each position within the foot. In this way, we might save considerable storage since we don’t have to store all the allophones in the lexicon. In exchange, we have to store foot structure, but this cost should be (relatively) minimal assuming that foot structure captures the relevant linguistic generalizations so that there will be enough sharing in the lexicon to make up for the additional overhead in pointers required to represent the structure.

43. The precise details vary from author to author. Liberman and Prince [78] describe a procedure that can recover the difference between primary and secondary stress from their definition of foot structure. In other formulations, such as Withgott’s [124], it is not necessarily true that feet begin with stressed syllables.
In summary, this section has discussed a parallel trend in speech and linguistic research toward richer and richer constituent structures. Selkirk [102 p. 580] summaries this trend toward larger and larger constituents in linguistic theory as follows:

(34) "It should also be evident from the foregoing that a new conception of the relation between segmental and suprasegmental phonology is now available. These two are tightly linked, given this theory, for it is claimed (with supporting evidence) that the proper characterization of the domains for the application of segmental rules is to be made in terms of the suprasegmental prosodic structure. I would like to make the further claim that the domains of rules of segmental phonology are never to be characterized in terms of the boundary elements #, # #, etc., of the standard theory. These are supplanted entirely by the notion of prosodic structure domains, and thus should be eliminated as a superfluous device of the theory. The information about the surface structure of a sentence which was encoded in boundaries (cf. SPE, Selkirk [100, 101]) is, in the present framework, encoded in the prosodic structure itself ..."

The notion of constituency is central to this thesis, both for engineering and for linguistic reasons. We concur with the general movement in linguistic theory in hypothesizing that constituent structure will provide the most elegant formulation of allophonic constraints. In addition, we hope to exploit constituent structure in order to achieve improved accuracy, faster recognition, and reduced storage costs.

1.4.2.2.2 Is Foot Structure Just a Notational Variant?

Before moving on to the next topic, a discussion of the parser, we might want to address a common criticism of foot structure. Is foot structure merely a notational variant of the traditional approach? It is, after all, extremely similar to the traditional representation, merely replacing the pre-stress position with the foot-initial position, and the post-stress position with the non-foot-internal position. It is well-known that prosodic relations such as stress can be represented about equivalently well with features such as [±stress] or with structural relations such [±foot-initial].

(35) "In American linguistics, the term [prosodic features] is used more or less synonymously with suprasegmental features. Suprasegmental features are usually either listed as the set of features consisting of pitch, stress, and quantity, or defined as features whose domain extends over more than one segment (Hamp, 1957)." [76, p.1]
Is metrical phonology just another notation for saying the same old thing? There are a number of very subtle arguments that attempt to address this criticism. For example, one could argue that the traditional approach allows for only two distinctions, pre-stress and post-stress, whereas foot structure allows for three: 44

(36a) foot-initial, as evidenced by the aspirated /t/ in Tom,
(36b) foot-medial, as evidenced by the flapped /t/ in butter,
(36c) and foot-final, as evidenced by the unreleased /t/ in cat.

Thus, foot structure is to be preferred over the traditional approach, if a three way distinction is necessary for descriptive adequacy. It appears to be the case that a three way distinction is called for, as evidenced by the three allophones of /t/ given above. Hence, we conclude that foot structure is to be preferred.

However, this argument is weakened by the fact that it is difficult to find other phonemes with three distinct allophones corresponding to the three positions: foot-initial, foot-medial, and foot-final. Although it is relatively easy to show, for a large set of phonemes, the necessity for at least a two way distinction between foot-initial and non-foot-initial, it is considerably more difficult to demonstrate the necessity for a three way distinction between foot-initial, foot-medial and foot-final. Perhaps the three way distinction is unmotivated, and hence the traditional position is to be preferred.

These sorts of arguments remain a current topic of linguistic research. This thesis will not attempt to decide the validity of foot structure. We will use both the traditional and the metrical terminologies more or less interchangeably in a theory-neutral way so as to avoid taking a stand on this controversial question.

1.5 Parsing and Matching

In the previous section, we presented a number of reasons for adopting a representation based on suprasegmental constituents (e.g., syllables and stress domains). In our framework, suprasegmental constituents play an important role in lexical retrieval, the module that converts segmentation lattices into word lattices. The lexical retrieval device has two components, a parser and a matcher. Our solution differs from many previous systems, which do not include an explicit parsing step.

---

44. Even this point is a subject of contention. A proponent of the traditional approach could capture a three way distinction by including a 'falling-stress' position which corresponds to the foot-medial position.
Fig. 5. Parsing and Matching

Output from Lower Level Processing
↓
Segmentation Lattice
↓
Syllable Structure and Stress Domains
↓
Lattice of Lexical Hypotheses
↓
Input to Higher Level Processing

Parsing

Matching

In our model of speech recognition, syllable structure and stress domains serve as an intermediate level of representation in between segmentation and lexical hypothesis. The gap between segmentation and the intermediate syllable level is considered to be a parsing task; the remain gap from syllable structure to lexical hypothesis is modeled as a matching problem.

Let us present our solution by way of an example. Our approach has been implemented and tested on several minutes of Walter Cronkite’s final CBS Newscast. The program starts with linguistic transcriptions as input because we do not have access to a labeling and segmentation module. (37) is a typical input. This is a transcription (by Meg Withgott) of the announcer on the Walter Cronkite Newscast saying "This is the CBS Evening ..."

(37) əɪslzəʊdʒɪvɪˈbiːesˈiːvniŋ

Phonetic transcriptions are represented internally as character strings. The internal representation of (37) is:

(38) 0|s|z|d|x|s|v|b|y|e|s|q|v|n|g

Figure 6 defines the notation.\textsuperscript{45}

\textsuperscript{45} This figure was constructed by David Shipman. According to Dennis Klatt, this figure ought to be entitled Shipman’s Dialect of Klattese. There are some disagreements on the assignment of A, Y, W, X. We will occasionally use Y and W as phonetic symbols to indicate on- and off-glides. Hopefully the use of notation will be clear from context.
## Klattese

<table>
<thead>
<tr>
<th>Klatt</th>
<th>IPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>k</td>
<td>k</td>
</tr>
<tr>
<td>G</td>
<td>ć</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
</tr>
<tr>
<td>g</td>
<td>g</td>
</tr>
<tr>
<td>j</td>
<td>j</td>
</tr>
<tr>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>G</td>
<td>ŋ</td>
</tr>
<tr>
<td>N</td>
<td>ŋ</td>
</tr>
<tr>
<td>M</td>
<td>ŋ</td>
</tr>
<tr>
<td>L</td>
<td>l</td>
</tr>
<tr>
<td>s</td>
<td>š</td>
</tr>
<tr>
<td>f</td>
<td>ď</td>
</tr>
<tr>
<td>T</td>
<td>ŏ</td>
</tr>
<tr>
<td>z</td>
<td>ż</td>
</tr>
<tr>
<td>Z</td>
<td>ź</td>
</tr>
<tr>
<td>D</td>
<td>ď</td>
</tr>
<tr>
<td>D</td>
<td>ď</td>
</tr>
<tr>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>w</td>
<td>w</td>
</tr>
<tr>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>e</td>
<td>e</td>
</tr>
<tr>
<td>@</td>
<td>æ</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>W</td>
<td>ć</td>
</tr>
<tr>
<td>Y</td>
<td>ć</td>
</tr>
<tr>
<td>a</td>
<td>ą</td>
</tr>
<tr>
<td>O</td>
<td>ą</td>
</tr>
<tr>
<td>O</td>
<td>ą</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>x</td>
<td>e</td>
</tr>
<tr>
<td>j</td>
<td>ɛ</td>
</tr>
<tr>
<td>X</td>
<td>ɛ</td>
</tr>
<tr>
<td>h</td>
<td>h</td>
</tr>
</tbody>
</table>

- Syllabic n, as in "button"
- Syllabic m, as in "bottom"
- Syllabic l, as in "bottle"
- As in "sit"
- As in "she"
- As in "fill"
- As in "thin"
- As in "zone"
- As in "azure"
- As in "van"
- As in "then"
- As in "late"
- As in "rat"
- As in "win"
- As in "yet"
- As in "meat"
- As in "pin"
- As in "met"
- As in "fate"
- As in "lack"
- As in "lock"
- As in "loud"
- As in "like"
- As in "luck"
- As in "bought"
- As in "boy"
- As in "note"
- As in "look"
- As in "cougar"
- As in "bird"

- Schwa or back schwa
- Front schwa; a reduced front vowel
- Retroflex schwa, as in "butter"
- As in "horn"
1.5.1 Parsing

This string is parsed into the lattice of segments

\[(39) \quad \ldots \, D \ldots \ldots \, s \ldots \ldots \, I \ldots \ldots \, z \ldots \ldots \, D \ldots \ldots \, x \ldots \ldots \, s \ldots \ldots \, iY \ldots \ldots \, b \ldots \ldots \, iY \ldots \ldots \, E \ldots \ldots \, s \ldots \ldots \, qY \ldots \ldots \, v \ldots \ldots \, n \ldots \ldots \, G \ldots \]
This is the C B S Evening

The lattice in this case is simply a linear list of segments. It would contain more branching if the hypothetical labeling and segmentation modules had proposed multiple alternatives across some section of the utterance. The program is designed to accept lattices with any amount of branching. Although none of our linguistic transcriptions contain any branching, we have tested the program on some others which have considerable branching. These transcriptions were obtained from Francine Chen’s notes on her spectrogram reading homework for Victor Zue’s seminar.

From the segment lattice above, the program eventually finds all possible syllables using a procedure to be explained later. (The version of the program that generated these lattices did not yet include Barnwell’s vowel constraint.)

\[(40) \quad \ldots \, D \ldots \ldots \, s \ldots \ldots \, iZ \ldots \ldots \, D \ldots \ldots \, x \ldots \ldots \, s \ldots \ldots \, iY \ldots \ldots \, b \ldots \ldots \, iY \ldots \ldots \, E \ldots \ldots \, s \ldots \ldots \, qY \ldots \ldots \, v \ldots \ldots \, n \ldots \ldots \, G \ldots \ldots \, I \ldots \ldots \, s \ldots \ldots \, I \ldots \ldots \, I \ldots \ldots \, qY \ldots \ldots \, v \ldots \ldots \, n \ldots \ldots \, G \ldots \ldots \]
This is the C B S Evening

Note that the lattice is very thin. There are rarely more than four paths active at any given point in the utterance. This property enables very efficient processing.

We can combine syllables into metrical feet in much the same way.

1.5.2 Matching

Once the syllable structure has been established, the structure is then matched against the dictionary to find word hypotheses. Matching ought to be a subject of future research; the solution taken here was quick-and-dirty, borrowing heavily upon the previous work of Smith [114]. We were primarily concerned in this thesis with the parsing task, and simply implemented enough of a matcher so that we could give a demonstration of a system which inputs linguistic transcriptions and outputs lexical entries. Our solution is intended to show plausibility for the basic hierarchical design.
The dictionary is stored in two levels along the lines suggested in Smith's thesis [114]: the first indexes words in terms of syllables, and the second indexes syllables in terms of its sub-constituents (sylparts). The program looks up every sylpart in the syllable dictionary, and then every syllable in the word dictionary. We should have added a third level of indexing at the level of foot structure. Thus, syllables would be indexed in terms of feet, and feet in terms of words. This has not yet been implemented.

1.5.2.1 Canonicalization

The look-up procedure undoes postlexical rules. In particular, allophones are replaced with their underlying phonemic form(s). For example, aspirated /t/ and glottalized /t/ are both replaced with /t/. This process will be referred to canonicalization. Canonicalization need not map an allophone to a unique phoneme. Flaps, for instance, map to /t/ and /d/ (and also /n/ in certain contexts). Canonicalization removes all phonetic detail from the lattice, a necessary operation since the lexicon is represented in terms of phonemes.

This operation, we hypothesize, does not sacrifice information, though, because the information content of the allophonic constraints has already been milked by the parser. Thus, whatever information there was in the allophonic variants in the segmentation lattice has be re-coded into the syllable (and metrical) structures. We could view canonicalization as an information reduction process. It reduces a phonetic segment to its distinctive features (e.g., place, voice, manner) and throws out all of its so-called non-distinctive features (e.g., aspiration, flapping, glottalization). In our framework (and many others such as Stevens [118]), the lexicon specifies only distinctive features and suprasegmental properties such as stress and syllable boundaries. Thus, canonicalizing segments to their distinctive features will not introduce any additional recognition errors, once we have extracted stress and syllable structure.\footnote{We have been assuming all along that non-distinctive features (e.g., aspiration) are useful for parsing syllables and stress domains. Thus, it would be a mistake to canonicalize segments before parsing. Some of the early speech understanding systems may have committed this error in their attempts to segment the acoustic signal into phonemic-like labels. In so doing, they would canonicalize out many of the phonetic cues that might have enabled them to constrain syllable structure and stress assignment.}

Canonicalization can also be used to ignore certain phonemic distinctions that the system does not yet understand. For example, there is a mode in which the system considers all vowels to be equivalent. This mode is useful for dealing with segmentation lattices where vowel errors are far more likely than consonantant errors. This use of canonicalization loses information. That is, phonemic features like vowel height are \underline{distinctive}, unlike phonetic features like aspiration. There are plenty of word pairs that differ in just one
distinctive phonemic feature (e.g., led and lid), but there are no word pairs that differ in just one phonetic feature (e.g., Tom with an aspirated /t/ does not contrast with an other word /tam/ with an unaspirated /t/). Thus, we lose constraining information if we canonicalize out distinctive phonemic features, but not if we canonicalize out non-distinctive phonetic features.

Canonicalization can also be used to ignore distinctions that tend to be neutralized (smeared over) by phonological rules. Suppose, for example, we had a rule which deletes voiced stops after homorganic nasals in words like find. How could we undo this transformation so that we could recognize the word find given the transcription /fain/? We could recover the /d/ by defining the canonicalization function of /n/ to return both /n/ and /nd/. In this way, we can implement a large number of deletion rules. Although this approach works, we believe that this use of canonicalization should be avoided. There appear to be extremely few attested cases where phonological rules really wash out distinctive phonemic features. Whenever possible canonicalization ought to conserve distinctive features. This position has been defended admirably in the literature on acoustic invariance of distinctive features (e.g., [118]). In the case of /nd/ clusters, for example, it is probably not the case that the /d/ is deleted. In fact, an /nd/ cluster can usually be distinguished from a single /n/ by human spectrogram readers on the basis of the duration of the nasal murmur, if nothing else. The difference between /nd/ and /n/ is distinctive in English (e.g., plan/planned, wine/wind, mine/mind, sin/sinned). Ideally, we would hope that the front end could capture distinctive contrasts such as the contrast between /nd/ and /d/. If so, it wouldn’t be necessary for the canonicalizer to wash these differences out.

---

47. Neutralization is a term used in phonology and phonetics to describe what happens when the distinction between two phonemes is lost in a particular environment. For example, in English, the contrast between aspirated (voiceless) and unaspirated (voiced) plosives is normally crucial, e.g., tip vs. dip, but this contrast is lost, or 'neutralized', when the plosive is preceded by /s/, as in stop, skin, speech, and as a result, there are no pairs of words in the language of the type skin vs. *squin. From a phonetic point of view, the explanation lies in the phonetic change which happens to /k/ in the position: the /k/ lacks aspiration, and comes to be physically indistinguishable from /g/. Flapping is another example of a neutralization process. The normally distinctive different between /t/ and /d/ is (essentially) lost in words pairs like writer/reader, for those speakers of American English who flap these medial stops.

48. The /d/ closure is often manifested by a short silence period (or dip) in total energy at the end of the nasal murmur.

49. Canonicalization ought to be dependent on the syllable structure and stress assignment. In the current implementation, canonicalization is independent of structure. This approximation works in more cases than one might expect. Suppose that we decided to assume a /d/ deletion rule and that canonicalization would be used to recover the /d/. Note that the structure independent formulation of canonicalization would also try to insert a /d/ after a pre-vocalic nasal (e.g., near – *nadear). This over-application of /d/ insertion will not lead to an error since /nd/ is not an onset in English. Consequently, there is no chance that *nadear will be found in the lexicon. Thus, one can get by in many cases without implementing the structure dependence condition on canonicalization, though there are also many cases where structure dependence is necessary. Suppose, for example, that we modeled /t/ deletion by defining the canonicalization of /t/ to include the null string as one of its possibilities. This would account for the fact that /t/ can delete in a word like near in certain dialects. Unfortunately, it would predict incorrectly that /t/ can also delete in onset position in words like rodeo and tree (e.g., rodeo → odeo, tree → tea). Obviously, /t/ deletion is dependent on structural relations. If we attempted to model /t/ deletion with canonicalization (and failed to implement structure dependence), then we would introduce recognition errors such as confusions between tree and tea. Structure dependent canonicalization should be implemented. We leave this for further research.
1.5.2.2 The Final Result

After canonicalization and dictionary look up, we end up with the following lattice:50

\[
\begin{array}{cccc}
| & \text{THE...} & \text{SAYS...} & | & \text{THE...} & \text{2:SEE.} & \text{2:BE...} & \text{2:S...} & | & \text{EVENING...} & | \\
& \text{3:THIS...} & \text{IS...} & | & \text{3:THIS...} & \text{UH.} & |
\end{array}
\]

This is the C B S Evening

In some cases, two or more words span over the same segments. For example, the edge printed as “2:SEE” actually represents two words, “SEE” and “C”, both having the same lexical representation: /si/. Rather than attempting to distinguish between homonyms, they program assigns a label (e.g., “2:SEE”) to the equivalence class as a whole, and uses that for processing. The number “2” indicates that the class has two members, and the word “SEE” is a member of the class picked in a purely arbitrary way.

Note that the branching factor is extremely small. In fact, the branching in this example is worse than usual because of the inadequate model of vowels in use in this earlier version of the program and because of the large number of few short words such as this is which can be easily confused with other short words like the says. Longer words, like Evening, are easier for this approach than short words, even with the inadequate model of vowels, because there are more segments to match against the dictionary. Longer words can be more easily distinguished from near misses because there is more information which to base the discrimination on.

Both dictionaries are implemented with discrimination networks. Hence look up time is logarithmic with the number of dictionary entries. (Actually, average case performance might be slightly better since some of the larger nodes in the discrimination network are implemented with hash tables instead of binary trees.) Smith [114] empirically verified the logarithmic behavior in his own implementation. He also found the performance degrades logarithmically with dictionary size.

---

50. In the version of the program illustrate here, vowel information was almost completely ignored. We have since incorporated a better understanding of vowels into a new version of the program which performs considerably better. The new version correctly excludes the says as an alternative for this is based on the difference between /ʌ/ and /ə/ as in /o/. This older version of the program is presented here for illustrative purposes.
1.6 Summary

We began this chapter with a brief outline of the general speech understanding problem and a discussion of various simplifications that have been suggested in the past. This thesis is intended as a small step toward continuous, speaker-independent, large-vocabulary recognition from high bandwidth clean signals produced by a cooperative speaker. In particular, we will focus on problems between segmentation and lexical retrieval. It is argued that we will be in a better position to take advantage of allophonic and phonotactic constraints if we insert an intermediate level of representation in between the segmentation lattice and the lexicon.

The intermediate level of representation was motivated with two different forms of argumentation. First, in section §1.2, we showed that syllable structure could be used to resolve some of the problems associated with the three sentences:

(42a) Did you hit it to Tom?
(42b) Tell the gardener to plant some more tulips.
(42c) We make all of our children.

which had previously seemed to be problematic for bottom-up solutions like our parsing and matching design. Secondly, the intermediate level of syllable structure and stress domains was argued in section §1.4 to be compatible with a general trend in both speech recognition research (e.g., POMOW → Noah) and MIT-style linguistics (e.g., SPE → Kahn’s Syllable Structure → Metrical Foot Structure).

Finally, in section §1.5, we introduced our suggestion for integrating this intermediate level of representation into a speech recognition device.

1.7 Outline of What’s To Come

Chapter 2 deals with the representation of segments. Our proposal differs from standard practice in two respects. First we argue that it is important to represent allophonic cues so that the parser can make use of allophonic constraints in order to parse the input segmentation lattice into suprasegmental constituents. Secondly, we decompose segments into bundles of features, enabling us to capture generalizations at the feature level. The means for capturing generalizations are introduced in chapter 3. Rewrite rules are proposed as a method for modeling allophonic variation.
Although we support the effort to model allophonic variation, we reject the rewrite rule formalism for two reasons. First, the rewrite rule formalism presupposes a flat representation of segments. We prefer to represent segments in a hierarchical constituent structure. Secondly, we object to rewrite rules on generative capacity grounds. Because they have the full generative capacity of a Turing Machine, rewrite rules are very difficult to parse. It is generally accepted that inverse transformational parsing is unlikely to succeed. Our alternative to rewrite rules is introduced in chapter 4. Phrase-structure rules have neither of the disadvantages of rewrite rules; they represent segments in a hierarchical way and they have sufficiently restricted generative capacity that they can be parsed easily. Phrase-structure rules might be considered too impoverished to account for the facts. We show that "phrase-structure rules bear more fruit than you would have thought." (In fact, most (if not all) allophonic processes can be accounted for within a finite state grammar.) We argue that phrase-structure rules are to be preferred because they capture generalizations in the classic sense of a constituency argument. In addition, phrase-structure rules have a number of attractive practical advantages (e.g., modularity, robustness, efficiency).

Chapter 5 discusses the implementation of the parser. Our parser is like a chart parser, but simpler in certain respects. Each non-terminal in the grammar is associated with a lattice. Concatenation, union and transitive closure are implemented as operations on these lattices. A sample grammar is given in Appendix IV.

Chapters 6 and 7 survey a number of constraints that we attempt to capture in our phrase-structure grammar. Chapter 6 focuses on phonotactic constraints in onset and coda position. Chapter 7 establishes the boundary between the coda of one syllable and the onset of the next syllable. In this chapter, we review the maximum onset principle and stress resyllabification. We have applied these constraints in order to assign syllable structure to one of our lexicons. Some of these results are given in Appendix V.

Chapter 8 deals with robustness issues. We will show that it is possible for our approach to deal with a certain amount of branching in the input segmentation lattice, but only within certain limits. We will discuss one example where our approach works fairly well, and one where it doesn’t. Some possible modifications to

---

51. A Turing Machine is an automaton (abstract machine) that can compute any effective procedure. In formal treatments, a Turing Machine is defined in architectural terms (e.g., a Turing machine is a machine with an infinitely long tape and a single head that reads the current value on the tape of 1's and 0's and performs one of a limited set of operations depending on the value on the tape). The reader is referred to [43] for a precise definition. Suffice it to say that a Turing Machine is extremely powerful; it is strongly equivalent to an arbitrary rewrite grammar, the most powerful class of grammars in the Chomsky Hierarchy (i.e., regular grammars → context-free grammars → context-sensitive grammars → arbitrary rewrite grammars). However, there are limits to the power of a Turing Machine. For example, it cannot be decided on any Turing Machine whether or not a particular procedure will halt. Such procedures are said to be undecidable.
our framework are suggested: relaxing phonological distinctions, probabilistic methods, alternative organizations of the grammar. The first two are rejected in favor of the third. We discuss how reliability can be improved by decomposing segments into distinctive phonological features.

Seven appendices are provided. In Appendix I, we discuss the organization of the lexicon. Three basic organizations are introduced: non-recursive discrimination networks, recursive discrimination networks and hash-tables. Most previous systems have adopted the first representation. We prefer recursive discrimination networks because they can take advantage of the phrase-structure constituent structure. We ought to investigate the hashing approach further.

Appendix II provides some of our motivations for avoiding syntax and semantics. First, experience has shown that an over-dependence on higher level constraints can lead to some embarrassing results. Secondly, many of the old arguments for higher level constraints don't seem as forceful as they once did. It was once believed that the speech signal was too impoverished to support direct recognition without first "understanding" the message. This position is not as popular as it once was. As we learn more and more about low level allophonic and phonotactic constraints, the speech signal appears to be much richer. It is currently possible for certain people such as Victor Zue to read spectrograms by hand; we believe that it is just a matter of time before machines can read them too.

Appendix III discusses the interaction between morphology (the study of prefixes, suffixes, compounds, inflection and their role in word formation) and phonology/phonetics (the study of phonemes and phones). It is well-known in linguistics that the two areas are highly interdependent; morphology is essential for the correct statement of a number of phonological processes (e.g., syllabification, stress assignment, tensing, laxing). These processes in turn control a number of lower level (post-lexical) rules such as flapping and schwa deletion. We discuss in some detail a recent proposal by Mohanan that assigns various morphological and phonological processes to a sequence of levels. There are three levels within the lexicon and one postlexical level. The first lexical level is reserved for low level (and relatively idiosyncratic) morphology and phonology (e.g., morphological processes such as irregular inflection and phonological processes such as vowel tensing). The second lexical level is reserved for more productive lexical processes such as "# affixation" (the morphological process that accounts for happy and -ness forming happiness) and related phonological rules such as the rule of compound stress. The third and final lexical level is reserved for regular inflection. There is a postlexical level after the three lexical levels for allophonic processes such as flapping, glottalization, and so forth. This proposal, called both Lexical Phonology and Level Ordered Morphology and Phonology, can serve practical speech recognition efforts because it provides a set of principled criteria
for classifying rules into levels. This organization can be useful as a tool to help the grammar writer keep various processes straight as he debugs the grammar. In addition, it can help justify certain implementation decisions. For example, Smith decided to implement schwa deletion (the rule that accounts for alternations such as *difference* → *dif-ference*) in one way and most other phonological rules (e.g., flapping and aspiration) in another way. With the aid of lexical phonology, we are in a better position to justify this implementation decision. Only a few of these ideas have been incorporated into the current implementation; we find postlexical phonology to be an exciting area for future research and plan to start the investigation upon the completion of this thesis.

Appendices IV, V and VI list a sample grammar, lexicon and output, respectively.

A glossary is provided in Appendix VII.
2. Representation of Segments

2.1 Stevens' Theory of Invariant Features

It is commonly agreed that there are two types of cues: those that vary a great deal with context (e.g., aspiration, retroflex, glottalization) and those that are relatively invariant to context (e.g., voicing, place, manner). The parsing and matching design proposed above is intended to take advantage of both types of cues in a uniform and modular fashion. The parser uses variant cues in order to locate suprasegmental constituents. Having found these constituents, the matcher canonicalizes them into sequences of invariant phonemic-like feature bundles and then looks them up in the lexicon to find word candidates.

In Stevens' theory of invariant features, invariant features (e.g., place, voice, manner) are assumed to be distinctive and therefore useful for retrieving the word from the lexicon. In contrast, it is argued, variant features (e.g., retroflex, rounding, aspiration) are non-distinctive and thus (essentially) superfluous for lexical retrieval. In our framework, it is assumed that invariant features are not the only source of constraint. We believe that both kinds of features are useful cues. Variant features are useful for parsing the utterance into syllables and metrical feet, and invariant features are useful for matching against the lexicon.

We have presented below a short excerpt of one of Stevens' arguments for the role of distinctive features in a speech recognition device.

---

Fig. 7. Examples of Features for Some Consonant Sequences

<table>
<thead>
<tr>
<th></th>
<th>t</th>
<th>b</th>
<th>m</th>
<th>θ</th>
<th>d</th>
<th>s</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>consonantal</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>coronal</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>anterior</td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>labial</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>sonorant</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>continuant</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>strident</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>voiced</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>nasal</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

(This figure appeared in Stevens [118].)
Fig. 8. Some Features for /t/ and /ʈ/  

<table>
<thead>
<tr>
<th></th>
<th>t</th>
<th>ʈ</th>
</tr>
</thead>
<tbody>
<tr>
<td>coronal</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>anterior</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>laminal</td>
<td>(−)</td>
<td>+</td>
</tr>
<tr>
<td>continuant</td>
<td>−</td>
<td>(+)</td>
</tr>
<tr>
<td>high</td>
<td>(−)</td>
<td>(−)</td>
</tr>
<tr>
<td>low</td>
<td>(−)</td>
<td>(−)</td>
</tr>
<tr>
<td>back</td>
<td>(−)</td>
<td>(+)</td>
</tr>
</tbody>
</table>

Some features for the consonants [t] and [ʈ]. The feature values in parentheses indicate situations where the feature may not be needed to make a distinction in English. (This figure appeared in Stevens [118].)

(43) "By way of introduction let us consider a few of the features that play a role in distinguishing between utterances in English. Table 1 [see figure 7] lists some features and indicates the assignment of values of these features for a few segments. Within this table we observe that the segments [t] and [d] are minimal pairs in the sense that they have all features in common except one — in this case the feature voiced. Likewise [θ] and [s] form a minimal pair (the feature strident) (at least with respect to the features given in this table), [s] and [ʃ] are a minimal pair (anterior), and [b] and [m] are close to forming a minimal pair (they differ by the two features in this table — sonorant and nasal). The features also group segments together to form natural classes, i.e., classes of segments that share a feature. In the table, for example, all of the segments shown form a natural class identified by the feature [+consonantal]. [b m d] in the table share the feature [+voiced], and thus form another natural class ...  

Associated with each feature there is a particular acoustic property. This property will be present in the sound whenever that feature is being used to make a distinction in an utterance in a particular language. It is assumed that these properties are matched in some sense to the auditory system, so that there is some kind of distinctive auditory response corresponding to each property. As speakers of the language, our task is to manipulate our articulatory system so as to achieve these acoustic properties or goals corresponding to the various distinctive features for each segment." [118, pp. 2-3]

In contrast, variant features (e.g., retroflex, rounding) play almost no role in Stevens’ framework.
(44) “Let us look at some examples of this type of contextual variation. In English no words are distinguished solely on the basis of retroflexion of a stop consonant. The feature representation for [t], in Table 2 [see figure 8], show the + value for the feature anterior in parenthesis, indicated that there is not a minimal pair with just this contrast in English (although there are, of course, languages for which such a distinction does exist). Consequently, in producing the initial consonant cluster [tr], as in the word trap, American English speakers anticipate the [r] by producing a retroflex [tʂ]. Similar arguments can be used to explain several variants of the [t], all of which occur because the feature specification for this segment is, in a sense, incomplete, such as the specification of the tongue body features and rounding as shown in the table. There are, however, certain acoustic properties that remain unchanged across all of these variants, since these are associated with features that are distinctive in the language. The feature coronal, for example, is one feature that all of the [t]'s have in common ...

Another example from English is the variation in the pronunciation of the back rounded vowel /u/ in different phonetic contexts. In English, however, the front-back feature is not distinctive for high rounded vowels, as shown in Table 3, and consequently it is not unreasonable for a speaker to exercise some latitude in actualizing this feature in this phonetic environment.” [118, pp. 5-6]

Stevens concludes the discussion with:

(45) “What are the implications of all of this for models of speech perception? We see what appear to be invariant acoustic correlates of distinctive features, and we also see acoustic attributes of segments that are context dependent. But it seems that much of this context dependence can be explained by the fact that languages do not require some of the features for some segments to be fully constrained. Under normal conditions, it is probable that most of the invariant properties for the relevant features are in fact present in the signal. The listener can extract the features from these properties and access items in the lexicon which are represented in terms of these feature ... it is suggested that the problem of context dependence is not as severe as one might think at first glance. Work is still needed to sharpen our understanding of some of the features and their acoustic correlates, but at least there is some hope that these remaining problems can be cleared up.” [118, pp. 12-13]
2.2 Our Position

As mentioned above, we believe that both invariant and variant cues are useful. In the proto-type program, a segment is implemented as a named structure data object with separate fields for phonetic and phonemic features. The inventory of phonetic features in the current version are:

(46a) stress markers '12
(46b) glottalization q
(46c) glide v
(46d) rounding w
(46e) aspiration h
(46f) lengthening :
(46g) nasalization -
(46h) unrelease >

though many more are certainly needed, especially: retroflex, palatalization, and color (for liquids).

On the whole, this representation has been satisfactory for our purposes, but it is not without its problems. In particular, it is difficult to express phonetic properties that relate multiple segments. For example, it is not easy to represent the cut-off frequency of a stop burst relative to the place of the following vowel. Similarly, it is difficult to represent the phonetic differences between /n/, /nt/ and /nd/. Admittedly, these facts are probably best accounted for in a theoretical framework at a suprasegmental level. Nevertheless, it would still be very useful for practical applications to have a more adequate mechanism for describing the facts at a very low phonetic level. Perhaps these difficulties are indicative of a more fundamental problem. Perhaps variant phonetic features should not be represented with the same mechanism as invariant phonological features.

It might be appropriate to say a words about the theoretical status of phonetic features. Phonetic distinctions (e.g., aspiration) are generally considered to be more descriptive than phonemic features (e.g., place of articulation). It is generally assumed that postlexical rules are realization rules or spell-out rules; they don't deal with theoretic entities in the same way that lexical rules do. In some sense, we seem to be bridging this distinction in this work, by stressing the importance of phonetic features and by representing them with more or less the same machinery as we used for phonemic features. Although we probably don't wish to make any theoretical claims for phonetic features, it is clear that they play an important descriptive role in speech recognition, and as such, they deserve a certain amount of attention.
“I would like to say a few words here explaining my attitude toward the phonetic evidence that I have taken some pains to assemble. For a linguist, phonetics in only a means toward an end, not a purpose in itself. The end is to provide reliable answers to linguistically relevant questions. However, for providing these answers, phonetics is indispensable. I believe firmly that true statements regarding phonological phenomena presuppose correct observation of their phonetic manifestation. A phonologist ignores phonetics at his own peril.” Lehiste [76, p. vi]

2.3 What’s New

2.3.1 Use of Parsing Constraints

Our representation differs from standard practice on two points. First, as we have said before, we believe that it is important to represent allophonic cues so that the parser can make use of allophonic constraints in order to parse the input segmentation lattice into suprasegmental constituents. In the past, it was believed that allophonic variation was not very useful; allophonic variation was considered to be a source of noise, not a source of constraint. Many systems would attempt to map allophones into phonemic-like equivalence classes. Thus, for example, an aspirated /t/, a retroflexed /t/, a glottalized /t/, and so forth, might be mapped into a single broad class of all /t/s. Even if this attempt were successful, we argue, it fails to take advantage of allophonic constraints.

2.3.2 Decompositional View of Segments

Secondly, in most previous systems, segments are generally modeled as atomic objects. We prefer to view segments as a bundle of features. We would rather not model an aspirated /t/ or a rounded /t/ with an atomic label like ASPIRATED-T or ROUNDED-T. It would require a large number of labels in order to represent all allophones in this way. As the number of labels increases, it becomes harder and harder to work with all of them. Increasing the label set leads to problems at the front end, in the parser, in the matcher, in canonicalization, in the representation of the lexicon, and so forth. We need some way to capture generalizations among allophones. Aspiration works more or less the same way in all voiceless stops. So does rounding. We hope that splitting segments into distinctive features will help to capture some of these generalizations so that each module (e.g., front end labeling and segmentation, parsing, matching, canonicalization) can be designed in a modular way so as to take advantage of the common generalizations among segments.
2.4 Motivations for Representing Phonetic Distinctions

2.4.1 Capture Phonetic Constraints on Suprasegmental Constituents

Why should we try to represent phonetic distinctions such as aspiration, rounding, glottalization, retroflexion, flapping, etc.? Why not look for invariant cues such as place, voice, manner, and be satisfied with that. Why should we look for variant cues? As we have said above, we need allophonic cues to constrain the parsing process. This is certainly the strongest (and perhaps the only) motivation for introducing an explicit representation of phonetic features. Recall some of the examples mentioned above such as at ease and a tease which are phonemically identical, but phonetically distinct. The parser can make use of the flap/aspiration distinction in order to assign the contrasting syllable boundaries and stress domains. We will discuss this pair in more detail in the next chapter when we examine flapping.

There are quite a number of additional examples like at ease and a tease which differ minimally in the phonetic realization of one segment. Recall examples like the following:

(48a) great rain / gray train
(48b) night rate / nitrate
(48c) átribute (noun) / átribute (verb)

which contrast stop-/r/ sequences with the same phoneme in cluster position. One can find similar examples to contrast (practically all) stop sonorant sequences. Consider the following table of /r/, /l/ and /w/ onset clusters. Each stop-sonorant sequence is presented twice, once in a cluster\textsuperscript{52} and once separated by a word/syllable boundary.\textsuperscript{53}

\textsuperscript{52} Notice that the word before the cluster always ends in a tense vowel. Tense vowels can be found in both open and closed syllables, but lax vowels cannot. We will return to this point in chapter 7.

\textsuperscript{53} There is some question whether the crucial juncture is at the syllable level or the word level or some other level. We prefer the syllable choice for theoretical and practical reasons. (Either choice, it seems, can be made to fit the data, more or less.) From a theoretical point of view, the syllable choice is the only choice available within the lexical morphology framework (see Appendix III). Post-lexical rules, such as retroflex and de-voicing, are blind to word boundaries. From a practical point of view, it is very convenient to drive the phonetic constraints off of syllable structure, and not off of word boundaries. It enables the parser to postpone difficult word boundary assignments until there are no more the phonetic constraints to be exploited.
<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>I</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>she prayed</td>
<td>gray plane</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>sheep raid</td>
<td>grape lane</td>
<td></td>
</tr>
<tr>
<td>l</td>
<td>gray train</td>
<td>*</td>
<td>gray twine</td>
</tr>
<tr>
<td></td>
<td>great rain</td>
<td></td>
<td>great wine</td>
</tr>
<tr>
<td>k</td>
<td>may train</td>
<td>may clean</td>
<td>see Quine</td>
</tr>
<tr>
<td></td>
<td>make rain</td>
<td>make lean</td>
<td>seek wine</td>
</tr>
<tr>
<td>b</td>
<td>bay bring</td>
<td>bay block</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>babe ring</td>
<td>babe lock</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>may drench</td>
<td>*</td>
<td>gray dwarf</td>
</tr>
<tr>
<td></td>
<td>made wrench</td>
<td></td>
<td>grade wharf</td>
</tr>
<tr>
<td>g</td>
<td>hey grade</td>
<td>hey glow</td>
<td>hey guana</td>
</tr>
<tr>
<td></td>
<td>Haig raid</td>
<td>Haig low</td>
<td>Haig wanna</td>
</tr>
</tbody>
</table>

In most cases, we ought to be able to find these junctures from phonetic correlates.

(49a) "... word as well as syllable boundaries are often marked by acoustic cues that listeners use for speech perception [Nakatani and Dukes, 1979]. A good understanding of the acoustic manifestations of word boundaries will enable us to resolve phonetic ambiguities and to propose potential boundaries from acoustic evidence. For example, the /pr/ sequence in the word pairs such as 'she prayed', 'sheep raid', and 'sheep preyed' have very different acoustic realizations although their phonemic transcriptions are essentially the same. Many of these cues are very robust; for example, a sonorant in a word initial consonant cluster with an unvoiced stop, such as /pr/ and /pl/, will be partly devoiced. In this context, the stop will generally have a lot of aspiration. If a clear /l/ is seen then there must be a boundary between the /p/ and the /l/." Lamel [72, p. 1]

We have presented one of Lamel's better examples of /pl/ in figure 12. Notice that /-pl/ is heavily aspirated in contrast with /p-l/. Lamel has also observed some other more subtle distinctions. For example, in a /pl/ cluster, the cut-off frequency of the /p/ is unusually high. In addition, as a result of devoicing in the /pl/ cluster, the /l/ has less of a steady state and $F_2$ starts higher. We hope that an automatic segmentation and labeling device will be able to make use of these cues in the near future.

---

54. **Cut-off frequency** is a term from acoustic phonetics used to describe stop bursts and strong fricatives. Imagine what a burst looks like on a spectrogram. The cut-off frequency refers to the frequency of the bottom of the burst.
"I walked along grape lane yesterday."
It is widely agreed that listeners can make use of acoustic correlates in order to disambiguate pairs such as these. Most of the arguments are usually supported with "armchair intuition". Lamel is currently examining acoustic evidence, hoping to evaluate proposed correlates in more rigorous way. Her preliminary study [72] looks positive, though there she has encountered some difficult pairs, especially bay block/babe lock.

The examples presented above dealt with onset clusters. Similar examples can be found for offset clusters such as /nt/ and /nd/.

\[(50a) \text{ can't Ann / can tan} \]
\[(50b) \text{ canned Ann / can Dan} \]

We would argue that the /nt/ and /nd/ will be realized differently depending on the syllable/word juncture. In particular, the nasal will be shortened in can't Ann by the unvoiced obstruent and lengthened in canned Ann by the voiced obstruent. These effects are much less pronounced in can tan and can Dan where the nasals are separated by a syllable/word juncture.

From this sort of evidence, we conclude that (at least some) variant cues are important for lexical retrieval. Although we have managed to integrate only a few phonetic correlates into the current implementation, it is clear to us that it is important to expand this approach in a more complete system.

### 2.4.2 Challenge to the Theory of Invariant Features

There are perhaps some additional motivations for representing phonetic features.

\[(51) \text{ "The importance and difficulty of a phonetic level of processing for speech recognition have been discussed elsewhere [10, 11], but we may summarize the main assertions here:} \]

\[\text{A phonetic stage is necessary because the allophonic variations within phonemes make direct recognition of phonemes impractical except in very limited cases of speech recognition. The same holds for syllable or word recognition. It also seems futile to expect higher levels of processing [e.g., syntax, semantics and pragmatics] to overcome poor phonetic processing." [12, p. 244]} \]

(We will dispense immediately with the discussion of domain restrictions. In restricted tasks, we could attempt to compensate for an inadequate model of phonetics by incorporating additional constraints at some other level(s). We could, for example, restrict the syntax and/or semantics and/or vocabulary so that hard
cases like *gray train/great rain* will never come up. Domain restrictions such as this are beyond the scope of this thesis.)

Broad and Shoup suggest that phonetic variation must be taken into account in order to determine phonemic features such as place, voice and manner. This appears to be a direct challenge to the theory of invariant distinctive features. If phonemic features such as place, voice and manner are really acoustically invariant, it ought to be possible to identify them in a bottom-up way without accounting for allophonic variation. We will discuss three cases where it seems to be necessary to account for contextual variation.

2.4.2.1 Example I: */t, d/*

Confusion matrices can be sensitive to phonetic environments. Consider, for example, a pair of phonemes like */t/* and */d/*. Some */t, d/* allophones are quite distinct in acoustic characterization (e.g., pre-stress) and some are not (e.g., flap). One would expect these allophones to have very different confusion statistics. Pre-stress */t, d/* are relatively less confusable and flapped */t, d/* are relatively more confusable. Other allophones of */t, d/* probably fall some place in between.⁵⁵

Performance would probably improve if the */t, d/* decision were factored up by allophone. For example, it might make sense in a probabilistic framework, to assign very confident scores (e.g., 95% for one phoneme and 5% for the other) in pre-stress environment, and less confident scores (e.g., 55% and 45%) in post-stress environment. In this way, in a probabilistic framework, we could take advantage of the fact that confusion matrices are sensitive to allophonic variation. It is not clear, though, how we could exploit this sensitivity in the invariant feature framework.

2.4.2.2 Example II: Nasals

The */t, d/* example illustrates that phonetic environments can make it easier or harder to detect voicing. We can also find cases where the phonetic environment influences the reliability of the place decision. Consider, for example, the */m, n, ƞ/* decision. The decision seems to be harder in post-stress position. */m, n/* are found in both pre-stress position and post-stress position; */ƞ/* is found only in post-stress position. The fact that distinctive features are harder to identify in post-stress position might explain why */ƞ/* seems to

---

⁵⁵ Zue and Laferriere [129], in their study of medial */t, d/*, found it profitable to consider six distinct phonetic environments: pre-stress (*return, reduce*), nasal-released (*sweeten, Sweden*), flapped (*nauer, raider*), unstressed (*parity, parody*), post-nasal (*tenting, tending*), and post-lateral (*mouled, molded*).
be relatively hard to recognize in the following matrices:86

(52)

<table>
<thead>
<tr>
<th></th>
<th>m</th>
<th>n</th>
<th>η</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>34</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>n</td>
<td>9</td>
<td>109</td>
<td>3</td>
</tr>
<tr>
<td>η</td>
<td>4</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>m</td>
<td>n</td>
<td>η</td>
</tr>
<tr>
<td>m</td>
<td>15</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>n</td>
<td>9</td>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td>η</td>
<td>0</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

We suggest that /η/ is unfairly treated by the statistics above. /η/ should be compared with just the post-stress /n, m/s, not the pre-stress ones, which are much easier. This seems to be another case where it might be profitable to factor confusion matrices by allophonic environment.

2.4.2.3 Example III: /t, k/

Let us consider a third and final example where it might be useful to factor phonemes into allophonic classes in order to determine an "invariant" feature. Consider the case of /t, k/. Burst frequency is considered to be a useful parameter for disambiguating between /t, k/. In general, the burst frequency for a /k/ is much lower than for a /t/.

(53) "Although the frequency locations of spectral concentrations vary from one speaker to another, the bursts of /t/ and /k/ exhibit some general spectral characteristics independent of speaker and, to some extent, context. The /t/ bursts are, in general, rather broadband and occupy primarily the high-frequency region of the spectrum: say, above 2000 Hz. The /k/ bursts show remarkably dissimilar shapes depending of the nature of the following vowel. The bursts of a /k/ preceding a front vowel has a more compact spectral shape than the /t/ burst, and the location of the burst is in the mid-frequency region. The burst of a /k/ preceding a back vowel has a predominant sharp peak in the low-frequency region ..." [127, §5.2]

However, it is important to control for the following vowel. The burst frequency for a /k/ before a front vowel is in the same range as the burst frequency for a /t/ before a back vowel.

---

86. The matrix on the left is taken from BBN's Final Report [126, vol. 2, p. 31]; the matrix on the right is taken from Senef's Study [106, p. 22].
"Averaging across all vowels, the mean burst frequency [of /t/] was found to be 3660 Hz... Excluding the rounded and retroflexed vowels, the mean burst frequency for /t/ was found to be 3900 Hz, whereas the mean burst frequency for /t/ preceding rounded or retroflexed vowels was found to be approximately 3300 Hz.

The burst frequencies for /k/ tend to be distributed in the low- to mid-frequency region... Averaging across all vowels, the mean burst frequency was found to be 1910 Hz. However, there appear to be three distinct peaks in the distribution. Closer examination of data reveals that these peaks are again attributable to the underlying features of the following vowel. The mean burst frequency for /k/ preceding front vowels was found to be 2720 Hz. Preceding back and unrounded vowels, the mean burst frequency was found to be 1770 Hz. The mean burst frequency for /k/ preceding back and rounded vowels was found to be 1250 Hz." [127, §5.2]

In short, there is a considerable overlap between /t/ and /k/. There is only about five hundred hertz difference between front environment /k/ (2720 Hz.) and back environment /t/ (3300 Hz.). However, if we factor for context, we find at least a thousand hertz difference between /t/ and /k/.

<table>
<thead>
<tr>
<th>Vocalic Context</th>
<th>Burst Frequency of /t/</th>
<th>Burst Frequency of /k/</th>
</tr>
</thead>
<tbody>
<tr>
<td>Averaging over front vowels</td>
<td>3900</td>
<td>2720</td>
</tr>
<tr>
<td>Averaging over back vowels</td>
<td>3300</td>
<td>1250 or 1770</td>
</tr>
</tbody>
</table>

Thus, if we knew the place of the following vowel, it would be much easier to make the /t, k/ decision. This provides a strong motivation for designing the labeling module to control for phonetic context.

Even if we didn't know the place of the following vowel, it might still be useful to organize the confusion matrix along phonetic features instead of phonemic ones. Suppose we had a confusion matrix of the form:

(55)

<table>
<thead>
<tr>
<th></th>
<th>back env k</th>
<th>front env k</th>
<th>back env t</th>
<th>front env t</th>
</tr>
</thead>
<tbody>
<tr>
<td>back env k</td>
<td>1/4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>front env k</td>
<td>1/8</td>
<td>1/8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>back env t</td>
<td>1/8</td>
<td>1/8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>front env t</td>
<td>1/8</td>
<td>1/8</td>
<td></td>
<td>1/4</td>
</tr>
</tbody>
</table>

Of the four cases (i.e., back environment k, front environment k, back environment t, back environment k), we assume the first and last are easy, but the middle two are extremely difficult. Victor Zue (personal
communication) has remarked that he is no better than chance at separating front environment /k/ from back environment /t/, when he doesn't know the place of the following vowel. (This is probably a slight exaggeration.)

Suppose that we tried to organize the confusion matrix along phonemic lines. We would essentially be replacing (55) with:

(56)

\[
\begin{array}{cc}
  & k & t \\
  k & 3/8 & 1/8 \\
  t & 1/8 & 3/8 \\
\end{array}
\]

It would be a mistake to organize the confusion matrix in this way. We can no longer deal with the easy cases in a confident way, because we have averaged some of the easy cases with some of the hard ones. A much better characterization of (55) is given in (57); it preserves all (and only) the distinctions that we are capable of making reliably.

(57)

<table>
<thead>
<tr>
<th></th>
<th>low burst</th>
<th>mid burst</th>
<th>high burst</th>
</tr>
</thead>
<tbody>
<tr>
<td>low burst</td>
<td>1/4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mid burst</td>
<td></td>
<td>1/2</td>
<td></td>
</tr>
<tr>
<td>high burst</td>
<td></td>
<td></td>
<td>1/4</td>
</tr>
</tbody>
</table>

This is an extreme example. By reorganizing the matrix as we have, we have obtained a 100% recognition rate. In practice, of course, we wouldn't see such high recognition rates. Nevertheless, we might expect to see large improvements in recognition rate, as the matrix is reorganized appropriately.

In this hypothetical example we found that descriptive labels (e.g., high, mid, and low burst) provided better performance than phonemic labels (e.g., coronal, velar). The phonemic labels could not be assigned as accurately without knowing the place of the following vowel. If this were a general trend, it would stand counter to the invariant feature hypothesis. It may be useful to consider the phonetic environment in making a decision on phonemic features such as place, voicing, and manner. We leave this matter open for further

---

57. We use the terms back environment and front environment in order to avoid confusion. We want to make it clear that we are discussing a descriptive difference, not a theoretical one. Front and back /k/’s have a theoretical status, not shared by /s/. Front and back /k/’s have a different place of articulation, and in some languages, they are treated as separate phonemes.
discussion.

2.5 Capturing Generalizations

2.5.1 Allophonic Proliferation

So far, we have argued that it is important to represent phonetic features primarily so that we can parse suprasegmental constituents (and secondarily so as to minimize labeling confusions). In this section we will show that there are a large number of allophones. Thus it will be useful to attempt to capture some generalizations among similar allophones so as to minimize the burden of allophonic proliferation upon the front end labeler, the parser, the matcher, the canonicalizer, and so on.

In order to demonstrate the gross magnitude of allophonic distinctions, it might be useful to construct a short list of some of the allophones of /t/ that we have encountered. Constructing such a list, though, is a very painful exercise because we are seriously lacking an adequate theory of phonetics. It is fairly easy to list a few distinctive features such as aspiration, rounding, and so forth, but one quickly runs into cases that are not easily classified in such a manner. So, for example, it is not clear what to do with a /t/ before /h/. Furthermore, one feels uncomfortable in classifying two different /t/s with the same label. For example, the /t/ is retroflexed in both the an attribute (noun) and to attribute (verb), but clearly it is more retroflexed in the first case than the second. We really don’t have an adequate terminology for discussing degrees of retroflexion.

With this disclaimer, this rather short list will be offered as a start. There are, of course, much more complete lists to be found in the literature (e.g., Heffner [42]), and hence we will not pursue the matter here.

(58a) aspirated: Tom, return, at all (British), attitude
(58b) retroflexed: tree, terse, the attribute, to attribute, winter
(58c) rounded: twenty, tuna
(58d) palatalized: at your, met your
(58e) flapped: butter, later, party, winter, altitude, repetitive

58. McCarthy discusses repetitive in [83, p. 582]. Three pronunciations offered: (a) both /t/s aspirated, (b) the first flapped and the second aspirated, and (c) both /t/s flapped. McCarthy argues that *rep[t]ive is ungrammatical. We are not sure of the facts. Zue (personal communication) has suggested that it might not be possible for two flaps to appear in a row. If his observation is correct (and we tend to believe that it is), then the facts reported in [83] are incorrect.
(58f) post-lateral: mælted, molten, altitude
(58g) post-nasal: winter, semantic, entice, scientific, can't, Keny, pentagon
(58h) nasal release: Burton, sweeten
(58i) syllable final: atlas, cat, atkins
(58j) before and after /s/: stay, stray, most, boasts, cats, at school, most schools, cents, sense
(58k) before /h/: at his
(58l) before and after fricatives: at the, hit sheep, hitch, soft, softly, often, list some

One might ask, “how many crucial distinctions are there in English?” The APRA speech projects employed about a hundred labels. Were these sets large enough? We suspect not. Would a thousand or ten thousand be enough? Well, we don’t know. The best answer that we have heard to this question is that the number of labels one chooses depends on one’s level of phonetic competence. That is, as one become more and more phonetically expert, there are more and more phonetic distinctions that can be defended with more and more refined linguistic argumentation. Perhaps there is no limit to the number of allophones in English. Rather, the number of allophones that one can argue for is a measure of one’s skill as a phonetician.

One word of caution is perhaps prudent at this point. This discussion should not be interpreted as a justification for inventing large numbers of labels in an arbitrary way. One could for instance quantize a continuous parameter such as duration into a thousand discrete intervals, thereby expanding the inventory with a thousand labels. However, a move such as this is completely unmotivated unless one can show that the language makes use of each of these thousand distinctions. Without such motivation, a move such as this is clearly mistaken.

With this large (possibly infinite) set of allophones, we need some way to capture generalizations among allophones. A large set of (unrelated) allophones will lead to problems for the front end labeler, the parser, the matcher, canonicalizer, and so forth. As the number of labels increases, the labeler’s recognition rate drops. More labels also leads to more complex parsing grammars, matching routines, canonicalization, etc.

In this work, we have dealt with these problems of scale by employing features to impose a structure among segments. As we will show in the discussion of robustness issues (chapter 8), we can alleviate some of the labeler’s recognition problems by decomposing the confusion matrix into distinctive features. This reorganization will improve recognition rate and will also provided an attractive method for estimating the probability of rare events. We suggest that phonetic features be treated in more or less the same way as phonemic features. We can decompose phonetic confusion matrices in much the same way as we did for phonemic confusion matrices in the previous chapter. As we will see, the parsing grammar, the matcher, and
the canonicalizer can all be designed to work in terms of these features, thus avoiding many of the problems attributed to large label sets.

2.5.2 Consistency in Transcriptions

So far we have argued that it is important to capture generalizations at the segmental level for basically engineering reasons. Without such generalizations, it would be impossible to assure robustness, consistency, modifiability, debuggability, and so on in the speech decoder. The same remarks hold for human transcribing. Without these generalizations, it is difficult (or impossible) to assure reliable results.

The current state of the art in transcription is so poor from an engineering point of view that linguists can't even agree among themselves as to what constitutes an error and what does not. For example, in one experiment [128, §2.1], it was found that three trained linguists would agree on the labeling of a segment in only 85% of the segments. Reddy [97, pp. 519-520] summarizes the problems with human transcriptions as follows:

(59) “Shockey and Reddy [112] show that subjective judgments of phoneticians seem to agree among themselves only about 51 percent of the time when labeling spontaneous connected speech in unfamiliar languages ... Carterette [14, §2.2] says that phoneticians disagree on phonemic labels up to 20 percent of the time on a familiar language. Shoup [personal communication] says that phoneticians with similar training and background disagree less than 10 percent of the time on phonetic labels while transcribing a familiar language.”

It is interesting to look further at Carterette and Jones' remarks on reliability of their corpus:

(60) “Transcription of the spoken language was very difficult and time-consuming. Each tape was first transcribed by a typist ... This often required many iterations ... The second step was to have the letter transcriptions checked by a research assistant. This always resulted in many changes. Next the phoneticians took over. When we could get them, there were two working. Because of the volume of material, it was not feasible to have them both transcribe all the text independently. Instead we had to be content with occasional reliability checks together with subsequent ironing out of discrepancies.

Two formal reliability checks were made. The number of disagreements in a passage were counted and this number was divided by the total number of phonemes in the passage. In the first instance there was six per cent discrepancy in 2300 phonemes transcribed. In subsequent discussion, agreement was reached. This involved listening together to the taped passage and
evolving new rules for transcription. Examples of these rules are: (1) Don't use schwa if you can hear the quality of the vowel; (2) if the sound is genuinely between a 't' and a 'd,' use the one which is normal for the word. A second check revealed a still greater discrepancy, sixteen per cent (1800 phonemes). Much of this was due to insistence by the authors that the transcription be as phonetic as possible within the bounds of the phonemic alphabet. Since the phoneticians were not used to this procedure, and also had used different systems in the past, some learning was necessary...

The problems of reliability of transcription were dealt with as effectively as possible under the constraints of time and availability of phoneticians. Phonetics is not an area in which Linguistics students in America are highly trained. It is therefore difficult to hire phoneticians with any but minimum experience. The turnover was likewise high, with long periods of vacancies and retraining... Detailed transcription rules are given in Section 4.1. The purpose of these rules was to permit generation of hypotheses about the natural units of language, since it became immediately obvious upon listening to the tapes that the language was quite different from linguistically ideal or written language...

Despite all this attention toward consistency and reliability, the quality of their transcriptions is not very high. Consider the transcription [14 p. 367]

(61) [wənəyəhætfəpudIn]

for the utterance "when they have to put in". First there is the transposition of the /t/ and /f/ in have to, which is presumably a typographical error. More seriously, the notation conflates an abstract phonemic alphabet with a desire to describe surface phonetic events. The result is extremely ambiguous. Consider the long nasal murmur in when they and the flap in put in. The phonemic alphabet is too abstract to represent this level of phonetic detail. There are no phonemic symbols to distinguish long nasals from short ones, or to distinguish flapped /t/ from aspirated or glottalized /t/. At the phonemic level, all allophones of /t/ are represented with the phonemic symbol /t/. Similarly, at the phonemic level, the /n/ in when they should be represented as /n/. In contrast, with a phonetic transcription, it is possible to represent these allophonic differences, but the much larger phonetic alphabet is necessary in order to make these distinctions.

Carterette and Jones asked for phonetic transcriptions, but restricted the notation to a purely phonemic alphabet. Since the phonemic alphabet is too abstract (impoverished) to represent certain phonetic events (e.g., long nasal murmurs and short flaps), the transcribers employed a number of notational innovations. For example, the long /n/ in when they was transcribed as /nn/ and the flap in put in was transcribed as /d/. 
These practices, unfortunately, introduce unwanted ambiguity. A /d/ might be used to represent a flap, a pre-stress /d/, some other allophone of /d/, a stop-like /ɹ/, and so on. If the representation were more phonemic we could assume that a flapped /t/ is a /t/ (not a /d/), and a pre-stress /d/ is a /d/, and a stop-like /ɹ/ is a /ɹ/. There would be none of these confusions. On the other hand, if the notation were more phonetic, we would have different symbols for each of these possibilities, and again we wouldn’t have any of these confusions. Unfortunately, with the current combination of phonemic abstractness and phonetic description, the transcriptions can be very difficult to decode.

There are other ambiguities in the notation, too. For example, the symbol /y/ is used as an off-glide in /ey/ as well as on-glides in other situations. Similarly, the symbol /w/ is used for the prevocalic consonant /w/ and for rounding in a word like so. Thus, if we have a single /y/ (or /w/) between two vowels, we can’t tell if the /y/ (or /w/) is a consonant or part of a diphthong. There are even some examples of two /y/’s (or /w/’s) in a row, one used in one way and one used in the other. Witness, for example, the transcription of so whatever [14, p. 367]

(62) [sɔwvrəvər]

There are other instances of symbols being used in multiple ways. Note, for example, that the symbol /ɹ/ is used to represent both word final glottalized stops and word initial glottalized vowels. All of these practices introduce confusions that are completely unnecessary. In short, the methods employed here for transcribing speech are probably unsuitable for use in a speech recognition device. At one point, we explored the possibility of using their transcriptions as an input to our prototype lexical access module, but abandoned the project when it became clear that their transcriptions were not sufficiently representative of the problems we wished to address.

2.5.3 Consistency Among Dictionaries

Dictionaries also seem to suffer from consistency problems. It is well-known that syllable boundaries and stress markings are notoriously unreliable, especially when one needs relatively more subtle arguments to assure consistency. So for example, the assignment of primary stress is likely to be fairly consistent from one dictionary to another. But secondary stress, is a completely different matter as Shane points out:
(63) "The multivalued notation utilized in SPE (a representation reminiscent of some of the structuralist approaches, e.g., Trager & Smith 1951, A. Hill 1958), recognizes at least four degrees of stress. (For words in isolation there are only three degrees, since [2 stress] is reserved for the phrasal level.) The [1 stress] of SPE is equivalent to Kenyon & Knott’s primary, whereas the latter’s secondary corresponds to the SPE [3 stress] and [4 stress], e.g., artificiality vs. artificiality. Where the two systems differ most radically is in the SPE treatment of Kenyon & Knott’s (and my) NON-REDUCED vowels in UNACCENTED syllables. In SPE, the majority of these vowels are considered stressed: thus stressed syllables may be contiguous ...

[For example, Daytona is 310 in SPE and WSW in Shane’s notation. In contrast, Dakota is 010 in SPE and WSW in Shane’s notation. Shane accounts for SPE’s 3/1 contrast with a distinction between reduced and non-reduced Ws. The first syllable in Daytona is a non-reduced W; the first syllable in Dakota is a reduced W.] Because of this difference (and even possible ambiguity) in regard to what may constitute a stressed syllable, I prefer, following Kenyon & Knott, to use the terms ‘accented’ and ‘unaccented’.” [108, p. 563]

Similarly, there is some disagreement from dictionary to dictionary on the placement of syllable boundaries. When there are strong phonotactic arguments as in a word like empty there is relatively more agreement. On the other hand, there is relatively little agreement in a word like meteor where there are no such phonotactic constraints.

Much greater consistency, robustness, and so forth could be achieved if we had a better understanding of linguistic generalizations, and if transcribers and lexicographers were forced to justify their decisions with linguistic argumentation. Consider the word meteor once again. The location of the syllable boundary seems to depend on linguistic assumptions which are currently under debate. One could argue for met-eor because it is possible to flap the /r/, and flaps are (usually) found in syllable final position. On the other hand, one could argue for me-teor since the first vowel is tense, and tense vowels are (usually) found in open syllables. Both of these positions can probably be defended. It is not clear which solution is correct. In the interest of consistency, it is important to adopt one of these solutions and apply it uniformly throughout the lexicon. We have found it profitable to do exactly this. All of the syllable boundaries in [84] were replaced with a set of automatically generated syllable boundaries, in order to assure a certain level of consistency.
2.6 Summary

In this chapter, we have argued for a representation of segments that differs from standard practice in two ways. We hope to include a large set of allophonic cues for the parser's benefit, and a rich substructure of phonetic and phonemic features for capturing linguistic generalizations and assuring consistency. The current implementation falls far short of these goals. There are only about a half dozen phonetic features and a half dozen phonemic features. This may be sufficient to demonstrate a few interesting cases in a prototype system, but it is clearly inadequate for a practical system. The next few chapters will discuss how the parser will take advantage of these allophonic cues and how the rich substructure will be put to use in order to capture some interesting linguistic generalizations and guarantee a certain level of consistency.
3. Allophonic Rules

As mentioned in the introduction, speech researchers [6, 21, 92] and linguists [15] have developed systems of allophonic rules for capturing the relevant generalizations. A typical rule of flapping might look something like:

\[(64) \quad t \rightarrow \text{[l]} / V_- V\]

which says that an intervocalic /t/ can be flapped.

Of course, this statement of the flapping rule is probably too simplistic. We may wish to generalize rule (64) to also apply to /d/ and perhaps /n/ which also undergo flapping in phrases like rider and want to pronounced [ra\textsuperscript{5}] and [w\textsuperscript{5}], respectively.

\[(65) \quad \left\{ \begin{array}{c} n \vspace{1em} t \\ d \end{array} \right\} \rightarrow \text{[l]} / V_- V\]

It will certainly be necessary to make many more modifications such as this.

We whole-heartedly support the attempt to model allophonic variation with rules, because these rules capture the generalizations that are so desperately needed in order to achieve the engineering desiderata of consistency, robustness, modularity, and so forth. However, we do take issue with the rewrite rule formalism because of the linear representation at the phonemic level and the transformational (side effect) mechanism. This chapter will argue that the linear representation misses crucial linguistic generalizations at the segmental, syllabic and suprasegmental levels, and that transformations are so unrestricted in generative capacity that they pose severe difficulties for practical implementations. Let us begin with a discussion of the missing generalizations and then move on to consider the processing difficulties.

3.1 Flapping and Syllable Level Generalizations

This section will show that rule (65) requires at least three modifications. The statement of these modifications is very awkward and ad hoc in the rewrite rule formalism, but quite natural in a framework that permits reference to syllable structure. Thus a hierarchical framework (e.g., Kahn's syllable structure) is to be preferred over a linear one (e.g., SPE). This argument is similar to the one proposed by Kahn [54].
3.1.1 Modification 1: Optional Consonant

First, it is necessary to expand the environment specification in rule (65) to allow an optional consonant for cases like party and torn apart which undergo flapping.

\[(66) \quad \{ \begin{array}{c} n \\ t \\ d \end{array} \rightarrow [L / V (C) - V] \]

Obviously this statement of the rule is too general. Although some consonants like /r/ and /n/ can appear before a flap

- (67a) party → par\text{ly} /r/ appears before flaps
- (67b) winter → wi\text{ller} /n/ appears before flaps

many other consonants like /k/ cannot

- (68) actor → ac\text{cor} /k/ does not appear before flaps

The exact characterization of consonants that can and cannot precede a flap remains an open question in the literature. It is fairly commonly agreed that sonorous consonants, especially /r/ and glides, are more likely to precede flaps than less sonorous ones such as fricatives and stops.

\[(69a) \quad \{ \begin{array}{c} n \\ t \\ d \end{array} \rightarrow [L / V (C) - V] \text{[+ sonorant]} \]

phonemic characterization

It is not exactly clear though where to draw the line in between. In fact, it is not clear that there exists a correct partitioning of consonants at the phonemic level into those that can precede flaps and those that cannot. The phoneme /n/ for example can precede /t/-flaps, but not /d/-flaps.

- (70a) winter → wi\text{ller} /nt/ flaps
- (70b) cinder → ci\text{ller} /nd/ does not flap

Perhaps the optional consonant should be determined in terms of a phonetic characterization: [-constriction].
\[ (70c) \{ \frac{n}{t} \}_{d} \rightarrow \mathbf{t} / V \quad (C) \rightarrow \mathbf{t} V \quad \text{phonetic characterization} \]

Flapping is possible after nasalization\(^{59}\) but not after a murmur.\(^{60}\) Flapping is possible in /nt/ because this /n/ can be realized as [−constriction] nasalization, but flapping is impossible in /nd/ where the /n/ is realized as a long [+constriction] murmur.

In summary, we find flaps after [−constriction] allophones (e.g., vocalized /y/, /w/, /r/, and /n/)

\[
(71a) \quad \text{later, lady, pain} \text{t} \text{er, shou} \text{t} \text{ing, loi} \text{ter} \\
(71b) \quad \text{party, part} \text{ing, torn apart} \\
(71c) \quad \text{winter, win} \text{ter, want} \text{ to, going to,} \\
\]

but not after [+constriction] allophones

\[
(72a) \quad ?\text{Washington, } ?\text{Mindy, } ?\text{cinder, } ?\text{wonder} \\
(72b) \quad ?\text{alter, } ?\text{Walker, } ?\text{mal} \text{ted, } ?\text{shelter, } ?\text{seldom,} \\
(72c) \quad ?\text{after, } ?\text{laught} \text{er, } ?\text{softer,} \\
(72d) \quad - \\
\]

after glides 
after /r/ 
after nasalized vowels

If this [±constriction] characterization of the flapping facts is correct (and we hypothesis that it is)\(^{61}\), then rewrite rule formulation would seem to be missing the relevant phonetic generalization. Rewrite rules would attempt to model the optional consonant with a phonemic description like [±sonorant], but this

---

59. The term nasalization is used in acoustic-phonetics to refer to nasal co-articulation with an oral vowel as in band. On a spectrogram, nasalization generally obscures the first formant of the vowel.
60. The term nasal murmur is used in acoustic-phonetics to refer to the closure portion of a released nasal. On a spectrogram, the nasal murmur has a distinctive plateau shape (like an extremely strong voice bar).
61. Some of the facts are perhaps somewhat debatable. Admittedly, some of the questionable cases, especially the post lateral /d/ as in seldom, undergo a weakening process that might be classified as flapping according to some researchers. We choose not to classify these medial stops as flaps because they have different phonetic properties. In particular, post-lateral stops display too much motion in the second formant and they are too long in duration to be flaps. Zue and Laferriere [129] found that post-lateral stops, for example, had a duration of approximately 50 ms., which is much longer than 10 ms. flaps after simple vowels [i, e, a, o, a\(^{W}\), a\(^{s}\), a, æ, ø], and even longer than the 35 ms. flaps found after high front glides [i, e\(^{Y}\), u, a\(^{Y}\), æ\(^{Y}\), yu]. (The difference between the two types of flaps was found to be significant at the 0.005 level. It is surprising that there should be such a difference. Zue and Laferriere suggest an explanation in terms of the mechanics of the tongue.) However, the post-lateral results must be interpreted with caution since the sample size for this condition was very small and because there were some difficulties in performing the measurements.
doesn't work very well for /n/.  

(Presumably this phonetic [±constriction] constraint can be motivated from an articulatory model of speech production. The articulators can't make a ballistic constriction from the position of total constriction. On the other hand, the phonemic [±sonorant] constraint is an ad hoc stipulation.)

3.1.2 Modification 2: Word Boundaries

It is well known that allophonic rules have to be sensitive to word boundaries as evidenced by a minimal pair such as a tease and at ease which differ minimally in the location of the word boundary. This pair might lead us to replace the rule above with

\[
(73) \quad \left\{ \begin{array}{c} \text{n} \\ \text{d} \end{array} \right\} \rightarrow \mathcal{L} / V \quad (C) \quad \overline{(- \text{constriction})} V
\]

which accounts for the fact that /t/ can flap before a word boundary in at ease or word internally in butter, but not after a word boundary as in a tease.

This boundary mechanism is extremely powerful. It can also be used to characterize phenomena that are sensitive to other kinds of boundaries including morpheme, syllable, derivational affix, inflectional affix, and so forth. The boundary mechanism is perhaps even too powerful. It can be used for instance to implement a Turing Machine, and thus it is completely unrestricted. It appears that the boundary mechanism is missing a generalization. According to Lexical Phonology (see Appendix III), boundaries do not have to be stated in postlexical rules (allophonic rules) because they all apply blindly across word boundaries as well as within words.  

---

62. In a rewrite rule framework, we could account for the /nt/-/nd/ contrast by posting a rule of nasal deletion that applies in /nt/ clusters and order this deletion rule before the flapping rule. This account has two drawbacks. First, it depends on rule ordering, a mechanism that we would prefer to abolish because it is difficult to implement. Secondly, deletion violates the conservation principle discussed in section §8.4.

63. It is possible for word boundaries to have an indirect influence on postlexical rules. For instance, the aspiration rule (which accounts for the aspirated stop in Tom) is often said to be a rule which does not apply across word boundaries. Hence, for example, it is not possible to aspirate the /t/ in at ease so that the pronunciation merges with that of a tease. Admittedly, there are no cases where this sort of aspiration rule applies across word boundaries, but this may be because the rule is restricted to syllable onsets, an environment which is certain to be word internal. Thus, we need not stipulate that the aspiration rule is a word internal rule, because it follows from the dependency between syllable structure and words.
3.1.3 Modification 3: Stress

This rule may look overly complicated (and it is), but matters become considerably worse when we consider stress. The stress facts seem to force us to split rule (73) into two rules.

\[(74a) \{ n \over d \} \rightarrow [ + \text{stress} ] / \text{V} \ (\text{C}) \ [ - \text{constriction} ] \ [ - \text{stress} ] \text{ word internally} \]

\[(74b) \{ n \over d \} \rightarrow [ - \text{constriction} ] / \text{V} \ (\text{C}) \ [ - \# \text{V} ] \text{ across word boundaries} \]

It seems necessary to split flapping into two rules because the stress conditions seem to be different in the two cases. Within words, there seems to be a "falling stress" condition. That is, flapping is possible in 10 and 20 stress contours,

\[(75a) \text{10 stress: party, strategy, beauty, lady, meteor, \text{addict} (noun), rodeo flapping possible} \]

\[(75b) \text{20 stress: automobile, international} \]

not not in 01, 02 or 12:

\[(76a) \text{01 stress: strategic, adapt, \text{addict} (verb) flapping impossible} \]

\[(76b) \text{02 stress: adaptability} \]

\[(76c) \text{12 stress: veto, reduce, retire, Barbra Parlee} \]

The stress condition across word boundaries is somewhat different. Notice, for example, that flapping is possible in ål ease despite its 01 stress contour. Thus it might appear that there are two different flapping rules one for word internal flaps and one for crossing word boundaries.

This two rule solution, though, is inelegant. We will see that syllable/foot structure can capture the generalization common to both inter- and intra-word flaps. In addition, it should not be necessary to mention word boundaries in the environment specification of a postlexical (allophonic) rule like flapping. (See the discussion of postlexical rules in Appendix III.) For these reasons, we reject solution (74a-b).
In summary, the rewrite rule formulation of flapping misses three generalizations:

(77a) No constrictions before flaps
(77b) Postlexical rules do not mention word boundaries.
(77c) Inter-word flaps are very similar to intra-word flaps

We will see that suprasegmental phonology is in a better position to capture these generalizations.

3.2 Non-Linear Formulations of Flapping

3.2.1 Kahn’s Ambi-Syllabicit

Kahn [54] and others (e.g., [105, 58]) have argued that certain allophonic rules including flapping would be materially simplified if they were reformulated in terms of larger units, e.g., syllables or perhaps feet. Let us consider Kahn’s particular suggestion in further detail. He argued that flaps occur in ambi-syllabic position, a position that is both syllable final and syllable initial. (In our terminology, ambi-syllabic position translates approximately to “syllable final and pre-vocalic”.) For example, Kahn would say the /t/ in the word butter can flap because it is ambi-syllabic (part of both the first and second syllables). The precise conditions for ambi-syllabicit are difficult to characterize simply (Kahn does so by defining a parsing procedure for determining syllable structure), but, oversimplifying grossly, we can say that an ambisyllabic segment must meet two conditions. First, the phonotactic constraints of the language must permit the segment to join both syllables, and secondly, the segment must be pulled into both syllables by a combination of constraints imposed by stress and word boundaries.

\[ I \ 0 \]

Consider, for example, a word like butter with an intervocalic /t/ in a 10 stress pattern. This /t/ is considered to be ambi-syllabic because the /t/ meets the phonotactic constraints imposed by both syllables and because it is pulled into both syllables. It is pulled to the left by the stress on the first syllable and it is pulled toward the right by a version of the maximal onset principle which Kahn assumes. In contrast, the /t/ in veto is not ambi-syllabic since stress resyllabification is restricted to 10 and 20 stress patterns.

64. We conclude in that discussion that it is worthwhile to phrase postlexical rules (e.g., flapping) in terms of stress and syllable structure, without reference to word boundaries. This enables the parser to undo the effects of postlexical rules without first hypothesizing word boundaries. This is an important practical advantage. Moreover, from a theoretical point of view it, postulating that postlexical rules are blind to word boundaries provides us with more compatibility with the theory of level ordered phonology.
The word boundary cases also work out nicely in Kahn’s framework. Consider the *a tease*/*at ease* pair once again. The */l/* is ambi-syllabic in the second case, but not in the first. Kahn assumes that consonants can be pulled to the right across word boundaries as in *at ease*, but not to the left as in *a tease*. (This asymmetry reflects the fact that consonants are more likely to undergo weakening processes in syllable final position than in syllable onset position.)

In short, the stress and word boundary constraints on ambi-syllabicity have the effect of unifying the two rules (72 a-b). Syllable structures have very natural ways of capturing the disjunction “post-stress or word final”, which is a fairly accurate (but probably not perfect) characterization of the stress and word boundary constraints on the flapping environment.

### 3.2.2 Metrical Foot Structure

More recently, there have been numerous attempts (e.g., [58]) to recast Kahn’s solution in terms of metrical foot structure. Kahn’s ambi-syllabic position translates (roughly) into foot-medial. Consider again the */l/* in *butter*. Recall that this */l/* was ambi-syllabic because the 10 stress pattern pulls the */l/* to the left. The 10 stress pattern will also assign this */l/* to foot-internal position. This */l/* will be foot-internal by almost the same reasoning (in this case). The 10 stress pattern will constitute a foot because feet are (generally) constructed over stressed syllables and all subsequent unstressed syllables. Thus, a */l/* between a stressed and unstressed syllable will be foot-internal. Hence the */l/* in *butter* is both foot-internal and ambi-syllabic.

In contrast, the */l/* is *véto* is neither ambi-syllabic nor foot-internal. Again the two approaches yield the same result by almost the same reasoning. In terms of syllable structure, the */l/* is not pulled to the right by stress because stress resyllabification only applies in 10 and 20 stress. In terms of metrical structure, the */l/* is not foot-internal because feet are not constructed over 12 stress patterns. Thus, in both frameworks, flapping is blocked by the 12 stress pattern.

Foot structure can also account for flapping across word boundaries. We could say that flapping occurs in *at ease*, for example, because the phrase forms a single foot [*at ease*] leaving the */l/* in foot internal position, and consequently eligible for flapping. One might ask why this foot is permitted to begin with an unstressed syllable, whereas all the other feet that we have seen have started with stressed syllables. In order to account for this, we will have to assume some version of stray syllable adjunction, which attaches stray unstressed syllables to an adjacent foot as in */lá[e̞s]/* or */át éase/*.
So far, syllable structure and metrical frameworks produce almost the same results with almost the same reasoning. It is possible to extend these non-linear approaches to deal with many other allophonic phenomena. Consider once again the following three allophones of /t/:

<table>
<thead>
<tr>
<th>Allophone</th>
<th>Theory Neutral</th>
<th>Syllable Theory</th>
<th>Metrical Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>[tʰ] as in Tom</td>
<td>pre-vocalic</td>
<td>syllable initial</td>
<td>foot initial</td>
</tr>
<tr>
<td>[ɾ] as in butter</td>
<td>inter-vocalic</td>
<td>ambi-syllabic</td>
<td>foot internal</td>
</tr>
<tr>
<td>[ɾ] as in cat</td>
<td>post-vocalic</td>
<td>syllable final</td>
<td>foot final</td>
</tr>
</tbody>
</table>

We will soon see that these same three positions appear in many different allophonic rules. This suggests that there are some significant generalizations in these positions. By capturing these generalizations, we are able to implement allophonic rules with the relatively simple parsing and matching algorithm proposed in the introduction. Such solutions are not usually available to systems based on rewrite rules.

3.2.3 Differences between Syllable and Foot Structure

3.2.3.1 Word Boundaries

There are some differences between the Kahn’s ambi-syllabicity and metrical foot structure. Consider a case like:

(78) The [kændɪdeɪt] [easil]y won the election.

Can the /t/ in candidate be flapped? Kahn would predict yes (on our reading of Kahn); foot structure would predict no. Our intuition sides with foot structure. We will not place much weight on this example. Kahn could clearly patch up this gap, if it is indeed a problem with his analysis.

3.2.3.2 Tri-Syllabic Feet

We find more interesting differences when we consider 00 stress patterns. Consider, for example, the second /t/ in repetitive. Metrical foot structure predicts that this /t/ can be flapped since it is internal to a 100 foot. Syllable structure, though, would predict flapping is blocked since there is no stress to pull the /t/ to the left. Metrical foot structure seems to be correct in this case. Again Kahn could patch his system to work just like the metrical solution. He could, for example, redefine stress resyllabification so that it applies in 10, 20, 100, 200, ... stress patterns.
On the other hand, metrical structure incorrectly predicts that there are no dependencies on the flapping of the two /t/’s in *repetitive. One could imagine four possible outcomes:

(79a) repetitive
(79b) *repetitive
(79c) repetitive
(79d) repetitive

McCarthy [83, p. 582] has suggested that (79b) is impossible. Zue (personal communication) has suggested that it is (generally) impossible for two flaps to appear in a row. Somehow the foot structure and syllable structure formulations will have to be modified in order to accommodate these facts, however they should turn out.

Despite these relatively minor differences, both of these non-linear approaches have much in common. They both improve upon the linear approach by capturing the generalization lost in the two-rule formulation of flapping (73 a-b). Furthermore, they both avoid the need to discuss word boundaries in the formulation of postlexical rules.

3.2.4 The Optional Consonant (Revisited)

In the discussion of non-linear frameworks, we have not mentioned the optional [−constriction] consonant that can appear before certain flaps: winter, party. We argued that the phonetic characterization (e.g., [±constriction]) of the optional consonant was superior to a phonemic one (e.g., [±sonorant]) for two reasons. First, the phonetic characterization provided a better description of the facts than a phonemic characterization (recall the discussion of /nt/ and /nd/), and secondly, the phonetic characterization was more motivated in terms of articulatory gestures. Despite its advantages, the phonetic account can be criticized on theoretical grounds. Phonetic features like [±constriction] are considered to be descriptive, and should not be necessary in the proper statement of the rule.

In a non-linear framework, we could replace the phonetic [±constriction] feature with a structural distinction. Suppose that [+constriction] allophones are assigned to one position (e.g., the syllable coda) and [−constriction] allophones are assigned to another (e.g., the syllable peak). In this way, we could replace

65. This proposal has not been implemented in the current prototype.
the rule:

(80) **Flapping rule**: /n, t, d/ flap in foot-internal (pre-vocalic) position unless preceded by a [+constriction] allophone.

with:

(81) **Flapping rule**: /n, t, d/ flap in foot-internal (pre-vocalic) position, unless preceded by a *second segment in coda position*.

Instead of saying that flapping is blocked after a [+constriction] allophone, we can say that flapping is blocked when there are multiple segments in coda position. Equivalently, flapping is blocked in branching codas. This structural statement is more suitable from a theoretical phonological point of view than a statement in terms of the phonetic feature [±constriction].

In order to assure that (81) produces the same results (more or less) as (80), the parsing grammar ought to be reformulated so that [vowel] [nasal] [obstruent] will parse as: 

(82) \[ \text{peak} [\text{vowel}] [\text{nasal}] [\text{coda} [\text{obstruent}]] \]

/nt/

when the obstruent is voiceless.\(^{66}\) The near parallel case of /nd/ clusters will have the constituent structure

(83) \[ \text{peak} [\text{vowel}] [\text{coda} [\text{nasal}] [\text{obstruent}]] \]

/nd/

In this way, [±branching coda] corresponds fairly closely with [±constriction].

We assume that the parser can make use of allophonic cues in order to decide where to assign the nasal. If the nasal is [+constriction], then it will be assigned to coda position. In contrast, if the nasal is [−constriction], then it will be assigned to peak position. In this way, the parser can take advantage of the allophonic rules of nasal deletion and lengthening in order to parse the utterance into a meaningful suprasegmental structure.

---

\(^{66}\) This bracketing structure can be motivated by assuming that voicing assimilation is a constraint in coda position.
3.2.5 Interaction with Morphology

3.2.5.1 #-Prefixes

All frameworks presented thus far (syllable, metrical, and linear) are relatively insensitive to morphological boundaries. We suspect that # boundaries are important. In terms of the metrical framework, they seem to delimit feet at least in some cases. (The discussion will assume the metrical framework for concreteness, though the problems are shared by all frameworks presented thus far.)

Consider words like:

\[(84a)\] auto # mobile
\[(84b)\] inter # dependent
\[(84c)\] inter # departmental
\[(84d)\] inter # collegiate
\[(84e)\] hydro # electric
\[(84f)\] extra # curricular
\[(84g)\] counter # attack
\[(84h)\] counter # intelligence

Each of these words begin with a bi-syllabic foot (e.g., [auto] # [mobile], [inter] # [department]), not a tri-syllabic foot (e.g., *[auto # mobile], *[inter # dependent]). The tri-syllabic foot is ungrammatical because it constructs a foot over a # boundary. Thus the rule for constructing feet must be reformulated to consider # boundaries. Notice that the tri-syllabic assignment would be preferred by the rule that we have been using: construct a foot over a stressed syllable and all following unstressed syllables.

---

67. See Appendix III for a tutorial on the linguistic distinction between "+" and "#" morpheme junctures.
68. The superscript \( S \) and \( W \) indicate "strong" and "weak" syllables. Strong corresponds to primary and secondary stress; weak corresponds to zero stress.
How do we know these words have bi-syllabic initial feet, and not tri-syllabic initial feet? In some cases, most notably *automobile, interdependent and *interdepartmental, we have allophonic arguments for selecting between the two possible foot structures. In other cases, e.g., counterattack and counterintelligence, we can justify the bi-syllabic foot in terms of syllable structure. In these cases, the correct syllabification (counter-attack and counter-intelligence) is different from the one predicted by the maximal onset principle (*counte-rattack and *counte-rintelligence). The correct syllabification will not be obtained if we assume a tri-syllabic foot and allow the default syllabification rules to apply as usual. On the other hand, the desired syllabification will result if we assume that counter- is a foot. (We will discuss syllabification in more detail in chapter 7.)

Let us return to *auto mobile. We claim that this word is parsed [auto][mobile], not *[auto][obile]. Why? Answer: in some dialects, the second vowel can be realized with a tense vowel: auto[w]mobile. There are only three places where a tense vowel can be found:

(85a) in a stressed syllable: rodeo, reduce, meteor, veto
(85b) before another vowel: rodea, meteor, area
(85c) word final (or #-morpheme final): rodea, happy, happy # ness, merry # ment (cf. experiment with a schwa), automobile, electromagnetic

Tense vowels are not found in the middle syllable of a tri-syllabic foot (except under condition (85b), e.g., area). If the second vowel of automobile were foot internal, we would predict that it would always be realized as a schwa (except when deleted by schwa deletion) as in extract. However, the second vowel of automobile can be realized with a tense vowel. Therefore, we conclude that automobile begins with a bi-syllabic foot, not a tri-syllabic foot.

Similarly we have allophonic arguments to show that interdepartmental and interdependent begin with bi-syllabic feet, not tri-syllabic feet. If these words did begin with the tri-syllabic foot interde then the /d/ ought to flapp just as it does in parody. But the /d/s in inter-departmental and interdependent do not flapp. Therefore, we conclude that #-prefixes must be taken into account in the assignment of feet. If we fail to account for #-prefixes, we will make mistakes in predicting flapping, aspiration, vowel tensing and reduction, etc.
3.2.5.2 #-Suffixes

The # boundary can also be important in suffixes. Meg Withgott [124] noticed a very interesting contrast between

\begin{align*}
(86a) \quad & \text{mili}[^b]t\text{ar} \ # \ \text{istic} \quad \text{aspirated} /\nu/ \\
(86b) \quad & \text{cap}i[^c]t\text{al} \ # \ \text{istic} \quad \text{flapped} /\nu/
\end{align*}

She argues very convincingly that the flapping facts demonstrate a contrast in metrical structures.

\begin{align*}
(87a) \quad & [\text{mili}] [\text{tar}] [\text{istic}] \quad \text{foot initial} /\nu/ \\
(87b) \quad & [\text{capital}] [\text{istic}] \quad \text{foot internal} /\nu/
\end{align*}

Presumably the difference can be attributed to their underlying forms:

\begin{align*}
(88a) \quad & \begin{array}{c}
2 \\
0 \\
1 \\
0 \\
\end{array} \\
& [\text{mili}] [\text{tary}]
\end{align*}

\begin{align*}
(88b) \quad & \begin{array}{c}
1 \\
0 \\
0 \\
\end{array} \\
& [\text{capital}]
\end{align*}

The contrast can be attributed to the difference in "#" boundaries.

\begin{align*}
(89a) \quad & [\text{mili}] [\text{tary}] \ # \ [\text{istic}] \\
(89b) \quad & [\text{capital}] \ # \ [\text{istic}]
\end{align*}

(We still need to discuss why the "y" drops out in \textit{tary} and why \textit{tar} loses its primary stress in the derivation. We can probably account for these facts in fairly straightforward ways.)

This example shows very nicely that "#" boundaries play an interesting role in the assignment of foot structure. It is important to understand these processes better if we are to model flapping correctly.

3.2.6 Summary of Argument for Non-Linear Analysis

We have just argued that linear rewrite rules miss crucial linguistic generalizations at the segmental, syllabic and suprasegmental levels. We presented a typical version of the flapping rule. The rewrite rule formulation of flapping required three modifications.
(90a) optional consonant before flap: \textit{winter, party}
(90b) optional word boundary: \textit{at ease}
(90c) a stress condition

These modifications are much more natural in a syllable or metrical based framework. It is very difficult to account for the optional consonant at the phonemic level, because /n/ plays a dual role. When /n/ is realized as a nasal murmur, it blocks flaps, and when it is realized as vocalic nasalization, it does not block flapping. We have seen how a syllable based framework can account for these facts by assigning the murmur /n/ to coda position and the vocalic /n/ to peak position. The syllable and metrical frameworks can also account for (90b-c) in a very natural way as Kahn has shown in his thesis. We have adopted an argument much like his.

Although the non-linear frameworks have much to recommend them, they are not without their problems. We do not currently have a nice method for handling the interaction with morphology. We leave this as an open problem.

3.3 Implementation Difficulties and the Lexical Expansion Solution

Recall that we had two objections to linear rewrite rules:

(91a) they miss crucial generalizations at the segmental, syllabic and suprasegmental levels, and
(91b) they are so unrestricted that they are difficult to implement.

We have just completed discussion of the first point. Let us move onto the second.

Rewrite rules are typically incorporated into a speech recognition system by pre-compiling the lexicon and the routines for accessing it. Within-word allophonic effects are accounted for in the HWIM system [126 vol. II, p. 3], which is representative of most speech recognition systems in this respect, by expanding the dictionary once at compile time to generate all possible stand-alone pronunciations of all words. Across-word phonological effects are later taken into account by compiling the lexical retrieval component.

Expanding the lexicon in this way incurs a clear cost in space. The HWIM system, for examples, reports [126 vol. II, p. 8] that their dictionary of 1138 roots and irregularly inflected forms generates a total of 8642 entries in the expanded dictionary. This cost may increase substantially as the allophonic component is enhanced to incorporate a more complete set of allophones and rules governing their application.
3.3.1 Generative in Nature

An obvious alternative to pre-compilation is to account for the allophonic rules at run time. The HWIM system had actually experimented with this possibility, but found the pre-compilation approach to be more successful [126 vol. II pp. 6-7].

(92) "In the first two years of the speech project (1971-73), we developed two different kinds of phonological rule mechanisms whose effectiveness we were able to compare. The first was a capability to inspect the segment lattice of phonetic labelings and apply phonological rules in reverse to add additional branches to the lattice. The second was a generative mechanism to apply phonological rules in the forward direction to the pronunciations of the words in the lexicon to produce additional possible pronunciations (i.e., dictionary expansion). We envisioned that certain phonological rules would be best implemented in one direction and others in the other direction. In the current system (with the exception of some of the rules used in the APR component itself), we are using only forward generative rules. The reasons for this shift to exclusively generative rules is a combination of the disadvantages of inverse rules and the discovery of an efficient solution to our major problem in the use of generative rules."

Their reasons for rejecting inverse rules are not very convincing. First, they were concerned about obligatory transformational rules. We don't understand why obligatory transformational rules should be more difficult than optional transformational rules. Both types of rules are difficult for parsing because they are extremely powerful. (For instance, transformations can delete all traces of having applied.) We don't see why it makes any difference whether the deletion rule is obligatory or optional. Secondly, the HWIM project was concerned about errors in the segmentation lattice. Although errors are problematic for recognition, it isn't clear to us why errors argue for a lexical expansion solution.

(93) "Applying phonological rules in reverse to the segment lattice has two principal disadvantages. First, there is no good way to implement obligatory rules in the reverse direction. There is in general no easy way to remove from the segment lattice just that path that is matched by a rule without also killing other paths that are not matched by the rule. Even if there were such a method, a path matched by a rule might arise accidentally and not by the mechanism corresponding to the rule, and therefore, in the reverse direction most rules must be considered optional and not obligatory. As a result, the effect of inverse rules is almost always to increase the subsequent processing required. Secondly, the identification of the segments in the lattice is only tentative and may be in error. Hence, many rule matches are also likely to be in error. Looking at such a lattice, one finds possible sequences of labelings that one would not expect to
find in idealized pronunciations, and consequently the rules must be applied in all sorts of cases that they would not have faced in the generative direction. Moreover, when the segment lattice includes a score for every possible phoneme at every point, as it currently does, many rules match very low scoring sequences, and it is difficult to decide how to assign scores to the additional branches added to the lattice.”

Essentially, Woods, et al., are saying that allophonic rules, as they envision them, are generative in nature and not easily re-coded into a form suitable for recognition. Similar sentiments have been expressed by Klatt in support of analysis-by-synthesis [61 pp. 324-325].

(94) “Rules that describe these effects are generative in nature and are partially ordered. Inverse analytic rules useful for sentence decoding are not easy to formulate: they require that one assume a specific phonetic, junctural or syntactic environment for rule applicability, when in fact this information is not available at early stages of the analysis. Some form of analysis-by-synthesis is therefore needed in a speech understanding system ...”

3.3.2 Restrictions on Allophonic Grammars

What does it mean to say that these rules are “generative in nature”? Why does it appear that generation is easier than recognition? Let’s consider this question in terms of formal generative capacity. Formally, a recognizer is a machine that decides whether or not a sentence (string of allophones) is induced by a grammar (set of allophonic rules). How hard is the recognition task? Well, the answer depends on the expressive power of the allophonic rules. If there were absolutely no constraints on the allophonic rules, the transformational grammar could take on the generative capacity of an arbitrary rewrite grammar. In this case, the recognition problem is formally undecidable. In contrast, if the allophonic rules are restricted in certain convenient ways, then recognition is relatively easy. If, for example, the allophonic rules were restricted to be context-free, then we could employ a general context-free recognizer such as Earley’s Algorithm [30]. We believe that it is profitable to restrict the generative capacity so that we can use well-known recognition algorithms.

In the past, researchers have avoided some of the hard recognition problems with a technique called “analysis-by-synthesis”. The machine would start synthesizing (generating) all possible sentences in the language until it finds the target sentence. In this way, analysis-by-synthesis can often analyze (parse) an input sentence even when the grammar is completely unrestricted. Analysis-by-synthesis is similar to the Al
paradigms of "generate-and-test" and "The British Museum Algorithm". These algorithms will find the solution in finite time if they are lucky and they happen to generate the target sentence relatively quickly. Unfortunately, if the language is infinite, it is possible to generate infinitely many sentences before the target. Consequently, it is possible for these algorithms to run forever without terminating.

Again, we can avoid this problem by restricting the language (or the grammar) in certain ways. For example, we can assure that the target sentence will be generated in finite time if the language is restricted to be finite. This assumption is taken in Harpy [79]. It may not be very attractive to approximate English with a finite language, but it does have the practical advantage of termination.

A somewhat more acceptable restriction is taken in HWIM and most other speech recognition systems. These systems assume that there are only a finite number of words in English and that rules don't apply across word boundaries, except for special rules called word boundary rules. If the proper restrictions are imposed on word boundary rules, then this assumption will guarantee decidability.

Woods et al. describe their restriction in a somewhat more favorable light [126, vol. II p. 7].

(95) "... it first appeared that the across-word phonological rules could not be applied to a word until one knew its context. Since the set of possible sentences that could be constructed is potentially infinite, the complete string of phonemes to which one would like to apply rules does not exist until a complete sentence has been hypothesized. Yet, one may need to have applied rules to a word in order to know whether it has occurred in the input in the first place. The problem appeared to be a vicious circle. However, the circle can be broken by a technique developed by Klovdstad [68], which permits across-word-boundary phonological rules to be applied when the in-core dictionary is being constructed. With the resulting compiled dictionary structure, all word matches (including those with word-boundary effects) can be found at run time and the contextual constraints implied by the use of a rule are automatically propagated to adjacent words. This technique is described more fully in [Klovdstad's Thesis (unavailable)]."

It is interesting to compare both of these proposed restrictions with analogous discussions in the literature on natural language syntax, where is is also necessary to restrict the grammars in certain ways to assure decidability. Recall that an ATN, at least as it was originally defined [125], had the power of a Turing Machine (i.e., the generative capacity of an arbitrary rewrite grammar), and hence the model also suffered from similar decidability criticisms. That is, it is possible to design an ATN which is so powerful that it cannot be decided if the machine will accept or reject a particular sentence. Since this is an undesirable situation for
practical applications, computational linguists have been attempting to rectify this situation\(^69\) by restricting
their grammatical formalisms. For example, there are suggestions in the computational literature that
grammar is context-free [36], or deterministic [81], or finite-state [19], etc. All of these restrictions guarantee
effective (and even relatively efficient) processing, but they do so in ways that preserve much of the
interesting structure of the grammar, unlike the assumptions in Harpy and HWIM.

Perhaps a similar approach should be taken at the allophonic level. We hope to develop a grammar of
allophonically well-formed utterances, and restrict the generative capacity of grammar so that it has desirable
computational properties. In particular, we hope to replace arbitrary rewrite rules with context-free rules,
which can be processed efficiently (e.g., by Earley’s algorithm [30]), and which generate parse trees that will
turn out to be very useful as we will see.

\(^69\) This is particularly interesting since there was considerable communication between the two groups of researchers, and because
some of the more influential researchers worked in both areas.

As mentioned above, we have attempted to reformulate allophonic rules in terms of rewrite rules. How can we reformulate these rules as phrase-structure rules? Basically, we will introduce additional constituent categories to encode contextual conditions. For example, one might want to say that a /t/ obligatorily aspirates before a vowel as in a word like Tom.

\[(96) \quad t \rightarrow t^h / \# - V\]

We can model this distribution of \([t^h]\) with a context-free grammar of the form:

\[(97a) \quad \text{utterance} \rightarrow \text{word}^* \]
\[(97b) \quad \text{word} \rightarrow \text{syllable}^* \]
\[(97c) \quad \text{syllable} \rightarrow \text{onset rhyme} \]
\[(97d) \quad \text{onset} \rightarrow t^h | k^h | p^h | ... \]
\[(97e) \quad \text{rhyme} \rightarrow \text{peak coda} \]
\[(97f) \quad \text{coda} \rightarrow t^1 | k^1 | p^1 | ... \]

where an onset can dominate an aspirated /t/, but a coda cannot.

4.1 PS Trees Bear More Fruit Than You Would Have Thought

There is a very widely held misconception that facts such as flapping, aspiration, and so forth, cannot be captured with context-free rules because contextual dependencies require more powerful mechanisms. Sometimes these dependencies are even called context sensitivities in order to emphasize the (incorrect) suggestion that they necessitate a context-sensitive grammar (as opposed to a context-free grammar). What is wrong with this reasoning?

The problem is that context-free grammars have alternative mechanisms for encoding contextual dependencies, even though they cannot encode contextual dependencies in the obvious way. The "trick" is to encode contextual dependencies into the category labels. So, for example, the contextual dependency "\(# - V\)" is found in many allophonic rules, especially those governing aspiration of voiceless stops and the realization of stop clusters like /str/, /tr/, /kl/, etc., is encoded into the category label onset. As we will see, a large number of contextual dependencies can be encoded into phrase-structure category labels in this way.
Perhaps the term context-free is unfortunate. A number of papers have been written attempting to clear up this confusion. Pullum and Gazdar’s so-called “string set” paper [96] documents a large list of such misconceptions that have appeared in the literature over the years. In addition, Joshi and his co-workers have written a number of papers on the power of context-free grammars [51, 52, 53]. They have shown that context-free grammars can be augmented with “local constraints” (Joshi’s term) without increasing the generative capacity beyond that of a context-free grammars. These local constraints seem to be sufficient for expressing most contextual dependencies found in syntax such as agreement facts. Local constraints are almost certainly sufficient for expressing contextual dependencies found in phonology and phonetics. One of these papers is entitled “Phrase Structure Trees Bear More Fruit Than You Would Have Thought”.

4.2 The Constituency Hypothesis

As Joshi, Gazdar, and many others have shown, it is often possible to reformulate rewrite rules into context-free grammars as we have. Of course, if this move were simply a notational reformulation, it would not be very interesting. However, the phrase-structure formulation has an interesting consequence. We hypothesize that the nodes in our phrase structure grammar capture interesting linguistic generalizations. It is possible, we hope, to construct classic linguistic constituency arguments for each of the phrase structure categories introduced by the reformulation of the allophonic grammar from the rewrite notation into phrase-structure notation. These constituency arguments attempt to show that each of the category labels such as onset, coda, syllable, and so forth, appear in numerous rules, and hence the grammar is materially simplified by defining the category in one place, as a phrase-structure rule does, instead of defining it redundantly in every allophonic rule that it is used in.

(98) The Constituency Hypothesis: Many allophonic and phonological processes share the same environments.

We will discuss a few examples of foot-internal, foot-initial and foot-final rules.

4.2.1 Foot-Internal Rules

As evidence that large numbers of rules share more or less the same environments, consider the following rules which apply in inter-vocalic position

70. We do not have an acceptable notation for a flap-like /v, ɾ, θ/ or a laterally released stop.
### Foot-Internal Rules

<table>
<thead>
<tr>
<th>rule</th>
<th>example</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>flapping</td>
<td>[butter]</td>
<td>bu[l]er</td>
</tr>
<tr>
<td>flap-like /v, ŭ, θ/</td>
<td>[liver]</td>
<td>li[v]er</td>
</tr>
<tr>
<td>/h/ deletion</td>
<td>[pro[n]bition]</td>
<td>pro[0]bition</td>
</tr>
</tbody>
</table>

Recall that there were some complicated stress conditions on the flapping rule. It will turn out (we hope) that these same stress conditions hold for the other rules as well. Notice that these inter-vocalic rules fail to apply when the second syllable is stressed.

<table>
<thead>
<tr>
<th>rule</th>
<th>example</th>
<th>near miss</th>
</tr>
</thead>
<tbody>
<tr>
<td>flapping</td>
<td>[beauty]</td>
<td>beau[t]ic</td>
</tr>
<tr>
<td>flap-like /v, ŭ, θ/</td>
<td>cons[erry][ative]</td>
<td>cons[erry][ation]</td>
</tr>
<tr>
<td>laterally released stop</td>
<td>[altitude]</td>
<td>all[iter]</td>
</tr>
<tr>
<td>/h/ deletion</td>
<td>[prohibit]</td>
<td>pro[h]bit</td>
</tr>
</tbody>
</table>

### 4.2.2 Non-Foot-Initial Rules

There is another class of rules which also apply in foot-internal position.

<table>
<thead>
<tr>
<th>rule</th>
<th>example</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>nasal lengthening</td>
<td>[wonder]</td>
<td>wo[n:]er</td>
</tr>
<tr>
<td>nasal deletion</td>
<td>[winter]</td>
<td>wi[~t]er</td>
</tr>
<tr>
<td>stop insertion</td>
<td>[sensitive]</td>
<td>sen[t]ive</td>
</tr>
<tr>
<td>dark /l, r/</td>
<td>[ally]</td>
<td>â[l+]ly</td>
</tr>
</tbody>
</table>

These rules, however, also apply in foot-final position (though perhaps their affects are somewhat different):

<table>
<thead>
<tr>
<th>rule</th>
<th>foot-internal</th>
<th>foot-final</th>
</tr>
</thead>
<tbody>
<tr>
<td>nasal lengthening</td>
<td>[wonder]</td>
<td>[canned]</td>
</tr>
<tr>
<td>nasal deletion</td>
<td>[winter]</td>
<td>[can't]</td>
</tr>
<tr>
<td>stop insertion</td>
<td>[sensitive]</td>
<td>[sense]</td>
</tr>
<tr>
<td>dark /l, r/</td>
<td>[ally]</td>
<td>[all]</td>
</tr>
</tbody>
</table>
Note that these rules do not apply in foot-initial position:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Example</th>
<th>Near Miss</th>
</tr>
</thead>
<tbody>
<tr>
<td>nasal lengthening</td>
<td>[vindicate]</td>
<td>vin[d]ictive</td>
</tr>
<tr>
<td>nasal deletion</td>
<td>se[mântic]</td>
<td>scienc[t]ific</td>
</tr>
<tr>
<td>stop insertion</td>
<td>[sénsitve]</td>
<td>sen[s]ation</td>
</tr>
<tr>
<td>dark /l, r/</td>
<td>[ álly]</td>
<td>all[y]</td>
</tr>
</tbody>
</table>

4.2.3 Foot-Initial Rules

In foot-initial position we find rules like aspiration, stop-like /v, ŋ, θ/, light /l, r/, and so on. We also find strong effects of retroflex, devoicing, and rounding in /r, l, w/ onset clusters. Notice, for example, the contrast between d[tribute] and [attri][bute]. When the /tr/ cluster is foot initial, the /t/ is extremely retroflexed. In contrast, when the /tr/ is foot internal, we see much less frication. The table below contrasts a number of onset clusters in foot-initial and foot-internal position.

---

Fig. 13. “Attribute” Examples

<table>
<thead>
<tr>
<th>d</th>
<th>l</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>de-prive</td>
<td>dip-lomacy</td>
<td></td>
</tr>
<tr>
<td>dep-rivation</td>
<td>dip-lomatic</td>
<td></td>
</tr>
<tr>
<td>a-tribute</td>
<td>att-ribute</td>
<td></td>
</tr>
<tr>
<td>de-crease</td>
<td>de-cline</td>
<td></td>
</tr>
<tr>
<td>dec-riment</td>
<td>dec-lination</td>
<td></td>
</tr>
<tr>
<td>cele-bration</td>
<td>o-bligatory</td>
<td></td>
</tr>
<tr>
<td>celeb-ry</td>
<td>ob-ligation</td>
<td></td>
</tr>
<tr>
<td>a-address</td>
<td>a-cquire</td>
<td></td>
</tr>
<tr>
<td>add-ress</td>
<td>acq-ui-sition</td>
<td></td>
</tr>
<tr>
<td>de-grade</td>
<td>deg-radation</td>
<td></td>
</tr>
</tbody>
</table>
4.2.4 Re-organization of the Grammar

The constituency hypothesis leads us to organize the allophonic grammar somewhat differently from how it has been done in the past. We choose to focus on environments whereas phonicians have traditionally organized their taxonomies based upon phonetic features of the segment that displays the alternation. So, for example, we would classify the aspiration rule under rules that occur in onset position, whereas it has been traditionally been placed under the class of rules that affect voiceless stops. Figure 14 outlines our suggestion for a taxonomy of phonological processes organized on environments. This should be contrasted with figure 15, which is organized along more traditional lines.71

4.3 Advantages of Phrase-Structure Formulation

The phrase structure grammar provides us with an intermediate level of representation between the segment and the lexical entry. This intermediate level of representation is useful for capturing constituency generalizations such as those discussed in the previous section. This is just one of many advantages of introducing a phrase-structure syntax. The phrase-structure grammar imposes much needed order on the data structures (e.g., the lexicon and the input transcriptions) and on the procedures (e.g., parsing, matching, canonicalizing).

4.3.1 Robustness

The grammar can be used to enforce principles of structured programming (e.g., consistency, robustness, modifiability) in each of these areas. For example, we can use the grammar to check the dictionary’s assignment of stress to make sure that it is internally consistent and that it agrees with the notion of stress that have been encoded in the parser. Similarly, we can use the grammar to check the dictionary’s assignment of syllable boundaries. In this way, we can increase the reliability of lexical entries. As mentioned in section §2.5.3, we have found many of our dictionaries to be very unreliable.

Similarly, we can use the grammar to increase the reliability of the input transcriptions. As discussed in section §2.5.2, human transcriptions are also in serious need of reliability checks. Presumably, mechanical transcriptions will also suffer in this way. It may be very useful to check the input transcription lattice for internal consistency. The grammar provides an elegant way to do this. For example, recall the example in

71. This figure is adapted from a slide that Victor Zue used in his 1982 summer class on spectrogram reading.
Fig. 14. A Taxonomy Organized by Syllable Structure

**PRE-VOCALIC** (onset)
- aspiration
- glottalized vowels
- onset clusters (e.g., retroflex)
  - stop-like /v, (vc, ː/)
  - undeleted /h/
  - light /l, prevocalic /r/

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>aspiration</td>
<td>Tom → [tʰ]om</td>
</tr>
<tr>
<td>glottalized vowels</td>
<td>evening → [ˈi]vening</td>
</tr>
<tr>
<td>onset clusters</td>
<td>attribute → a [t ɪ ˈtrɪbudə]</td>
</tr>
<tr>
<td>stop-like</td>
<td>cðnsɜːr [væʃən] (as opposed to cðn [sɜrvətɪv])</td>
</tr>
<tr>
<td>undeleted /h/</td>
<td>prð [hɪbɪt] (as opposed to [prə] [bɪʃən])</td>
</tr>
<tr>
<td>light /l/</td>
<td>light (as opposed to all)</td>
</tr>
</tbody>
</table>

**POST-VOCALIC** (codas or foot-final)
- glottalized stops
- unreleased stops

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>glottalized stops</td>
<td>cats → ca [tʃ]s</td>
</tr>
<tr>
<td>unreleased stops</td>
<td>cat → ca [tʃ]</td>
</tr>
</tbody>
</table>

**INTER-VOCALIC** (ambi-syllabic or foot-internal position)
- flapping
- flap-like /v, (vc, ː/)
- laterally released stops
  - /h/ deletion (or voicing)

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>flapping</td>
<td>butter → bu [ˈbʌr]</td>
</tr>
<tr>
<td>flap-like</td>
<td>cðn [sɜrvətɪv] (as opposed to [cðnsɜːr] [væʃən])</td>
</tr>
<tr>
<td>laterally released stops</td>
<td>seldom</td>
</tr>
<tr>
<td>/h/ deletion</td>
<td>[prə] [bɪʃən] (as opposed to [prə] [hɪbɪt])</td>
</tr>
</tbody>
</table>

**WORD-FINAL** (appendix, affix)
- long sequences of dentals
- voicing assimilation

**POST- or INTER-VOCALIC** (non-foot-initial)
- nasal lengthening
- nasal deletion
- stop insertion
- dark /l/, post-vocalic /r/
- nasal assimilation

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>nasal lengthening</td>
<td>canned → ca [n:rd]</td>
</tr>
<tr>
<td>nasal deletion</td>
<td>can't → ca [0]nt</td>
</tr>
<tr>
<td>stop insertion</td>
<td>sense → sen [t]se</td>
</tr>
<tr>
<td>dark /l/</td>
<td>all (as opposed to light)</td>
</tr>
<tr>
<td>nasal assimilation</td>
<td>hypnotize → hip[ə]ntəzɪˈ</td>
</tr>
</tbody>
</table>
Fig. 15. A Taxonomy Organized by Phonological Class

**VOEWS**
- vowel reduction
- schwa deletion
- schwa de-voicing
- syllabification
- vowel lengthening
- diphthongization
- schwa insertion
- vowel raising
- vowel lateralization
- vowel nasalization
- vowel retroflexion
- front rounding (Umlauting)

**SEMI-VOEWS**
- glide insertion
- palatalization of vowel
- ruh reduction
- ruh-lessness
- r insertion and deletion

**CONSONANTS IN GENERAL**
- geminate reduction
- palatalization
- voicing assimilation

**STOPS**
- stop insertion
- stop deletion
- gottalization
- stop aspiration

**FLAPS**
- alveolar stop flapping
- n flapping
- flap deletion

**NASALS**
- nasalization of vowels
- nasal deletion
- -ing reduction
- nasal assimilation

**FRICATIVES**
- s deletion
- fricative clusters simplification
- sibilant fronting
- fricative de-voicing

**OTHERS**
- h deletion
- cluster phonotactics
- article selections

- spectrogram → spectr[ə]gram
- boun-da-ry → boun-dry
- multiply → mul[tə]ply
- suddenly → sudd[ə]ly
- e.g., before a pause or voiced consonant

- pure → /pyuər/
- compute → comp[ü]ute
- the angel → the [ʌ] angel; blowing → blo[ŋ]ing
- California → Californ[œ]
- introduction → int[ə]duction
- surprise → s[ə]prise
- idea → ide[r]; forward → fo[ʊ]ward

- bus stop → bu[s:][t]op
- did you → di[t]ou; this year → thi [ʌ]ear
- have to → ha[ʊ] to
- we eat → we[ɪ] eat; console → con[t]sole
- softly → sof[0]ly; kindness → ki[n:0]ess
- gotten → go[ʊ]n

- data → da[ʊ]a
- how many → how ma[d]v
- little → li[ʊ]

- camp → c[æ]mp
- spent → sp[ɪ]nt
- simplifying → simplif[i][n]
- hypnotize → hip[m]otiz
- give me → gi[m]e
- is there → is[0]ere
- sixths → si[k]ths
- tunafish sandwich → tunaf[0] sandwich
- e.g., fricative sonorant onset clusters
- huge → [ʊ]uge
- e.g., fricative stop clusters
- e.g., a/an and th[ə]/th[i]
section §1.2.1.3.

(99)  dh3h11h]t]am

In this case, the transcription is ill-formed because the lax vowel /l/ cannot end a syllable, and yet the following aspirated /t/ must start a new syllable. Therefore, we have a contradiction. This sort of inconsistency should be caught by the parser; there is no way to assign the lax vowel and the aspirated /t/ to a sequence of well-formed syllables.

The phrase-structure grammar can also be used to enforce explicit conditions on the parser, the matcher and the canonicalizer. It is possible, for example, to use (almost)\(^{72}\) the same grammar to parse transcriptions as we use for checking the dictionary. In this way, we can be sure that they agree on the exact definition of syllable structure, stress assignment and so on.

We have actually caught a number of interesting flaws in the parser by checking it on all words in the lexicon. Consider, for example, the word exchange. This is one of very few words in English where we have a /ks/ offset cluster in word internal position. Most other cases of /ks/ can be split into two syllables (e.g., extra) or can be split into a coda and a word final appendix (e.g., box). It turns out that /ks/ can also be found before a few strong fricatives such as /z/. We had originally overlooked this possibility.

The phrase-structure grammar improves the reliability of matching as we discussed in section §1.2.3. The matcher will consult the lexicon with allophonically and phonotactically well-formed candidates. There is no chance that the matcher will accept an ill-formed string. If the matcher did not filter out ill-formed strings before consulting the lexicon, there is always the chance that an ill-formed string might match something due to some bug or oversight in the matcher. (Our system also saves the time required to consult the dictionary will ill-formed candidates.)

The canonicalizer can also benefit from the structure imposed by the grammar. As we have discussed in section §1.5.2.1, it would be advantageous for the canonicalizer to make use of structure. Recall the example of /tr/ deletion, where it might be reasonable to postulate a rule of /tr/ deletion in certain contexts (e.g., near) but not in others (e.g., rodeo). This has not been implemented, although it certainly should have been.

\(^{72}\) There is a slight difficulty in running the grammar on the lexical entries since the grammar is designed to work with phonetic labels, whereas the lexical entries are represented in terms of phonemic labels. The grammar has been kludged to get around this problem in a few places.
In short, the phrase-structure grammar imposes much needed order in practically all aspects of lexical retrieval:

(100a) location of junctures
(100b) inverting phonological rules
(100c) lexical access
(100d) organization of lexicon
(100e) segmentation and labeling

4.3.2 Efficiency

The phrase-structure grammar also improves efficiency in several ways. Smith [113, 114] showed how a hierarchical representation of the lexicon could improve accuracy and reduce space and time costs. We will review his arguments in Appendix 1. Basically, for any database, it is well-known that we can achieve improved performance by imposing a discrimination structure on the keys. It is not necessary that the discrimination structure be a natural one; performance will improve even if the discrimination structure is purely arbitrary (e.g., alphabetic order). However, if the discrimination order is a natural one based on natural units such as syllable structure and stress, then we expect to find even better performance. In this way, the phrase-structure grammar improves matching efficiency.

The grammar also improves the efficiency of the procedure for locating junctures. If we didn't have a parser, we would have to assume that every node in the segmentation lattice is a potential juncture. Thus, if we had a segmentation lattice with n nodes, we would have on the order of n² word arcs. With our syllable grammar, we can do much better. Syllables have a finite length. Suppose the longest syllable has 10 segments. Then there cannot be more than 10n syllable arcs. This is a big improvement over n². However, the average case is even better because there are usually some very powerful allophonic and phonotactic constraints that improve matters substantially. As we will show in section §6.4.1, there are usually only one or two places where a syllable boundary can be found between two vowels. For example, if we had the sequence:

---

73. As we will see in chapter 6, it is possible to reduce this number substantially down to something like 5.
(101) ... Vowel t r Vowel ...

we could find a syllable boundary in just two places:

(102a) ... Vowel # t r Vowel ...
(102b) ... Vowel t # r Vowel ...

The other possibility

(103) *... Vowel t r # Vowel ...

is out because /tr-/ is not a possible syllable coda in English. Suppose most consonant sequences are like /tr/ in the sense that they can only be syllabified in at most two ways. Then there are at most 4n syllable arcs. The lattice is even more constrained when we account for allophonic constraints such as aspiration and glottalization. Recall that most of the examples of syllable lattices have been very thin. On the average, there are usually fewer than four arcs spanning across a vowel.

4.3.3 The Chicken or Egg Paradox

Context-free grammars are well-suited for coping with the so-called “Chicken or Egg Paradox”.

(104) “A paradox seems to exist which might prevent listeners from making effective use of durational cues [or any other type of context-dependent cues] in forming hypotheses about sentence structure. In order, e.g., to perceive the lengthening at a phrase boundary as a cue to the end of a constituent, it seems necessary to know the identity of the lengthened segments. Durations are lengthened relative to the inherent durations for the segments in question. But a listener can’t know the inherent duration until he has identified the vowel, so we have a version of ‘the chicken or the egg’ paradox” [63, p. 1220]

Although it is correct that we cannot decide which came first, the chicken or the egg, there is no particular reason why we should get upset about it. There will be times when the system cannot decide between two alternatives. In Klatt’s example, there are two possible explanations for the long vowel:

---

74. This turns out to be a consequence of the *genre* hierarchy as we will show in section §6.4.1.
75. In this discussion, we have been assuming no branching in the input transcription. As the input lattice becomes bushier, there are more possible syllables.
(105a) inherent lengthening
(105b) contextual lengthening

There may not be any evidence for deciding between these two cases. The best we can do is to limit the space of possibilities to the obvious two. Context-free grammars permit us to do this in a natural way. Suppose, for example, that we had a grammar like the following:  

\[(106a) \quad \text{peak} \rightarrow \text{vowel} \quad [+] \text{contextual-lengthening} \quad [+] \text{inherent-lengthening} \quad [+] \text{contextual-lengthening} \]

\[(106b) \quad \text{peak} \rightarrow \text{vowel} \quad [+] \text{contextual-lengthening} \quad [-] \text{inherent-lengthening} \quad [+] \text{contextual-lengthening} \]

\[(106c) \quad \text{peak} \rightarrow \text{vowel} \quad [-] \text{contextual-lengthening} \quad [+] \text{inherent-lengthening} \quad [-] \text{contextual-lengthening} \]

\[(106d) \quad \text{peak} \rightarrow \text{vowel} \quad [-] \text{contextual-lengthening} \quad [-] \text{inherent-lengthening} \quad [-] \text{contextual-lengthening} \]

Suppose there are three classes of vowel lengths:

\[(107a) \quad \text{vowel} \quad [-] \text{inherent-lengthening} \quad [-] \text{contextual-lengthening} \quad \text{short class} \]

\[(107b) \quad \text{vowel} = \text{vowel} \quad [+ \text{inherent-lengthening}] \quad [- \text{inherent-lengthening}] \quad [- \text{contextual-lengthening}] \quad [+ \text{contextual-lengthening}] \quad \text{medium class} \]

\[(107c) \quad \text{vowel} \quad [+ \text{inherent-lengthening}] \quad [+ \text{contextual-lengthening}] \quad \text{long class} \]

---

76. We use features for convenience; they do not take the generative capacity of the system outside that of context-free grammars, at least the way that we use them.
If we are given a vowel from either extreme, then we can deduce both the inherent length and the context. On the other hand, if we are given a vowel from the medium class, we can't deduce either property. The parser will quickly realize that both cases are possible and simply put them both into the chart. This is the best that can be expected under the circumstances. If, at some later time, we should find some other cue that bears on the matter, then we might be able to use it to disambiguate between the two possibilities. We might discover, for example, that the vowel is inherently short because those are the only vowels that are found in the lexicon. We could use this information then to rule out the first possibility in (107b), leaving us with an inherently-long contextually-short vowel. Thus, if we are given sufficient cues to come to a unique solution, we can. But, on the other hand, if we are not given enough information, we can recognize the situation as ambiguous, and deal with it accordingly. This is one of the virtues of context-free parsers (and of the more general class of algorithms known as dynamic-programming).

4.4 Summary

This chapter introduced the phrase-structure grammars as an alternative to rewrite rules. In section §4.1, we dispelled the common misconception that context-free and finite-state phrase-structure rules are inadequate to capture contextual dependencies. Phrase-structure rules capture contextual dependencies by encoding them into the non-terminal symbols. For example, instead of saying that an /s/ is realized differently before /tr/ in words like street, we could say that an /s/ is realized different in retroflexed onset clusters. In this way, we have encoded the contextual dependency “/\_ trV” with the non-terminal symbol “retroflexed onset cluster”. Thus it is not necessary to represent contextual dependencies with primitives like “/\_ trV”. (In section §5.7.2, we will re-introduce contextual primitives. For now we will do without this mechanism.)

So far, we have argued that phrase-structure rules are capable of representing more facts than you might have thought. In section §4.2, we hypothesized that phrase-structure rules are preferable to rewrite rules because phrase-structure rules capture generalizations that cannot be captured in the rewrite rule formalism. If many rules share the same environments (and we hypothesize that they do), then it makes sense to package up these environments into phrase-structure constituents.

For our practical purposes, we can use the constituent structure as an intermediate representation in the decoding process. The first step parses the input transcription lattice into a constituent structure. The second process matches the constituent structure against the lexicon. The motivation for this intermediate constituent structure representation lies with the constituency hypothesis. It makes sense to look for
constituents if they capture lots of generalizations (i.e., there are relatively few constituents compared with the number of rules).

In addition, because of the constituency hypothesis, there ought to be a number of redundant cues for telling us how to assign foot structure. Consider the pair [prò] [hibìt] / [pròhì] [bìtìón] once again. The standard argument for the contrast in constituent structure notes the different /h/ cs. In [prò] [hibìt], the /h/ cannot delete, suggesting that the /h/ is foot initial. In contrast, the /h/ in [pròhì] [bìtìón] can delete suggesting that it is foot internal. This is just one of many arguments for the contrast in foot structures. How many additional arguments can we construct? Note that there are differences in all of the vowels. The vowels in foot final position are tense or reduced; the vowels in the first syllable of a foot lax and unreduced.77

(108a) [prò] [hibìt] / [prohì] [bìtìón]  
(108b) [prò] [hibìt] / [prohì] [bìtìón]

In addition, there are differences in the consonants that can probably be attributed to the contrast in syllabification. Thus, we have quite a large number of reasons for assigning the constituent structure the way we did. Hopefully the parser can make use of these redundant cues. In this way, the decoder can take advantage of the constituency hypothesis in order to improve robustness. The robustness issue was discussed in §4.3 along with a number of additional engineering desiderata.

---

77. The first vowel in [prghì] [bìtion] is tensed by another mechanism. In addition to the rule that accounts for tense vowels in foot final position, there is another rule which tenses vowel when it precedes another vowel (or deleted /h/).
5. Parser Implementation

In the previous chapter, we proposed phrase-structure rules as an alternative to rewrite rules. Recall that aspiration could be restricted to syllable initial position (a reasonable first approximation) with a set of rules of the form:

(109a) utterance $\rightarrow$ syllable$^*$
(109b) syllable $\rightarrow$ onset rhyme
(109c) onset $\rightarrow$ aspirated-t | aspirated-k | aspirated-p | ...
(109d) rhyme $\rightarrow$ peak coda
(109e) coda $\rightarrow$ unreleased-t | unreleased-k | unreleased-p | ...

This sort of context-free phrase-structure grammar can be processed with well-known parsers like Earley’s Algorithm [30]. Thus, if we completed this grammar in a pure context-free formalism, we could employ a straightforward Earley parser to find syllables, onset, rhymes and so forth.

However, it has been our experience that the context-free formalism is not exactly what we want for this task. Context-free rules are generally considered to be awkward for manipulating features. For example, in order to express subject-verb agreement constraints in context-free rules, we probably have to say something like:

(110a) $S \rightarrow$ singular-NP singular-VP  \hspace{1cm} \textit{singular case}
(110b) $S \rightarrow$ plural-NP plural-NP  \hspace{1cm} \textit{plural case}

where much of the grammar will have to be repeated in multiple places. In this example, the grammar of plural noun phrases will likely duplicate much of the grammar of singular noun phrases. There is no known way to avoid this repetition in a pure context-free framework.

The same sort of agreement problems arise in phonology. Consider, the example of homorganic nasal clusters (e.g., \textit{camp}, \textit{can’t}, \textit{sang}), where the nasal agrees with the following obstruent in place of articulation. We could express this agreement constraint with a set of rules of the form

(111a) homorganic-nasal-cluster $\rightarrow$ labial-nasal labial-obstruent  \hspace{1cm} \textit{labial case}
(111b) homorganic-nasal-cluster $\rightarrow$ coronal-nasal coronal-obstruent  \hspace{1cm} \textit{coronal case}
(111c) homorganic-nasal-cluster $\rightarrow$ velar-nasal velar-obstruent  \hspace{1cm} \textit{velar case}
Ideally, it should not be necessary to split these sorts of rules into cases; there ought to be ways to discuss agreement constraints in terms of features (e.g., *place*, *manner*, *voicing* in phonology and *person*, *number*, *gender* in syntax). This observation has led many researchers to introduce augmentations of various sorts (e.g., ATN registers [125], LFG constraint equations [55], GPSG meta-rules [36], bit vectors [29, 82]). To serve this purpose, we will introduce a simple language of lattice operations. In our opinion, this proposal seems to be easier to work with than many others.

5.1 An Introduction to Chart Parsing

Before we discuss the details of our own proposal, it might be useful to review some basic background material. The organization of this section strongly follows that of [37, 94, 99].

Let us define a very general context-free algorithm. A context-free parser takes as input a context-free grammar and a sentence and produces as output a chart\(^{78}\) of labeled phrases. A labeled phrase is a sequence of words delimited by two brackets and labeled with a category symbol. Let the triple \(<i, j, c>\) denote a phrase of category \(c\) spanning the words between the \(i^{th}\) position and the \(j^{th}\). For illustrative purposes these triples will be represented in a two dimensional matrix (i by j) as illustrated below for the sentence (112a). For example, the entry \({\text{NP}, \text{VP}}\) in Chart(2, 4) represents two analyses of the words between positions 2 and 4, namely \([\text{NP}, \text{flying planes}]\) and \([\text{VP}, \text{flying planes}]\). (More efficient representations will be discussed shortly.)

\[(112a) \text{ Input Sentence: } 0 \text{ They } 1 \text{ are } 2 \text{ flying } 3 \text{ planes } 4\]

\[(112b) \text{ Grammar:} \]

\[
\begin{align*}
\text{N} & \rightarrow \text{they} & \text{V} & \rightarrow \text{are} & \text{N} & \rightarrow \text{flying} & \text{A} & \rightarrow \text{flying} & \text{V} & \rightarrow \text{flying} & \text{N} & \rightarrow \text{planes} \\
\text{S} & \rightarrow \text{NP} \text{ VP} & \text{VP} & \rightarrow \text{V} \text{ NP} & \text{VP} & \rightarrow \text{V VP} & \text{NP} & \rightarrow \text{N} & \text{NP} & \rightarrow \text{AP NP} & \text{NP} & \rightarrow \text{VP} & \text{AP} & \rightarrow \text{A}
\end{align*}
\]

\[(112c) \text{ Chart:} \]

\[
\begin{array}{cccccc}
0 & \{\} & \{\text{NP,N,they}\} & \{\text{S}\} & \{\text{S}\} & \{\text{S}\} \\
1 & \{\} & \{\} & \{\text{VP,V,are}\} & \{\text{VP}\} & \{\text{VP}\} \\
2 & \{\} & \{\} & \{\} & \{\text{NP,VP,AP,N,V,A,flying}\} & \{\text{NP,VP}\} \\
3 & \{\} & \{\} & \{\} & \{\} & \{\text{NP,N,planes}\} \\
4 & \{\} & \{\} & \{\} & \{\} & \{\}
\end{array}
\]

---

\(^{78}\) Some authors prefer the term well-formed substring table (wfst) to chart.
If there is a complete parse of the sentence, the chart will have an S in the top-most right hand corner to represent the fact that the parser found an S spanning the words from before the first one (0) to after the last one (n). Otherwise, the sentence is rejected. There are no entries in the lower left half of the chart because there no phrases which end before they start. The diagonal entries correspond to phrases of zero words (eg. trace and other empty categories).

There are many well-known parsing algorithms that produce a chart like that above in time $O(n^3)$ (proportional to the cube of the number of words). One such algorithm is given below. It finds a phrase between positions $i$ and $j$ by picking a position $k$ in between and testing whether there are two phrases, one from $i$ up to $k$ and another from $k$ to $j$, that can combine according to the grammar. For example in the sentence, _They are flying planes_, the algorithm will determine that there is an S from 0 to 4 because there is an NP from 0 to 1 and there is a VP from 1 to 4 and the two phrases can combine according to the grammar rule $S \rightarrow NP \ VP$. The general entry in the chart is:

\[
(113) \quad chart(i, j) = \bigcup_{i < k < j} chart(i, k) \cdot chart(k, j)
\]

where $\alpha \cdot \beta$ combines phrases from $\alpha$ and $\beta$ according to the rules of the grammar. That is, it returns the set of phrases whose left daughter is from $\alpha$ and whose right daughter is from $\beta$. (For the present discussion we will assume that phrases have one or two daughters, or more formally, that the grammar is in Chomsky Normal Form [2].)\(^79\) This algorithm can be performed in $O(n^3)$ time by choosing all combinations of $i$, $j$ and $k$, each of which have $n$ possible values. (The multiplication step requires constant time, independent of the actual input words. It only depends on the grammar and hence it is often known as the grammar constant.)

\[
(114) \quad \text{for } j := 1 \text{ to } n \text{ do} \\
\quad \quad chart(i-1, j) := \{A \mid A \rightarrow \text{word}\} \quad \text{basis} \\
\quad \quad \text{for } i := j-2 \text{ downto } 0 \text{ do} \\
\quad \quad \quad chart(i,j) := \bigcup_{i < k < j} chart(i, k) \cdot chart(k, j) \quad \text{invariant} \\
\quad \quad \text{if } S \text{ is in chart}(0,n) \text{ then accept} \\
\quad \quad \quad \text{else reject}
\]

\(^79\) This Chomsky Normal Form assumption is imposed for expository convenience. There is a straightforward generalization of the foregoing which does not depend upon the assumption. See [82, pp. 11-14] for details.
This formulation of chart parsing is convenient for showing the parallelism between context-free parsing and matrix multiplication. This is an important result, originally due to Valient [122], which allows us to take advantage of advances in matrix multiplication algorithms, currently a very active area of research in computer science. Intuitively, the parallelism comes from a very strong similarity in invariants: (115a) is the invariant for chart parsing and (115b) is the invariant for multiplying two matrices $A$ and $B$ to produce a resulting matrix $C$.

\begin{align*}
(115a) \text{chart}(i,j) &= \bigcup_k \text{chart}(i,k) \star \text{chart}(k,j) \quad \text{invariant for parsing} \\
(115b) c_{ij} &= \sum_k a_{ik} \star b_{kj} \quad \text{invariant for matrix multiplication}
\end{align*}

Algorithm (114) is similar to several well-known algorithms dating back to the early 1960’s (e.g. Harvard Predictive Analyzer [71] and the Cocke-Younger-Kasami [2]), though these algorithms tend to enumerate the chart in slightly different orders. From a computational point of view, the enumeration order is of surprisingly little importance. Sheil [109] showed that the $O(n^3)$ time bound can be achieved by any algorithm that limits its search space to points in $<i, j, k>$-space. From this perspective, there is no difference between depth first enumeration of $<i, j, k>$-space, breadth first enumeration, best first, or even random order, for that matter.\textsuperscript{80}

The same parsing methods can be used to find syllable structure from an input transcription.\textsuperscript{81}

\begin{align*}
(116a) \text{Input Sentence:} & \quad _0 \overset{1}{f} 2 3 4 5 \\
(116b) \text{Grammar:} & \\
& \quad \text{syl} \rightarrow \text{(onset) rhyme} \\
& \quad \text{onset} \rightarrow \overset{1}{f} | s | z \\
& \quad \text{rhyme} \rightarrow \text{peak coda} \\
& \quad \text{peak} \rightarrow \overset{1}{f} | l
\end{align*}

\textsuperscript{80} This claim needs a slight qualification: it doesn’t hold if the equality relation in the invariant is replaced with an assignment statement because the former is associative while the latter is not. That is, every order of evaluating equality relations produces the same results, but this is not necessary true of assignment statements. Hence the equality in the algorithm really means equality and not assignment. (There is a recent trend in computer science to replace assignment statements with equality constraints [117], thus sidestepping a large number of ordering problems and similarly, in linguistics there is a trend to replace strict ordering of transformations with well-formedness constraints such as Chomsky’s conditions on binding, case, government, and thematic relations [16] or Bresnan-Kaplan’s completeness, coherence, and consistency [55].)

\textsuperscript{81} The input transcription need not be a flat string of segments, as it is in this case. The input could be a transcription lattice, with only minor modifications to the parsing methods.
5.2 Representation Issues

There is a rich literature discussing alternative representations of the chart. We find a proposal by Martin [82] to be particularly attractive. He suggested that the category symbol be factored out of the chart, by decomposing the chart into a sum of smaller matrices, one for each part of speech. For example, the chart in (116c) above could be decomposed into

\[(117) \quad \text{Chart} = \text{syl} M_{\text{syl}} + \text{onset} M_{\text{onset}} + \text{peak} M_{\text{peak}} + \text{coda} M_{\text{coda}} + \text{rhyme} M_{\text{rhyme}} + \ldots \]

where $M_{\text{syl}}$ is a binary matrix describing the location of syllables and $M_{\text{onset}}$ is a binary matrix describing the location of onsets, and so forth. Each of these binary matrices has a 1 in position $(i, j)$ if there is a constituent of the appropriate part of speech spanning from the $i^{th}$ position in the input sentence to the $j^{th}$ position. Thus, for example, $M_{\text{syl}}$, $M_{\text{onset}}$, and $M_{\text{rhyme}}$ are:

\[
\begin{array}{cccccccc}
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

(For historical reasons, we will occasionally refer to these matrices as lattices. This usage of the term lattice is...
technically incorrect because the matrices do not necessarily have a unique top and bottom).

5.2.1 The Chart Tends to be Sparse

These matrices tend to be very sparse (very few 1's) for a number of reasons. All of the lower diagonal entries are certain to be 0 because phrases don't end before they start. Furthermore, since the branching factor tends to be small and bounded (recall that there are rarely more than four active paths at any point in the utterance), it is unlikely that there would be more than a few edges leaving any particular node, and hence it is unlikely that the matrix would have more than a few non-empty sets on any particular row. Similarly, it is unlikely that there would be more than a few edges entering any particular node, and hence it is unlikely that there would be more than a few non-empty sets on any particular column of the matrix.

In principle, e (the number of edges or 1's) could be on the order of \( n^2 \) (the number of nodes or the number of rows in the matrix). The worst case, \( e = \frac{n(n-1)}{2} \), arises when all substrings of the input sentence are grammatical phrases. Consider, for example, a grammar like:

\[
(118a) \text{ rhyme } \rightarrow \text{ peak coda} \\
(118b) \text{ peak } \rightarrow \text{ segment}^* \\
(118c) \text{ coda } \rightarrow \text{ segment}^*
\]

In this case, for an input sentence of five segments, the rhyme matrix will be:

\[
\begin{array}{cccccc}
0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

82. For an explanation of this terminology, see the glossary.
83. We could have also used the simpler grammar rule, rhyme \( \rightarrow \) segment*, in order to prove the point. We use the more complex grammar with the ambiguity between where the peak ends and where the coda begins to insure that the chart will be dense. There is a condition in Earley's algorithm which doesn't add edges to the chart when they cannot be part of a complete parse. (We will discuss a similar condition in section §5.7.3.) This condition would keep the lattice sparse with the grammar rule, rhyme \( \rightarrow \) segment*, since the “usefulness” condition rules out most sequences of segments. With the ambiguous grammar, the usefulness condition is ineffective.
where all the upper diagonal entries are 1. There are a large number of 1's in this case because the grammar is prolific. Any string of segments is a grammatical rhyme (as long as its starting point precedes its end point). Fortunately, though, linguistically motivated syllable grammars tend to be more restrictive than (118), and therefore, \( e \) tends to be much smaller than \( n^2 \).

In a probabilistic framework, one could replace all of the 1's and 0's with probabilities. A high probability in location \((i, j)\) of the syllable matrix would say that there probably is a syllable from position \(i\) to position \(j\); a low probability would say that there probably isn't a syllable between \(i\) and \(j\). Most of what we plan to say for binary matrices will follow straightforwardly for probability matrices, with one major exception. Probabilistic matrices will tend to be dense (because many of our 0's will be assigned a small, but non-zero value). Thus, the efficiency optimizations to be discussed in the next section will not apply.

### 5.2.2 Taking Advantage of the Sparseness

The printer has been optimized to take advantage of the sparseness of the matrices. It will display one arc for each 1 in the matrix. For example, the rhyme matrix above will appear as:

\[
(119a) \quad \begin{array}{c|c|c|c|c}
0 & 1 & 2 & 3 & 4 & 5 \\
\hline
0 & 1 & . & . & . & . \\
1 & . & 1 & . & . & . \\
2 & . & . & 1 & . & . \\
3 & . & . & . & 1 & . \\
4 & . & . & . & . & 1 \\
5 & . & . & . & . & .
\end{array}
\]

(This display looks much more attractive on the Lisp Machine where the arcs are represented with inverse video rectangles and the labels appear in IPA notation.)

The internal representation is also optimized to take advantage of the sparseness of the matrices. Matrices are stored in adjacency format (abbreviated as \texttt{adimat}). For example, the rhyme lattice above would look like:

\[
(120) \quad ((1\ (2\ [i])\ (3\ [s]))) \quad ; \text{This}
\]
\[
(3\ (4\ [i])\ (5\ [Iz])) \quad ; \text{is}
\]

Adjacency coding saves space when \( e \) is much smaller than \( n^2 \), as it usually is for reasons discussed above. In example (120), the savings are extremely large; there are only 4 edges (rhymes), which is much smaller than 36 (the square of the number of segment junctures).
In addition to saving space, adjacency coding also makes it easier to iterate over the edge set when \( e \) is much less than \( n^2 \). Adjacency code enables iteration in \( O(e) \) time, as opposed to \( O(n^2) \) time in standard array format. This savings is important to the overall performance of the system because many of the basic operations tend to iterate over the edge set.

Unfortunately, there is a problem with adjacency coding: it is very expensive to access the entries in random order. In general, the system might have to iterate down most of a dimension in order to find a particular index. To alleviate this problem, the indices are stored in sorted order so that the accessing functions can stop once they have gone too far. In addition, there is a cache associated with each dimension to remember the location of the last reference. This helps operations that tend to access the same locations repeatedly. It also helps operations that tend to access locations in order, since the accessing functions can start searching for the desired index from where they last left off, and not have to start from the beginning every time.

5.3 A Parser Based on Matrix Operations

5.3.1 Concatenation and Union

In the previous section, we showed how the chart can be viewed as a sum of binary matrices. In this section, we will proceed to formulate a parsing algorithm which operates directly on these matrices. The parser takes as input a lattice (matrix) of phonetic segments and produces as output a matrix of syllables. All grammar rules are implemented as matrix to matrix operations. For example, the grammar rule

\[(121) \quad \text{syllable} \rightarrow \text{onset \ rhyme} | \text{rhyme} \]

is implemented by computing the onset matrix (in some way to be specified later) and computing the rhyme matrix (in some way) and then concatenating the results. Thus, the Lisp code for constructing the syllable matrix is:

\[(122) \quad (\text{setq syllable-lattice} \ (M + (M* \text{onset-lattice rhyme-lattice})) \text{\ rhyme-lattice})) \]

We are making use of the well-known reduction of parsing to matrix manipulations discussed above. Terminal and non-terminal symbols (e.g., syllable, onset, rhyme) are implemented as matrices and productions are implemented as operations on matrices (e.g., \( M* \), \( M+ \)).
Let us return to the example discussed previously in section §1.5.1.

(123) |.D.|.|s.|.|I.|.|z.|.|D.|.|x.|.|s.|.|iY.|.|b.|.|iY.|.|E.|.|s.|.|q1Y|.v.|.|n.|.|.|.G.| This is the C B S Evening

In this example, the phonotactically and allophonically well-formed onsets are:

(124) |.D.|.|s.| |.|D.|.|s.|.|b.| |.|n.| This is the C B S Evening

and the well-formed rhymes are:

(125) |.|.|.|I.| |.|x.| |.|iY.| |.|iY.|.|E.| |.|q1Y| |.|.| |.|s...|.|Iz...| |.|xs...|.|iYb...| |.|Es...|.|q1Yv...| |.|G...| This is the C B S Evening

The matrix of well-formed syllables (repeated from 40 above) is formed by concatenating the onset matrix with the rhyme matrix and adding the result to the rhyme matrix. Concatenation (M*) is implemented with standard matrix multiplication\(^{84}\) and addition (M + ) is implemented with standard matrix addition.

(126) |.|D|.|sI...| |.|Dx...|.|siY...|.|b1Y...|.|E.| |.|q1Y| |.|n|.| |.|s|.|D|s...|.|I.| |.|Dxs...|.|iY.| |.|iY.|.|Es...|.|q1Yv...|.|n|G...| |.|.| |.|s...|.|Iz...| |.|x.|.|siYb...| |.|n|G...| This is the C B S Evening

5.3.2 Optionality

Pure context-free rules may be somewhat awkward for certain applications. This can be alleviated (to some extent) by introducing certain notational conveniences, informally known as syntactic sugar.

Suppose that we wanted to write grammar (127a) in the more convenient form (127b).

(127a) syllable → onset rhyme | rhyme
(127b) syllable → (onset) rhyme

---

84. Intuitively, we say that there is a syllable from position i to position j just in case there is an onset from position i to some position k in between, and there is a rhyme spanning the rest of the from position k to j. In other words, the (i,j) location is the syllable matrix is formed by multiplying the \(i^{th}\) row of the onset lattice with the \(j^{th}\) column of the rhyme lattice.
We could say:

(128) (setq syllable-lattice (M* (Mopt onset-lattice) rhyme-lattice))

Mopt is not formally necessary. Expression (128) is equivalent to:

(129) (setq syllable-lattice (M* (M + onset-lattice identity-lattice) rhyme-lattice))

where the identity lattice is the identity element under M* (a matrix with 1's along the diagonal and 0's elsewhere). This expression can be shown to be algebraically equivalent to (122). The proof is straightforward, assuming only the distributive law and identity relations.

Internally, Mopt is often removed by macro expansion for efficiency's sake. In particular, M* is defined so that:

(130) (M* (Mopt <sexp_0>) <sexp_1> ... <sexp_N>)

will expand at compile time^85

(131)  (let ((temp (M* <sexp_1> ... <sexp_N>)))
        (M + temp (M* <sexp_0> temp)))

Similarly, it is possible to remove Mopt at compile time with no loss in efficiency when it is in final position:

(132)  (M* <sexp_0> <sexp_1> ... (Mopt <sexp_N>))

When Mopt is found in any other position, (Mopt <sexp>) is implemented by adding the identity matrix to <sexp> as discussed above. We have found this to be a fairly efficient and elegant way to implement optionality.

^85. Internally the variable temp is implemented with a gensym variable so as to avoid any possible name clash.
5.3.3 Transitive Closure

Transitive closure is used for implementing phrase structure rules with Kleene stars. So for example, if we wanted to define an inflectional affix to be a sequence of /t/ and /s/, or a sequence of /d/ and /z/, we could write the following phrase structure rule:

\[(133) \text{inflectional-affix} \rightarrow (\{t\}_s)^* \mid (\{d\}_z)^*\]

This could be implemented as:

\[(134) \text{setq inflectional-affix-lattice} \]
\[
(M + (M** (M + (phoneme-lattice #/t)) \]
\[
(phoneme-lattice #/s))) \]
\[
(M** (M + (phoneme-lattice #/d)) \]
\[
(phoneme-lattice #/z))))\]

(Phoneme-lattice selects the edges in the input segment-lattice that are allophones of the given phoneme.)

There are two versions of transitive closure: \(M^{**}\) and \(M^{++}\). The first includes null transitions and the second excludes them. Thus

\[(135) (M^{**} \langle \text{sexp} \rangle) = (M^{opt} (M^{++} \langle \text{sexp} \rangle))\]

There are well-known algorithms for computing the transitive closure of a graph or a lattice. \(M^{**}\) and \(M^{++}\) are about as efficient as \(M^*\). \((M^+\) is much faster). Both transitive closure and concatenation require about \(n \cdot e\) time, where \(n\) is the number of nodes in the lattice and \(e\) is the number of edges. In the worst case, \(e\) will be approximately \(n^2\). Thus, the computation can require \(n^3\) time. Fortunately, though, the worst case does not arise very often (if at all) in our domain for reasons discussed above.

Theoretically, there are some more efficient algorithms for computing these operations. It is known how to multiply matrices (which can be shown to be a generalization of concatenation and transitive closure) in approximately \(n^{2.66}\) time. However, these algorithms may not be practical for our purposes because they have some very large constant overheads, and because the constraints on the lattices that we are dealing with are such that the average case is much better than the theoretical worst case. For these reasons, we have not found it to be worthwhile to implement these more complicated algorithms. The straightforward \(n \cdot e\) algorithms work well enough for our purposes.
Any finite-state grammar could be implemented in this way with just three matrix operations: \(M\ast, M+, \) and \(M^{**}\) (transitive closure). For speech applications, finite state grammars are adequate since grammars of syllable and stress structure do not seem to require additional generative capacity.\(^86\)

5.4 No Recursion

The way that lattices are combined is slightly less general than Earley's algorithm. We cannot easily represent a recursive context-free grammar such as:

\[
(136) \quad \text{sexpr} \rightarrow \text{NIL} \mid (\text{sexpr})
\]

which generates sentences of the form: \( (^{n} \text{NIL} )^{n} \). It is easy to prove that this grammar is strictly context-free (i.e., not finite state). Therefore, it cannot be characterized completely with just the three operations presented thus far: union, concatenation and transitive closure. If, for example, we attempted to implement this grammar with the lisp code:

\[
(137) \quad \text{setq sexpr-lattice } (M + \text{NIL-lattice} \\
(M* \text{ open-parenthesis-lattice} \\
\text{sexpr-lattice} \\
\text{close-parenthesis-lattice}))
\]

we would discover that the lisp variable \text{sexpr-lattice} is unbound. Lisp variables have to be bound before they are referenced. There is no way to define \text{sexpr-lattice} in terms of itself without introducing a mechanism for processing recursion. Earley's algorithm, of course, has such a mechanism. We have not found it necessary to introduce it in our parser since syllable grammars tend to be non-recursive. If we had found it useful to introduce recursion, we would have been motivated to implement the full Earley algorithm. (We have avoided this move in order to enhance debugging capabilities.)

Many (perhaps all) formulations of metrical structure assume recursion, but only a very restricted type of recursion. The recursion is typically strongly biased toward right branching. Right branching can often be replaced with iteration (transitive closure) as discussed in [19] and references therein.\(^87\) In short, we can replace a recursive grammar like:

---

\(^86\) Although, for natural language applications, finite state grammars are generally considered to be inadequate. In [19], we have attempted to counter this argument.

\(^87\) Recursion removal such as this is similar in many respects to a well-known compiler optimization called \textit{tail recursion}.
(138a) feet → foot | foot feet

with an iterative formulation such as:

(138b) feet → foot*

It is also possible to remove recursion in certain left recursive cases, which we will not discuss here. In short, the non-recursive assumption is not too much of an imposition because we can often use recursion removal optimizations to reformulate a recursive grammar into an iterative one. There are, of course, grammars where recursion removal cannot be applied. The best examples (and perhaps the only examples) are strictly context free grammars such as (136). However, we have not found these grammars to be necessary (or even to be convenient) for describing syllable and metrical structures. For this reason, we feel comfortable without recursion. In exchange for recursion, we obtain a much simpler control structure. This simplicity greatly enhances our ability to construct useful debugging aids, display routines, and diagnostic tools.

5.5 Order of Evaluation

As we have been using the lisp setq operation above, it is important that a lattice variable such as syllable-lattice be bound before it is referenced. As we have discussed in the previous section, this assumption excludes recursive grammars. In addition, though, it requires us to order the rules in our grammar so that constituents will be computed in a bottom-up way. It may not be very convenient to order the grammar in this way.

We have implemented a memoization scheme which avoids the inconvenience of having to order the definition of the grammar. We will no longer use setq to define lattices and lisp variables ending with -lattice to reference lattices. Instead, lattices will be defined with the form:

(139) (def-lattice <lattice-name> <body>)

Def-lattice tells the system how to compute a lattice of the name <lattice-name>. Def-lattice is implemented as a macro. The expansion of

(140a) (def-lattice syllable (M* (lattice onset) (lattice rhyme)))
is:

(140b) (defun (syllable create-lattice) nil
      (M* (lattice onset) (lattice rhyme)))

In this way, we can create a new syllable lattice by applying the create-lattice function property on the symbol syllable.

The lattice function is used for referencing lattices. If the lattice has not been computed yet, it will create a new one by calling the create-lattice property. The resulting lattice will be saved away in a memoization table. Very often, though, the lattice function does not have to compute the lattice since it has already been computed and memoized. In this case, the lattice function can simply fetch the appropriate lattice from the memoization table, instead of calling the create-lattice function.

With this memoization scheme, it is no longer necessary to define the grammar in any particular order. If the parser (i.e., the lisp interpreter) should encounter a use of a lattice that has not been computed, the parser will find its definition and compute it recursively and then continue from where it left off. In this way, the lisp interpreter will manage the order of evaluations for us, and we need not be concerned with it.

In summary, we have seen in this section that our parser is a simplified formed of a chart parser. The lattice representation provides a data structure which is very similar to a chart. The difference between our parser and a chart parser becomes evident when we consider recursion. We have purposely excluded recursion because we didn’t need it (our grammars are only finite state) and we felt that eliminating it would enhance debugging capabilities (see section §5.8). However, in eliminating recursion, we do not want to impose an ordering relation on the definition of the grammar. If we used the lisp setq operation, as we did in the beginning of this chapter, it would be necessary to define the grammar in a purely bottom-up way, a possibly awkward and inconvenient imposition. In this last subsection, we provided a more sophisticated memorizing scheme that solves this problem.

5.6 Feature Manipulation

So far, our parser is much like any other finite-state parser that one might find in the literature. As it currently stands, this parser will have just as much trouble dealing with agreement facts as any other parser. Recall the example of homorganic nasal clusters mentioned in the introduction to this chapter. With ordinary unaugmented phrase-structure rules, it seems to be necessary to introduce three cases in order to express the
place assimilation constraint.

(141a) homorganic-nasal-cluster → labial-nasal labial-obstruent
(141b) homorganic-nasal-cluster → coronal-nasal coronal-obstruent
(141c) homorganic-nasal-cluster → velar-nasal velar-obstruent

labial case
corononal case
velar case

With the mechanisms introduced thus far, the three way split also seems to be unavoidable.

(142) (def-lattice homorganic-nasal-cluster
       (M + (M* (lattice labial-nasal) (lattice labial-obstruent)))
       (M* (lattice coronal-nasal) (lattice coronal-obstruent))
       (M* (lattice labial-nasal) (lattice velar-obstruent)))

labial case
corononal case
velar case

Ideally, it should not be necessary to split homorganic nasal clusters into three cases. We would really like to say that homorganic nasal clusters are nasal clusters subject to place assimilation. In our language of matrix operations, we can say just exactly that. (Let M& (element-wise intersection) read subject to.)

(143) (def-lattice homorganic-nasal-cluster
       (M& (lattice nasal-cluster)
           (lattice place-assimilation)))

Nasal-cluster and place-assimilation are defined as:

(144a) (def-lattice nasal-cluster
        (M* (lattice nasal)
            (lattice obstruent)))

(144b) (def-lattice place-assimilation
        (M + (M++ (lattice labial))
            (M++ (lattice dental))
            (M++ (lattice velar))))

For a second example of M&, consider the inflectional affix (plural and tense marking). Many linguistic treatments would define an inflectional affix as a sequence of coronal consonants subject to voicing assimilation. With M&, this could be implemented in a very natural way:
(145) (def-lattice inflectional-affix
   (M& (M ++ (lattice coronal))
       (lattice voicing-assimilation)))

where voicing assimilation contains all portions of the input lattice that agree in voicing. It could be constructed with the lisp code:

(146) (def-lattice voicing-agreement
   (M + (M ++ (lattice voiced))
       (M ++ (lattice unvoiced))))

Without M&, we would have to introduce two cases in order to define the inflectional affix:

(147) (def-lattice inflectional-affix
   (M + (M** (M + (phoneme-lattice #/t)
                   (phoneme-lattice #/s)))
       (M** (M + (phoneme-lattice #/d)
                   (phoneme-lattice #/z)))))

M& seems to be well suited for describing agreement constraints (e.g., place and voice assimilation in speech and person/number agreement in syntax). It allows us to nest conjunctive constraints (e.g., M&) outside of disjunctive constraints (e.g., M+). The pure phrase-structure rule formalism has no attractive way to accomplish this. For some applications, M& may be much more convenient than ordinary phrase-structure rules. Obviously, though, M& is not strictly necessary, since all of these agreement facts can be expressed without it.

5.7 Additional Lattice Operations

5.7.1 Over-generate and Filter

We only need the three operations, concatenation, union, and transitive closure, in order to specify any finite state grammar; other operations such as M& are added purely for convenience. M& provides the grammar with an ability to over-generate and filter. We have also implemented a subtraction operation (M −). Both M& and M − filter a matrix by a set of other matrices. Intersection copies the edges from the first matrix argument that are also in all of the other matrices. Subtraction copies the edges from the first matrix argument that are not in any of the other matrices.
The ability to over-generate and filter provides the grammar writer some additional freedom in selecting the most efficient formulation of the grammar. In the previous section, we have discussed a number of examples of M&. M— is also very convenient for certain applications, though it has a serious theoretical problem. Suppose, for example, that we wanted to say that an onset is a non-flapped stop.\(^8\)

(148) (def-lattice onset (M— (lattice stop) (lattice flap)))

This is a very natural formulation, and it almost works. The problem arises when the parser is given an input lattice such as:

\[ \begin{array}{cccc} \text{\texttt{t}} & \text{\texttt{t}} & \text{\texttt{z}} & \text{\texttt{t}} \\ 0 & 1 & 2 & 3 & 4 \end{array} \]

where a flap and a non-flap segment span across the same region. Rule (149) will exclude the region from 1 to 2 from the onset lattice because there is a flap there. This turns out to be incorrect because the /t/ ought to be considered a possible onset. For this reason, subtraction should be avoided when this type of ease might arise.

There is a similar well-known problem for intersection. Consider an input lattice of the form:

\[ \begin{array}{cccc} \text{\texttt{k}} & \text{\texttt{t}} & \text{\texttt{z}} & \text{\texttt{z}} \\ 0 & 1 & 2 & 3 & 4 \end{array} \]

The region from 2 to 4 should not be considered to be an inflectional affix. Unfortunately, the grammar outlined above will consider the region from 2 to 4 to be an inflectional affix because it is coronal (e.g., /tz/) and it is unvoiced (e.g., /tk/). M& does not check to see that both of its arguments analyze the input in the same way. This problem with M& is not as severe as the problem with M—; it is better to err in the direction of over-accepting as M& does, than to err in the direction of over-rejecting as M— does.

In some ways, the problem with M& is similar to the well-known problem for bit vector systems with a sentence like **Flying planes can be dangerous.** Flying planes can be singular (as a gerund) or plural, and can be dangerous can be singular or plural. Note, though, that there are only two interpretations. A bit vector system would find four interpretations. Both interpretations of flying planes are grammatical because there is an

---

88. This definition is oversimplified for the sake of exposition.
interpretation of the predicate which is consistent, and both interpretations of the predicate are grammatical because there is an interpretation of the subject that is consistent. There is no check that all interpretations of the subject are consistent with all interpretations of the predicate.

Despite these problems with over-generating and filtering, we have found these operations to be useful and worthy of further investigation.

5.7.2 Context Primitives

We have also provided mechanisms for specifying contexts. These mechanisms do not increase the generative capacity, though they provide simple and efficient formulations of certain commonly performed computations. The context predicates are: adjmat-copy-when-before, adjmat-copy-unless-before, adjmat-copy-when-after, adjmat-copy-unless-after, adjmat-copy-when-starts-with, adjmat-contains, and adjmat-doesnt-contain. The specifications of these operations should be clear from their names. For example, a word final /s/ could be specified as

(151) \((\text{adjmat-copy-when-ends-with (phoneme-lattice #/s)}\text{)}\text{)}\text{ (lattice word)\)}\\

and an /s/ before a word could be specified as:

(152) \((\text{adjmat-copy-when-before (phoneme-lattice #/s)}\text{)}\text{ (lattice word)\)}\\

These two examples select almost the same lattices except when the utterance ends with an /s/. An utterance final /s/ will be included in (151) but not in (152).

---

89. These contexts can be encoded into the states of the finite state machine FSM along the lines suggested in [51, 52, 53].
90. Adjmat is an abbreviation for adjacency matrix.
5.7.3 Garbage Collection

One last operation, adjmat-gc, is provided to weed out edges that are not part of a complete path through the lattice. We like to view this operation as "garbage collecting" [91] (hence the name adjmat-gc). For example, garbage collecting the syllable lattice

\[
\begin{array}{cccc}
|D| & s & .I & .E \mid q & Y & n \mid \\
\end{array}
\]

produces

\[
\begin{array}{cccc}
|D| & s & .I & .E \mid q & Y & n \mid \\
\end{array}
\]

Many of these edges that are excluded by gc-ing would not have been constructed in the first place if we employed a chart parser like Earley’s algorithm [30] which has a prediction mechanism to focus attention on edges that can fit into path through the lattice (complete parse of the sentence). Earley’s algorithm essentially interleaves the GC loop with the rest of the program, providing certain improvement in performance. However, we have decided to keep the GC loop separate in order to enhance debugging capabilities. We have observed that GC-ing tends to amplify an error in the program, so that after GC-ing, it is very hard to localize the problem. In particular, if there is a small section of the utterance that cannot be parsed (due to some error in the grammar or in the transcription, for example), the GC algorithm will eventually throw out all the other edges in the lattice on the grounds that they aren’t part of a complete path through the lattice. We have found that an easy way to locate the trouble spot is to check the pre-GC lattice for a section with no edges.

---

91. A computational term for a procedure that reclaims storage that can no longer be referenced. The procedure marks all memory cells that can be referenced by marking all cells on the stack and then recursively marking all cells that they reference and all cells that they reference and so on. The remaining cells are swept away and placed on the free list. This algorithm is also called a mark-sweep.
92. GC is a common abbreviation for garbage collection.
5.8 Debugging Capabilities

The parser was designed this way in order to enhance debugging capabilities. We could have used Earley's algorithm, but our previous experience with Earley's algorithm has convinced us that it can be very difficult to develop grammars in that formalism. If something goes wrong and the parser fails to find a parse or finds too many parse trees, it is very difficult to track the problem down. Most implementations of Earley's algorithm have very crude debugging tools. They typically have mechanisms for printing a few parse trees, but this won't help us when the parser isn't finding any parse trees. Printing is also fairly useless in the case where the parser is finding thousands of parse trees. There is no way that a person is going to look at a few thousand parse trees.

Some implementations of Earley-like parsers provide certain limited debugging mechanisms (e.g., tracing packages, break points, single stepping). Most of these mechanisms trigger on primitive parsing operations (e.g., references to the chart), and consequently, most of these mechanisms tend to be too low level to tell us what went wrong on a global scale. As a consequence, program development proceeds in a kind of "batch-mode" fashion. Many grammar writers (e.g., Bill Martin) do not use these debugging tools at all. Instead they run their parsers on large corpuses in batch mode, and then compare the output listing with the source code. It might be possible to encourage a more interactive style of grammar development, if debugging tools were more sophisticated.

In our system we provide a number of basic tools like those above. (Actually, many of these tools come with the basic Lisp Machine system.) In addition, we have found that it is important to be able to display a large amount of relevant information in an interactive way. The matrix view of parsing makes this relatively easy to do. At any point, the grammar writer can request to see any matrix with a one line command. For example, if the grammar writer types:

(155) (admat-print (lattice syllable))

the machine will compute the syllable lattice (or retrieve it from the memoization table) and display it on the screen as best as it can. (On the Lisp Machine, the edges will be printed in the IPA font against an inverse video background.)

We have also found it very useful to be able to look at a lattice before and after a suspicious operation. Suppose, for example, the programmer wanted to see the syllable lattice before and after garbage collecting (an operation that often magnifies a small problem into a major one for reasons mentioned above). This can
be accomplished with the three line expression:

(156) (adjmat-print-matrices *graphics-port* *echo-port*
      (lattice syllable) "before"
      (adjmat-gc (lattice syllable)) "after")

This will display the syllable lattice before and after garbage collecting in the *graphics-port*. The string before will appear above the first lattice and the string after will appear above the second lattice.

In this way, the grammar writer can look at lattices before and after literally any operation. Imagine, for example, that the grammar writer wanted to see sylparts before and after they were combined into syllables. He could say something like:

(157) (adjmat-print-matrices *graphics-port* *echo-port*
      (lattice onset) "onset"
      (lattice rhyme) "rhyme"
      (M* (Mopt (lattice onset)) (lattice rhyme))) "syllables")

This would display the onset lattice below the word onset, followed by the rhyme lattice below the word rhyme, followed by the product under the word syllables. This mechanism makes it fairly easy to determine where a problem might be coming from. One simply displays the lattices before and after the questionable operation and makes sure that they are doing what they ought to be doing.

In short, the lattice implementation of parsing enhances debugging capabilities because there are no side-effects; all intermediate results are available (in the memoization table) for immediate interactive inspection. At any point in the processing of an input string, the programmer can enter a break point and ask the system to display all of the relevant lattices. In this way, it is relatively easy to find bugs in the grammar, in a highly interactive programming environment. In other systems, one is often forced to trace through large listings of irrelevant steps in batch mode, as many of us know all too well.

---

93. *graphics-port* is bound to a window on the lisp machine, the top half on the screen on Maclips that support split screen, and the TTY on other systems.
5.9 Summary

In summary, we have outlined a novel parsing design. The chart is modeled as a sum of sparse matrices. Each step maps a number of input matrices into an output matrix. We have provided a large number of matrix functions (e.g., $M\ast$, $M+$, $M^{++}$, $M\ast\ast$, $M\&$, $M-$, adjmat-copy-when-before, adjmat-copy-when-after, adjmat-copy-unless-after, adjmat-copy-when-starts-with, adjmat-contains, adjmat-doesnt-contain, adjmat-gc). Examples of these functions can be found in the sample grammar presented in Appendix IV.

This programming style, rich in lattice computations, is reminiscent of the programming language APL. This style has the major advantage that the code tends to be very short and readable because a few straight line lattice computations can perform the same task that might require loops in a more traditional programming style. The programming style is amenable to a highly interactive programming environment. The standard objection to this way of coding is efficiency. Lattice operations tend to produce considerable numbers of unnecessary temporary lattices to hold intermediate results. A few efficiency optimizations (e.g., memoization) have been implemented. More could be added if necessary.

From a computational linguistics perspective, our matrix view of parsing is intriguing in its focus. We tend to focus on the chart, whereas most discussions of parsing are concerned with parse trees. In a sense, there are no parse trees in our parser. They could be constructed from our matrices, but we generally don’t bother to do so. In an old version of the parser, the user could select an edge from a matrix and display some or all of the parse trees that the edge dominates. For example, if the user selected the edge [NP $flying$ $planes$], the machine would display the parse tree [NP [A $flying$] [N $planes$]]. Although this display makes for an impressive demonstration (especially on the Lisp Machine which supports fancy graphics), we tend not to use parse trees because they don’t display what we are interested in, especially when the grammar is broken. Parse trees are useless when the parser can’t find a parse, or when it finds too many.

So far, we have motivated the usefulness of our segmental parser and discussed some of the implementation details. Let us now turn to some of the issues involved in formulating an appropriate grammar.
6. Phonotactic Constraints

Many speech recognition devices operate with an inadequate model of phonotactic constraints. In particular, most recognition devices incorporate a model of word structure of the form:

(158a) word → syllable*
(158b) syllable → initial-cluster vowel final-cluster

This model assumes a list of permissible initial- and final-clusters. How do we estimate the set of permissible clusters? It is widely assumed that the set of syllable initial-clusters are identical to the set of word initial-clusters, and that the set of syllable final-clusters are identical to the set of word final-clusters.

(159) "Delimitation of phonological syllables may be performed by the method of permissible initial and final clusters, set up by Hjelmslev (1936, p. 52) and developed further by, e.g., Kurylowicz (1948, p. 83ff). According to this method English words like inhere, husband, subscribe, anxious may be syllabified /ɪn-ɪhər/, /hʌzd-ənd/, /səd-ə skrai/ɪb/, /æn-ək-sət/, since these divisions are the only ones yielding syllable-initial and -final consonants and consonant clusters which are permitted initially and finally in: words." [26, p. 16]

The permissible initial and final clusters method, though, is inadequate for two reasons. First, it will be necessary to add some more constraints (e.g., the maximal onset principle and stress resyllabification) so that words like extra are assigned just one syllabification /ek-strə/, not three: /ek-strə/, */eks-strə/ and */ekst-strə/.

We will discuss the maximal onset principle and stress resyllabification in the next chapter.

Secondly, though, the set of syllable final clusters is far more constrained than the set of word final clusters. Note, for example, if any word final cluster could end a syllable, then we might expect to find a word final cluster like /rnt/ to appear in word medial position followed by another syllable. But we do not find sequences of consonants like /rntstr/ in word medial position. We will see in this chapter that sequences of six consonants can be excluded for principled reasons. In practice, one rarely finds sequences with as many as four consonants within the same morpheme. Phonotactic constraints such as this might improve the performance of a speech recognition device in many ways. In particular, these constraints could lead to more

---

94. Davidsen-Nielsen [26] mentions that O'Connor & Trim (1953) suggested that statistical methods may provide a useful supplementary tool for scoring competing syllabifications.
95. Long sequences of consonants almost always cross a morpheme boundaries (e.g., instructive, explanation, thanksgiving). The only morpheme with four sequential consonants is monstratry, as far as we know.
compact representation of lexical entries, and more efficient search procedures for matching portions of the utterance against the lexicon.

According to Kiparsky’s review paper of recent developments in suprasegmental phonology [57], there are three types of phonotactic constraints on syllable structure (excluding morpheme final syllables for the moment):

(160a) **Length**: there are at most three consonants before the vowel, and two afterward
(160b) **Sonority**: the relative prominence or sonority (Jespersen) must decrease from the nucleus towards the margins
(160c) **Idiosyncratic systematic gaps**: English happens not to allow certain possibilities, e.g. voiced stops after nasals or [u]. even though these meet the previous two conditions.

we might add two more:

(160d) **voicing assimilation**: In general, two adjacent consonants must agree in voicing.
(160e) **place assimilation and dissimilation**: In some contexts (e.g., nasalized codas), two adjacent consonants must agree in place of articulation. In other contexts (e.g., onset clusters), two adjacent consonants must disagree in place of articulation. Hence the “ungrammaticality”\(^{96}\) of */tl/, */dl/, */pw/ and */bw/ in onset position.

Each type of constraint has many intricate caveats, some of which will be discussed more carefully in what follows.

6.1 The Affix Position

All three of these constraints can be violated in morpheme final syllables. Thus, in morpheme final position, we have exceptions to:\(^{97}\)

(161a) **length restrictions**: wild, world, paint, exempt, lounge, texts, (thou) estrangedst
(161b) **sonority profile**: apse, axe, bets, width, depth, paths, laughs
(161c) **idiosyncratic restrictions**: bind, sand, mend, wood, hood, could, should (recall that English happens

---

96. The status of these clusters is controversial.
97. These examples are borrowed from Kiparsky [57]. Kiparsky credits the *estrangedst* example to Jane Simpson, noting that it is probably a record for English (six consonants in the final cluster). Indeed, we have not found any examples with more than three affix consonants in our computer readable dictionaries [84, 85].
not to permit voiced obstruents after nasals or [u])

(161d) place assimilation: seemed, seems, sings, hanged (nasals usually agree with the following stop in English).

In order to account for these exceptions, many analyses introduce a morpheme final affix position (also called the appendix or extrametrical segments). This affix position was originally intended to hold regular inflectional affixes (e.g., plural marking /s, z/ and past tense marking /t, d/), though it has been generalized to apply whenever there is an apparent violation of length, sonority, idiosyncratic gaps or place agreement. Thus, for example, the /d/ is analyzed as an affix in both fined and find,\(^98\) even though the latter /d/ is not a regular inflectional affix. The affix position has been generalized further yet so that in principle it can hold any number of coronal consonants subject to voicing assimilation.

In practice, though, one finds a much more restricted array of possible affixes. There are no attested cases with more than three consonants (except for estrangedst, which we will ignore).\(^99\) Moreover, if there are three consonants, the middle one will be a stop or /θ/ and the outer two will be fricatives. If there are only two consonants, one of them will be a stop and the other will be a fricative.\(^100\) Thus, the only affixes that might be found in practice are:\(^101\)

\[(162a)\text{s, t, z, d, }\theta\]
\[(162b)\text{st, ts, dz, zd, }?\theta, \theta\text{s}\]
\[(162c)\text{sts, }?\text{dz, }s\theta\text{s}\]

Actually, it is possible to construct a syllable grammar with a much shorter list of affixes. We will not go into the arguments here, because it is not clear that they are very important for our purposes.

\(^{98}\) Note that find violates the length constraint since it has three post-vocalic consonants: /yd/.
\(^{99}\) We suspect though that estrangedst has only three morpheme final consonants, at least in modern English where most speakers would probably insert a schwa between the /ə/ and the /d/.
\(^{100}\) There might be an account for this alternation between stops and fricatives at the phonetic level. Obviously, if there were two fricatives or two stops in a row, they would geminate into one. But, why must sequences of three coronal consonants start with an /s/? Perhaps there is a constraint that limits the affix position to a single stop constriction. The /s/ before the stop arises from co-articulation with the stop gap, and the /s/ afterward arises from co-articulation during the release.
\(^{101}\) Question marks are placed in front on the cases that have not been attested.
6.2 The Length Restriction

Excluding morpheme final affix clusters, there are at most two post-vocalic consonants. This explains why we do not find morpheme internal sequences with more than five consonants in a row such as */mrntsthr/. In particular, we do not find post-vocalic clusters such as /mrnt/, /rsrp/, /lymph/, /lymph/, and so forth, except in morpheme final position (e.g., /mrnt, first, find, sixths/). In pre-vocalic position, we do not find more than three consonants, and if there are three, the first one must be an /s/. For this reason, and others that we will come to, a number of authors (e.g., [105]) model /sp/, /st/, /sk/, and other /s/-clusters as constituents. This enables us to say that there are at most two constituents before the vowel, and two afterward.

The length facts can be encoded into context-free rules in a straightforward way. Many analyses (e.g., [57, 105]) accomplish this by decomposing syllables into smaller units, an onset and a rhyme. The rhyme is further decomposed into a peak and a coda. For example, a word like strike would be analyzed as:

\[(163) \text{ syllable } \text{ onset } \text{ str } \text{ rhyme } \text{ peak } a^Y \text{ coda k } \]\n
This decomposition could be formulated within a simple phrase-structure grammar as:

(164a) syllable → onset rhyme

(164b) rhyme → peak coda

With this decomposition, it is fairly straightforward to capture the length facts, by restricting the onset to at most 3 consonants (or 2 constituents), and the peak and coda to a total of at most 2.\(^{102}\)

We would, of course, hope to find that onsets and codas capture a number of other generalizations in addition to the length facts. We have have good reason to suspect that this is indeed the case, and that there are some other interesting constraints on the constituents (e.g., sonority, place and voicing assimilation and dissimilation). In particular, we have observed that 99% of the possible strings of consonants of length 0, 1, 2, or 3 are not found in onset position in our computer readable dictionaries.\(^{103}\) These dictionaries could

\(^{102}\) This second stipulation is somewhat awkward to formulate in our framework. It would be very straightforward to put both consonants in the coda, or one in each place, but in our framework, neither of these options are available to us. We want to be able to put some consonants in the peak and some in the coda for reasons that will become apparent.

\(^{103}\) How many sequences are there of three or less consonants? Assume there are 21 consonants (6 stops, 8 fricatives, 3 nasals, and 4 liquids and glides). Hence there are \(21^3\) sequences of three consonants + \(21^2\) sequences of two consonants + \(21^1\) sequences of one consonant + \(21^0\) sequences of no consonants. This totals to 12,144 possible onset clusters, of which, only about 70 (1.34%) are found in practice.
probably be stored much more efficiently if a few of these constraints could be captured in the lexical representation. This is just one of the many bottom-line benefits mentioned above for capturing generalizations.

6.3 The Sonority Hierarchy

The sonority constraint is the second of Kiparsky’s phonotactic constraints on syllable structure.

(165) **Sonority**: The relative prominence (vowel-li' ness) must decrease from the nucleus towards the margins.

In other words, the center of a syllable will contain a vowel, and moving away from the center toward the edges of the syllable, we will find less and less vowel-like consonants. More precisely, sonority imposes a total order on phonemes:

(166a) stops < fricatives < nasals < liquids < glides < vowels

The sonority order must be strictly decreasing as one moves away from the center of a syllable toward its edges.

6.3.1 Exceptions to the Sonority Hierarchy

There are three classes of exceptions to the sonority constraint:

(167a) In English, /s/ doesn’t fit into the hierarchy. It appears to be less sonorous that a stop in words like *stress* and *its*, but more sonorous than a stop in many other words such as *most*. Hence, there is an ordering paradox; /s/ is both more and less sonorous than /t/.

(167b) In just a few words, such as *Carlson*, there is an ordering relation between the two liquids /r/ and /l/. Sonority, being a strict relation, should not permit an element of one class (e.g., /r/) to be related to another element of the same class (e.g., /l/).

(167c) Morpheme final syllables (as in *sixths*) can end with several coronals. This is yet another case of a reflexive ordering relation, which is disallowed in strict order. This time the reflexive ordering relation is between /s/ and /θ/ and between /θ/ and /s/.

We can ignore the third problem now, because it can be attributed to the affix position. The first two problems, though, elicit serious faults in the sonority hierarchy. A modification will be necessary. Two
proposals seem promising. Either, one can extend the sonority hierarchy as follows:

(168) s < stops < fricatives < nasals < r < l < glides < vowels

and then say something special about the few exceptions that remain (e.g., /sp/, /st/ and /sk/). We will not pursue this approach because the extension to the sonority hierarchy is unmotivated, and because /s/ clusters will still have to be treated as exceptions.

An alternative approach, and one that we are more sympathetic with, allows some segments to cluster together into constituents before applying the sonority hierarchy. For instance, we could say that /sp/, /st/ and /sk/ are constituents and thus immune to the sonority principle. This approach is becoming more popular in the literature. See Selkirk [105] and Fujimura and Lovins [35] for two recent treatments.

We now have two reasons for considering /s/-clusters to be constituents. First, /s/-clusters are the only exceptions to the principle that there are at most two consonants before and after the vowel, and secondly, /s/-clusters are one of the few cases which systematically violate the sonority principle. We will soon see two more motivations for considering /s/-clusters to be constituents. First they exhibit strong co-articulation affects, and secondly, they show clear evidence of voicing assimilation (unlike other clusters which exhibit no evidence of either voicing assimilation or voicing dissimilation). Nevertheless, the evidence is far from conclusive. It remains an open question how to best account for the exceptions to the sonority hierarchy.

6.4 Practical Applications of Phonotactic Constraints

Let us consider two practical applications of phonotactic constraints. First we will show that the sonority hierarchy limits the branching factor. Because of the sonority hierarchy, the number of arcs in the syllable lattice tends to be bounded by $4n$ ($n = 1 +$ the number of segments), instead of the theoretical limit of $n^2$. After discussing the sonority bound, we will move on to a suggestion by Fujimura and Lovins for

---

104. It may not be appropriate to dismiss these exceptions so quickly. Admittedly, sonority violations such as /ps/, /ts/ and /ks/ only appear in affix position. This account, though, is weakened by the fact that the limited distribution of these clusters can be explained (in part) by the principles of syllabification, especially the maximal onset principle. Moreover, it appears to be the case that all fricative + stop clusters, not just the ones that violate sonority, have more or less the same distributions. That is, just as one does not find /ks/ in morpheme internal codas, one also does not find /sk/. Thus, it can be debated whether or not it is appropriate to distinguish between the /ks/ case and the /sk/ case in this way.

105. There is one more reason for considering /s/-stop clusters to be constituents as opposed to stop-/s/ clusters. It appears that /s/-stop clusters are much more common than stop-/s/ clusters in English. For example, the frequency estimate of /st/ is 0.013214 which is considerably higher than that of /ts/ (0.001977). Similar trends are found between /sp/ and /ps/ and between /sk/ and /ks/. These estimates were obtained by David Shipman [personal communication] by counting the number of times these consonant sequences appeared in word internal position in the Brown Corpus [70].
storing the lexicon in terms of distinctive features. This suggestion takes advantage of several phonotactic constraints (such as sonority and length) so that it will not be necessary to specify this information in the representation itself.

6.4.1 The Sonority Bound on Syllable Ambiguity

Excluding the exceptions for the moment, note that the sonority hierarchy makes a prediction which may be important for practical applications:

(169) The Sonority Bound on Syllable Ambiguity: There are at most two places in a string of consonants where a syllable boundary can be found.

This is a most welcome prediction, because it severely limits the width of our syllable lattices. If the sonority bound is correct, there would be at most four arcs spanning across each vowel. To see why this is so, imagine that we had an infinite string of consonants and vowels:

(170) ... (C)* V_i (C)* V_j (C)* V_k (C)* ...

such that there were two alternative locations to insert a syllable boundary between each vowel. Thus we have:

(171) ... (C)* V_i (C)* (#) (C)* (#) (C)* V_j (C)* (#) (C)* (#) (C)* V_k (C)*...

Note that each vowel may belong to one of four possible syllables. For example, V_i can start with either the first or the second (#) to its left and it can end with either the first of the second (#) to its right.

Let's see how this applies in the familiar example:

(172a) Tell the gardener to plant some more tulips.
(172b) /tɛləˈɡɑrdnərtəplaɪnsʌmˈtʌlpz/

and recall that we had difficulty explaining how to syllabify the words plant some. However, by the sonority bound on syllable ambiguity, we know that there can be at most two places in the consonant sequence /ns/ where a syllable boundary can be found.106 (It is probably possible to further disambiguate these two choices on phonetic grounds, as we have mentioned elsewhere.)
Note that the branching factor in our lattices will not exceed 4 (ignoring morpheme boundaries for the moment).

We have also taken advantage of the sonority bound in the design of an efficient algorithm for checking syllable boundaries in our computer readable dictionaries that mark syllable boundaries, and for inserting syllable boundaries in those dictionaries that do not provide syllabification information. The algorithm computes a table of all ways to insert a syllable boundary into all sequences of intervocalic consonants. For example, for the consonant sequence /rt/, the table would list /rt-/, /r-t/, /r/-t, and /r-t/). By the sonority hierarchy we know that the table will never have to list more than two possible syllabifications for any given sequence of consonants. Given this table, the algorithm need only select which of these two entries is appropriate. This decision depends on a number of factors (e.g., stress, vowel reduction and tensing), which we will come to.

The sonority bound is extremely useful. Since it limits the syllabification possibilities to just two, we only need a single bit of information to make the decision.

The sonority bound seems extremely unlikely. One might think that /st/, for example, could have a syllable boundary in three places, *st, s-t, and st*, not just two, as predicted. Why might the prediction be correct?

The prediction is based on the sonority principle. According to the sonority principle, the syllable boundary will always be located at a point of minimum sonority. Note that there are at most two points of minimum sonority in an arbitrary sequence of consonants. Suppose we had a sequence of consonants with three points of minimum sonority as in the ungrammatical word *aktor*. Note that there would be no way to insert a syllable boundary such that sonority was strictly increasing up toward both vowels. Hence, there can be no word like *aktor* with more than two points of minimum sonority.

The bound is actually a consequence of the assumption that sonority is a strict ordering relation. In general, between two peaks (vowels), there can be at most two valleys (possible locations of syllable boundaries). If there were three valleys, then it would not possible to climb out of the middle valley toward either peaks in a strictly increasing fashion. No matter which peak you head toward, you would have to cross another valley. The *aktor* example illustrates just this point. There are three such valleys in this example: aktor, atktor and atktor. From the middle valley, it is not possible to move toward either vowel by a path of

106. Note, though, that we are excluding two sources of potential counter-examples. First, we are not including the appendix of word final syllables, and secondly, we are not including phonological rules, especially gemination. We need to exclude appendixes because of examples like *first rate* which breaks the /str/ cluster at a point of non-minimal sonority. We need to exclude gemination because of examples like *some more* which breaks the /m/ in the middle.
strictly increasing sonority. No matter which vowel you move toward, you will have to pass another point of minimum sonority.

In contrast to *aktor, there are plenty of words like deprive with two points of minimum sonority. These cases arise where there is a unique segment that is strictly less sonorous that its neighbors. In the case of deprive, the underlined /p/ is strictly less sonorous than both of its neighbors. Thus, there are two syllabification points, either to the immediate left or the immediate right of the /p/: de-prive and de-priv-ation, but not *depriv-ation because this would leave the first syllable violating the sonority hierarchy. Other words like actor have only one syllabification point because there is only one point of minimum sonority (i.e., ac-tor).

Thus, to the extent that sonority is a strict ordering relation, we have a theoretical argument demonstrating the correctness of the sonority bound of syllabification. From a practical point of view, we tested the sonority bound on syllabification in the following way. For each word in the MITalk lexicon, we found all intervocalic strings of consonants, and checked if the syllable boundary was in one of the two places predicted by the rule. In fact, there was only one class of counter examples in the corpus (of about 10,000 morphemes). The words, western, northwestern, and other words containing the morpheme west, all have a syllable boundary after the /w/ as in west-ern. Assuming that /s/ is the point of minimum sonority, the syllable boundary should have been adjacent to the /s/. Note that our testing procedure does not include word final syllables.

6.4.2 Removing Redundancy From the Lexicon

In the previous section, we saw that the sonority hierarchy constrained the grammar in such a way as to improve parsing performance by reducing ambiguity. In general, all phonotactic constraints will have this affect. In this section, we will see that phonotactic constraints can also be used to improve the representation of lexical items. (These ideas are extremely speculative and have not been implemented.)

Fujimura and Lovins [35] suggest that a word like limp might be represented in the lexicon as [+lateral; +hi & +front; +nasal & +closure & +labial & +tense]. This representation employs enough features to distinguish each syllable from some other and no more. Thus, it is not necessary to specify the +labial feature for both the nasal and the stop in limp, because the two phonemes have to agree in place. Similarly, it

---

107. The MITalk morph lexicon is large enough to cover all of the morphemes in the Brown Corpus [70].
is not necessary to specify the ordering of the nasal and the stop because the order is a consequence of the sonority hierarchy.

Let us consider in more detail how this approach might work in coda position. The coda position is interesting because it is highly constrained in length, sonority, place, and voicing. The place and voicing constraints can be accounted for by assuming a single place and a single voicing feature in the specification of a coda. The length and sonority constraints can be accounted for by assuming at most two manner features in the specification of a coda.

Thus, we have the following array of possibilities (where \([-\text{stop}]\) is an abbreviation for \([+\text{nasal} \& +\text{fric}]\), and \([-\text{nasal}]\) is an abbreviation for \([+\text{stop} \& +\text{fric}]\), and \([-\text{fric}]\) is an abbreviation for \([+\text{stop} \& +\text{nasal}]\)):\(^{108}\)

<table>
<thead>
<tr>
<th>lab</th>
<th>cor</th>
<th>vel</th>
<th>pal</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ stop</td>
<td>p</td>
<td>t</td>
<td>k</td>
</tr>
<tr>
<td>+ fric</td>
<td>f</td>
<td>s</td>
<td>š</td>
</tr>
<tr>
<td>+ nasal</td>
<td>m</td>
<td>n</td>
<td>nš</td>
</tr>
<tr>
<td>- stop</td>
<td>mř</td>
<td>nš</td>
<td>nč</td>
</tr>
<tr>
<td>- fric</td>
<td>mb</td>
<td>nd</td>
<td>nč</td>
</tr>
</tbody>
</table>

Note that no slot is filled with two different clusters. This suggests that these four features (two manner values, one voice, and one place) are sufficient to uniquely determine the coda.\(^{109}\) On the other hand, these four features may not be necessary. There are a number of empty slots, four of which are due to the absence of velar fricatives in English (either voiced, unvoiced or in clusters). There are also quite a few other feature combinations whose status is dubious in English. For example, many people argue, Kiparsky among them, that voiced nasal clusters are absent from English.\(^{110}\) It can also be argued that voiced fricative clusters are also not English.\(^{111}\) These facts indicate that our system of 48 clusters overgenerates the set of English codas. A similar analysis can be applied for obstruents in onset position.

\(^{108}\) Our analysis of /θ/, /ð/ with two manner features is somewhat non-standard. There are precedents, though, for assigning multiple manner features to single phonemes. For instance, Withgott [124] has argued that /η/ also has two manner features \([+\text{stop} \& +\text{nasal} \& +\text{velar} \& +\text{voice}]\). Her analysis of /η/ avoids the need to posit a rule like /g/-deletion to explain alternations like long/longer. It can be incorporated into our scheme by moving /η/ from \([+\text{nasal} \& +\text{velar} \& +\text{voice}]\) to \([-\text{fric} \& +\text{velar} \& +\text{voice}]\).

\(^{109}\) We will assume that this table is complete, and that there are no other codas in English. Clusters such as /ps/, /tʃ/, /ts/, /ks/, /stʃ/, /stʃs/, etc. are accounted for by placing the final coronals in the affix.
Removing Redundancy From the Lexicon

These tables can be adapted as a framework for storing the lexicon. By the argument above, the lexicon can store a coda core by specifying only the place, voicing and manner features. (If the system needed the phonemic representation of the cluster, it could compute it from the feature specification by looking it up in the table above.) This representation captures the constraints imposed by place and voicing assimilation in a reasonable way so that the lexicon need not specify multiple values for place and voicing. It also captures the sonority hierarchy in a reasonable way, so that it is unnecessary to specify which order the manner features are to concatenated in. Thus, this representation saves storage over a system that explicitly specified multiple place and voice features and ordering dependencies over the manner features. (This representation can probably also be used to reduce the error rate as we will see in section §8.6.6.)

This new representation almost completely dispenses with the notion of the segment. There is no attempt to couple features at the segmental level. For example, the representation does not say where the nasal ends and where the stop begins in the /nt/ coda. It merely says that both features are present in the /nt/ coda. Perhaps manner, place and voicing should be modeled as asynchronous processes, with only a loose coupling relationship at some suprasegmental level.

6.4.2.1 A Note on Liquids and Glides

We have avoided liquids and glides in this discussion of assimilation facts because liquids and glides are governed by a different set of constraints. Liquids and glides show no evidence of voicing assimilation; we find words with /ls/ and /lz/ (e.g., false and falls), /lt/ and /ld/ (e.g., salt and bald), /lf/ and /lv/ (e.g., self and selves), and so on. Likewise there is no place constraint after liquids in rhyme position as evidenced by the existence of /lp/, /lt/ and /lk/ (e.g., help, felt and silk). Instead of an assimilation constraint, there appears to be a place dissimilation constraint in the onset, explaining the absence of /tl/, /dl/, /pw/, /bw/, /stl/, /spw/, /sr/, and so on. Fujimura and Lovins [35, p. 33] note, though, that the usual distinctive feature representation is not entirely suited to expressing this generalization. Note for instance that the coda /lt/ is grammatical, but the onset */tl/ is not. From a phonological point of view, these facts may appear very

110. We prefer to leave the voiced nasals clusters in the chart on the grounds that voicing does affect the phonetic duration of the nasal murmur. Thus we want to leave the /d/ in the phonological representation of a /nd/ cluster so that it will be available at the postlexical level where the /d/ lengthens the nasal murmur and reduces the stop gap and burst.

Admittedly, there are cases like lamb, where it isn't so clear that the /b/ is influencing the preceding nasal. In these cases, we might want to say that there is a historical process in the lexicon which tends to merge voiced nasal clusters with a single nasal segment. This process is probably limited to morpheme-final position, where the evidence for the voiced obstruent is weakest, and therefore, most apt to be missed by the language learner.

111. We would also like to include voiced fricative clusters in the chart even though their status is questionable. They marginally appear in some derived contexts (e.g., despire, disdain, excuse).
difficult to explain.

The difference, though, between the acceptability of /l/ in post-vocalic position and the unacceptability of /l/ in onset position may become more apparent at the phonetic level, where the two /l/ are quite distinct. The post-vocalic /l/ will be dark in English, whereas the pre-vocalic /l/ will be light. From an articulatory point of view, it seems reasonable to assume that a light /l/ might not be found after the tongue is already raised (e.g., /tl/ or /dl/, as opposed to /sl/), since the light /l/ gesture requires the raising of the tongue. In general, it is difficult to follow a total constriction with a partial constriction at the same place of articulation. Thus, after the tongue is already raised for a /t/, it can't be raised anymore for a light /l/. This explains the absence of /tl/, /dl/ and /stl/. Similarly, after the lips are already fully closed for a /p/, we can't make a partial constriction for a /w/. This explains the "absence" of /pw/, /bw/ and /spw/. On the other hand, the possibilities in post-vocalic position are less constrained because it is possible to follow a partial constriction with a total constriction at the same place of articulation. As Fujimura and Lovins point out, explanations of this sort seem plausible but we need more articulatory data in order to substantiate any claims.

6.5 Summary

We have discussed in this chapter a large number of constraints on syllable structure. Five constraints imposed by phonotactics were introduced:

(173a) **length restrictions**: there are at most two constituents in onset position and two in coda position
(173b) **sonority**: there is a total order on phonemes (or perhaps phonetic segments) so that sonority decreases as we move away from the syllable peak
(173c) **idiosyncratic gaps**: e.g., no voiced obstruents after nasals, limited distributions of /h, η, y, w/.
(173d) **voicing assimilation**: obstruent clusters agree in voicing in the onset and coda and in the affix.
(173e) **place assimilation and dissimilation**: obstruent clusters agree in place in onset and coda; glides seem to undergo dissimilation in the onset.

Two practical advantages were discussed. First, it was shown that these constraints are (often) sufficient to limit the branching factor to no more than four syllable arcs per vowel. In addition, we saw that the constraints can be used to re-organize the representation of the lexicon in order to save storage costs and

112. The facts are not entirely clear on this point. /pw/, for example, may be an onset in certain loan words (e.g., poire).
increase potential for error recovery. The organization of the lexicon is discussed in more detail in Appendix I.
7. When Phonotactic Constraints are Not Enough

7.1 Basic Principles

7.1.1 Maximize Onsets and Stress Resyllabification

Phonotactic constraints are not sufficiently strong to produce a unique syllabification in all cases. For example, in words like *western, winter, water, vego, divinity*, and so forth, phonotactic constraints do not tell us which way the /t/ will attach. Most theories of syllable structure invoke "tie-breaking principles" for words such as these. Two very common "tie-breaking principles" are The Maximize Onset Principle and Stress Resyllabification. By the first principle, there is a preference toward assigning consonants to the onset of the following syllable rather than than the coda of the previous syllable. Thus, for example, the /t/ in *retire* will be assigned to the onset of the second syllable (i.e., *ré-tire*), and not to the coda of the first syllable (i.e., *rē-tire*). By the second principle, there is a preference toward assigning consonants to a stressed syllable over an unstressed one. Thus, for example, the /k/ in *record* will be assigned to the first syllable when it is stressed (i.e., *rē-cord*), and to the second syllable when it is stressed (i.e., *rē-cörd*).

(174a) **Maximize Onsets**: When phonotactic and morphological constraints permit, maximize the number of consonants in onset position (e.g., *re-tire*).

(174b) **Stress Resyllabification**: When phonotactic and morphological constraints permit, maximize the number of consonants in stressed syllables (e.g., *rē-cörd / rē-cörd*).

The maximal onset principle is relatively uncontroversial. Stress resyllabification, on other hand, has been the subject of considerable debate.

Stress resyllabification is intended to adjust syllable boundaries so that it will be possible to account for stress dependent phonetic and phonemic facts at the level of syllable structure without direct reference to stress. Thus, in order to account for the phonetic and phonemic differences in a pair like *rēcörd / rēcörd*, the stress resyllabification principle will syllabify the first one as *rēc-ōrd* and the second as *rē-cörd*, and then define the phonetic and phonemic rules to predict the observed phonetic and phonemic differences from the different syllable structures. In a very similar fashion, Kahn and other proponents of ambi-syllabic would employ two different syllable structures to *rēcörd / rēcörd* in order to account for the different phonetic and phonemic possibilities. They would, though, invoke a slightly different form of stress resyllabification which produces an ambi-syllabic /k/ in *rēcörd* instead of a syllable final /k/.
We will discuss three alternatives to stress resyllabification. These methods all account for constraints between stress and phonetic and phonemic outcomes. Before suggesting these proposals, though, let us introduce some of the basic issues.

### 7.1.2 Morphology

It should be noted that neither principle applies into #-prefixes and compounds. Consider the following violations of the maximal onset principle:

- (175a) counter # attack (not *counte-rattack)
- (175b) counter # intelligence (not *counte-rintelligence)
- (175c) bank # rupt (not *ban-krupt)
- (175d) bed # ridden (not *be-dridden)

and the following violations to stress resyllabification:

- (176a) counter # act (not *counte-ract)
- (176b) dis # interested (not *di-sinterested)
- (176c) dis # like (not *di-slike)
- (176d) inter # action (not *inte-raction)

It is not clear if level 1 ++-prefixes block these principles. There are some cases where the two principles are blocked:

- (177a) in + attention, in + articulate, in + advertent
- (177b) in + alienable, in + adequate

but there are also some cases where the two principles seem to apply (e.g., con + mission → co-mmission). Level 1 "+-" affixes pose a number of difficult problems. We leave these issues for future research.

### 7.1.3 Maximize Onsets

The maximize onset principle is widely accepted because it is quite robust. We will illustrate the principle with examples where stress resyllabification does not apply. In this way, we can isolate out the effects of the maximal onset principle.
First consider words with two stress syllables in row:

(178) veño, reducес, produсе (verb). Barbra Partee

Stress resyllabification does not apply in this environment (at least in most formulations that we know of). Thus, these examples provide a good test of the maximal onset principle. How do these words syllabify? The "facts" seem to support the maximal onset principle. It appears that the medial consonants tend to form part of the following onset, not part of the previous coda.

(179a) ve-Jo, re-duce, pro-duce (verb), Barbra Par-tee

(179b) *ve-J-o, *red-uce, *prod-uce (verb), *Barbra Par-t-ee

Thus, it seems that the maximal onset principle makes the correct prediction in these cases where stress resyllabification does not apply.

The maximal onset principle also seems to make the right predictions in examples like:

(180) monkey, donkey, trinket, blanket, bronchus, anchor, conquer, bunker, canker, bronco, Lincoln, handkerchief, anchorite

where the maximal onset principle is opposing stress resyllabification:

(181a) Maximal Onset: mon-key, don-key, trin-ket, blan-ket, bron-chus, an-chor, con-quer, bun-ker, can-ker, bron-co, Lin-co ln, hand-kerchief, an-chorite


Thus it appears that the maximal onset principle correctly captures the "fact" that the consonant sequence /ηk/ will almost always be split into two different syllables despite the stress pattern. In the MITalk Lexicon, the only cases where /ηk/ do not split are forced by morphological/phonotactic constraints:

113. The noun préduse has different stress pattern: prédùce.
114. For now let us simply stipulate these syllable assignments. Later, we will attempt to defend these judgments with allophonic arguments, possibilities for hyphenation, and agreement with authorities.
115. These syllable boundary judgments were taken from the MITalk Lexicon [85].
(182a) ink, pink, stink, wink, drunk, funk, punk,
(182b) drunk-ard, frank-furt, bank-rupt, thanks-giving
(182c) punct-tuality, anxious (an[k-s]jous), func-tion, instinc-tive

There are many other consonant sequences like /ŋk/ which tend to split into different syllables. The best examples are homorganic nasals (except perhaps /nt/), and /ks/, though all multiple consonant sequences which appear in coda position exemplify the basic tendency to split as the maximize onset principle predicts. We will provide a better characterization of these consonant sequences shortly.

In summary, we have seen two cases where the maximal onset principle seems to be necessary. First, we showed that maximal onset principle provides the correct predictions in cases like veto with two stressed syllables in a row. Secondly, the maximal onset principle seems to be necessary to account for words like monkey where stress resyllabification doesn't seem to work very well.

### 7.1.4 Stress Resyllabification

There are a few cases where we probably want to say something in addition to the maximize onset principle. It has long been noticed that certain syllable boundaries tend to alternate with stress alternations. Witness bi-syllabic words like:

(183) addict, affix, annex, collect, digest, essay, minute, present, produce, project, rebel, record

These words have two (or more) stress patterns and two syllabifications with a dependence between the stress pattern and the syllabification. Thus, for example, when primary stress falls on the first syllable of récord, it is syllabified as ré-cord, whereas when the primary stress is on the second syllable, it is syllabified as re-córd. The other words behave similarly.

(184) 10 stress (nouns): add-ict, aff-ix, ann-ex, pres-ent, prod-uce, proj-ect, reb-el, rec-ord

(184d) 01 stress (verbs): a-ddict, a-ffix, a-nnex, pre-sent, pro-duce, pro-ject, re-bel, re-cord

In general, it appears that consonants are attracted to the more stressed syllable. Thus, the /k/ in record falls in the first syllable when the first syllable has primary stress, and on the second syllable when the second syllable has primary stress.
The evidence for syllabification in these cases consists mainly of phonetic observations. That is, the /k/ is assigned to the first syllable in récord because it will be realized with a front unreleased allophone, like the /k/ in wreak. In contrast, the /k/ in recórd is assigned to the second syllable because this /k/ will be realized with a back and aspirated allophone, like the /k/ in cord.

(185a) ré-c-ord by analogy to wreck syllable final /k/
(185b) re-córd by analogy to cord syllable initial /k/

Allophonic reasoning such as this produces syllable boundaries that correspond very closely with our intuition about syllable structure and with our computer readable dictionaries in the cases discussed thus far. It also produces syllable boundaries that agree very closely with judgments on the appropriate locations for hyphenating words. Thus, record can be hyphenated as rec-ord when it is a noun, and as re-cord when it is a verb.

7.2 Against Stress Resyllabification

Stress resyllabification may work well for words like record with a single medial consonant. On the other hand, when there are multiple medial consonants, stress resyllabification seems to be suspect. As we discussed above, stress resyllabification seems to make the wrong predictions in words like:


In general, when there are multiple consonants with falling sonority such as /ηk/, stress resyllabification fails to apply. It is important to distinguish the falling sonority case (e.g., /ηk/) from the rising sonority case (e.g., /gr/). In the latter case, stress resyllabification does seem to apply. Notice, for example, that stress resyllabification does induce a syllabification alternation in a pair like de-grade / deg-radation.

In summary, there seem to be four possible sonority environments:

116. The so-called expletive infixed argument [83] can also be used to justify syllable boundaries in certain cases. Thus, we know that fantastic has a syllable boundary (and a foot boundary) between the fan and the tastic because one can say: fan-fucking-tastic.
117. The hyphenation judgments on record are reported in the TeX Manual [69, p. 180] as the correct practice.
(187a) constant sonority: latter, butter, record, affix
(187b) rising sonority: degrade, degradation, attribute
(187c) falling sonority: monkey
(187d) falling and rising sonority: accomplish, children

As a general rule of thumb, stress resyllabification seems possible in the first two cases and not in the last two. This characterization of the facts seems to agree with our intuitions, our computer readable dictionaries, and the possibilities for hyphenation. It does not explain the allophonic facts; in all three cases, stress seems to have an affect on the allophonic realization of the medial consonants. We will attempt to capture those facts at the level of foot structure.

Why do we feel that it is necessary to explain these facts at the level of foot structure? Syllable structure alone cannot explain the allophonic facts. Consider words from (187d) such as accomplish. We can't explain the realization of the /mpl/ with a single juncture. If we assign the /p/ to the left, then we are hard pressed to explain why it doesn't "delete" as in the "word" comp. If we assign the /p/ to the right, then we are hard pressed to explain why it doesn't aspirate as in the word plush. Thus, there is no easy way to account for the allophonic facts with just syllable structure.

These problems don't arise in a foot structure framework because we can also make reference to the stress (foot structure).

(188) a[ccom-lish]

We assign the /p/ to the right and explain the lack of aspiration by reference to the foot structure. Aspiration occurs in foot initial position, not in foot internal position as we have here.

We will now discuss three proposals for assigning syllable structure more or less as outlined in (187).

(189a) No Resyllabification (Use foot structure instead)
(189b) Limited Stress Resyllabification
(189c) Vowel Resyllabification

118. We also want to include words like lady, and meteor in this class and not in (187a). We want to say that these words should be syllabified as la-dy and me-tor for the same reason that monkey should be syllabified as mon-key.
119. comp is slang for complementizer in syntactic circles.
We do not believe that one of these solutions is correct and that the others are incorrect. Rather, we believe that they are all correct (in spirit if not in detail). They all elicit slightly different sets of constraints which all converge on more or less the same basic notion of constituency. Hopefully our parser will someday be able to exploit all three sets of constraints.

7.2.1 Alternative 1: No Resyllabification

Stress resyllabification may not be necessary. A foot structure based solution is available for all the cases where the stress resyllabification principle is appropriate, as Withgott points out in her thesis [124]. Consider the record example once again. This example seemed to illustrate stress resyllabification very well. In (190a), the /k/ is resyllabified to the left by stress resyllabification. In (190b), the /k/ is syllabified to the right by the maximal onset principle.

(190a) réc-ôrd
(190b) ré-côrd

How can we account for this contrast in a metrical framework? In a metrical framework, examples of stress resyllabification are foot internal. (191a) corresponds to (190a).

(191a) [régôrd]
(191b) ré[gôrd]

Thus, we can replace stress resyllabification with the metrical notion of foot-internal. Any principle that we could formulate in terms of stress resyllabification can be reformulated in terms of foot structure. Recall for example that stress resyllabification accounted for the fact that record (noun) should be hyphenated as rec-ord whereas record (verb) should be hyphenated as re-cord. In a metrical framework, we can account for this contrast by saying that:

(192) **Hyphenation of Foot-Internal Consonants:** Hyphenation occurs after foot-internal consonants (subject to morphological and phonotactic constraints).

(192) is so close to stress resyllabification that it makes many of the same errors. Both (192) and stress resyllabification predict that all four cases in (187) will show hyphenation alternations, not just the first two.
The metrical and syllable based analyses can be made to look very similar in most other respects too. Both analyses agree that phonetic weakening is to be expected in foot-internal / stress resyllabification / ambi-syllabic position, and not in foot-initial / maximal onset position. Thus, we would expect to find short VOT, flaps, unreleased stops, glottalized stops, dark /l/ and so forth in foot internal (or stress resyllabification) position, and stronger allophones such as aspirated stops and light /l/ in foot initial (or maximal onset) position. (As we discussed in the accomplish example, we prefer foot structure over syllable structure for discussing certain aspects of allophonic variation.)

So far we have managed to make the foot structure approach work almost exactly like stress resyllabification (complete with many of the same flaws). In some sense, this is a trivial achievement since metrical foot structure is little more than a notational recoding of the stress pattern. Thus, there is little empirical content to the claim that it is possible to account for stress dependent facts in terms of foot structure. On the other hand, it might turn out that foot structure is a more convenient representation of stress, than say the traditional numbering scheme employed by most dictionaries. Thus, foot structure might be a more appropriate representation even though it is completely derivable in an information theoretic sense from the stress numbers.

It is well-known (especially in Artificial Intelligence) that the choice of representation is very important. It is possible for one representation to be more suitable than another for a particular application, even though the two representations might be "notationally equivalent". For example, in certain signal processing applications, it turns out to be more convenient to represent the signal in the Fourier domain than in the time domain, even though the two representations are equally capable of describing the signal. (In other signal processing applications it turns out just the other way around.) Thus it is possible for one representation (e.g., foot structure) to be superior to another (e.g., stress numbers), even though they may be "equivalent" in some descriptive sense. Indeed, this has been been our experience. Foot structure is more suitable in our parsing and matching framework because it is more naturally described with a set of phrase-structure rules.

However, it is also possible to formulate metrical foot structure so that it corrects some of the flaws discussed above. In particular, we can redefine the metrical hyphenation rule (192) so that it corrects some of the errors in (187). Suppose we restrict (192) so that it only applies to one foot-internal consonant. Thus in applies in (187a,b) but not in (187c,d).

(193) **Limited Hyphenation of Foot-Internal Consonants:** Hyphenation occurs after one foot-internal consonant (subject to morphological and phonotactic constraints).
This limited form of resyllabification is a metrical version of a proposal by Fujimura and Lovins. Let us now turn to this proposal.

7.2.2 Alternative 2: Limited Stress Resyllabification

Suppose that stress resyllabification is restricted so that it only applies to the first consonant after a stressed vowel. Thus, it will apply to the /k/ in réc-órđ, but not to the /k/ in món-kēy. Thus, we could replace stress resyllabification with:

(194) “All stressed syllables must have final specifications (among these may be glides or vowel elongation).” Fujimura and Lovins [35, p. 19, proposal (1)]

This particular formulation was proposed by Fujimura and Lovins as an alternative to stress resyllabification. They argue that it cannot be ruled out given the limited allophonic evidence that is currently available. Fujimura and Lovins’ proposal accounts for the two syllabifications of record and the single syllabification of monkey. In the record case, there are two syllabifications: one when the first syllable is stressed (pulling the /k/ into the first syllable as a final specification) and one where the first syllable is not stressed (thus permitting the /k/ to be sucked into the second syllable under the maximal onset principle). In contrast, the word monkey has just one syllabification (i.e., mon-key). The limited stress resyllabification principle will not pull the /k/ into the first syllable over the intervening /n/, because the /n/ already fulfills the final specification clause. Thus, Fujimura and Lovin’s modification predicts that stress resyllabification is limited to a single foot-internal consonant.

The parenthetical remark about glides and vowel elongation is intended for cases like reduce, ladý and mgieor where the underlined vowels are followed by a glide fulfilling the role of final specification. These glides block the primary stress from pulling an addition consonant into the first syllables. Thus we have re-duce, la-dý and me-teor as opposed to *red-u-ce, *lad-ý, and *met-eor. These cases illustrate very nicely the need to account for allophonic variation at the level of foot structure, not at the level of syllable structure. Notice that [la-dý] and [me-teor] flap because the /l, d/ is foot internal. If we had defined flapping in terms of syllable structure, we would be forced to assume a different syllable structure (e.g., lad-ý and met-eor) and a different account of resyllabification.

---

120 On the other hand, they don’t argue very strongly for their proposal either. In the article, they don’t seem to take a position either for their resyllabification proposal or for Kahn’s. They are more concerned with the paucity of evidence.
7.2.3 Alternative 3: Vowel Resyllabification

Barnwell\textsuperscript{121} has proposed an alternative model of stress resyllabification that appears to be quite different from the models discussed thus far, but turns out to be much more compatible than one might have expected. He partitioned the set of vowels into two classes:

(195a) Class I: The set of vowels that can end an English word (e.g., reduced vowels and diphthongs).
(195b) Class II: the set of vowels that cannot end an English word (e.g., lax vowels except schwa).

If there are multiple ways to syllabify a word so that phonotactic and morphological constraints are satisfied, preference is given to the syllabification that assigns Class I (tense or reduced) vowels to open syllables and Class II (lax unreduced) vowels to closed syllables.\textsuperscript{122} For example, when the first vowel of \textit{record} is realized with the Class I vowel /i/, Barnwell's algorithm will produce the syllabification \textit{re-cord}. In contrast, when the first vowel is realized with the Class II vowel /e/, Barnwell's algorithm will produce \textit{rec-ord}.\textsuperscript{123} Thus Barnwell's algorithm determines the syllable boundaries from the surface realization of the vowel. We will sometimes refer to Barnwell's suggestion as vowel resyllabification by analogy to stress resyllabification which determines the syllable boundaries from stress assignment.

Vowel resyllabification is similar to the limited form of stress resyllabification presented in the previous section. Vowel resyllabification will pull one consonant toward a lax vowel. Limited stress resyllabification will pull one consonant toward a stressed vowel (unless it is followed by a glide). Thus, they both pull one consonant toward a lax vowel. They accomplish this with different mechanisms. Stress resyllabification triggers off stress, whereas vowel resyllabification triggers off [±tense] and [±reduced]. It is hard to find any examples though that manifest these differences (probably for principled reasons). Thus vowel resyllabification and limited stress resyllabification make (more or less) the same predictions.

\textsuperscript{121} We attribute vowel resyllabification to Barnwell because his formulation pre-dates others that we have found. The basic idea that lax vowels tend to be found in closed syllables and that reduced and tense vowels tend to be found in open syllables is well-known. Umeda \cite[p. 436]{umu} observes that /I/, /e/ and /ʌ/ do not appear in open syllables. Nakatani, in a set of slides dated 1978, observed that six lax vowels /I e ʌ U ɔ/ are found only in closed syllables. Barnwell's formulation is a bit stronger than both Umeda's and Nakatani's. In addition to the claim that lax vowels (except schwa) are closed, Barnwell hypothesizes that reduced and tense vowels have a preference for open syllables.

\textsuperscript{122} It is interesting to note that Barnwell's partitioning of the vowels is similar to the one that one required to explain the vowel contrast that we learn in grade school in pairs like: \textit{diner/dinner, anal/annal, later/latter, hater/hatter, hoping/hopping}. Class I vowels appear before singleton consonants; class II vowels appear before doubled consonants. (Presumably doubled consonants indicated closed syllables.)

\textsuperscript{123} There are quite a number of words like \textit{record} that have two stress patterns (e.g., \textit{abuse, addicted, affect, affix, annex, attribute}). In each case, the first syllable will be open when it is realized with a class I (tense or reduced) vowel and closed when it is realized with a class II (lax unreduced) vowel.
Syllable and foot based analyses can also account (more or less) for the vowel distributions predicted by Barnwell's algorithm. Syllable and foot based analyses start with the underlying constituency structure and generate the surface vowels with rules like vowel reduction, tensing, laxing and so forth. Barnwell's algorithm works in just the reverse direction. It starts with the surface form of the vowel and parses the constituency structure.

7.2.4 Summary of Alternatives to Stress Resyllabification

In summary, we have presented three alternatives to stress resyllabification:

(196a) No Resyllabification (use foot structure)  \[\text{suprasegmental}\]
(196b) Limited Stress Resyllabification  \[\text{segmental}\]
(196c) Vowel Resyllabification  \[\text{subsegmental}\]

Although all three are very similar to one another, we have found them all useful because they all look at the problem in slightly different ways. (196a) views the problem from a suprasegmental perspective, (196b) takes a narrower segmental view, and (196c) works with distinctive features.

7.3 Practical Applications of Vowel Resyllabification

7.3.1 Alternative Points of View

In some applications, some of these cues may be more readily available than others. So for example, from the parser's point of view, (196c) is more useful than (196a) because features like \([\pm \text{tense}]\) and \([\pm \text{reduced}]\) are more readily available than features like \([\pm \text{foot-internal}]\).

In other tasks, we might find (196a) to be more useful than (196c). Consider, for example, the task of checking lexical entries for consistency. We have found that the assignment of primary stress to be more reliable than the assignment of \([\pm \text{tense}]\). For example, we might find the word *America listed “incorrectly” as *Amér[il]-ca (cf. Amér[ó]-ca). This entry is internally inconsistent because the lax Class II vowel /l/ is assigned to an open syllable. We could correct this entry in one of two ways: replace the class II vowel with a class I vowel or re-assign the stress/foot structure so that the offending class II vowel appears in a closed syllable.
(197a) *Amèr[l]-ca → Amèr[ə]-ca
(197b) *Amèr[l]-ca → *[əmər] [ica]

correct the vowel

correct the stress

If the assignment of primary stress is more likely to be correct (as we have suggested), then (197a) is the more promising correction. Thus, for this correction task, it appears that stress resyllabification is suitable than vowel resyllabification. In general, it is advantageous to have these alternative viewpoints so that we can select the most appropriate one depending on the task.

7.3.2 Vowel Class and Allophonic Cues are Often Redundant

In some situations, we might not want to choose just one point of view. It might be reasonable to try several of them in parallel and compare results at the end. There could be one detector looking for each type of cue. If one detector doesn’t find what it is looking for, maybe one of the other detectors will. In this Pandemonium-like framework, the decoder can take advantage of the redundancy in the speech signal.

Consider, for instance, a minimal pair like:124

(198a) [æt̚l - trə - but̚l]
|   |   |   |
|   |   |   |
(198b) [ə - tʰr̩l - ur̚l]
|   |   |   |

attribute

attribute

where there are several redundant cues for the different syllable/foot structures. From Barnwell’s vowel class argument, we know that (198a) has a closed first syllable (because /æ/ is class II) and an open second syllable (because /ə/ is class I). Similarly, we know that (198b) has an open first syllable (because /ə/ is class I) and a closed second syllable (because /l/ is class II). These same facts can also be determined from the allophonic variation: of /l/ and /b/ in the two pronunciations. In (198a), we know that the first syllable is closed because the /l/ is unreleased and we know the second syllable is open because the /b/ is released. In (198b), we know that the first syllable is open because the /l/ is released, and we know that the second syllable is closed because the /b/ is unreleased.

124. Some people may object to our phonetic transcriptions. We want to make three important phonetic distinctions. In the attribute case, (a) the /l/ has a longer stop gap, (b) the /l/ is less retroflexed, and (c) the /b/ is more likely to be released.
In general, there will be two redundant cues for the syllable/foot structure before /tr/-clusters. First, we can look at the vowel before the /tr/-cluster to see if it is class I or class II. Secondly, we can look for a release in the stop before the /tr/-cluster. The same multiple cues hold for other onset clusters as well. For each cluster, we have found a minimal pair with contrasting syllable/foot structure. The first element of the pair (e.g., de-prive) has an open first syllable, a class II vowel, and a released stop. The second element of the pair (e.g., dep-rivation) has a closed first syllable, a class I vowel, and an unreleased stop. These correlations are very robust.  

\[(199)\]

\[\begin{array}{lll}
 & \text{i} & \text{w} \\
\text{d} & \text{de-prive} & \text{di-plomacy} \\
 & \text{dep-rivation} & \text{dip-lomatic} \\
\text{l} & \text{a-tribute} & \text{de-cline} \\
 & \text{att-ribute} & \text{a-cquire} \\
\text{k} & \text{de-cresce} & \text{dec-lination} \\
 & \text{dec-ri-ment} & \text{ob-ligatory} \\
\text{b} & \text{cele-bration} & \text{ob-ligation} \\
\text{d} & \text{a-ddress} & \text{acq-uisition} \\
\text{g} & \text{de-grade} & \text{a-cquire} \\
 & \text{deg-radation} & \\
\end{array}\]

7.3.3 Reduction of Branching Factor

These redundant viewpoints can be used to reduce the branching factor in our lattices. In Chapter 1, we discussed a number of examples illustrating the effectiveness of allophonic constraints in restricting the search space. Let us now consider an example to illustrate how the parser can take advantage of vowel

---

125. It is interesting that there are as many gaps in the paradigm as there are. Some of the gaps can be explained on phonotactic grounds. Perhaps English simply lacks the onsets: /\text{kl}/, /\text{dl}/, /\text{pw}/ and /\text{bw}/. (This is a controversial point.) However, there are still some remaining gaps left to be explained. We suspect that /\text{gl}/, /\text{lw}/, /\text{dw}/ and /\text{gw}/ have a limited distribution with respect to stress. Thus, we can find them in pre-stress onsets (e.g., glow, agglutinate), but we don’t seem to find them in post-stress environments. It is possible that these clusters cannot be found in foot medial environments, just as we don’t find voiceless /\text{h}/ and intervocalic /\text{t}/ in foot medial position.
resyllabification. Suppose we were given the transcription:

\[(200) \ \{\text{\texttt{\textipa{[\textipa{\textipa{T}E\textipa{\textipa{n}s\textipa{\textipa{k}\textipa{\textipa{n}}}\textipa{\textipa{m}}}\textipa{\textipa{l}}}\textipa{\textipa{q}r\textipa{\textipa{\textipa{\textipa{\textipa{p}}}w\textipa{\textipa{z}}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}]\}

for the phrase *the president's economic proposals*. Vowel resyllabification can be used to syllabify (200) into

\[(201) \ \{\text{\texttt{\textipa{[\textipa{\textipa{\textipa{\textipa{T}E\textipa{\textipa{n}s\textipa{\textipa{k}\textipa{\textipa{n}}}\textipa{\textipa{m}}}\textipa{\textipa{l}}}\textipa{\textipa{q}r\textipa{\textipa{\textipa{\textipa{\textipa{p}}}w\textipa{\textipa{z}}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}\}

because these are the only syllable boundaries that leave schwas and tensed vowels in open syllables and lax vowels in closed syllables. (We are, of course, ignoring the affect of morpheme boundaries for the moment.)

There are only two regions remaining to be syllabified:

\[(202a) \ \{\text{\texttt{\textipa{\textipa{\textipa{\textipa{T}E\textipa{\textipa{n}s\textipa{\textipa{k}\textipa{\textipa{n}}}\textipa{\textipa{m}}}\textipa{\textipa{l}}}\textipa{\textipa{q}r\textipa{\textipa{\textipa{\textipa{\textipa{p}}}w\textipa{\textipa{z}}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}\}

\[(202b) \ \{\text{\texttt{\textipa{\textipa{\textipa{\textipa{\textipa{q}r\textipa{\textipa{\textipa{\textipa{\textipa{p}}}w\textipa{\textipa{z}}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}\}

Phonotactic constraints uniquely determine case (202b), and limit the syllabification possibilities in (202a). The resulting lattice of syllabification possibilities would be:

\[(203) \ \{\text{\texttt{\textipa{\textipa{\textipa{\textipa{\textipa{\textipa{D}}}x.\textipa{\textipa{\textipa{\textipa{\textipa{p}}}r.\textipa{\textipa{\textipa{\textipa{E}}}z.\textipa{\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}]}\textipa{\textipa{z}}}\}

\[\text{The president's economic proposals}\]

This lattice is considerably thinner than it would have been without the constraint on vowel classes. With the transcription parsed into syllables, it is relatively easier to hypothesize words.

7.3.4 Schwas Enrich the Speech Signal

It is interesting to contrast our use of schwas in order to reduce the branching factor with the widely held view that schwas are a source of “noise”. Admittedly, it is very difficult to determine the underlying identity of vowel given just a schwa on the surface. That is, a so-called “fast speech rule” like

---

126. The current version of the parser takes the weaker position that class I (tense or reduced) vowels can be either open or closed, but class II (lax unreduced) vowels must be closed. We have not employed the stronger version for reasons that will be discussed in section §7.3.5.
(204) \( V \rightarrow \emptyset \)

would seem to be increasing recognition perplexity because it maps many underlying vowels into the same non-descript surface representation. On the other hand, though, it doesn't seem to be necessary to know the underlying identity of the vowel. It really doesn't matter, for example, if the first vowel in *pilato* is front, back, high, or low. The syllabification and stress information encoded in the schwa is far more important because it controls a number of allophonic and phonological processes including aspiration, flapping, and tensing. Furthermore, it is important to determine the syllabification, at least in our framework, because syllabification is the basis of the organization of the lexicon.

There is some recent psycholinguistic evidence which supports the claim that schwas enrich the speech signal. Cutler and Clifton [25] contrasted mis-pronounciations in stress with mis-pronounciations in vowel reduction. They concluded:

(205) "The distinction between full and reduced vowels, it is true, appears to be very important indeed: the results of Experiment 4 clearly show that reducing a full vowel, or giving full weight to a reduced vowel, makes a word very difficult to recognize ... [Primary] stress marking ... we have found to be of primary importance for word recognition." [25, p. 13]

Our own informal experience concurs with Cutler and Clifton's finding. We have found it to be extremely difficult to understand certain foreigners when they apply a foreign rule of vowel reduction. We have also found it difficult to read spectrograms of function words in citation form where they receive a stress that they would not have in context. For instance, the words *to* and *chew* can look very similar in citation form, but not in context where the vowel in *to* would be reduced.

Cutler and Clifton attribute this distinction between stress and vowel to a well-known linguistic fact about word stress.\(^{127}\)

(206) "It is a fact well known to linguists that word stress can shift from its citation form location in response to the demands of sentence rhythm; thus 'it is autoMATic' but 'AUTomatic focus'. However, such stress shifts are strictly constrained; the stress can shift only to a full syllable ..." [25, p. 14]

---

127. Cutler and Clifton also discuss a left/right asymmetry in stress alternations. We have omitted this discussion since it does not relate to our point about the usefulness of schwas.
We might speculate that mis-pronouncements in vowel reduction are more serious because this type of error will cause the parser to assign the wrong syllable structure. This speculation needs to be backed up with firm data before we reach a conclusion.

7.3.5 Stronger Form of Vowel Resyllabification

So far we have been assuming that vowel resyllabification is subject to phonotactic constraints. Thus, we can find class I (tense and reduced) vowels in closed syllables in words like accept /'æk-sept/ and achieve. In these cases, phonotactic constraints prohibit us from assigning the vowels in question to open syllables: *a-cept and *achieve. There are only a few cases where we need to resort to phonotactic constraints. Perhaps we should special case those exceptions:

\[ (207) \text{Strong Form of Vowel Resyllabification: Except for certain special cases, Class I (tense or reduced) vowels are found in closed syllables and Class II (lax unreduced) vowels are found in open syllables.} \]

There are no special exceptions for class II (lax unreduced) vowels\(^{128}\) and only four special cases for class I (tense or reduced) vowels. The four special cases are:

- (208a) in function words (e.g., in, on, and, an),
- (208b) in morpheme final syllables (e.g., rancid, achieve, rancidness, achievement),\(^{129}\)
- (208c) in certain prefixes (e.g., ad-, ac-, con-, des-, dis-, ex-, in-, per-, sub-, suc-).\(^{130}\)

---

128. The MITalk Lexicon lists numerous entries with lax (class II) vowels in open syllables. However, the dictionary is probably incorrect in these cases. For example, the MITalk Lexicon lists words like America, pentagon, testicle, receptacle, spectacle, swastika and furniture with a lax /a/ vowel in the open penultimate syllable. There are good reasons, though, for suspecting that lax vowels (class II) are blocked from the penultimate syllable when the primary stress is assigned to the antepenultimate syllable. (This is a consequence of the English main stress rule which assigns main stress to the antepenultimate only when the penultimate is open, ignoring sonorant destressing for the moment.) Thus we find reduced vowels in the open penultimate syllable of America, pentagon, testicle, etc., and tensed vowels in the open penultimate syllable of area, rodeo, and meteor where the penultimate vowel is followed by another vowel.

129. Schwas and tense vowels are rarely followed by two morpheme final segments. Thus, we don't seem to find cases of /i/tsp/ or /e/ypsp/, for example. (We also don't find cases of class I vowels in prefixes like trans- which end in two consonants.) In general, a class I vowel can be followed by most one morpheme final segment (except for a few exceptional cases like paint, saint, range, arrange which have two morpheme final segments after a class I vowel). Ignoring these exceptions, we can account for the fact that a single morpheme final segment can follow a class I vowel, if we analyzed the morpheme final segment as extrametrical, and consequently, it does not count as closing the final morpheme. This extrametrical analysis is motivated by Hayes' use of the notion in order to predict foot structure from syllable weight [40, 41].

130. Many dictionaries list these prefixes with schwas, but it is not clear that this is correct. Despite the fact that these vowels are short, one finds even shorter vowels in prefixes with open syllables. Contrast, for example, gress/contest, gconscious/excuse, receive/prceive, and gird/wet/illegal. Perhaps the vowels in openyllables are schwas, whereas the vowels in closed syllables are lax, despite the fact that they are very short.
It seems promising to look for some special methods to deal with these cases, so that we could adopt the stronger form of vowel resyllabification. (This approach has not been implemented.)

Let us summarize the discussion of vowel resyllabification. Vowel resyllabification was originally proposed as an alternative to stress resyllabification. It has some advantages over stress resyllabification (it parses *monkey as mon-key, not *mon-ey). But it is also useful as simply an alternative point of view. For certain applications, [±tense] and [±reduced] are more readily available than [±stress]. Thus, with alternative viewpoints, the system can select the most appropriate one for a particular application. In addition, alternative formulations can be used to exploit redundancy. As we have seen, [±tense] and [±reduced] often agree with [±stress]. The system only has to find one of these multiple cues in order to parse the syllable structure correctly. In this way, we can take advantage of the fact that vowel constraints and allophonic constraints are often redundant.

Although vowel resyllabification works well in conjunction with other constraints, it is also effective on its own. Recall the example, the president's economic proposals, which was intended to demonstrate that schwas enrich the speech signal. This example assumed a slightly stronger version of vowel resyllabification, where schwas and other class I vowels are blocked from open syllables except in certain special cases. This strong position can probably be maintained, but we leave the details for future research.

7.4 Automatic Syllabification of Lexicons

We have applied vowel resyllabification in conjunction with a number of additional constraints on syllable structure to clean up the syllable boundaries in our computer readable dictionaries. As we have mentioned previously, our computer readable dictionaries are not very consistent in marking syllable boundaries. For this reason, we have found it useful to design an automatic procedure for marking syllable boundaries. This procedure starts with a string of phonemic symbols (without syllable boundaries, morpheme

131. In these case, one might not want to say that the sonorant consonant closes the syllable, since the segment is almost completely vocalized. Perhaps, we might represent the entire vowel as a single segment, and say that these reduced vowels cluster along with schwas in class I.
132. Some of our dictionaries (e.g., [66]) have found it so hard to mark syllable boundaries in a consistent fashion that they do not even attempt to mark them.
boundaries, or stress) and returns a unique assignment of syllable junctures.

Consider, for example, the word abandon. The algorithm looks at each string of inter-vocalic consonants, one by one. The algorithm will try to syllabify the strings /b/ and /nd/. The first one will be syllabified as /-b/ and the second will be /nd/. Thus the final result will be a-ban-don.

The algorithm is extremely simple. There is a procedure called parse-medial-cluster that returns a list of all the ways that a string of inter-vocalic consonants can be syllabified subject to phonotactic constraints and resyllabification possibilities. Thus, for example, parse-medial-cluster would say that /b/ can be parsed as either /-b/ or /b/ and that /nd/ can only be parsed as /nd/. (/nd/ is not a possible resyllabification of /nd/, assuming the limited form of resyllabification discussed above: /nd/ will be syllabified as /n-d/ in all stress patterns.) By convention, parse-medial-cluster returns the maximal onset parse (e.g., /-b/) as the first choice and the resyllabified parse (if it exists) as the second parse (e.g., /b/). There can be no more than two parses because of the sonority bound on syllabification (see section §6.4.1). Four examples of parse-medial-cluster are given below.

<table>
<thead>
<tr>
<th>Sonority</th>
<th>Cluster</th>
<th>Maximal Onset</th>
<th>Resyllabified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>b</td>
<td>-b</td>
<td>b-</td>
</tr>
<tr>
<td>Rising</td>
<td>gr</td>
<td>-gr</td>
<td>g-r</td>
</tr>
<tr>
<td>Falling</td>
<td>nd</td>
<td>n-d</td>
<td></td>
</tr>
<tr>
<td>Rising and Falling</td>
<td>mpl</td>
<td>m-pl</td>
<td></td>
</tr>
</tbody>
</table>

We are now left with a single binary choice between the maximal onset parse and the resyllabified parse. Barnwell’s vowel classification will be used to make the choice. The maximal onset parse is selected if the preceding vowel is class #3. Otherwise, the resyllabified parse is taken.

(209a) Class I Vowel → Maximal Onset Parse

(209b) Class II Vowel → Resyllabified Parse

In this way, we can take advantage of the sonority hierarchy on syllabification and Barnwell’s vowel resyllabification algorithm in order to parse lexical entries into syllables.

There are two cases where parse-medial-cluster will produce incorrect results. Parse-medial-cluster has no provision for morpheme boundaries and affixes. In the present system, morpheme boundaries have to be marked with an asterisk (e.g., bank*rupt) in order to over-ride parse-medial-cluster. If we did not mark the
morpheme boundary in this way. *Parse-medial-cluster* would produce *ban-krupt*. Ideally, these boundaries would be parsed automatically by a morphological decomposition algorithm like the one in MITalk [85].

The second problem, affixes, can be ignored in the syllabification task. Affixes are only found in word final position (e.g., *can’t*) and sometimes in morpheme final position, (e.g., *bail*-*man*). *Parse-medial-cluster* would produce the wrong results in these cases. For example, *parse-medial-cluster* would say that the /nt/ in *can’t* ought to be parsed as /n-t/ which is clearly wrong. Fortunately, we don’t need to consult *parse-medial-cluster* in word final and morpheme position, because it is trivial to syllabify word/morpheme final consonants (final consonants are part of the final syllable). Thus we don’t need to worry about affixes.133

We have introduced a few modifications for efficiency’s sake. In particular, the calls to *parse-medial-cluster* are memoized so that we can take advantage of the fact that it will be called many times with the same argument. In addition, *parse-medial-cluster* has been redefined to return two parses in all cases. When it is given a cluster like /ktf/ (e.g., *actor*) that has only one parse, it will return the one parse twice as both the maximal onset parse and the resyllabified parse. In this way, we can avoid the special case for unambiguous clusters. These sorts of efficiency issues are very important for large lexicons.

We have implemented the algorithm outlined above and tested it on Webster’s Pocket Dictionary [84] (20k entries) and the Klatt Lexicon [66] (3k entries). Appendix V shows some of the results from the Klatt lexicon.

7.5 Summary

We began this chapter with the observation that phonotactic constraints are not sufficiently strong to produce a unique syllabification in all cases. For example, in words like *western*, *winter*, *water*, *veto*, *divinity*, and so forth, phonotactic constraints do not tell us which way the /t/ will attach. In order to disambiguate these cases, we introduced two “tie-breaking principles”: maximal onsets and stress resyllabification.

The first principle was found to be extremely useful and effective, especially in words like *veto* with a 12 stress pattern. Stress resyllabification seemed to work well in words like *rē-cōrd* / rē-cōrd, but not so well in words like *mon-key* (not *monk-ey*). In general, it seems that resyllabification applies in constant and rising sonority, but not in any other cases.

133 In Appendix V, we ran *parse-medial-clusters* on word final clusters anyway and marked the results with "+" signs. Thus, *can’t* would be parsed as can+*t* and *abandon* would be parsed as a-ban-do+n. We suspect that we might be able to make use of these final affix constituents, but we have not done so in the current implementation.
(210a) constant sonority: latter, butter, record, affix
(210b) rising sonority: degrade, degradation, attribute
(210c) falling sonority: mon-key, lo-y-dy, me-y-teor
(210d) falling and rising sonority: accom-plish, chil-dren

To correct this deficiency in stress resyllabification, we proposed three alternatives:

(211a) No Resyllabification (Use foot structure instead) suprasegmental
(211b) Limited Stress Resyllabification segmental
(211c) Vowel Resyllabification subsegmental

All of these alternatives can account for facts as laid out in (210), but in very different ways. We suggested that it was useful to have these three different perspectives, so that we could select the most appropriate one for a particular task, and so that we could exploit the redundancy in order to recover from certain labeling and segmentation errors.

All of these constraints are very useful. We presented yet another example in this chapter, the president’s economic proposals, in order to demonstrate how we can make use of syllabification principles in order to reduce the search space. In addition, these principles where put to work in order to correct the syllable boundaries in our computer readable lexicons. The syllabification algorithm introduced in the final section makes use of a number of principles discussed throughout this thesis, especially the sonority bound on ambiguity and Barnwell’s vowel resyllabification.
8. Robustness Issues

In a practical speech recognition system, we could not expect the front end acoustic processor to be as good as a linguist. It is unlikely that the acoustic processor will be able to provide a segmentation lattice of the same caliber as those obtained from our linguistic consultant, Meg Withgott. More realistic segmentation lattices are likely to contain a large number of errors and alternative choices. How can we modify our system to deal with these realities?

Before addressing this question, we should say that we have not been concentrating on robustness concerns per se. We originally wanted to see what could be done with linguistic transcriptions as they were. This seemed to be a reasonable starting point. After all, if we couldn't deal with this problem, it seemed unlikely that we could handle the more difficult problem assuming errors and branching in the input lattice.

This chapter will begin by showing that the current implementation can deal with a limited degree of uncertainty in the input. However, if there is too much uncertainty, there will be a dramatic decrease in performance. Two examples will be discussed, one where the current implementation works well, and one where it does not. Within limits it is possible to tune the parser to deal more effectively with ambiguity. The parser should depend more strongly on cues that the front end is likely to label correctly, and less on cues that the front end is likely to confuse. For instance, if the front end is not very good at distinguishing vowels, then it might make sense for the parser to group all vowels into a single equivalence class and not bother to exploit vocalic constraints. Of course it would be more advantageous to improve the front end so that we can depend on distinctive features. In the second half of the chapter, we outline a proposal for improving the front end. Segments should be decomposed into distinctive features. In this way, the front end can benefit from the rich structure imposed at the feature level.

8.1 Alternatives in the Input Lattice

The system developed here can deal with multiple choices in the input lattice, though we have not had much opportunity to test this aspect of the program. We have looked at some of Francine Chen's notes while reading spectrograms, which tend to have one, two or three choices at each point. Here is, for example, a transcription of "...we're not promising any instant..."

\[(212) \text{[wR]} \text{n.} \text{[O]} \text{.d} \text{.} \text{.k.} \text{[r]} \text{[O]} \text{m.} \text{s} \text{I[G]} \text{[E]} \text{.m.} \text{.i.} \text{[g]} \text{.i.} \text{n.} \text{.z.} \text{[t][n][t]} \text{[G.} \text{.t.} \text{.e.} \text{.p.]} \text{[n.} \text{.e.} \text{.q}{.} \text{.i.} \text{.1.} \text{.s.} \text{]} \text{[f.} \text{.f.} \text{.]}
\]

we're not promising any instant
In this case, there don't turn out to be very many possible syllables, even though there are a number of alternative labels in the input segmentation lattice. There is no way, for example, for a syllable to span across the 5th and 6th segments. We are fortunate in this example that /d, t, f/ cannot be in the same syllable as /k, p/. There will, of course, be other examples where bushy segmentation lattices lead to even bushier syllable lattices. For now, let us focus on an easier case like (212) above.

The parser produced the following syllable lattice:

\[(213) \begin{array}{cccccccc}
\vdots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\vdots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}\]

we're not promising any instant

Many of these arcs can be removed because they don't contribute to a complete path through the lattice.

\[(214) \begin{array}{cccccccc}
\vdots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}\]

we're not promising any instant

Thus we see that branching in the segment lattice did not lead to an unmanageable amount of branching in the syllable lattice, at least in this example.

### 8.2 Problems for Parsing

On the other hand, we have been much less successful in our attempts to parse some of the lattices from the BBN speech project into syllables.\(^{134}\) These lattices have much higher branching factors and are much more likely to contain serious mistakes. Consider, for example, the segmentation lattice in figure 18. Our parser does not work well on this lattice for two reasons. First, it is difficult to parse syllables because the transcription doesn't make enough distinctions to enable allophonic and phonotactic constraints to do much work. Secondly, the lattice is sufficiently errorful that we can't place much confidence in the resulting parse.

---

\(^{134}\) Richard Schwartz of BBN provided us with approximately 100 segmentation lattices produced during the ARPA speech project.
Fig. 18. A Hard Segmentation Lattice

"+" denotes alternative path; "*" denotes sequence. The symbols are defined in the BBN final report [126, vol. II, appendices 1-2, pp. 72-74]. "nvobs" denotes a non-velar (non-strident) obstruent (e.g., a weak fricative); "ftdh" denotes either an /f/, /θ/, /h/ or /ð/. "kg" denotes a /k/ or a /g/. Many of these broad categories were further refined (statistically) in the actual BBN system. Unfortunately, we were not able to obtain this data.

(* (+ ao (* w uh)) ; what
 kg
 iy
 nvobs
 ih
 (+ (* nvobs ftadh) jht) ; the
 uw
 (+ (* flap el) ao (* el l))
 (+ (* vdh bv) bv) ; total
 ah
 (+ (* nvobs dh) jh)
 ix
 (+ nvobs pk)
 plos
 ih
 jht
 (+ (* (+ iy (* y ix)) flap) er eh)
 (+ er r))

8.2.1 Allophonic Constraints on an Impoverished Lattice

First, allophonic constraints don’t help us as much as they might because the lattice doesn’t mark enough allophonic distinctions. For instance, the lattice doesn’t say whether the stop in what is glottalized or aspirated. Some of these phonetic distinctions are (often) relatively easy to detect from a spectrogram. We maintain that there are many instances where it is easier to detect aspiration than place or voicing.

In some cases, it is possible to guess which allophone might have been employed by looking at the alternative hypotheses proposed at the same point. For example, the first /t/ in total was probably aspirated because it was confused with a /ʃ/ whereas the final /t/ in budget was probably not aspirated because it was confused with weak fricatives. Thus, by clever guessing, we might be able to take advantage of certain allophonic constraints. However, it would be much more elegant and reliable for the front end to supply the allophonic cues directly.
8.2.2 Phonotactic Constraints Aren’t Helping Either

Secondly, we can’t use phonotactic constraints to parse the lattice into syllables because the lattice is too bushy and errorful. Consider the juncture between *budget* and *figure*. If the lattice were more constrained than it is, we could use phonotactic constraints to find the juncture between *budget* and *figure*. As we have seen, the only case where two obstruents can be found in the same syllable is when the second is in a word final affix [35] as in a word like *baked*. In such a case, the second obstruent will be coronal and will agree in voicing. If these place and voicing constraints are not met as in *budget figure*, we know that there must be a syllable juncture (and possibly a word juncture as well) between the two obstruents. Unfortunately, the place of the second obstruent is not given in the lattice\textsuperscript{135} so it is not possible to use the phonotactic cues to find the juncture. Thus, we cannot be sure whether the two obstruents are in the same syllable or not.

Moreover, the program will spend a lot of time and effort before it realizes that both syllabifications are possible. The program, as it is currently implemented, will literally try all combinations of stops and fricatives, ruling out those combinations that violate place and voicing constraints. This takes a long time. When the lattice doesn’t have any place and voice information, the program shouldn’t try to enforce agreement constraints. Although it wastes a lot of time trying all place and voice combinations, it doesn’t introduce any additional errors to do so.

However, if there are any errors in the input segmentation lattice (214) above (and there are certain to be many), they will also be present in the output parse. So, for example, there are several errors in the transcription of *is the*. The transcription for *is the* is given as: [i n v o b s l]. We suspect that both vowels and both consonants in *is the* are transcribed incorrectly. On theoretical grounds, we predict that The first vowel should be lax and the second vowel should be reduced since the first vowel is in a closed syllable and the second vowel is in an open syllable. In addition, we would hope that the front end could one day distinguish the */z̃/ sequence as two consonants, not as a single geminated segment.

(215a) The first vowel is probably lax.\textsuperscript{136}

(215b) The nvoobs is really two segments, a */z/ followed by a */èmes/.

(215c) The third vowel is probably a schwa.

\textsuperscript{135} The place information was available in the BBN system, but not in the lattices that we were given.

\textsuperscript{136} The labeler may have failed to account for the following */z/ which would tend to pull 1-2 up.
The first error will not cause problems for the parser, as currently implemented, but it will cause difficulties later when we try to match syllables against the lexicon. The second and third errors, though, will cause the parser to produce incorrect results. The parser will fail to consider the possibility that the nvosbs might be a so-called “geminated” segment; it will attempt to attribute all of the nvosbs to just one of the adjacent syllables. The third error, labeling the vowel in *the* as lax, will cause the parser to assign the vowel to a closed syllable. Thus, some errors in the input segmentation lattice such as (215b–c) will lead to parsing errors.

### 8.3 Relaxing Phonological Distinctions

There are a large number of ways in which we could recover from these sorts of errors. We could, for instance, recover from the vowel class error above, by tuning the parser and the matcher to relax certain vowel distinctions since the front end can’t provide this information with perfect reliability. In this example, the parser should not trust the front end labels of /I/ and /ə/. If the parser completely ignored the difference between /I/ and /ə/, then it would find the correct syllabification. Moreover, if the matcher also ignored the front end’s assignment of vowel labels, then it would find the word *the* as one of the possible decodings.

The disadvantage of this solution, though, is clear; it loses the distinction between /I/ and /ə/ which is presumably distinctive in English. Consequently, the system is likely to find a large number of words, only a few of which should have been considered to be valid decodings of the transcription. We find ourselves facing the classic trade-off between type I and type II errors. The same trade-off arises in a probabilistic system, as we will see in section §8.5.

It has been our experience that segmentation confusions are very different in each of our corpuses. Although we have not attempted to evaluate the quality of the corpuses in a careful statistical study, it is our impression that there are relatively few errors in the linguist’s transcriptions, somewhat more errors in the spectrogram reader’s notes, and many more errors in the computer system’s segmentation lattices. Our lexical retrieval system would be at a great advantage to know which corpus it is working with. When it is given one of the linguist’s transcriptions, it can pretty safely assume that the transcribed vowels will match those in the lexicon, especially when the vowel is front and stressed. For instance, there is relatively little chance that the linguist will transcribe the /æ/ in *band* as an /e/. Thus, when working with the linguist’s transcriptions, the matcher should not accept a confusion between /e/ and /æ/. On the other hand, a confusion such as this is much more probable in the spectrogram reader’s notes, and extremely likely in the computer system’s lattices.

---

137. It is assumed that a tense vowel can be open in word final position (e.g., *reduce*).
For this reason, the matching routine is implemented with a few switches for specifying (in a very limited sense) which confusion errors are acceptable and which are not. In this way, the system will decode the linguist's transcription of *band* as *band*, not as *bend*. In contrast, when the system is decoding the same transcription from the spectrogram reader, it will return both *band* and *bend*. In this way, the relaxation of the parsing and matching routines can be tuned (in a very primitive way) to the reliability of front end transcriber.

These relaxation methods can also be applied to correct the second error in the BBN lattice presented above. Recall that a */\theta/* was deleted in the context of */z/* in the phrase *is the*. The matcher can be relaxed so that it overlooks certain distinctions which are likely to be confused at earlier stages in the segmentation lattice. In this case, we could say that a segment labeled as */z/* could actually be a */z\theta/*. If this sort of confusion is considered very likely, we might add an optional */\theta/* arc after all */z/*s in the segmentation lattice. Modifying the input lattice in this way would allow us to find the correct decoding, at least in this case. Unfortunately, in addition to *is the*, we will also find a number of spurious phrases such as *is a*. Over-matching is the danger of relaxing the constraints to overlook distinctive features.

This general relaxation approach could be adapted to account for a large number of deletion, assimilation and gemination “facts” such as

(216a) deletion of */d/* after */n/*:  
(216b) deletion of */t/* after */f/*:  
(216c) deletion of */t/* after */s/*:  
(216d) deletion of */m\eta/* before a voiceless obstruent:  
(216e) assimilation of */s\eta/*:

and so forth. That is, we could add an optional */t/* after all */f/*s and */s/*es, an optional homorganic nasal before all voiceless obstruents, an optional */s\eta/* path around all */s/* arcs in the segmentation lattice, an optional */s\eta/* path around all */s/* and so forth. This approach will undo these transformations and retrieve the correct decoding.

We are reluctant, though, to implement this approach (and only a few cases have actually been implemented), because as we relax more and more constraints, there is a greater and greater chance of over-matching. Each of the rules above wipes out distinctions that are phonologically distinctive. In so doing, each of the rules above lose valuable parsing and matching constraints.
8.4 Conservation of Distinctive Features

We do not believe that it is necessary to give up these constraints. People (and trained spectrogram readers) can distinguish pairs like:

(217a) *is the* $\neq$ *is a* /z也希望 /= /z/
(217b) *find* $\neq$ *fine. opened* $\neq$ *open* /nd/ $\neq$ /n/
(217c) *stuffed* $\neq$ *stuff* /stf/ $\neq$ /tfl/
(217d) *missed* $\neq$ *miss* /stfl/ $\neq$ /tfl/
(217e) *hint* $\neq$ *hit* /ntfl/ $\neq$ /tfl/
(217f) *this ship* $\neq$ *the ship* /$sfl$/ $\neq$ /$sfl$/

(Obviously, we don’t have an automatic front end segmentation module that can make all of these distinctions. Let us assume, for the remainder of this section, that someday (not too far in the future) we might be able to construct such a front end.)

We suggest that many of these so-called deletion rules are unnecessary. In many cases, it is possible to find alternative analyses that do not delete features, but merely recode them in some other way. Thus, instead of employing a deletion rule that applies in /nd/ clusters, we prefer a reanalysis rule that recodes the /nd/ cluster into a single unusually long nasal murmur. Similarly, we would replace a deletion rule for /nt/ clusters with a reanalysis rule that encodes the nasal into the preceding vowel. Ideally, we might strive for a set of phonological rules that obey the “conservation law”:

(218) **Conservation of Distinctive Features:** Phonological rules cannot insert or delete distinctive features

We might refer to (218) as a “no neutralizational condition”. Obviously, it is stated too strongly. It seems fairly clear (but not certain) that flapping neutralizes the voicing distinction in a pair like *writer/*rider. However, we hypothesize that neutralization is far less common than previously believed. This is the thrust of (218).
8.4.1 Deletion in Homorganic Nasal Clusters

How can we account for the difference between /nd/ and /n/, and the difference between /nt/ and /t/ without neutralization rules? We could have a rule to say /nd/ (as in find) will have a longer nasal murmur than /n/ (fine), and /nt/ (hint) will have a nasalized vowel unlike /t/ (hit). The conservative formulation does not lose distinctive features (e.g., nasality, voicing, stop). In this way, it is more consistent with our view (and Stevens’ theory of invariant distinctive features). The conservative view enables us to maintain our hypothesis that phonological rules are a source of constraint, not of noise.

Moreover, not only does the conservative view permit us to recover the underlying invariant distinctive features, but it also turns out to be useful for recovering suprasegmental boundaries. Recall minimal pairs such as:

(219a) can’t Ann / can Jan
(219b) canned Ann / can Dan

In our framework, we can distinguish these pairs on the basis of the allophonic variation of the underlined stop. If we were only looking at invariant cues, we would miss the crucial distinctions.

8.4.2 Deletion in Fricative Clusters

In the previous section, we have suggested that three so-called deletion rules may actually be information-preserving upon closer phonetic investigation.138

(220)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>/nd/ → /n:/</td>
<td>find → /faːn:/</td>
</tr>
<tr>
<td>/nt/ → /t/</td>
<td>hint → /hɪt/</td>
</tr>
<tr>
<td>/z̩/ → /z:/</td>
<td>is the → /lz̩/</td>
</tr>
</tbody>
</table>

In this section, we will consider a few more so-called deletion rules that we have mentioned recently. Recall the /t/ deletion rule which applies in words like soft and list.

---

138. We have attempted to represent IPA characters as well as we could with the Dover printer. The results are not ideal. [θ] is a schwa reduced vowel and [–] is nasalization.
(221) 

<table>
<thead>
<tr>
<th>Rule</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>/ft/ → /f:/</td>
<td>soft</td>
</tr>
<tr>
<td>/st/ → /s:/</td>
<td>soft</td>
</tr>
</tbody>
</table>

The conditions under which these rules apply are not precisely known. The /ft/ deletion rule is more likely to apply in words like softly than in words like softer. But, even in a word like softly, it may not be correct to say that the /t/ is deleted. In this environment, it is very difficult to tell whether or not there is a /t/. It is hard to argue that there is /t/, but it is also hard to say that there isn’t one. Perhaps, there really is a /t/ in this environment, but we just can’t see it because we do not know enough about phonetics yet. It seems that pairs like laughed/laugh and missed/miss are phonetically distinct, and that people can often hear the differences, although they can be confused in a few cases. Expert spectrogram readers can also see the differences in some cases. As we learn more, we may be able to find the /t/ in more and more of these cases. We believe there is a rich source of bottom-up phonetic evidence that can be of considerable use in these sorts of cases.

8.4.3 Tunafish Sandwich

Consider the /$$s/ → /s:/ rule. This rule may work out the same way. It has traditionally been argued that the /$$s/ in tunafish sandwich merge into a single phonetic segment.

(222) "It should be noted that the formal description of this process in terms of rules have been proposed by linguists. Heffner, for example, describes an assimilation process by which gas shortage would be gashshortage, i.e., s$$ = =>$$s. In a similar vein, Hill has suggested a 'de-palatalization' process by which tunafish sandwich would become tunafishsandwich, i.e., $$s = =>ss." [130, pp. 1-2]

However, in a recent study, Zue and Shattuck-Hufnagel [130] have found some low level acoustic evidence against a gemination analysis of /$$s/ as tunafish sandwich. It seems that the first half of the /$$s/ spectrum tends to be more /s/-like and that the second half tends to be more /s/-like. Zue and Shattuck-Hufnagel demonstrate this by measuring the cut-off frequency at two points along the time axis. The first sample is taken at the 1/4 point. This presumably corresponds to the /s/ portion of the /$$s/. The second sample, taken at the 3/4 point, presumably corresponds to the /s/ portion of the /$$s/. They find that the first sample has a cut-off frequency of about 3 kHz (well within the range for a typical /s/), and the second has a cut-off frequency of about 4 kHz (well within the range for a typical /s/). From this spectral tilt, they conclude that
/ss/ should be analyzed as two segments, not one.

The /sʃ/ case (as in gas shortage) did not work out so well. They did not find any spectral correlates to distinguish the two segments in question. From this failure-to-find, they conclude that /sʃ/ might be a true example of gemination. We might prefer to say that the case remains open. Perhaps (with a bit of luck), someone might find an acoustic correlate in the future. (In fact, Zue [(personal communications)] has informed us of a more recent unpublished study which seems to suggest that /sʃ/ can be separated into two segments.)

(223a) "The qualitative differences between the /sʃ/ sequence and the /ʃs/sequence are illustrated in the next slide. In the /sʃ/, or gas shortage case, the sequence of fricatives tends to have a homogeneous acoustic structure very similar to that of either a single /ʃ/ or a geminated /ʃʃ/. On the other hand, the /ʃs/, or tunafish sandwich case, the /ʃs/ sequence clearly shows two separate phases, the first having acoustic characteristics similar to that of /ʃ/ and the second similar to that of /s/. The acoustic differences between the two sequences are also reflected in the cut-off frequency computed at 10 msec intervals and plotted below the spectrograms.

The results are further illustrated in the next slide, where we present histograms summarizing the distribution of cut-off frequencies for a typical speaker ... it is clear that the means and distributions for the 1/4 point measurements lie predominantly in the /ʃ/ range, whereas the mean and distribution for the 3/4-point measurement lie predominantly in the /s/ range ..." [130, pp. 4-5]

Thus in the tunafish sandwich case, as in other examples of apparent deletion, gemination, and/or assimilation behavior, we see that there is growing reason to believe that there are low level phonetic cues which may help us to separate the two segments. We may not need to posit deletion analyses, at least in many cases. Alternative feature-preserving accounts are preferable when they apply; they preserve the important distinctive features in pairs like find/fine and hit/hint.

Unfortunately, we do not have feature-preserving accounts for all phonological rules. For instance, we do not have a method (or even a suggestion) for distinguishing /sʃ/ as in gas shortage from /ʃs/ as in cash shortage.139 Until we find a solution, we will have to allow for the possibility that allophonic processes can occasionally obscure certain distinctive phonemic features. We maintain, though, that this is a relatively rare

---

139. Zue and Shattuck-Hufnagel account for the asymmetry between /sʃ/ and /ʃs/ by assuming that palatalization in an anticipatory rule. Thus, palatalization applies in gas shortage but not in tunafish sandwich.
possibility; it is not nearly as common as has been traditionally assumed. In addition, the situation seems to be improving. Recent studies, such as Zue and Shattuck-Hufnagel, are finding low level evidence for two segments in many cases which have been traditionally analyzed as gemination. We expect this trend to continue.

8.5 Probabilistic Methods

In this project, when we encounter two cases which we do not know how to distinguish (e.g., /$s$/ versus /$s$/), we do not attempt to guess between them. Rather, we enter them both into the lattice data structure as alternative possibilities. Our approach might be characterized as categorical. In many other projects, most notably at IBM [7, 45, 46, 47, 48, 49], the two cases would be viewed as competing hypotheses, each one having a certain probability of being correct. This probabilistic view is well suited for dealing with errors of the sort mentioned above, as the IBM project has demonstrated. We can associate a certain probability with each vowel confusion, for example, and thereby recover from many vowel confusions.

Let us illustrate an example of how this might work. Imagine that the speaker uttered the word band. Suppose the correct transcription is /bæn:/, but for some reason, the acoustic processor produced the segmentation lattice /bɛn:/ . The linguistic decoder might, nevertheless, find the correct decoding from this erroneous decoding because the confusion /æ/ $\rightarrow$ /e/ is fairly common, and would therefore have a fairly high probability. Let us assume that of 100 utterances of the vowel /æ/, 55 of them will be labeled correctly by our hypothetical acoustic processor as a /ɛ/, 40 of them will be labeled incorrectly as an /ɛ/, and other 5 will be labeled as something else. Given these sorts of confusion statistics, we could assign a probability that the segmentation lattice produced by our acoustic processor was generated from the word band. In this case, the word band will score fairly well because the /b, n, d/ all match exactly and the /ɛ/ matches /æ/ pretty well since this sort of confusion is quite common. As long as no other word in the dictionary matches the segmentation lattice better, the system will select band as the favorite decoding (ignoring the language model for the moment). By relaxing the /æ/-/ɛ/ distinction in this way, the probabilistic approach can recover from certain types of labeling and segmentation errors.

8.5.1 Problems with Relaxing Distinctions

However, in this case with the /æ/ having been transcribed as an /ɛ/, the word bend will match the segmentation lattice even better than the word band. Thus, the system is likely to report erroneously that it heard the word bend instead of band. If the system is going to distinguish correctly between pairs like
bend/band is will have to have a model of the difference between /æ/ and /ɛ/. It doesn’t matter whether the difference is stated in probabilistic or categorical terms.

Note that the probabilistic approach described above is depending on gaps in the dictionary in order to compensate for an inadequate front end. In some cases it may be necessary to depend on gaps in the dictionary, but, as we discussed in section §1.2.3, this practice should be kept to a minimum. It is likely to lead to trouble if one routinely smears out phonemically distinctive differences (e.g., /æ/ versus /ɛ/). In a probabilistic framework, we could alleviate the dependence on gaps in the dictionary by accepting all words that match with sufficiently high probability. In this way, both band and bend might be returned by the system since they both match about equally well.

How do we decide what is “a sufficiently high probability”? If we set the probability threshold too low, the system is likely to return too many candidate matches. On the other hand, if we set the threshold too high, the system is likely to reject the correct choice. It seems that the probabilistic system is faced with the trade-off between type I and type II errors, just as we saw with the categorical system. It is not absolutely clear that the probabilistic system is, in fact, more suitable for dealing with segmentation errors.

8.5.2 The Similarity of Probabilistic and Categorical Approaches

It seems to us that the two approaches are more similar than it might appear at first glance. In both frameworks, we seem to see the same issues coming up. It is possible to construct models, in both probabilistic and categorical frameworks, to account for all of the constraints that we have discussed so far. Both frameworks can model confusions at the front end, phonological processes, distributions of variant and invariant features, and suprasegmental generalizations.

The difference between the probabilistic and categorical approaches is more a matter of style (or religion), than one of substance. The probabilistic approach starts with the assumption that the ultimately “correct” model is so difficult to discover (if it exists at all), that it is hopeless to try to find it. In contrast, the categorical approach starts with the assumption that the “correct” model exists, and with a lot of effort, we might actually characterize it with explicit constraints. Proponents of both views argue back and forth. Proponents of categorical approaches see their rules as principled “explanations” that they should be proud of. On the other hand, these “explanations” are called ad hoc “rules of thumb” by proponents of probabilistic approaches. This debate has raged for many years and will certainly continue for many more.
The following two entries from *A First Dictionary of Linguistics and Phonetics* discuss a slightly different dispute, but one that is not totally unrelated.

(224a) "God's truth: A phrase coined in the 1950s to characterise one of two extreme states of mind of a hypothetical linguist when he sets up a description of linguistic data; opposed to hocus-pocus. A 'God's truth' linguist approaches his data with the expectation that the language has a 'real' structure which is waiting to be uncovered. The assumption is that if one's procedure of analysis is logical and consistent, the same description would always emerge from the same data, any uncertainty being the result of defective observation or logic on the part of the analyst. In a hocus-pocus approach, by contrast, no such assumption is made. See Robins 1971." [24, p. 164]

(224b) "hocus-pocus: A phrase coined in the 1950s to characterise one of two extreme states of mind in a hypothetical linguist when he sets a description of linguistic data; opposed to God's truth. A 'hocus-pocus' linguist approaches his data in the expectation that he will have to impose an organization on it in order to show structural patterns. Different linguists, on this view, could approach the same data, and by virtue of their different backgrounds, intuitions, procedures, etc. arrive at differing descriptions. In a God's truth approach, by contrast, the aim is to demonstrate an underlying structure really present in the data over which there could be no dispute. See Robins 1971: Ch. 2." [24, p. 174]

In this work, we have adopted a categorical approach because we are attempting to defend the thesis that there are interesting constraints at the allophonic and phonotactic level. This thesis is easier to defend in a categorical framework where the constraints can be stated explicitly. Moreover, we find it comforting to believe that these constraints have fundamental explanations in terms of more basic linguistic fundamentals. It would be disappointing if these constraints turned out to be statistical accidents.

### 8.6 Distinctive Features

In the previous section, we discussed probabilistic and categorical methods for dealing with segmentation errors. Both of these approaches treated segments as atomic objects. In the probabilistic case, a probability was assigned to each possible confusion. For instance, there was a certain probability that a /æ/ would be confused with an /ɛ/ and another probability that a /t/ would be confused with /θ/ and so forth. These probabilities would be used in the matcher to compute a score for each possible hypothesis, as in the

example discussed above. In the categorical case, there would be switches in the matcher to control the set of acceptable confusions. In this way, it is possible to relax phonemic distinctions in the categorical system, as well as in the probabilistic system.

8.6.1 Modeling Confusions at the Segmental Level

So far, we have only considered confusions at the segmental level. Perhaps it would be more appropriate to represent confusions in terms of distinctive features. For instance, instead of computing confusion matrices in terms of phonemes as we normally do, it might be more profitable to compute them in terms of distinctive features. It is well-known that most segmentation devices are more likely to make single feature errors (e.g., labeling a /t/ as a /d/ or a /k/ than double feature confusions (e.g., labeling a /t/ as a /g/). In this section, we will attempt to take advantage of this intuition. (None of these ideas have been incorporated into the demonstration program.)

Consider the following confusion matrix, which appeared in Seneff [106, p. 21]:

\[
\begin{array}{cccccc}
 & p & b & t & d & k & g \\
p & 19 & 4 & 4 & 2 & 4 & 1 \\
b & 3 & 9 & 0 & 1 & 0 & 3 \\
t & 4 & 0 & 39 & 1 & 4 & 0 \\
d & 4 & 3 & 9 & 10 & 1 & 0 \\
k & 4 & 1 & 2 & 0 & 17 & 8 \\
g & 1 & 1 & 0 & 1 & 5 & 8 \\
\end{array}
\]

There are some interesting trends in the matrix which support our intuition that single feature errors are more probable than double feature errors. In particular, notice that there are 9/173 (5%) double feature errors and 63/173 (36%) single feature errors. Keep these figures in mind; we will return to them shortly.

This matrix is extracted from one of four confusion matrices reported in Seneff’s study.\(^{141}\) Seneff’s unpublished study is considered by many [128] to be much more comprehensive than most published studies (e.g., [23]) since Seneff collected a much larger sample of speech (1153 segments). These confusion matrices

---

141. There is one matrix for vowels, one for fricatives, one for stops and weak fricatives, and one for nasals and semi-vowels. Her labeling system is basically phonemic. There are 40 labels: 16 vowels, 5 strongly fricatives, 12 stops and weak fricatives, and 7 nasals and weak fricatives.
measure Seneff's ability to segment and label spectrograms "correctly". Seneff's labels were graded by Victor Zue.

8.6.2 A Comparison of Seneff's Performance with BBN's Front End

At the time, Seneff was a novice spectrogram reader having previously seen only about 30-40 spectrograms. She reports that 64.4% of the phonemes and 60.6% of the words were correctly identified. On stops, her performance is somewhat lower. Based upon the matrix given above, we have computed a recognition rate of 58% over stops.\textsuperscript{142} This level of performance is comparable to that of mechanical segmentation and labeling devices of the ARPA speech project. The BBN front end [126, vol. II, § 3.3] correctly labels 52% of the segments with the first choice.\textsuperscript{143} From these recognition rates, it appears that Seneff (64%) is slightly better than BBN's front end (52%). These numbers may be slightly deceptive, though, since the BBN results are computed over a set of 71 labels, whereas Seneff's results are computed over only 40 phonemes. (In general, accuracy decreases with the number of labels.) Their performance appears much more comparable when recognition rates are computed over the same class of labels.\textsuperscript{144}

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
Phonemes & Seneff & BBN's Front End \\
\hline
/m n η/ & 80\% & 83\% \\
/s z/ & 74\% & 73\% \\
/p t k/ & 77\% & 77\% \\
\hline
\end{tabular}
\end{center}

On the basis of this evidence, it appears that their performance is roughly comparable.

\textsuperscript{142} Stops are relatively hard. Vowels are even harder. Nasals, glides, and weak fricatives are relatively easy.

\textsuperscript{143} With more choices, performance obviously improves. BBN reports recognition rates of 69\%, 75\%, 80\%, 83\%, 85\%, and 86\% with each additional choice. We discuss BBN's performance on the first choice for comparison to Seneff's results which were based on a single choice.

\textsuperscript{144} These figures were computed from the confusion matrices reported in [106] and [126, vol. II, § 3.3]. We only considered the submatrix for segments that were labeled as a member in the class in question (e.g., /m n η/) and that the correct label was a member of the class in question. Thus, these scores indicate the number of times a phoneme was labeled correctly, given that it was labeled as a member in the class in question and that the correct labeled was a member of the class in question.
Fig. 19. Two Confusion Matrices for /p t k/

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>t</th>
<th>k</th>
<th></th>
<th>p</th>
<th>t</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>19</td>
<td>4</td>
<td>4</td>
<td></td>
<td>46</td>
<td>11</td>
<td>18</td>
</tr>
<tr>
<td>t</td>
<td>4</td>
<td>39</td>
<td>4</td>
<td></td>
<td>1</td>
<td>91</td>
<td>11</td>
</tr>
<tr>
<td>k</td>
<td>4</td>
<td>2</td>
<td>17</td>
<td></td>
<td>9</td>
<td>5</td>
<td>45</td>
</tr>
</tbody>
</table>

Two confusion matrices for /p t k/ are presented above. The matrix on the left is compiled from Seneff’s results [106]; the matrix on the right is compiled from the BBN final report [126, section 3.3].

8.6.3 Practical Application of Confusion Matrices

These confusion matrices could be used for estimating the confusion probabilities in the probabilistic framework outlined above. Thus, for example, we could estimate the probability of a /p/-/b/ confusion with 4/172 (~2%) from the fact that Seneff found 4 such instances in her corpus of 172 stops. However, we shouldn’t have much confidence in this estimate since it is based on only four instances. The probably that /p/ will be recognized as /p/ is much better estimated in this way because there are 19 instances as opposed to just 4. In contrast, the probably that /b/ will be confused with /t/ is extremely poorly estimated, since there are no instances of this confusion. We should certainly not conclude that /b/ and /t/ will never be confused in the future, just because we haven’t seen any cases of it in our sample of 172 stops. In this case, we could solve the “sparse data” problem by simply gathering more data. However, in a real system, we might not want to ask each new speaker to go through a long training sequence. Thus, for certain practical applications, it may be advantageous to have the ability to approximate the probability of rare events from little (or no) direct evidence. One could place a threshold on the minimum probability. So for example, we might say that all confusions have a probability of at least 1%. This sort of solution is ad hoc. It would be more aesthetic to model the estimation errors in some way that could be justified in terms of acoustic-phonetics.

8.6.4 Modelling Confusions at the Distinctive Feature Level

Suppose that we broke phonemes down into their distinctive features and attempted to model confusions in terms of distinctive features rather than in terms of phonemes. We could then interpolate the phonemes confusions in terms of distinctive feature confusions. Of course, we are introducing an approximation here, that distinctive features are statistically independent. Perhaps this independence
approximation is more appropriate than other approximations that might be suggested to deal with rare events.

We need less data to estimate confusions at the feature level because there are fewer different feature values than different phoneme values. Notice, for example, that there are no zeros in either of the following two confusion matrices.

\[
\begin{array}{ccc}
\text{voiced} & \text{unvoiced} \\
\text{voiced} & 42 & 33 \\
\text{unvoiced} & 22 & 76 \\
\end{array}
\]

\[
\begin{array}{ccc}
\text{labial} & \text{coronal} & \text{velar} \\
\text{labial} & 35 & 7 & 8 \\
\text{coronal} & 11 & 59 & 5 \\
\text{velar} & 7 & 3 & 38 \\
\end{array}
\]

These two confusion matrices are formed by summing the corresponding entries of the confusion matrix for stops given as (226) above. In formulating the model of confusions at the level of distinctive features, we have reduced the number of degrees of freedom. The phoneme based model of stop confusions had 36 degrees of freedom, whereas the feature based model has only 13 (4 for voicing and 9 for place of articulation).

Reducing the number of degrees of freedom has a number of advantages. With fewer degrees of freedom, it is easier to estimate the parameters. Moreover, we have more faith in our ability to estimate these parameters since there are fewer very rare events. Reducing the number of degrees of freedom also increases Seneff's performance. Note, for instance, that Seneff correctly recognized only 58% of the stops, but she correctly recognized 68% of the voicing decisions and 76% of place decisions.\(^{146}\)

---

\(^{145}\) At IBM, this sort of interpolation is sometimes referred to as smoothing, since it removes glitches and gaps (zeros) from the probability distribution. In some ways, estimating the probability of rare events is similar to the problem of grammar induction (also called acquisition and learning). These are all schemes for inducing the possibility (and frequency) of rare events given insufficient evidence. Most proposed solutions to the estimation problem assume some sort of pre-determined (possibly innate) model of grammar with a few parameters to be estimated from the data. The training device (or the child) examines whatever data it can and sets the parameters according to some evaluation metric (e.g., parsimony, maximum likelihood).
8.6.5 Feature Integration

In moving to the more constrained feature based model of confusions, it will be necessary to develop a model for integrating multiple cues. Let us assume that we have separate processes for detecting voicing and place. From these feature detectors, we could construct phoneme detectors in the obvious way. That is, the /p/-detector would look at the outputs of the voicing and place detectors, and light up when the voicing detector found a voiceless segment and the place detector found a labial segment.

This design is heavily influenced by Stevens' invariance theory of distinctive features. If Stevens' theory is correct, then we could build the features detectors to be completely autonomous. This would be a very elegant design. On the other hand, if Stevens' theory turns out to be false, and it turned out that the feature detectors had to be coupled, we could stretch the model accordingly. Assume, for instance, that the voicing decision depended in some way on the place of articulation. One could imagine that voicing and place might be coupled when the decisions hinge on VOT, which seems to be an important cue for both features. We could account for this coupling by implementing the feature detectors to recompute everything that they depend on. If both place and voice depend on VOT, then the place and voice detector will compute VOT. If voicing depends on place, then the voicing detector will have a copy of the place detector inside of it. (There may be more efficient solutions, but we will not concern ourselves with efficiency issues just now.) Thus we can account for dependencies, if they should be necessary.

We could estimate the performance of the feature based phoneme detectors from the confusion matrices of the feature detectors. Let us assume that confusions in voicing are independent from confusions in place of articulation. From this assumption, we can predict the probability of single and double feature errors. The probability of a double feature error is the product of the probability of error in voicing (32%) and the probability of error in place (24%). Thus, we predict a 7.7% probability of a double feature error. In the confusion matrix above, Seneff actually displayed a 5% double feature error rate. In a similar fashion, we can estimate the probability of a single feature error. It is the probability of error (32% + 24% - 7.7% =

---

146. Seneff's increase in performance should be taken with a grain of salt. Error rate is generally related to the size of the label set. As Reddy [97, p. 519] observes, "The finer the desired phonemic transcription, the lower the accuracy." Perhaps, recognition rate is not the relevant measure. An appropriate (information theoretic) measure of performance would take into account both recognition rate and the size of the label set. In this way, increasing or decreasing the size of the label set (in an arbitrary way) should have relatively little influence on performance.

147. Voice Onset Time (VOT) is a acoustico-phonetic measure of the time between the release of a stop and the voicing of the following vowel. VOT is related to voicing (short for voiced stops and long for voiceless ones), place of articulation (shortest for labial stops, intermediate for coronal stops and longest for velar stops) and metrical structure (short in foot-internal position; long in foot initial position). It is not known how these trends interact. (The magnitude of the correlation with place is very small.)
48%) minus the probability of a double feature error (7.7%). Thus, we expect 40% chance of single feature error. Recall that Seneff displayed a 40% single feature error rate.

<table>
<thead>
<tr>
<th>Interpolated</th>
<th>Counted</th>
</tr>
</thead>
<tbody>
<tr>
<td>voicing error</td>
<td>32%</td>
</tr>
<tr>
<td>place error</td>
<td>24%</td>
</tr>
<tr>
<td>single error</td>
<td>40%</td>
</tr>
<tr>
<td>double error</td>
<td>7.7%</td>
</tr>
<tr>
<td>no errors</td>
<td>52%</td>
</tr>
</tbody>
</table>

(Notice that the model supports the intuition that double errors are much less likely than single errors.) The predicted recognition rate is 52% which is somewhat lower than Seneff's performance of 58%. In general, these estimates are fairly close, though they are overestimating the probability of errors.

If we put these matrices back together again, we obtain the following matrix:

\[
\begin{pmatrix}
  p & b & t & d & k & g \\
  p & 15.38 & 4.45 & 3.08 & 0.89 & 1.02 & 3.51 \\
  b & 6.68 & 8.50 & 1.34 & 1.70 & 1.94 & 1.53 \\
  t & 4.83 & 1.40 & 25.92 & 7.50 & 0.64 & 2.20 \\
  d & 2.10 & 2.67 & 11.25 & 14.32 & 1.21 & 0.95 \\
  k & 1.34 & 1.70 & 0.57 & 0.73 & 9.23 & 7.25 \\
  g & 3.08 & 0.89 & 1.32 & 0.38 & 4.83 & 16.69
\end{pmatrix}
\]

If we subtract the matrix above from the original we obtain the following table of estimation residues:

---

148. Each element of the estimated stop confusion matrix is computed by multiplying the appropriate locations of the feature confusion matrices. For example, the /p/ → /b/ element of the stop confusion matrix is computed by multiplying the unvoiced → voiced location in the voice confusion matrix with the [labial] → [labial] location in the place confusion matrix. (The feature confusion matrices have to be normalized so that they sum to unity and then the resulting matrix has to be normalized so that it sums to 173 (the total number of stops)).
Although the predicted values are fairly close to the observed values, there are some clear trends in the
residues. The model, as it currently stands, tends to underestimate the probability of frequent phonemes (e.g.,
/\textipa{t}/) and overestimate the probability of infrequent phonemes (e.g., /\textipa{g}/). We could correct this problem by
normalizing the estimates by the frequency of each phoneme in the corpus.

We now adjust the estimated confusion matrix (228) by these percentages to obtain a more accurate estimate:
This corrected matrix provides a recognition rate of 54%. We are still underestimating Seneff's performance, but we are considerably closer. We would expect the interpolation algorithm to decrease the recognition rate since it is filling in the rare events, and rare events tend to be errors.

The correction does indeed improve the estimate. Note that the differences between this estimated matrix and the one presented in Seneff's study are much smaller.\textsuperscript{149}

\begin{equation}
\begin{array}{cccccc}
\text{p} & \text{b} & \text{t} & \text{d} & \text{k} & \text{g} \\
\text{p} & -2.03 & 0.91 & -0.61 & -1.02 & -2.88 & 2.88 \\
\text{b} & 0.47 & -4.59 & 0.69 & -0.12 & 1.01 & -2.21 \\
\text{t} & 3.53 & 2.18 & 1.40 & -10.69 & -3.01 & 3.42 \\
\text{d} & -2.16 & -0.66 & 0.87 & 2.56 & 0.06 & 0.84 \\
\text{k} & -2.61 & 0.77 & -1.41 & 0.76 & -7.41 & -0.47 \\
\text{g} & 0.60 & -0.54 & 0.68 & -0.80 & -2.49 & 0.67 \\
\end{array}
\end{equation}

Thus, we can achieve a fairly accurate prediction of the stop confusion matrix from the combination of three smaller matrices:

(233a) voice confusion matrix
(233b) place confusion matrix
(233c) apriori distribution of stops in the language

Thus, with these 19 parameters, we can estimate the 36 confusions with fairly high agreement. Moreover, with limited training data, our 19 parameters can be estimated more accurately since they are based on much more frequent events. For instance, we will see many more voicing confusions than /p/-/b/ confusions. We hypothesize that it is useful to consider all types of voicing confusions in the estimation of /p/-/b/, assuming that we do not expect to see enough instances of /p/-/b/ confusions to estimate them directly. In this way, distinctive features might provide a more robust estimation of confusion probabilities.

\textsuperscript{149} The largest residue 10.61 clearly indicates that the model is missing some important aspect(s) of Seneff's labeling strategy. In the /t/-/d/ decision, her errors are heavily biased (9 d→t and only 1 t→d) whereas her other voicing errors were much more evenly distributed (3 b→p vs. 4 p→b and 5 g→k vs. 8 k→g). We have no explanation for this interaction.

(It seems that Seneff tends to over-use the /d/ label, applying it about twice as frequently as it occurs in the corpus. Perhaps her performance would improve if she checked her labels against the apriori probability of occurrence, to make sure that she isn't using one label too often.)
8.6.6 Practical Applications

How could this observation be put to work in a practical system? If we were to adopt a probabilistic framework, we could use the scheme outlined above in order to estimate confusion probabilities. Even if we believed that we had enough training data in order to estimate many of confusions that were assumed to be too rare to estimate directly, we still might find the foregoing discussion useful for estimating a few rare events, and then we could combine the two estimation techniques to obtain an optimal compromise.\(^{150}\)

The distinctive feature approach can also be of service in categorical based systems. We could for example index words (or syllables) in clever ways so that we could easily recover from single feature errors. Suppose for example that each word is indexed three times: once as a sequence of manner features, once as a sequence of place features, and once as a sequence of voicing features. For example, the word *splint* would be indexed under the three keys:

\[
\begin{align*}
\text{(234a) } & \text{[fricative] [stop] [sonorant] [vowel] [nasal] [stop] } & \text{manner} \\
\text{(234b) } & \text{[coronal] [labial] [coronal] [??]} & \text{place} \\
\text{(234c) } & \text{[voiceless] [voiceless] [voiced] [voiced] [voiced] [voiceless]} & \text{voicing}
\end{align*}
\]

We would access the lexicon on all three of these keys. Words that were found on all three keys would be considered to be extremely plausible candidates. Words found on two out of three would be considered to be less plausible, but still possible. Words found on only one key might be considered too implausible to be worth further investigation. In this way, we might augment the matching procedure in order to recover from certain transcription errors.\(^ {152}\) This approach could be justified in terms of the probabilistic discussion above.

The probability score of words with no feature confusions is almost certain to be higher than the probability of words with a single feature error. The probability score of words with two feature errors is almost certain to be so low that there will always be some word with fewer errors that matches better (assuming a large lexicon). Thus we can ignore words with two feature errors because there is no hope of recovering in this case.\(^ {153}\)

---

150. The IBM group has developed maximum likelihood techniques for combining multiple estimators such as these [49].
151. We leave the place of a vowel unspecified for expository convenience. The place features that we have been assuming for consonants may not be appropriate for vowels. The place of a vowel is usually described in terms of the position of the tongue (e.g., ±front, ±back, ±tongue, ±coronal) whereas the place of a consonant is usually described in terms of the position of the constriction (e.g., ±velar, ±labial, ±coronal). There have been some efforts to unify the feature system.
152. This algorithm is a generalization of an idea proposed in [111].
153. This assumption may not be correct if the lexicon is small and/or the language model is highly constrained, so that it is may be possible to recover from considerably more errors. In which case, it may be reasonable to pursue words with double and triple feature errors. We can imagine indexing the lexicon so that it is possible to look up all words that are within \(n\) feature errors of a given key.
So far, we have been assuming that indexing is applied at the word level. Suppose instead that indexing is applied at the sylpart level. Thus, we might re-organize the features in example (234) as:

\[(235)\]

<table>
<thead>
<tr>
<th></th>
<th>onset</th>
<th>peak</th>
<th>coda</th>
</tr>
</thead>
<tbody>
<tr>
<td>manner</td>
<td>[− nasal]</td>
<td>[+ vowel]</td>
<td>[− fric]</td>
</tr>
<tr>
<td>place</td>
<td>[+ labial]</td>
<td>[??]</td>
<td>[+ coronal]</td>
</tr>
<tr>
<td>voicing</td>
<td>[− voice]</td>
<td>[+ voice]</td>
<td>[− voice]</td>
</tr>
</tbody>
</table>

In this way, we can combine the advantages of syllable structure with the advantages of multiple indexing. The resulting scheme takes advantage of phonotactic constraints (e.g., sonority, length, place and voicing assimilation) and also obtains some ability to recover from single feature errors.

8.7 Summary

It was shown that the currently implementation could deal with a limited degree of uncertainty in the segmentation lattice. Two examples where discussed, one where the current implementation works well and one where it does not.

Within limits it is possible to tune the parser to deal more effectively with ambiguity. The parser should depend more strongly on cues that the front end is likely to label correctly, and less on cues that the front end is likely to confuse. It was shown that performance could be improved by clustering vowels into a single equivalence class when the front end is very likely to confuse vowels. Relaxing vocalic distinctions has the obvious disadvantage; by sacrificing certain distinctive features, the parser and matcher will find too many candidates. There is no way to beat the trade-off between type I and type II errors, short of improving the front end.

One might argue that a probabilistic system is more suitable than a categorical one for dealing with errors in the front end. Indeed, there are automatic procedures for tuning (optimizing) the decoder to the front end such as those developed at IBM [48]; no such techniques have been developed for categorical frameworks. (Perhaps our categorical system should be recast into a probabilistic framework in the final version.) Of course the probabilistic system still can't get around the trade-off between type I and type II errors. There is room for real improvement at the front end.
In the second half of this chapter, we outlined a proposal for improving the front end. By decomposing segments into distinctive features, it is possible to exploit the rich structure imposed at the distinctive feature level. In the example of stops, the analysis in terms of distinctive features has a number of advantages. First, it has many fewer degrees of freedom (4 + 9) than the analysis in terms of segments (36). Secondly, the parameters are easier to estimate since they are based on more frequent events. Thirdly, recognition scores will be higher because there are fewer alternatives at each step.\footnote{This third advantage may be artificial.} Fourthly, the analysis in terms of distinctive features is compatible with Stevens' Invariance Theory. Fifthly, the analysis can be incorporated into a lexical organization as proposed in section 8.6.6, so that it will be possible to recover from single feature errors in an efficient way.
9. Conclusion

This thesis has argued that allophonic constraints provide an important source of information that can be useful for speech recognition. Allophonic cues reveal important properties of the suprasegmental context because they are variant. In this respect, allophonic cues are more useful than invariant cues; invariant cues can’t tell us anything about the syllable structure and foot structure, since, by definition, they are insensitive to context.

9.1 Review of the Standard Position

We have presented a number of arguments for viewing allophonic cues as a source of constraint. Three arguments for the standard position were reviewed in chapter 1. Recall the discussion of the sentences:

(236a) Did you hit it to Tom?
(236b) Tell the gardener to plant some tulips.
(236c) We make all of our children.

These examples were presented in order to show that lexical retrieval is difficult because of The Invariance Problem: “there are extensive rule-governed changes to the way that words are pronounced in different sentence contexts.” These authors concluded that it was necessary to introduce higher level constraints (e.g., syntax and semantics) in order to decode such sentences. Recall that Klatt, for instance, concluded with the following:

(237) “All of these phonological phenomena result in lexical ambiguity so that even the best lexical hypothesis routines will propose many words that are not in the original sentence, simply due to fortuitous matches. The third step in the process [speech understanding] would therefore be to use syntactic - semantic modules to weed out the false lexical hypotheses and put together a word string that represents what was spoken.” [64, p. 1346]

We have been advocating a different view. We believe that it is not necessary to resort to syntax and semantics in order to compensate for weaknesses in the front end. Doddington makes this point very forcefully:
(238) "A favorite myth among many speech recognition scientists is that speech must be truly understood before it can be adequately recognized. Although intimate knowledge of the language is certainly crucial to human recognition in adverse environments, such knowledge cannot hope to compensate for a devastating lack of perceptual adequacy at the acoustic/syllabic level. Emphasis on 'understanding' what is being said needs to be deferred until the basic level of perceptual adequacy is substantially improved." [28, p 557]

We agree with Doddington; emphasis should be placed on improving the front end. Only then should we pursue higher level constraints. Admittedly, syntax and semantics would probably help. Recall that we had trouble disambiguating between competing hypotheses such as see, sea and the letter C. Although higher level constraints might solve this problem for us, we have chosen to explicitly avoid the application of higher level constraints in order to explore the capabilities of lower level constraints at the segmental, suprasegmental, and distinctive feature levels. See appendix II for more discussion of the application of syntax and semantics in speech recognition.

Instead of resorting to the application of syntax and semantics in examples like (235), we advocated a parsing and matching strategy. We argued that it is possible to parse transcriptions such as:

(239a) [d][ø][h][l][t][h][ã][m]

into

(239b) [d][j][ø] # [h][l] # [t][h] # [t][h][ã][m]

because of phonotactic and allophonic constraints such as:

(240a) /h/ is "always" syllable initial,
(240b) [l] is "always" syllable final,
(240c) [t] is "always" syllable final, and
(240d) [t][h] is "always" syllable initial.

At this point, with the transcription parsed into manageable syllable sized constituents, it is relatively easy to match constituents against the lexicon in order to decode the utterance.
Hence if we exploit the phonotactic and allophonic constraints on syllable structure, we are in a superior position to hypothesize word candidates.

9.2 Review of Nakatani’s Position

As mentioned in the introduction, !loyd Nakatani and his co-workers have been advocating a similar position for many years [87, 88, 89, 90, 91]. They argue that allophonic and prosodic cues should be applied to constrain the possible locations of word boundaries.

(241) “We see the fission and fusion processes fitting into a comprehensive model of speech perception... For example, a word boundary must occur somewhere between the syllabic nuclei of ‘gray twine’... The precise location of the word boundary is indicated by the allophonic cues. In this example, the word boundary must fall before the /t/ because its strong aspiration is characteristic of word-initial allophones of voiceless stops. [In contrast, in ‘great wine’, the word boundary must fall after the /t/ because its realization is characteristic of word-final allophones.] ...a word boundary can be detected from allophonic cues alone...” [91, p. 3]

We have formalized Nakatani’s approach from a computational point of view with our parsing and matching strategy. It is very exciting to us that our computational model can be so close to a theory of human perception such as Nakatani’s. In the past, most computational models (e.g., IBM’s statistical model) have little in common with psychological models. It might be interesting to pursue these questions of “psychological reality” in future research.

Many of Nakatani’s examples provide strong evidence in support of our model. For example, gray twine/great wine demonstrate very nicely the use of variant allophonic cues into to constraint the suprasegmental context. We have presented quite a number of example like his. The following table appeared in section §2.4.1.
It is widely agreed that listeners can make use of acoustic correlates in order to disambiguate pairs such as these. Most of the arguments are usually supported with "armchair intuition". In our laboratory, Lamel is currently examining acoustic evidence, hoping to evaluate proposed correlates in more rigorous way. Her preliminary study [72] looks positive, though there she has encountered some difficult pairs, especially bay block/babe lock.

Our parsing and matching approach will be able to take advantage of whatever constraints that should come out of acoustic studies like Lamel's. On the other hand, we are not completely dependent on uniformly positive results. If Lamel should find, for instance, that she cannot reliably distinguish /b-l/ (e.g., bay block) from /b-l/ (e.g., babe lock), then our parsing and matching strategy can treat the sequence /b/ followed by /l/ as ambiguous.

9.3 Review of the Constituency Hypothesis

In the previous section we reviewed the claim that word/syllable junctures are (often) marked with acoustic correlates. Thus it is (often) possible to parse constituents in a bottom-up way. In this section, we want to say something stronger: parsing captures important linguistic generalizations. Let us consider the following definition of linguistically significant generalization from *A First Dictionary of Linguistics and Phonetics* [24 pp. 211-212] and discuss how parsing fits in.
"linguistically significant generalization": A term used especially in generative grammar to refer to the kind of analytic statement which it is hoped the grammatical analysis will provide. The aim of the grammar is not just to generate all and only the grammatical sentences of a language, but to do this in such a way that those relationships felt to be significant by native-speakers are expressed in an economical way. For example, a grammar which generated active sentences separately from passive once, or questions from statements, and which failed to show how these are inter-related, would be missing linguistically significant generalizations.

Phrase-structure parsing is based on the assumption that phrases (or constituents) are 'significant'. This claim has been formulated as follows:

(243) The Constituency Hypothesis: Many allophonic and phonological processes share the same environments.

As evidence for the constituency hypothesis, we discussed a number of allophonic rules and argued that they shared more or less the same environments.

(244a) foot-initial: aspiration, retroflex, sonorant devoicing,
(244b) foot-internal: flapping, flap-like /v, ð, ʈ/ /h/-deletion
(244c) foot-final: glottalized stops, unreleased stops
(244d) non-foot-initial: nasal deletion, nasal lengthening, epenthetic stop insertion

For each of these rules, we presented a number of examples. Recall the table presented in sections §4.2.3 and §7.3.2. This table is intended to show that rules like retroflex, sonorant devoicing and rounding mark apply only in foot-initial onset position. If this is correct, then we can use these rules to find foot junctures both in examples like (245) below and in examples like (243) above.

Insertion
Similarly, we can use foot-internal, foot-final and non-foot-initial rules as cues for foot structure.

9.4 Review of Phonotactic Constraints

We can also take advantage of a number of phonotactic constraints in order to parse the segmentation lattice into suprasegmental constituents. Excluding the affix position for the moment, there are six basic types of phonotactic constraints on syllable structure:155

(246a) **Length**: there are at most three consonants before the vowel, and two afterward

(246b) **Sonority**: the relative prominence or sonority (Jespersen) must decrease from the nucleus towards the margins

(246c) **Idiosyncratic systematic gaps**: English happens not to allow certain possibilities, e.g. voiced stops after nasals or [u], even though these meet the previous two conditions.

(246d) **Voicing assimilation**: In general, two adjacent consonants must agree in voicing.

(246e) **Place assimilation and dissimilation**: In some contexts (e.g., nasalized codas), two adjacent consonants

\[\begin{array}{l}
(245) \\
<table>
<thead>
<tr>
<th>d</th>
<th>p</th>
<th>t</th>
<th>k</th>
<th>b</th>
<th>d</th>
<th>a</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>de-prive</td>
<td>de-crease</td>
<td>cele-bration</td>
<td>a-ddress</td>
<td>de-grade</td>
<td>di-plomy</td>
<td>dip-lomatic</td>
<td>a-cquire</td>
</tr>
<tr>
<td>dep-ivation</td>
<td>dec-riment</td>
<td>o-bligatory</td>
<td>add-ress</td>
<td>deg-radation</td>
<td>de-cline</td>
<td>dec-lination</td>
<td>acq-ui-sition</td>
</tr>
</tbody>
</table>
\end{array}\]

---

155. Actually, it is not clear that phonotactic constraints should be stated as the level of syllable structure. There seem to be a number of facts that can be stated more naturally at the level of foot structure. For instance, a number of consonants do not appear in foot internal position (e.g., voiceless /h/, /y/, /w/, light /l/, light /r/) and a number of other consonants do not appear in foot initial position (e.g., voiced (deleted) /h/, intervocalic /η/, Y off-glides, W rounding, dark /l/, dark /r/). These facts are difficult to explain in a syllable based theory without making reference to the stress structure. (One could push these problems off to the phonetic component of the grammar. They do fall in the crack between phonology and phonetics.)
consonants must agree in place of articulation. In other contexts (e.g., onset clusters), two adjacent consonants must disagree in place of articulation. Hence the ungrammaticality of */tl/, */dl/, */pw/ and */bw/ in onset position.

(246f) Barnwell’s Vowel Resyllabification: Lax vowels (except schwa) appear only in closed syllables.¹⁵⁶

These constraints fix the syllable structure in more cases than you would have thought. In fact, they can even be redundant in certain cases. In these cases the parser can make use of the redundancy into the check the transcription for internal consistency.

9.5 Comparison with Syntactic Notions of Constituency

Despite all of this evidence for constituents in phonology, the battle over the constituent has been long and stormy. In the literature, there are two very common objections to the syllable (and most other suprasegmental constituents). First, there doesn’t seem to be a very good acoustic correlate, as we observed in section §1.4.2:

(247) “... the syllable is probably the most elusive of all phonological/phonemic notions. There is a vast and inconclusive literature on the subject ... The primary reason for the great confusion surrounding the syllable is a lack of any adequate phonetic definition. At this point the only thing that can be said with any confidence is that the syllable is an abstract programming unit in terms of which speech is articulated. The chest pulse (units of musculature activity controlling the flow of air from the lungs) is the only articulatory correlate for the syllable that has been discovered. Furthermore, studies by Ladefoged (1967) indicate that the chest pulse is not always an accurate diagnostic for the syllable. ...” [56 pp. 255-256]

Secondly, the syllable (and other constituents) are not strictly necessary. Recall Anderson’s summary of the “Classical” position.

(248) “The ‘classical’ model of generative phonology (as presented in, e.g., Chomsky & Halle, Schane) recognized only one sort of structural unit in phonological and phonetic representations: the segment. There was thus no explicit provision for syllables (or other units, such as prosodic feet and the like) as significant elements contributing to the organization of speech. This was not, as some have suggested, simple oversight or failure of imagination, but rather a matter of principle: while traditional phonetic descriptions of course frequently refer to syllabic structure, and many

¹⁵⁶. We might also want to add that tense and reduced vowels appear in closed syllables, but this needs more work.
informal statements of processes in Chomsky & Halle do so as well, the convenience of this unit for ordinary language description does not ipso facto establish its linguistic significance. If it were to turn out that all the statements we might want to make in terms of syllables were, when expressed formally, representable simply in terms of (strings of) segments, without important loss of generality, this would suggest that the more parsimonious and restrictive theory which only allowed reference to such units was in fact essentially correct, and thus to be preferred. It was the attempt to establish this program that lay behind the exclusion of syllable structure from the formalism of early generative phonology.” [5, p. 546]

Neither of these objections are taken as seriously today as they once were. But why? No one has yet discovered a good acoustic correlate of a syllable. Nor has anyone found a fact that cannot be accounted for in the “classical” model.157

So why do we believe in the syllable? We might argue for the syllable by analogy to constituents in syntax. Consider the sentence. One rarely finds objections to the constituent-hood of the sentence. And yet, both objections to the constituent-hood of the syllable apply equally well to the sentence. First, there are no surface (acoustic) correlates of the sentence; the sentence is an abstract unit in the underlying structure.158 Secondly, the sentence constituent is not strictly necessary. We can probably replace the sentence constituent with a complicated description in terms of noun phrases and verb phrases and the like, just as we can replace the syllable with a complicated description in terms of segments. Thus, there must be something wrong with the arguments against the syllable, because they would also seem to exclude the sentence from grammar.

Why is the sentence considered to be a constituent? There are two standard reasons in the literature. First there is a strong intuition that sentences have “meaning” (whatever that is). Secondly, the sentence seems to simplify159 a number of syntactic processes, such as question formation, wh-movement, and so forth. A typical text book formulates these two arguments as follows:

157. Kahn argues quite persuasively that flapping illustrates a case where the “classical” model misses crucial generalizations. Despite the fact that we have adopted many of his arguments, they are not entirely convincing. They may not stand up to the test of time.
158. It seems sort of strange to require that a constituent have a unique 1-1 onto correlate. The point of constituency is to capture generalizations. There can be no generalization to capture unless there are multiple correlates. If syllables were perfectly correlated with some acoustic property (e.g., chest pulses), then we might as well replace the term syllable with the term chest pulse, since they would be equivalent.
159. The weak formulation “seems to simplify” is probably more appropriate than the strong language found in the quotation (“impossible to state ... without”) until a formal proof can be constructed.
(249) "We have now cited two kinds of evidence in favor of the hypothesis that sentences are structured. First, if we do not assume that sentences are structured — that words are grouped together in constituents — then we have no way to explain how a sentence consisting of a set of unambiguous words can nevertheless have an ambiguous meaning. Second, it is impossible to state certain grammatical rules such as the Question Rule for English without appealing to a constituent structure." [4 p. 147]

We maintain that both arguments justify the constituent-hood of the syllable about as "well" as they justify the constituent of the sentence. The syllable satisfies the first criterion: intuition. We all agree (more or less) on what a syllable is. Naive speakers have a sense of what is a possible syllable and what is not, even for syllables that they haven't heard before.

(250) "The notion of syllable, in short, is very real to native speakers, and is often used in a quasi-technical sense in everyday conversation (e.g., shall I put it in words of one syllable?)." A First Dictionary of Linguistics and Phonetics [24 p. 342]

The syllable also passes the second criterion: a common unit found in many "transformational" processes. The syllable seems to be clearly responsible for hyphenation facts, and (if one accepts the arguments of Kahn and his followers) for a number of allophonic facts. Thus we see no reason for considering the syllable to be any less of a constituent than the sentence. Having adopted this conclusion, then it only seems natural that we should construct syllable parsers for processing speech just as we have constructed sentence parsers for processing text.

9.6 Contributions

To our knowledge, this thesis is the first attempt to provide a phrase-structure framework as a computational model for processing well-known allophonic and phonotactic constraints. We have shown in concrete terms how all of these well-known constraints can be utilized within a uniform structure, and by virtue of numerous examples, that all of these levels contribute to the quality of the overall parse. We have put together existing knowledge in a way that leads to new insight and demonstrable improvement in performance, at least at the example level. There are no numbers, we realize, and so a full assessment must

---

160. Admittedly, none of these arguments seem to be very good.
161. Admittedly syllables don't have meaning. However, meaning is probably not the relevant test for constituency, as Chomsky [17, p. 15] attempted to demonstrate with the sentence: Colorless green ideas sleep furiously. This is a syntactically well-formed sentence despite the fact that it is semantically meaningless.
await the arrival of a decent front end.

We would have liked to have had the chance to construct the front end ourselves, but this turned out to be infeasible. Speech recognition has progressed to the point that no one person can do it all. Back in the mid 1970's, it was possible to build a one-thesis complete system, as Bruce Lowerre demonstrated with HARPY, but that that's probably the end of an era. In the next few years, we will see a number of highly focused doctoral dissertations dedicated to very small sub-parts of the total problem (e.g., Lori Lamel's forthcoming study of sonorant clusters, Mike Riley's formant finder). This trend is indicative of a growing awareness in speech recognition research that we need to develop a much better understanding of speech.

There has been a similar trend in Artificial Intelligence. Most of the early work sought to discover general purpose searching methods (e.g., beam search, hill climbing, theorem proving). However, history has shown that domain specific expertise is an essential ingredient to a truly successful AI project (e.g., Macsyma and Dendral).

(251) "The study of generality in problem solving has been dominated by a point of view that calls for the design of 'universal' methods an 'universal' problem representations. These are the GPS-like and Advice-Taker-like models. This approach to generality has great appeal, but there are difficulties intrinsic to it: the difficulty of translating specific tasks into the general representation; and the tradeoff between generality and the power of the methods." Feigenbaum, Buchanan, and Lederberg [34, p. 187]

In arguing this thesis, that the phrase-structure framework provides an attractive computational model for allophonic and phonotactic constraints, we have touched upon a number of secondary points, which may be important contributions in their own right. In this category, we might place the novel treatment of parsing in chapter 5 and the automatic syllabification of dictionary pronunciations in section §7.4. The parsing implementation may turn out to be important because it allows one to state many agreement facts in a very natural way by nesting conjunctive constraints (e.g., M&) outside of disjunctive constraints (e.g., M+), in a way that is generally not possible in many phrase-structure formalisms. The syllabification algorithm has potential to be significant in practical applications because dictionaries are notoriously unreliable with respect to syllable boundaries.
References


6. Informally Collected Notes on Phonological Rules gathered from the participants in the ARPA speech project.


85. MITalk Lexicon, course notes from special summer course, 1979.


Appendix I - The Organization of the Lexicon

This thesis has concentrated more on the parsing aspects of lexical retrieval than on the matching aspects. Nevertheless, it might be useful to say a few words about the matching routines as there are in the current implementation, and how they might be improved in future implementations.

An important recurrent theme in the theory of algorithms is “Divide and Conquer”. To illustrate the issues, consider the following example borrowed from Smith's thesis [114, pp. 17-22]. Suppose we had the following “toy” lexicon:

(252a) Word₁: abcde
(252b) Word₂: dedfg
(252c) Word₃: abcdfg
(252d) Word₄: deabc

This “toy” lexicon has only four entries. Our demonstration system is typically run with about 3k dictionary entries. The MITalk lexicon has about 15k morphemes. Webster's pocket dictionary lists approximately 20k entries. Even larger lexicons may be required in the ultimate recognition system. With such large lexicons, it is extremely important that they be organized for efficient retrieval. The difference between a binary search and a linear one is extremely significant. Let us first discuss linear time retrieval algorithms and then move on to more efficient techniques.

I.1. Linear Representation and Linear Search

Smith observes that simply listing the words as we have done above is perhaps the simplest representation of this “toy” lexicon. The recognition algorithm could proceed in a straightforward fashion, comparing each lexical entry in sequence with the utterance, and selecting the closest match. Smith assigns a storage cost of 25 symbols to the linear representation of the example above (21 symbols for the lexical entries + 4 symbols to denote the starting points of each entry) and a recognition cost of 21 matches (the sum of the length of the lexical entries). We will refer to this organization of the lexicon as a linear representation because storage costs grow linearly with the size of the lexicon.¹⁶² Similarly, this recognition algorithm is

¹⁶² That is, if one lexical entry requires a constant amount of memory (k bytes), then n lexical entries will require kn bytes.
called a linear search because its time costs grow linearly with the size of the lexicon.

Linear methods are currently employed in holistic systems\(^{163}\) (e.g., dynamic time warping [44]), because the holistic assumption blocks any attempt to “divide and conquer”. On the other hand, in compositional frameworks, it is possible to decrease storage and recognition costs by re-organizing the lexicon so as to increase sharing among common parts of lexical entries. Thus, for example, we might want to represent the morpheme \(re\) just once in the lexicon, not multiple times for each lexical entry that it appears in (e.g., \(redo\), \(reduce\), \(return\), ...). In this way, sharing can reduce memory costs in the obvious way; but in addition, sharing also decreases recognition cost because the shared data will be compared against the utterance once, not multiple times for each lexical entry that it appears in. We will discuss three types of sharing that have been introduced in the past.

(253a) Non-Recursive Discrimination Networks
(253b) Recursive Discrimination Networks
(253c) Hash Tables Based on Equivalence Class Abstractions

I.2. Non-Recursive Discrimination Networks

The lexical entries can be placed into a discrimination network as illustrated below:

(254) \(\text{branch} (a \ b \ c \ d \ (\text{branch} (e \ \text{Word}_1) \ (f \ g \ \text{Word}_2)) \ (d \ e \ (\text{branch} (c \ f \ g \ \text{Word}_3) \ (a \ b \ c \ \text{Word}_4))))\)

A discrimination network is a tree structure, formed by merging all common initial sequences.\(^{164}\) The point at which two lexical entries diverge is denoted by the key word \textit{branch}.\(^{165}\) Discrimination networks can save both asymptotic storage costs and asymptotic recognition costs; for very large lexicons, discrimination

\(^{163}\) Holistic systems are opposed to compositional systems such as our own proposal. Holistic systems tend to work by matching the input utterance against a set of templates in the lexicon (Templates are used to represent lexicon entries: a template is a speech utterance with some (but not much) data compression.) Holistic systems have enjoyed some success in small vocabulary tasks, especially for speaker dependent, isolated words. For large vocabulary tasks, it is questionable whether holistic systems are appropriate because of the space and time requirements, and because of problems associated with the collection of suitable templates.

\(^{164}\) Alternatively, one could share all common final sequences, which might be more reasonable for English, because of the way that morphological processes seem to work.

\(^{165}\) Actually this representation can be made somewhat more compact by removing the branch symbols and the NILs at the ends of the lists. These simplifications are included merely for convenience.
networks consume logarithmic space and time. However, there may be a significant initial overhead due to the pointers. Smith assigns a storage cost of 15 symbols and 19 pointers\textsuperscript{166} to the structure above, and a recognition cost of 15 matches per input symbol.

1.3. Recursive Discrimination Networks

Note that the phrases "abc", "de" and "dfg" appear twice in the network above. It is possible to take advantage of these common phrases by introducing an intermediate level of representation in between the segment level and the word level. Then we could split the discrimination network presented in (254) above into two: (255a) tells us how segments can be put together to form phrases, and (255b) tells us how phrases can be combined to form words.

\[(255a) \text{ (branch (a b c Phrase\(_1\)) (d (branch (e Phrase\(_2\)) (f g Phrase\(_3\))))}}\]

\[(255b) \text{ (branch (Phrase\(_1\)) (branch (Phrase\(_2\) Word\(_1\)) (Phrase\(_3\) Word\(_3\)))) (Phrase\(_2\)) (branch (Phrase\(_3\) Word\(_2\)) (Phrase\(_1\) Word\(_4\))))} \]

If the phrase level is capturing a significant generalization, then splitting the discrimination network in this way will save storage and recognition costs. Smith argues quite convincingly that it is worthwhile to treat syllables as an intermediate level of representation between segments and words because certain frequent syllables are common to many words. However, in this toy example, the phrase level is of marginal utility. The phrases "abc", "de" and "dfg" are common to just a few words; it may not be worthwhile to maintain the extra pointer structure overhead required by the intermediate phrase level. Smith notes that the storage costs are actually somewhat higher, because of the overhead incurred by introducing the additional phrases. (Storage cost: 20 symbols + 20 pointers; recognition cost: 9 matches per input symbol.)

Smith actually introduced two intermediate levels: the syllable level and the syllpart level (i.e., onset, peak, coda). In our implementation, we have essentially replicated Smith's design. We might like to extend this approach somewhat; we might represent words in terms of feet, feet in terms of syllables, syllables in terms of onsets and rhymes, and so forth. In this way, we hope to keep postlexical processes (e.g., aspiration, flapping, retroflex) out of the lexicon (recall the discussion in section §1.4.1.2). Ideally, the notion of

\textsuperscript{166} This count does not include the three pointers to the atom branch, or the pointers to NIL, because these pointers are unnecessary.
constituency that motivates the most elegant formulation of the allophonic rules is compatible with the notion of constituency that maximizes sharing in the lexicon.

1.4. Hash Tables Based on Equivalence Class Abstractions

Finally, we will consider a method that was not discussed by Smith, but has been put to some use in a number of the ARPA speech projects, and the MIT Morse Code Project [38], and is being revived in some recent work by Shipman and Zue [111]. Discrimination networks treat lexical entries as a string of atomic segments. Sharing is limited to the case where a substring of segments is found in two or more entries. We might want to arrange our lexicon so that sharing reflects similarity among entries. Constituents (e.g., words, morphemes, feet, syllables, clusters) that “sound similar” will share relatively more with each other and constituents that “sound distinct” will share relatively less. So for example, one might argue that bend and band are relatively similar, differing in only one feature. Much of the representation of these two words should be shared. This will not result with the current discrimination schemes outlined above. In this section, we will explore ways in which the organization of the lexicon can reflect our intuitive notion of similarity.

Recall the example discussed in §8.6.6. We proposed in that example that lexical entries could be indexed three times: once as a sequence of manner features, once as a sequence of place features, and once as a sequence of voicing features. For example, the word splint could be indexed under the three keys:

\[(256a) \text{[fricative] [stop] [sonorant] [vowel] [nasal] [stop]} \quad \text{manner}\]
\[(256b) \text{[coronal] [labial] [coronal] [??] [coronal] [coronal]} \quad \text{place}\]
\[(256c) \text{[voiceless] [voiceless] [voiced] [voiced] [voiced] [voiceless]} \quad \text{voicing}\]

We originally presented this scheme in order to show how a categorical scheme could recover from single feature errors in the transcription. Recall that we would access the lexicon on all three of these keys. Words that were found on all three keys would be considered to be extremely plausible candidates. Words found on two out of three would be considered to be less plausible, but still possible. Words found on only one key might be considered too implausible to be worth further investigation. In this way, we might augment the matching procedure in order to recover from certain transcription errors.

Note that this representation scheme captures the generalization between words like bend and band. Suppose that each set of features is stored in a separate discrimination network. In this way, bend and band would share quite a bit. In the manner and voicing discrimination networks, they would be identical; the only differences would be found in the representation of place features.
I.5. Shipman and Zue

Shipman and Zue have proposed a scheme much like the one above. They map words into sequences of gross manner characterizations. So for example, the word *splint* would be represented as the sequence:

\[(257) \quad \text{[strong fricative]} \ [\text{stop}] \ [\text{liquid or glide}] \ [\text{vowel}] \ [\text{nasal}] \ [\text{stop}]\]

We can now *invert* that dictionary on this mapping function (hashing function), so we can quickly retrieve the set of words with a particular gross specification. This inverted dictionary (or hash table) helps the recognizer narrow down the set of potential candidates with a very quick-and-dirty first pass filter. On the first pass, the front end produces a labeling and segmentation lattice of gross manner features. The lexical retrieval device looks up all sub-lattices in the inverted dictionary and returns all word candidates that match the gross manner specifications. These candidates are examined more carefully by a more expensive second pass.

This approach depends on three basic assumptions. First, Shipman and Zue are assuming that it is possible to construct an accurate front end acoustic analyzer that can obtain the gross labels in a bottom-up way from the speech signal. This assumption is supported by experience with human spectrogram reading. Confusion statistics such as those reported in [106] indicate that manner features are relatively easy to achieve in most cases. However, no such front end has ever been built, as far as we know.

Secondly, Shipman and Zue are assuming that the first pass will weed out enough of the lexicon so that the first pass is worth doing. In order to justify this second assumption, Shipman and Zue have gathered some distribution statistics on Webster’s Pocket Dictionary [84]. They found that the first pass would reduce the search space by two or three orders of magnitude depending on how you count. About one third of this 20,000 word lexicon is uniquely specified by the gross specification. This would almost certainly reduce the search space down to the point where holistic methods would be practical as a second pass.

I.6. Morse Code

The Shipman and Zue approach is similar to the algorithm implemented in the Comdec Morse Code decoder [38]. The Comdec decoder also employed a gross characterization in order to reduce the search space with a quick-and-dirty first pass. Comdec uses the run-length sequence of dots and dashes as the gross characterization. It turns out that this is a very good hashing function. Almost all words in Comdec’s lexicon (~4,000 entries) are uniquely identified by the run-length sequence of dots and dashes. On the second pass, Comdec would look more carefully at the spaces between the dots and dashes in order to find the best
decoding.\(^\text{167}\)

Indexing the lexicon on gross classifications (e.g., the run-length sequence of dots and dashes in the morse code domain or manner features in the speech domain) has two major advantages:

(258a) search space reduction
(258b) error recovery

Shipman and Zue stress the first advantage. By classifying by their gross classifications, it is possible to reduce the space of possible candidates from 20,000 to less than 100 (often down to a unique word). In the morse code domain, the gross classification is an even better hashing function. The gross classification narrows the search space down from a few thousand words to a unique hypothesis in almost all cases.

The second advantage is also important. We have discussed how gross feature classifications can help recover from errors in the speech domain. The same is true in morse code. By indexing the lexicon in terms of run-length sequences, Comdec could easily recover from space confusions. Dot and dash confusions are harder for Comdec to recover from.\(^\text{168}\) Fortunately, they are not as common as space confusions.

The Shipman and Zue approach is very similar to the one we proposed above. They both group words (or constituents) into gross equivalence classes based on distinctive features before accessing the lexicon. There are a number of differences of course. First, we have generalized the approach to index along several gross classifications in parallel. This modification improves the potential for error recovery. It also makes the gross classification approach more compatible with the theory of distinctive features. Secondly, we have generalized the Shipman and Zue approach to index constituents, not just words. In this way, we can make use of the constituent structure (and allophonic constraints).\(^\text{169}\) Thirdly, we have been assuming that the inverted lexicon is stored as a recursive discrimination network, a generalization of hash tables. These

---

\(^{167}\) In morse code, there are three types of spaces: inter-word, inter-letter and inter-element. Ideally, inter-word spaces are longer than inter-letter spaces which are longer than inter-element spaces. In practice, the mean word space is longer than the mean letter space which is longer than the mean element space. However, there is a large overlap in the distribution of letter spaces and elements spaces, and some overlap between word spaces and letter spaces.

\(^{168}\) For dealing with occasional dot/dash confusion, Comdec did have a second hash-table, called the mark-error dictionary. This dictionary stored all words that were one mark-error away from matching a particular run-length sequence. In this way, Comdec could recover from a single mark confusion (in certain cases).

\(^{169}\) Constituent structure also turns out to be useful in matching, when the matcher doesn’t know the location of word boundaries. The constituent structure is useful for decomposing words into smaller and smaller constituents. The matcher consults the lexicon for the smallest constituents first. Only if they are in the lexicon will the matcher consider larger constituents on top of them. In this way, the lexicon can avoid looking up words in the lexicon if their sub-pieces aren’t there. Without a constituent structure, the matcher has to take all substrings of the input lattice and look them up in the lexicon.
improvements, though, are relatively minor. It is much more important to select the appropriate gross classification.

1.7. Selecting the Appropriate Gross Classification

The optimal classification\textsuperscript{170} must meet two constraints:

(259a) sensitive to (most) distinctive features
(259b) insensitive to (most) non-distinctive features

The classification ought to be rich enough so that it can discriminate among the words in the lexicon. The first criterion (259a) will assure this; only a few words will have the same sequence of distinctive features. On the other hand, the classification ought to be abstract enough so that it can abstract away from features that are likely to cause more trouble than they are worth. We don’t want the first pass to be concerned with features that tend to be confused by the front end, that don’t distinguish lexical entries very well, that depend on extraneous factors such as speaker, dialect, mood, etc. The second criterion assures this.

Some people have objected to Shipman and Zue’s scheme along criterion (259b). It has been observed that manner features are not always distinctive. Manner features can be “deleted” in many contexts (e.g., /h/ deletion is prohibition, /n/ deletion in can’t, /b/ deletion in lamb, gemination in gas shortage) and “inserted” in others (e.g., /U/ insertion in sense, hence and false). The same objection can be raised against our proposal. It might be reasonable to take context into account in the gross classification scheme. Perhaps, we might want to abstract away from potential trouble spots in the gross classifications. For example, two fricatives in a row as in gas shortage might be classified as a single fricative. In this way, the gross classification scheme would be insensitive to gemination, a potentially troublesome area for the front end. We might take a similar approach with many of the other “deletion” and “insertion” rules. These classifications schemes, though, conflate distinctive features that will make a difference. If we do this, it will be important that these oversights be corrected in the second pass.

\textsuperscript{170} There are some strong parallels between this gross classification function and the canonicalization function employed in our matcher. Both functions try to preserve important robust distinctions and smooth over irrelevant/unreliable ones.
1.8. Summary

In summary, we have briefly mentioned four representation systems in this appendix: linear lists, non-recursive discrimination networks, recursive discrimination network, and hash tables based on equivalence classes. The first two were quickly rejected. It was observed that linear lists are too expensive in space and time. We also discarded non-recursive discrimination networks because recursive discrimination networks seem to be so much richer for expressing linguistically meaningful levels of representation. The current implementation employs recursive discrimination networks. However, we are not very committed to this design choice. One could probably show that hash tables are more suitable for certain purposes. (An earlier version of the program used the hash table scheme.)
Appendix II - Don't Depend Upon Syntax and Semantics

Throughout this thesis we have avoided the use of higher level constraints. In this appendix, we will attempt to justify this decision. At first, in the early 1970's, there was a strong emphasis on higher level constraints. More recently, many researchers are changing their minds for two reasons. First, experience has shown that an over-dependence on higher level constraints can lead to some embarrassing results as we will see. Secondly, many of the old arguments for higher level constraints don't seem as forceful as they once did. It was once believed that the speech signal was too impoverished to support direct recognition without first “understanding” the message. This position is not as popular as it once was. As we learn more and more about low level allophonic and phonotactic constraints, the speech signal appears to be much richer. There is much greater hope that the speech signal can be recognized without being understood first. It is currently possible for certain people such as Victor Zue to read spectrograms by hand; we believe that it is just a matter of time before machines can read them too.

II.1. Higher Level vs. Lower Level Constraints

During the ARPA speech project, there was considerable debate between those advocating the engineering of syntax and semantics to patch up the inadequacies of a quick-and-dirty front end, those emphasizing basic research into low-level phonetics. Klatt summarized the two positions as follows:

(260) “There were two possible ways to meet the ARPA goals: (1) simplify the general speech recognition problem by finding ways to apply syntactic and semantic constraints and (2) improve upon previous speech recognition capabilities. As noted above, the steering committee emphasized the first alternative and recommended that funding be given to research groups that were composed mainly of computer scientists, not speech scientists. It turned out that various research groups tried different combinations of the two strategies, but the only clearly successful speech understanding system, Harpy, relied heavily on the first technique. In fact, if the ARPA project were to be judged on its contributions to speech recognition and the speech sciences, rather than judging it against its stated goals, a more negative appraisal might have to be given.” [64, p. 1345]

As Klatt notes, the ARPA speech project emphasized the use of syntax and semantics, and it was not a great success. Perhaps it would have been more profitable to have adopted the second position, that of improving speech recognition capabilities at the phonetic level.
II.2. Too Much Dependence in the Past

It is probably unreasonable to expect syntax and semantics to compensate for an inadequate front end. During the ARPA speech project, it is clear that too much dependence was placed on syntax and semantics. There was, for example, the reported incident where someone coughed in front of one of the ARPA speech understanding systems and the machine responded "I heard pawn to king four". (The machine happened to have been tuned for the chess domain.) It seems that the semantics constraints were so tight, that the system didn't need the acoustic signal. One might go so far as to say that this machine was practicing telepathy. In fairness, this is a most extreme incident. It is probably reasonable to say that most of the ARPA speech systems made considerable use of the acoustic signal, but perhaps, not as much as they should have.

Let us consider a word lattice produced by Hearsay and ask whether semantic considerations are really the best criteria for distinguishing between competing hypotheses, or whether phonetic considerations might be preferable. The word lattice given in [31, p. 373] contains the words Weizenbaum and Feigenbaum competing for roughly the same region of the utterance: Are any by Feigenbaum and Feldman?

(261) Are any by {Weizenbaum, Feigenbaum} and Feldman?

Should these two words be distinguished on top-down semantic grounds or on bottom-up phonetic grounds? We suggest the bottom-up approach. The words feigenbaum and weizenbaum differ in two places: /f/ versus /w/ and /z/ versus /g/. Both of these differences are extremely easy to distinguish on a spectrogram. They are not a minimal pair like /v/ and /b/ where a confusion might be expected. They are about as different as two segments can be. Both confusions differ on two distinctive features.

(262a) /f/ and /w/ differ on voicing and manner;
(262b) /z/ and /g/ disagree on place and manner.

Double feature errors are quite rare as we have discussed in section §8.6.1. We suspect that two double feature errors in the same word ought to be essentially impossible. In addition, spectrogram reading experiments (e.g., [106]) suggest that manner confusions are the least common of all confusions. In short, there is no reason why these two words should have been considered to be competing hypotheses.\textsuperscript{171}
The Weizenbaum/Feigenbaum confusion is just one of many acoustically distinct pairs found in the lattice reported in [31, p. 373]. Consider some more pairs from the same word lattice

(263a) done / given
(263b) cite / thought
(263c) models / ones
(263d) monitor / Mostow

and ask how these words might have been considered to be acoustically similar?

The HWIM word lattices also contain many acoustically dis-similar words as competing hypothesis. Consider the competing hypotheses

(264a) ten / been / did / give
(264b) we / give
(264c) greater / dealing / metal
(264d) led / not

taken from [9, p. 7]. These confusions do not seem reasonable. Higher level constraints cannot be expected to recover from these sorts of gross confusions at the front end (except in a "toy" domain).172 Perhaps higher level constraints should be expected to disambiguate among sea, see and C which really do sound alike.

II.3. How Much Can Higher Constraints Help?

On the other hand, higher level constraints bear less fruit than one might have thought. Part of the reason that higher level constraints were not very successful in the ARPA speech projects is that they are not as fruitful as one might have thought. From our own experience with large syntactic grammars (e.g., [82]), we have acquired a certain appreciation for the productivity of English syntax. It is our suspicion that syntax is so productive that many strings of random words can be organized into a syntactically grammatical parse tree.

171. Admittedly, these two words were proposed by the Verify module, which is working "top-down". However, there are numerous other examples from the same lattice, some of which we will mention, where acoustically distinct words are proposed in a bottom-up way for the same region of the utterance.

172. We should not criticize the ARPA speech project for assuming "toy" domains because that was part of the guidelines set up by the steering committee. On the other hand, since we are not assuming a "toy" domain, we do not want to adopt techniques from the ARPA speech project that cannot be scaled up to an unrestricted domain.
Note, for example, that syntax and semantics are unlikely to exclude either:

(265a) Did you hit it to Tom?
(265b) Did you hit it at Tom?

because they are both well-formed. English syntax and semantics are not sufficiently restrictive to compensate for an inadequate front end. It is not possible to "Hear what I mean": you have to listen.

Instead of resorting to syntax and semantics, we advocate a deeper investigation of allophonic processes. We suspect that the /t/’s will be different in the two sentences:

(266a) hit it to Tom $\rightarrow [\text{hit}^{\text{th}} \text{am}]$
    $\uparrow \uparrow$

(266b) hit it at Tom $\rightarrow [\text{hit}^{\text{th}} \text{am}]$
    $\uparrow \uparrow$

Thus allophonic cues are (probably) sufficient to distinguish between the two sentences. It is our suspicion that these low level cues are more robust than high level cues for deciding between decoding (266a) and decoding (266b).

One can improve the effectiveness of higher level constraints in various ways. For example, one can introduce domain restricted grammars with arbitrarily strong top-down constraints. Alternatively, one can introduce a probabilistic model such as the IBM tri-gram model. Both of these methods are somewhat successful within certain ranges. We have little to say about these techniques. They are probably worthwhile; but they are not the complete solution.

II.4. Detraction from the Important Low-Level Issues

One of the strongest arguments against the use of higher level constraints is that they detract from the important low level issues. Even if higher-level constraints were sufficient by themselves without lower-level constraints (which they aren’t), it would still be useful to focus on low-level constraints in order to explore the potentials of low-level constraints.
(267) "A favorite myth among many speech recognition scientists is that speech must be truly understood before it can be adequately recognized. Although intimate knowledge of the language is certainly crucial to human recognition in adverse environments, such knowledge cannot hope to compensate for a devastating lack of perceptual adequacy at the acoustic/syllabic level. Emphasis on 'understanding' what is being said needs to be deferred until the basic level of perceptual adequacy is substantially improved." [28, p 557]

This thesis has explicitly avoided the application of higher level constraints in an attempt to explore the capabilities of constraints at the phonetic, segmental, and suprasegmental levels.

II.5. New Directions: Recognition without Understanding

It was generally felt during the ARPA Speech Project that syntax and semantics were required in order to compensate for the inadequacies of phonetic level processing, which was extremely poor at that time. When the ARPA project began in the early 1970's, there was a strong emphasis on higher level constraints. Participants were encouraged to focus on the understanding aspects of speech understanding. However, in more recent years, more and more researchers are beginning to question the usefulness and necessity of syntax and semantic constraints in speech recognition. By the late 1970's, the consensus had moved much more toward a focus on low level recognition. Consider the following except taken from the IPS-77 Open Discussion as summarized in [75, pp. 562-563]:

(268) "A lively discussion showed the following to be a popular and controversial issue:

What is the significance of the distinction between 'understanding' versus 'recognition' of speech? Which type of systems should be pursued now, or in the foreseeable future?

It was observed that Harpy, the most successful of the ARPA SUS systems, was more of a recognizer than an 'understanding system', in that it used the acoustic data to select word sequences, admittedly within strict syntactic constraints, but with no (or almost no) semantics. Despite two decades of strong advocacy of the use of higher-level linguistic information to constrain the recognition task, controversy still exists about whether you absolutely need syntax and semantics, or whether our knowledge of them is sufficiently advanced to warrant their inclusion in recognizers of continuous speech. Some noted that syntax, semantics, and discourse increase recognition accuracy and are used by the successful human prototype system, and that speech has linguistic communication of intended meanings as its ultimate intent. Others argued that the basic pattern matching
techniques, statistical analyses, and mathematical methods (without elaborate syntax, semantics, or pragmatics) have produced the best successes in actual recognizers. Advocates of the more mathematical view criticized available semantic and syntactic models as ad hoc. They noted the diversion that such higher-order modules have been, in detracting from needed work on the acoustic ‘front-ends’ of recognizers. The question of which knowledge sources really warrant extensive effort and money being spent seems to be open for further discussion, though there seems to be strong agreement that a primary area was the ‘front-end’, involving acoustic, phonetic, prosodic, and phonological analyses, or the equivalent in acoustic pattern matching processes.”

The move away from syntax and semantics is becoming increasingly popular for two reasons. First, the ARPA speech project proved to be a bitter lesson. Many of the ARPA systems depended too severely on higher level constraints, and they suffered for it. Systems like HWIM and Hearsay, which tended to rely more on higher level constraints, were not as successful as systems like Harpy, when tended to employ fewer high level constraints. Secondly, the old arguments for higher level techniques are not as convincing as they once were. Recent developments indicate that lower level constraints bear more fruit than previously thought.

Having completed discussion of the first point, let us now move onto the section, the utility of lower-level constraints.

II.6. Lower-Level Constraints Bear More Fruit

We have developed a strong faith in low level constraints from watching Victor Zue read spectrograms and teach spectrogram reading to his students. He reads spectrograms in a bottom-up way without employing syntax and semantics. Several spectrogram reading experiments have been reported in the literature (e.g., [23, 106]), demonstrating that it is possible to correctly identify phonemes with approximately 80-90% agreement with linguistic transcriptions. We cite the abstract of one of these papers:

(269) “This paper presents a summary of several spectrogram reading experiments designed mainly to uncover the amount of phonetic information that is contained in the speech signal. The task involved identifying the phonetic contents of an utterance only from a visual examination of the spectrogram. The results generally support the notion that there is a great deal of phonetic information in the speech signal that can be extracted by the proper application of phonetic rules. From these results, it is argued that phonetic recognition in speech recognition systems can be improved substantially, and that improved phonetic recognition will lead to speech recognition systems of greatly increased complexity and sophistication.” Zue [128]
Our own experience with spectrogram reading supports Zue's conclusion. The front end can and will be improved substantially in the future.

In addition, it has been argued that the speech signal is too impoverished to permit direct recognition without the application of higher level constraints. Recall the three examples discussed in the introduction:

(270a) Did you hit it to Tom?
(270b) Tell the gardener to plan some more tulips.
(270c) We make all of our children

In each case, it has been argued in the literature that the lower level constraints are inadequate to disambiguate the speech signal, and therefore the listener/recognition device must employ some higher level syntactic and/or semantic inference. We have repeated one of the three arguments below:

"Consider the sentence 'Tell the gardener to plant some more tulips' spoken with the following phonetic structure:

/teIðgærdnəðplænsʌmʃrtULips/

The spoken version of the sentence contains several alternative word-level segmentations. 'Plant some' can be perceived as 'plan some' (due to stop deletion of the word final /t/ in 'plant'), 'some more' may be perceived as 'some ore and 'tulips' may be parsed as 'two lips.' As we will see, the intended segmentation is provided by the listener's use of context, and not by acoustic information." [22, p. 134]

We are not convinced that lower level constraints are inadequate. In this thesis, we have discussed a large number of allophonic and phonotactic constraints that may be very useful in examples such as this.

II.7. Summary

This appendix has presented two reasons for rejecting high level syntax and semantics. First, experience with high level constraints has not been promising. A few examples where presented to illustrate how previous systems may have depended too much on higher level constraints. The chess-cough example is a particularly amusing demonstration. The Weizenbaum/Feigenbaum example is more representative of the competing hypotheses that would be given to semantics. We argued that higher level constraints ought to be reserved for more acoustically similar hypotheses such as sea, see and C. Syntax and semantics constraints are far less restrictive than one might imagine. From our experience with syntax [19, 82], we have gained a certain
amount of respect for the productivity of English. Even if one could manage to take advantage of higher level constraints (and only the IBM tri-gram model seems to show positive results), it might be advisable to focus on low-level constraints in order to explore the potentials of low-level constraints.

Secondly, low-level constraints have been gaining favor because they seem to be much more powerful than once believed. It was once thought that people could not read spectrograms. Victor Zue has proved this to be incorrect. It was once thought that allophonic variation introduced so much “noise” into the signal that the phonetic transcription cannot be decoded with just lower level constraints. Nakatani and others are currently proving this to be mistaken. In short, low level constraints seem to be gaining favor for good reason.
Appendix III - Lexical Phonology

In many speech systems, the lexicon is viewed as large, rather uninteresting, database of words and their
pronunciations. Morphology (the decomposition of words into morphemes) is sometimes introduced as an
efficiency move; it clearly saves considerable (about an order of magnitude) storage to remove entries like cats
from the base lexicon, and generate them on the fly as they are needed. There are a number of other
advantages, though, to morphological decomposition. It turns out that many phonological processes (e.g.,
tensing, laxing, and maybe even schwa deletion) are very tightly coupled with morphology. Thus, for
example, the rule that accounts for the vowel alternation in divine/divinity does not apply in other
morphological environments as we will see. Similarly, the phonological rule that deletes the vowel in
experiment does not apply in words like merriment which have a different morphological structure. If we
are to understand these processes better, it will be necessary to consider morphology more carefully.

The connection between morphology and non-linear phonology has been an active topic of research for
some time now. As is to be expected, there have been many proposed variants. One variant that we find to
be particularly interesting and relevant to speech recognition research is Lexical Phonology (also known as
Level Ordered Phonology and Morphology [59, 86, and references therein].

III.1. Difference Between + and #

This framework extends the dichotomy between “+” and “#” morphemes, which has been well-
known for a long time, at least as far back as Whitney (1889) according to Kiparsky [59]. It is not easy to
define “+” and “#”. Informally, “+” morphemes are (generally) historically derived from Latin and “#”
from Greek and German. This historical trend is only a rough correlation and has numerous counter-
example (e.g., the German suffix -ist behaves like “+”). Most linguists (at least at MIT) would prefer to find
a more robust characterization between the two classes of morphemes. We will discuss a number of
differences between the two classes.

First, though, let us establish a common ground with the following mini-lexicon:

173. There are a number of morphological differences between merriment and experiment. In particular merry is a free morpheme (word), whereas experi- is a bound morpheme (non-word).
(271a) in-, -ion, -ity, -ic, -ive, -ate, -al-er, -est, -ist
(271b) un-, auto-, mon-. -hood, -ness, -ment, -less, -ful, non-
(271c) -able

As a matter of notational convention, we will adopt the + and # signs in our description of morphological decomposition. Thus, we will write the decomposition of divinity and unhappiness as:

(272a) divinity → divine + -ity
(272b) unhappiness → un- # happy # -ness

(The "-" sign will be used to distinguish bound morphemes from free morphemes.) 174

A number of phonological rules distinguish between "+" junctures and "#" junctures. For examples, stress shifts and vowel alternations apply across + but not across #. Note the stress shift and vowel alternation in divine/divin + ity but not in happy/happiness. Level ordered phonology makes considerable use of this interaction between morpheme types and phonological rules.

There is also a well-known precedence relation between + and # morphs. With very few exceptions, # morphemes nest outside of + morphemes. Thus, we have

(273a) non- # [in- + moral]

but not

(273b) *in- + [non- # moral]

The precedence175 relations yield some very subtle (but correct) predictions. Note that -able can be "+" affix in some cases (e.g., comparable)176 and a # affix in other cases (e.g., employable).177 Notice that the contrast

174. A free morpheme can stand alone as a word; a bound morpheme must be attached to another morpheme to form a word.
175. Precedence is a formal computation device for expressing ordering dependencies among operations. In linguistic terminology, one might say that "+ affixes" apply before "# affixes". We prefer the computation terminology here, though, because we do not wish to imply that the processing model has to work in the specified order. All that is required is that it produce the same results that would have been produced had it proceeded in the specified order.
176. The -able in comparable must be a "+" affix because it induces a stress alternation; compère has primary stress on the second syllable.
177. This -able is not a "+" affix because it does not lead to a stress alternation.
between *incomparable* and *unemployable*; the "+" marked *comparable* takes the "+" marked prefix *in-* whereas in contrast, the "#" marked *employable* takes the "#" marked prefix *un-*. The same contrast is brought out by the famous pair: *indivisible* vs. *undividable*.

In summary, we have mentioned but a few of the many contrasts between + and # morphemes. Here is a short list of distinguishing tests:

(274a) + morphemes feed certain phonological rules (stress assignment, tri-syllabic laxing); # do not.
(274b) + morphemes take precedence over #
(274c) + morphemes can attach to bound morphemes; # cannot
(274d) + morphemes are more exceptional than #
(274e) + morphemes are (often) historically correlated with Latin; # with German or Greek

The differences between # and + remain an active topic of research; see [59, 86, 104] for some recent treatments.

### III.2. Pipeline Design

Level ordered morphology extends the +/# distinction into a series of levels:

(275)  

underlying morphological representation  
↓
irregular inflection and "+ boundary" derivation  
↓
"# boundary" derivation and compounding  
↓
regular inflection  
↓
syntax  
↓
surface morphological representation

---

178. There is a secondary pronunciation of *comparable* as *compárable* where the "-able" is "#" marked (since it does not alternate stress from the underlying form *compàre*). This pronunciation takes the "#" marked prefix *un-*, i.e., *uncompárable*
This pipeline design provides a nice account for the fact that "+ affixes" and irregular inflection take precedence over "# affixes" and regular inflection. Thus, for example, we have an account of the difference between the non-immoral / *in-non-moral pair mentioned above: the former is grammatical since it attaches a level 2 # affix (non-) outside a level 1 + affix (in-), whereas the latter is ungrammatical since it attaches a level 1 affix outside a level 2 affix.

Secondly, though, and far more interesting from our point of view, is the extension of the same level-ordered pipeline design to phonology. Lexical phonology organizes phonological rules into a flow chart much like the flow chart for morphological rules:

(276)

underlying phonological representation
    ↓
stress assignment, shortening, laxing, ...
    ↓
compound stress, ...
    ↓
...
    ↓
flapping, glottalization, palatalization, ...
    ↓
postlexical phonology
    ↓
surface phonetic representation

Furthermore, it is assumed that the morphological and phonological levels are tightly coupled, so that all level 1 morphology and phonology takes precedence over level 2, and that all level 2 rules take precedence over level 3, and all level 3 rules take precedence over level 4 (if there is a level 4), and so on. All of these lexical rules must precede postlexical rules such as aspiration and flapping. Moreover, it is assumed that only level 1 rules can specify "+ boundaries" in their environments, and only level 2 rules can specify "# boundaries" in their environments, and so forth.

This level ordering hypothesis makes a number of interesting predictions. In particular, the hypothesis that level 1 rules precede level 3 provides an account of the vowel alternation in kept versus keeps. The explanation makes use of the fact that kept is derived at level 1, since it is an irregular form. Hence kept is subject to the level 1 phonological rule of vowel shortening. On the other hand, keeps is derived at level 3 since it is a regular form, and hence it is not subject to the level 1 phonological rule of vowel shortening.
Lexical phonology attempts to assign phonological rules to one or more levels, depending on the environmental specifications. Thus, a rule that applies across a word boundary would be assigned to the postlexical component, a rule that applies across a level 3 boundary would be assigned to level 3, a rule that applied across a level 2 "#" boundary would be assigned to level 2, and so on. Mohanan motivates this attempt as follows:

(277) "It is well known that [lexical] rules such as trisyllabic laxing and \(t \rightarrow s/_{-}i\) [as in \textit{president} \rightarrow \textit{presidency}] require morphological information for their correct application, while [postlexical] rules such as the aspiration of voiceless stops and the flapping of \(t\) do not. Laxing, for example, applies across + in \textit{divinity} (\textit{divin}+\textit{ity} \rightarrow \textit{divinity}), not across # in \textit{maidenhood}^{179} (\textit{mæden} # \textit{hud} \rightarrow \textit{*mædenhud}). \(t \rightarrow s/_{-}i\) applies only when \(t\) is followed by +: \textit{rezident} + \(i\) \rightarrow \textit{residensi} (\textit{residency}), but \textit{kritik} \rightarrow \textit{*krisik} (\textit{critic}). In contrast, the [postlexical] rules of aspiration and flapping are governed by factors which are strictly phonological [as opposed to morphological], such as the foot and the syllable (Kahn [54])." [86, pp. 1-2]

III.3. Distinctions Between Lexical and Postlexical Rules

This framework will force us to re-apportion some of the phonological grammar in certain ways, though these moves may lead to desirable results. After these moves, we will be in a position to capture generalizations about the distinctions between rules at different levels. A typical list of distinctions might look like:^{180}

(278a) Postlexical rules give precedence to (apply "after") lexical rules.
(278b) Postlexical rules apply across word boundaries; lexical rules do not.
(278c) Postlexical rules produce allophones (e.g., flaps); lexical rules do not.\(^{181}\)
(278d) Lexical rules can change the spelling; postlexical rules cannot.
(278e) Lexical rules are sensitive to morpheme boundaries; postlexical rules are not.
(278f) Lexical rules apply in derived environments; postlexical rules apply across the board.
(278g) Lexical rules have lexically marked exceptions; postlexical rules do not.

\(^{179}\) Mohanan is assuming that there are two [æ] vowels, a tense one (e.g., \textit{tab, pat, tag, badge, ham, man, staff, save, path, pass, jazz, cash}) and a lax one (e.g., \textit{tap, pai, rack, patch, pal}). This assumption is discussed in [39, p. 613], who gives credit to Trager (1930) for the first discussion of the distinction.

\(^{180}\) All but one of these were taken from Kiparsky's class lectures (fall 1982). The claim that only lexical rules can change spelling is our own proposal.
(278b) Lexical rules are cyclic; postlexical rules are not.
(278i) Lexical rules can specify the part of speech; postlexical rules cannot.
(278j) Lexical rules can block each other; postlexical rules do not.

It might seem that there are a number of problems with these generalizations. Consider, for example, the claim that postlexical rules are blind to word boundaries. It is assumed that postlexical rules apply exactly the same way within words as they do across word boundaries. This is an intriguing claim, since if it were correct, it would mean that we could process phonological rules without first solving the very difficult problem of segmenting the utterance into words. But at first glance, at least, the claim appears to be false. There are well-known apparent differences say, in the treatment of flapping, for instance, when there are word boundaries and when there are not. Within words, flapping seems to require falling stress:

\[(279a) \text{stratég}y \rightarrow \text{str][ég}y \quad \text{falling stress}\]
\[(279b) \text{stratégic} \rightarrow \text{str][égic} \quad \text{rising stress}\]

but no such requirement seems to be imposed across word boundaries as evidenced by the fact that \textit{at ease} flaps despite rising stress. We have seen in section §3.2 that these facts can be accounted for in terms of syllable and foot structure, thus avoiding the necessity to split the flapping rule into two rules, and stipulate that one case applies only within words and the other applies only across word boundaries. We will permit postlexical rules to look at syllable and foot (stress) structure, but we will not allow postlexical rules to refer to word and morpheme boundaries. If this is correct, then we will not have to stipulate for each rule whether or not it can see word boundaries. In addition, the speech decoder will not have to check each rule to see if they do or do not apply across word boundaries. Moreover, it will not be necessary to find word boundaries in order to "undo" the phonological rules.

181. It has been pointed out to us that this statement may appear to be circular since we have not yet said what an allophone is. The statement would clearly be circular if we defined an allophone to be a segment that does not appear in the lexicon. This criticism is well taken. From a formal point of view, it is impossible to evaluate the statement without precise definitions. We will depend on the reader’s intuition for the distinction between allophones and phonemes.

We have not attempted to define this distinction more carefully because it is extremely difficult to do so. Suppose for example that we defined allophones to be segments with non-distinctive features (e.g., features other than place, voice, manner). Even this, though, is a bit circular, because it is unclear what a distinctive feature is. Perhaps a distinctive feature is a feature that is found in the lexicon, and a non-distinctive feature is one that is not. This definition, of course, would lead to a circularity. Hopefully there is a more fundamental definition of distinctive features that does not depend on the lexicon.
III.4. Which Rules are Lexical and Which are Postlexical?

Let us give a slight glimpse of how the grammar is to be re-organized. Consider the palatalization rule. Some cases will clearly have to be postlexical, especially the case that accounts for the well-known assimilations:

(280a) miss you → mI[s]ou
(280b) did you → di[s]ou
(280c) gas shortage → ga[s:]ortage

because these cases cross word boundaries. It will also be necessary to assign the palatalization of /k/, in words like circular, to the postlexical level since a palatalized /k/ is an allophone (written [kʰ]), and allophones, by hypothesis, are produced at the postlexical level. On the other hand, not all cases of palatalization are postlexical. Words like church and shirt come out of the lexicon with palatalization already marked, as evidenced by the spelling among other things.

The distinction between lexical and postlexical rules parallels the distinction in many speech recognition projects between those phonological processes that apply within a word and those that apply across word boundaries. Thus most systems in the past would distinguish between the /ɛ/ in conscience and the [ɛ] in met your. The first /ɛ/ would be represented in the underlying representation of church whereas the second [ɛ] would be derived through some sort of phonological rule.

We, though, would like to de-emphasize the role of word boundaries in phonological rules. We hypothesize that rules are not marked for whether they apply within words or across word boundaries. Rather lexical rules are restricted for principled reasons to apply within lexical entries, and postlexical rules apply blindly across the board without reference to lexical structure and word boundaries. Thus for example, the lexical rule that palatalizes the /ɛ/ in church cannot apply across word boundaries, whereas the postlexical rule that palatalizes the /k/ in circular must apply across word boundaries as in a phrase like work your way. We believe postlexical rules are insensitive to word boundaries (though they may be sensitive to syllable and foot (stress) structure).  

---

182. We assume that the spelling is part of the lexical entry. Thus, if a phonological processes is manifest in the spelling, the phonological process must be located within the lexicon.
The word boundary distinction is but one test in a large battery of diagnostics for deciding whether a rule is lexical or postlexical. These tests provide us with a principled justification for partitioning the grammar one way rather than another. Consider the schwa deletion rule. This rule accounts for the alternation between difference with three syllables and difference with just two syllables. This rule is said to operate on words like difference and opera which have a stressed syllable followed by two unstressed syllables (i.e., 100 stress).\textsuperscript{184} Note, for instance, that the rule does not apply when the stress pattern is shifted as in differential or in operatic.\textsuperscript{185}

Recall that Smith [114] implemented this rule with lexical expansion, unlike most of the other phonological rules like flapping which he implemented with his training procedures and lexical organization. Why should schwa deletion be implemented differently from the aspiration rule? Smith did not provide us with a justification for this implementation decision. With our greater understanding of the lexical/postlexical distinction, we are in a better position to justify Smith's implementation of schwa deletion. We believe that schwa deletion is a lexical rule, unlike aspiration which is a postlexical rule. Thus, it is not surprising that schwa deletion should be implemented differently.

\textsuperscript{183} Many frameworks would depend on the word boundary to explain the phonetic difference between a tease/at ease, gray train/great rain, night rate/nirate and so forth. This solution is not available in lexical phonology where it is assumed that post-lexical rules (such as flapping and retroflexion) are blind to word boundaries. Instead, we will draw the contrast in terms of syllable structure and stress domains. Thus, we will say that the /l/ in train is retroflexed because it is in a /sr/ syllable onset cluster. In contrast, the /l/ is not retroflexed in great rain because the /l/ is not part of a /sr/ onset cluster.

\textsuperscript{184} There is also said to be a condition on the consonants surrounding the consonant. Note, for instance, the word triplicate does not undergo schwa deletion because doing so would produce a /plk/ cluster which cannot be parsed since /pl/ is not a syllable offset and /lk/ is not a syllable onset. This same condition on consonants is often used to account for the unacceptability of schwa deletion in words like circulate, articulate, accurate and so forth. Schwa deletion cannot apply in these words because of a /ly/ in the second syllable. With the /ly/, it becomes impossible to parse the remaining consonants (/rcyI/, /cyI/, and /cyr/, respectively) into a syllable offset followed by a syllable onset.

\textsuperscript{185} This rule can be used to elicit very subtle stress judgments. For example, it is said that words like experiment, approximate, alternate, moderate, separate and other -ate words have two different stress patterns depending on their part of speech. When they are verbs, they have 102 stress, and when they are nouns and adjectives, they have 100 stress. We would predict that schwa deletion should apply in the 100 case (nouns and adjectives) but not in the 102 case. This prediction seems to be correct.

It is also interesting to consider words ending in the suffix -ing. Take a word like recording, where the root has two stress patterns depending on the part of speech: 10 (noun) and 01 (verb). If we could add the -ing suffix to nouns, we would get a 100 stress pattern, which would feed the schwa deletion rule producing *rec-ding. Of course, we can’t add the -ing affix to nouns, only to verbs. The stress of the resulting -ing form will be 010, and consequently, schwa deletion does not apply.

But we do see schwa deletion in words like suffering, handling, labeling and so forth. In these cases, the underlying verbal roots have the stress pattern 10, so that after we add the -ing we have a 100 stress pattern, meeting the stress conditions on schwa deletion. These 100 -ing forms are complicated though by a “#” boundary in one of the possible derivations. For example, it is said [15, p. 86] that participle twinkling can be derived from twinkle # ing. As we will see, the “#” will block schwa deletion, so that under this derivation, there will be three syllables. There is also another form of twinkling which does not exhibit the “#” boundary. This form is subject to schwa deletion.

- 229 -
Why is schwa deletion a lexical rule? Note that schwa deletion does not cross word boundaries and that it does not introduce allophones. These observations are compatible with the view that it is a lexical rule, but they are not in themselves convincing. We find more compelling evidence when we consider morpheme boundaries. Notice that schwa deletion is sensitive to "#" boundaries. It is not possible for example to delete the middle syllable in merry # ment or happy # ness, whereas Merimack and experiment, which lack the "#" boundary, can be pronounced with just two syllables. The suffix -able is particularly interesting in this regard because it can act as a "#" affix in words like employ # able and as a "+" affix in other words like cômpar + able.\textsuperscript{186} Notice that schwa deletion is possible in the "+" -able case (e.g., préferable, cômparable)\textsuperscript{187} but not in the "#" case (e.g., employable, removable).

(281a) Schwa deletion is possible at level 1 (across + boundary): cômparable, préferable, experiment, Merrimack, boundary

(281b) But not at level 2 (across # boundary) or level 3 (across compounds): compàrable, préfèrable, happiness, merriment, employable, penniless, candyland

Thus it appears that schwa deletion is a lexical rule because it is sensitive to the internal morphological structure of lexical entries. Moreover, it appears to apply at level 1.\textsuperscript{188}

There are quite a number of rules that must be re-examined from this point of view. For example, there are some cases of nasal assimilation that are probably lexical

(282a) in + legal → illegal  \hspace{1cm} level 1

(282b) in + portant → important  \hspace{1cm} level 1

because, among other things, the rule changes the spelling.\textsuperscript{189}

\textsuperscript{186} The two -ables can be defended by looking at stress alternations. The "#" -able preserves the stress pattern of the root (employ → employable) whereas the "+" -able induces an alteration in the root (compère → cômparable).

\textsuperscript{187} It is possible that some "+" boundary cases of -able might resist schwa deletion. This cases can be accounted for by assuming that they are lexically marked exceptions.

\textsuperscript{188} Actually, there is another solution. It is possible that #-boundaries determine foot structure, which is available at the post-lexical level. We will not explore this possibility here. See section §3.2.5 for a discussion of the dependence between #-boundaries and foot structure.

\textsuperscript{189} As mentioned in the church example above, we assume the spelling is part of the lexical entry. Thus, if the effects of a rule can be found in the spelling, the rule must be lexical.
Nasal assimilation is probably a level 1 rule because it is blocked\textsuperscript{190} by \#-affixes and compounds:

\begin{align*}
(283a) \text{un \# fortunately } & \rightarrow \text{i[m]fortunately} \quad \text{level 2} \\
(283b) \text{in \# put } & \rightarrow \text{se[i[m]put} \quad \text{level 2}
\end{align*}

In contrast to these lexical cases of nasal assimilation, there are also some cases of nasal assimilation that are postlexical:

\begin{align*}
(283c) \text{infested } & \rightarrow \text{i[m]fested} \quad \text{postlexical} \\
(283d) \text{seven } & \rightarrow \text{se[i[m]ven} \quad \text{postlexical}
\end{align*}

These cases are very different. These cases seem to apply at word boundaries,

\begin{equation}
(283e) \text{in fact } \rightarrow \text{i[m]fact} \quad \text{postlexical}
\end{equation}

they are optional and more common in casual speech, they do not change spelling, they are independent of morphological structure, and they are probably dependent on stress. Admittedly, though, it is difficult to determine, in an objective experimental way, the place of the nasal in this context, and therefore we will not pursue the matter here.

III.5. The Implementation of Lexical and Postlexical Rules

From our point of view, this level-ordered pipeline design is helpful because it allows us to distinguish in a principled way the postlexical phonological rules that concern us such as flapping, glottalization, palatalization, and so forth, from the lexical phonological rules that play a less important role in speech recognition, such as laxing, stress assignment, vowel shift, and so forth. Our design of a speech recognition device will make considerable use of this distinction. The lexical rules will be used in the parser in an indirect fashion as cues for constituent boundaries. So, for example, we will use the restricted distribution of lax vowels, which is an indirect consequence of lexical rules such as laxing, in order to constrain the possible foot structures.

\begin{footnote}
190. There is another postlexical form of nasal assimilation that may apply in cases like this. It has many different properties as we will see.
\end{footnote}
Fig. 20. Summary of Lexical/Postlexical Distinctions

In summary, we have alluded to some of the distinctions between lexical and postlexical rules. In addition to the ordering constraint (lexical rules take precedence over postlexical rules):

<table>
<thead>
<tr>
<th>ENVIRONMENT SPECIFICATIONS</th>
<th>Lexical Rules</th>
<th>Post-Lexical Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitive to Part of Speech</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Sensitive to word boundaries</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Sensitive to morpheme boundaries</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Sensitive to syllable boundaries</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Sensitive to stress</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Sensitive to Allophones</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

| DOMAIN of APPLICATION                                           |               |                    |
| Apply across word boundaries                                    | -             | +                  |

| OUTPUT SPECIFICATIONS                                           |               |                    |
| lexical entries (spelling)                                      | +             | -                  |
| feet (stress)                                                   | +             | -                  |
| syllables                                                       | +             | -                  |
| phonemes                                                        | +             | -                  |
| allophones                                                      | -             | +                  |

| FLOW of CONTROL                                                 |               |                    |
| Apply in Derived environments                                   | +             | -                  |
| Cyclic                                                          | +             | -                  |

| EXCEPTIONS                                                      |               |                    |
| Lexically marked exceptions                                     | +             | -                  |
| Lexical Blocking                                                | +             | -                  |

On the other hand, postlexical rules will play a much more major and direct role for a variety of reasons. First, postlexical rules are easier to apply in the reverse direction than lexical rules because postlexical rules are

(284a) acyclic,
(284b) free of lexically marked exceptions, and
(284c) blind to the lexical category and internal morphological structure of the word.
Moreover, in addition to the fact that postlexical rules are easier to run in the reverse direction, there are several reasons why they almost have to be run in reverse at run time\textsuperscript{191} by the speech recognition device. In particular, postlexical rules are

(285a) the sole source of phonetic allophones, and
(285b) the only rules that cross word boundaries.

Hence, we have some very good reasons to interpret the postlexical component. We might now proceed to ask how much of the lexicon should be interpreted\textsuperscript{192} and how much should be compiled? In this work, for purely pragmatic reasons, we have decided to draw the line between level 2 and level 3. This compromise enables us to compile out many of the hard phonological rules, especially stress assignment and irregular morphology, while obtaining the advantages of interpreting morphological decomposition for the relatively easy cases of regular inflection and compounding.

We have found Lexical Phonology to be helpful because it has provided a meaningful framework for discussing the differences between phonology and phonetics. In order to design a speech recognition device, we wish to focus on the crucial allophonic processes at the postlexical level and avoid many of the complex questions of phonology encountered by lexical rules, namely the issues of cyclicity, lexical exceptions, and morphology. In so far as lexical phonology permits this, we find it to be a welcome advance over past frameworks such as SPE, which conflate lexical and postlexical processes freely:

(286) In SPE, the two kinds of phonological processes intermingle freely and apply as part of the derivation from the underlying to the phonetic representation: no theoretical status is assigned to the distinction between the two types of processes. The model of Lexical Phonology assumes trisyllabic laxing and velar softening apply in the lexicon, as part of the word formation component; aspiration and flapping apply post lexically. The principle governing the distinction between lexical and post lexical applications of rules is: (1) Post lexical operations are blind to the internal structure of words.” [86, pp. 1-2]

\textsuperscript{191} Run time is a computational term. It is being used here in opposition to compile time. That is, we are advocating that postlexical rules be applied when the speech recognition device is processing a particular utterance, not when the device is being constructed (compiled).

\textsuperscript{192} Interpretation is a computational term which is generally contrasted with compilation. In compilation, much of the work is performed in advance for all possible inputs. An interpreter, in contrast, waits until it has a particular input, before it goes to work.
In summary, lexical phonology provides us with a principled set of criteria for distinguishing postlexical allophonic processes from lexical phonological rules. Moreover, it offers a number of good reasons for designing a speech recognition device which interprets the postlexical rules at run time and does not expand them into a compiled list by applying all possible rules to all possible underlying forms as is the standard practice in speech recognition research. In our own implementation, we have decided to compile the first two levels of the lexicon and to interpret the third level and the postlexical level.
Appendix IV - A Sample Grammar

This grammar is intended to demonstrate the parsing, canonicalization and matching approach. It just barely scratches the surface of the wealth of linguistic constraints that one could make use of. Nevertheless, it gives a feeling for what is involved in writing a comprehensive grammar in this framework.

This code is written in a lisp macro package called LSB. LSB is currently supported on most MIT brands of lisp including Maclisp (both ITS and system 20s), NII., Lisp Machine (both LMI and Symbolics), Multics Lisp, etc. LSB is intended to enhance transportability. All of this code should (but probably won’t) run in most of the before mentioned dialects of Lisp with no modifications. Most of this code was debugged in Maclisp (both ITS and System 20) and Lisp Machine Lisp (both LMI and Symbolics). (On the small address space machines, there can be difficulties in loading the large dictionaries into core memory.)
; Top Level
(declare (special segment-lattice *graphics-port* *echo-port*
  *lattice-env* *phoneme-lattice-env*))

(defsysr (parse-segment-lattice segment-lattice)
  ; Keep upper case characters (e.g., A) distinct from lower case (e.g., a)
  ; over the dynamic scope of the following.
  (with-lower-case
    ; Create a new lattice environment (memoization table)
    ; and bind it over the dynamic scope of the following.
    (with-lattice-env
      ; Print syllable level constituents.
      ; Recall that lattices are stored in the memoization table
      ; bound above by WITH-LOWER-CASE. (Lattice xxx) will look
      ; in that memoization table for the appropriate lattice.
      ; If the lattice is not found in the memoization table,
      ; a new one will be computed and memoized.
      ; The code describing how to compute a new lattice is
      ; defined with Def-lattice. (See below).
      (adjmat-print-matrices
        *graphics-port* *echo-port*
        segment-lattice "segments"
        (lattice onset) "onsets"
        (lattice peak) "peaks"
        (lattice coda) "codas"
        (lattice inflectional-affix) "inflectional affix"
        (lattice syllable) "syllables"
        (lattice gc) "gc-ed syllables" ; garbage collected
      )
      ; print metrical level constituents
      (adjmat-print-matrices
        *graphics-port* *echo-port*
        segment-lattice "segments"
        (lattice stray-syllable) "stray syllable"
        (lattice foot1) "foot1"
        (lattice foot2) "foot2"
        (lattice foot3) "foot3"
      )
      ; lexical entries
      (load-dictionary-files)
      (adjmat-print-matrices
        *graphics-port* *echo-port*
        segment-lattice "segments"
        (lattice canonicalized-onset) "canonicalized onsets"
        (lattice canonicalized-rhyme) "canonicalized rhyme"
        (lattice canonicalized-syllable) "canonicalized syllables"
        (lattice sylparts-in-dictionary) "sylparts in dictionary"
        (lattice words-in-dictionary) "words in dictionary"
        (lattice gc-words-in-dictionary) "gc-ed words in dictionary"
      )))
: features

: This function defines how to compute a new aspirated lattice.
: It maps over the stop lattice (see definition below) and
: selects those segments that pass the ASPIRATED? predicate.
: That predicate is not defined here. It performs a memq test
: for the aspiration feature in the phonetic features of the segment.
(defun lattice aspirated (m-filter (lattice stop) 'aspirated?)
  (def-lattice flapped (m-filter (lattice dental) 'flapped?)
  (def-lattice glottalized-stop (m-filter (lattice stop) 'glottalized?)
  (def-lattice glottalized-vowel (m-filter (lattice vowel) 'glottalized?)
  (def-lattice unreleased-stop (m-filter (lattice stop) 'unreleased?)
  (def-lattice released-stop (m-filter (lattice stop) 'released?)
  (def-lattice light (m-filter (lattice liquid) 'light?)
  (def-lattice dark (m-filter (lattice liquid) 'dark?)
  (def-lattice retroflex (m-filter (lattice liquid) 'retroflex?)

:vowels
(defun lattice short-vowel (phoneme-lattices "cEIiou")
(defun lattice schwa (phoneme-lattices "æ/æ")
(defun lattice y-glide (phoneme-lattices "æl")
(defun lattice long-vowel (phoneme-lattices "o")
(defun lattice rounded (phoneme-lattices "OUWYuo")
(defun lattice tense-vowel (m+ (lattice y-glide) (lattice rounded))
(defun lattice lax-vowel (phoneme-lattices "cEI@o")
(defun lattice vowel
  (m+ (lattice schwa)
  (lattice tense-vowel)
  (lattice lax-vowel)))
(defun lattice syllabic-segment (phoneme-lattices "RLWw"))
; manner features of consonants
(def-lattice nasal (phoneme-lattices "nmG"))
(def-lattice stop (phoneme-lattices "&tpkbdbgJC"))
(def-lattice glide (phoneme-lattices "wy"))
(def-lattice liquid (phoneme-lattices "ly"))
(def-lattice strong-fric (phoneme-lattices "ssZ"))
(def-lattice weak-fric (phoneme-lattices "fwTD"))
(def-lattice fric (m+ (lattice strong-fric) (lattice weak-fric)))
(def-lattice obstruent (m+ (lattice nasal) (lattice stop) (lattice fric)))
(def-lattice semi-vowel (m+ (lattice liquid) (lattice glide)))
(def-lattice consonant (m+ (lattice obstruent) (lattice semi-vowel)))
(def-lattice voiceless-stop (phoneme-lattices "ptk?C"))
(def-lattice voiced-stop (phoneme-lattices "bdgJ?&"))
(def-lattice voiceless-fric (phoneme-lattices "sTf"))
(def-lattice voiced-fric (phoneme-lattices "zSDv"))
(def-lattice unvoiced-obstruent (phoneme-lattices "ptkCsSfT"))
(def-lattice voiced-obstruent (phoneme-lattices "bdgJzZvDmnG"))
(def-lattice voiced (m+ (lattice semi-vowel)
  (lattice vowel)
  (lattice voiced-obstruent)))
(def-lattice unvoiced (lattice unvoiced-obstruent))
(def-lattice unvoiced-obstruents (m+ (lattice unvoiced-obstruent)))
(def-lattice voiced-obstruents (m+ (lattice voiced-obstruent)))
(def-lattice dental (phoneme-lattices "dtTDnszNly&"))
(def-lattice velar (phoneme-lattices "kgG"))
(def-lattice labial (phoneme-lattices "pbvmmWw"))
(def-lattice palatal (phoneme-lattices "JCSzn"))
(def-lattice dentals (m+ (lattice dental)))
(def-lattice velars (m+ (lattice velar)))
(def-lattice labials (m+ (lattice labial)))
(def-lattice palatais (m+ (lattice palatal)))
(def-lattice homorganic
  (m+ (lattice dentals)
  (lattice velars)
  (lattice labials)
  (lattice palatais)))
::: Sylpart Level
: Onset and Offset Clusters

: Clusters with fricatives and stops
: there are really only 3 possibilities: sp, st, sk
(def-lattice fric-stop-cluster
  (m* (phoneme-lattice #/s)
    (lattice voiceless-stop)))

: Clusters with nasals and stops
(def-lattice nasal-stop-cluster
  (m& (m* (lattice nasal) (m- (lattice stop)
    (lattice aspirated)
    (lattice flapped))
    (lattice homorganic)))

: Clusters with nasals and fricatives
(def-lattice nasal-fric-cluster
  (m& (m* (lattice nasal) (lattice fric))
    (lattice homorganic)))

(def-lattice s-onset-cluster
  (m* (phoneme-lattice #/s)
    (m+ (lattice voiceless-stop)
      (phoneme-lattices "nm")))

: onsets which can be followed by a semi-vowel
(def-lattice semi-vowel-onset
  (m& (m* (lattice onset-core) (lattice semi-vowel))
    (lattice non-homorganic-onset)))

: rules out tl, dl, pw, bw, fw, sr, Sr
(def-lattice non-homorganic-onset
  (m* (mopt (phoneme-lattice #/s))
    (m+ (m* (lattice non-dental-stop) (phoneme-lattice #/l))
      (m* (lattice non-labial-stop) (phoneme-lattice #/w))
      (m* (lattice velar) (lattice semi-vowel))
      (lattice non-sonorant-semi-vowel)))

(def-lattice non-dental-stop (phoneme-lattices "bgpk")
(def-lattice non-labial-stop (phoneme-lattices "dgtk")
(def-lattice non-sonorant-semi-vowel (phoneme-lattices "lwy"))
(def-lattice onset-core
  (m+ (lattice fric-stop-cluster)
      (lattice voiceless-fric)
      (m- (lattice stop) (lattice unreleased-stop))))

(def-lattice onset
  (m+ (lattice onset-core)
      (lattice semi-vowel-onset)
      (lattice voiced-fric)
      (lattice semi-vowel)
      ? (m* (mopt (phoneme-lattice #/s))
           (phoneme-lattices "nm"))
      (phoneme-lattice #/h)))

(def-lattice coda
  (m+ (m- (lattice stop) (lattice aspirated))
      (lattice fric)
      (lattice nasal)
      (lattice nasal-stop-cluster)
      (lattice fric-stop-cluster)
      (lattice nasal-fric-cluster)))
Peaks

We want to distinguish codas in foot internal position from those in foot final position. For now, there aren't many differences between the two types of codas. There should be many more.

(defun foot-internal-coda (lattice coda))
(defun foot-final-coda (m- (lattice coda) (lattice flapped)))

(defun lattice reduced-peak
  (m+ (m* (lattice schwa)
          (mopt (lattice liquid)))
       (lattice syllabic-segment)))

(defun lattice tensed-peak (lattice tense-vowel))

(defun lattice lax-peak (m* (lattice lax-vowel)
                           (mopt (lattice liquid)))

(defun lattice non-lax-peak (m+ (lattice reduced-peak)
                              (lattice tensed-peak)))

(defun lattice peak (m+ (lattice lax-peak)
                     (lattice non-lax-peak)))

Lax peaks must be closed.
Non-lax peaks can be open or closed.

(We would have liked to have taken the stronger position that non-lax peaks must be open, but we have not worked out the details sufficiently well to make this position tenable.)

(defun lattice rhyme
  (m+ (lattice stressed-rhyme)
       (lattice reduced-rhyme)))
(def-lattice stressed-rhyme
  (m* (m* (lattice tensed-peak) (mopt (lattice coda)))
       (m* (lattice lax-peak) (lattice coda))))

(def-lattice reduced-rhyme
  (m* (lattice reduced-peak) (mopt (lattice coda))))

(def-lattice syllable (m* (mopt (lattice onset))
                          (lattice rhyme)))

(def-lattice foot-final-syllable
  (adjmat-copy-when-ends-with
   (lattice syllable)
   (m* (lattice peak)
        (lattice foot-final-coda))))

(def-lattice unstressed-syllable (lattice syllable))

(def-lattice stressed-syllable (m* (mopt (lattice onset))
                                   (lattice stressed-rhyme)))

(def-lattice reduced-syllable (m* (mopt (lattice onset))
                                (lattice reduced-rhyme)))
(def-lattice stray-syllable (lattice reduced-syllable))

; a single syllable foot
(def-lattice foot1
  (m& (lattice stressed-syllable)
       (lattice foot-final-syllable)))

; a two syllable foot
(def-lattice foot2
  (m* (lattice stressed-syllable)
       (m& (lattice unstressed-syllable)
            (lattice foot-final-syllable))))

; a three syllable foot
(def-lattice foot3
  (m* (lattice stressed-syllable)
       (lattice reduced-syllable)
       (m& (lattice unstressed-syllable)
            (lattice foot-final-syllable))))

(def-lattice inflectional-affix
  (m+ (m-after (m++ (phoneme-lattices "zd"))
          (lattice voiced))
       (m-after (m++ (phoneme-lattices "stT"))
              (lattice unvoiced-obstruent))))

(def-lattice syllable-with-affixes
  (m* (lattice syllable)
       (lattice inflectional-affix))

(def-lattice gc (adjmat-gc (lattice syllable-with-affixes)))

(def-lattice foot
  (m* (lattice stressed-syllable)
       (mopt (lattice unstressed-syllable))))

(def-lattice word-tree
  (m* (m+ (lattice stray-syllable)
          (lattice foot1)
          (lattice foot2)
          (lattice foot3))
       (mopt (lattice inflectional-affix))))
; Canonicalization

; There should be different canonicalization functions in each
; structural position, but this has not been implemented yet

(def-lattice canonicalized-onset
  (canonicalize-lattice (lattice onset)))

(def-lattice canonicalized-rhyme
  (canonicalize-lattice (lattice rhyme)))

(def-lattice canonicalized-syllable
  (canonicalize-lattice (m- (lattice gc)
          (lattice inflectional-affix)))))

; These next two s-expressions perform the canonicalization and match
; steps. Most of the work is done in performed in ad-hoc lisp routines,
; which we will not present here.

(def-lattice sylparts-in-dictionary
  (adjmat-transduce (lattice canonicalized-syllable)
          #'lookup-sylparts))

(def-lattice words-in-dictionary
  (lookup-adjmat-in-dnet
    (lattice sylparts-in-dictionary)
    *syllable-to-word-dnet*))

(def-lattice gc-words-in-dictionary
  (adjmat-gc (m* (lattice words-in-dictionary)
          (mopt (lattice inflectional-affix))))
Appendix V - Sample Lexicon

This lexicon was derived from the Klatt Lexicon of 3432 entries. (This is an old version taken with permission.) The entries listed below begin with the letter a or the letter b. The syllable boundaries were inserted by the algorithm discussed in section §7.4. Foot structure brackets were inserted purely on the basis of the stress marks. The phonetic symbol set was originally designed by Meg Withgott and modified by the author. It is likely that these character translation have introduced some errors.

\[
\begin{array}{llll}
a & \partial & \text{abandon} & \partial \ [\text{b\'\'n d\'\']} + n \\
abide & \partial \ [\text{b\'\'y}] + d & \text{ability} & \partial \ [\text{b\'\'ll t u}] \\
able & \partial \ [\text{b\'\' b t}] & \text{aboard} & \partial \ [\text{b\'\' w r}] + d \\
about & \partial \ [\text{b\'\' w t}] + t & \text{above} & \partial \ [\text{b\'\' v}] \\
abroad & \partial \ [\text{br\'\'d}] & \text{absence} & [\text{\'\'b s\'\'n}] + s \\
absent & [\text{\'\'b s\'\'n}]+ t & \text{absolute} & [\text{\'\'b s\'\' lu}]+ t \\
absorb & \partial b \ [\text{zo \text{w} r}] + b & \text{abundant} & \partial \ [\text{b\'\' m n d\'\'n}] + t \\
abuse & \partial \ [\text{by\'u}] + z & \text{accept} & \partial k \ [\text{\'\'p}] + t \\
access & \partial \ [\text{\'\'k ses}] & \text{accompany} & \partial \ [\text{\'\'k m p\'\' n}] \\
accomplish & \partial \ [\text{\'\'k m pl\'\'}] + \$ & \text{accord} & \partial \ [\text{\'\'k w r}] + d \\
accuse & \partial \ [\text{\'\'k yu}] + z & \text{accustom} & \partial \ [\text{\'\'k s t\'\'}] + m \\
ache & \partial \ [\text{\'\' e k}] + k & \text{achieve} & \partial \ [\text{\'\'ef}] + v \\
achievement & \partial \ [\text{\'\'civ m\'\'n}] + t & \text{acid} & [\text{\'\'as t}] + d \\
acknowledge & \partial k \ [\text{\'\'n t}]+j & \text{acquaint} & \partial \ [\text{\'\'k w\'\' n}] + t \\
acquaintance & \partial \ [\text{\'\'k w\'\'n t\'\'}] + s & \text{acquire} & \partial \ [\text{\'\'k w\'\' y}] + r \\
across & \partial \ [\text{kr\'\'s}] & \text{act} & [\text{\'\'ek}] + t \\
action & \partial \ [\text{\'\'ek s\'\'}] + n & \text{active} & [\text{\'\'ek t}] + v \\
actor & \partial \ [\text{\'\'ek t\'\'}] & \text{actual} & [\text{\'\'ek s\'\'}] \\
adapt & \partial \ [\text{\'\'d\'\' p}] + t & \text{add} & [\text{\'\'ad}] \\
additional & \partial \ [\text{\'\'d\'\' s t\'\' n}] + t & \text{address} & [\text{\'\'ad res}] \\
administration & \partial d \ [\text{m n t}] [\text{str\'\' y s\'\'}] + n & \text{admiration} & [\text{\'\'ad m\'\'d}] [\text{\'\'y s\'\'}] + n \\
admire & \partial d \ [\text{m\'\' y}] + r & \text{admit} & \partial d \ [\text{m n t}] \\
adopt & \partial \ [\text{\'\'d\'\' p}] + t & \text{adore} & \partial \ [\text{doi w r}] \\
advance & a\partial d \ [\text{v\'\'n}] + s & \text{advantage} & a\partial d \ [\text{v\'\'n t}] + j
\end{array}
\]
adventure $\partial d$ [vɛn tɛr]
advice $\partial d$ [vɛr]t+z
affair $\partial$ [fɛr]
afford $\partial$ [fɔ[r]t]+d
after $[\text{af tɛr}]
afterward $[\text{af tɛr wɛr}]+d$
against $\partial$ [gɛn]+st
agency $\partial$ [gɛn sɪ]
ago $\partial$ [ɡɔ[w]
agree $\partial$ [ɡri]
agriculture $\partial$ [ɡrɪ kwɛl tɛr]
aid $\partial$ [ɞɔ]+d
ain't $\partial$ [ɞɔn]+t
alarm $\partial$ [lɛr]+m
alive $\partial$ [lɛr]+v
allow $\partial$ [lə[r]
almost $[\text{ɔl mo[w}s]+t$ along $\partial$ [lɛŋ]
 aloud $\partial$ [lɔ[w]t]+d
also $[\text{ɔl so[w]}
although $[\text{ɔl do[w]}
always $[\text{ɔl wɛr}]+z$
amaze $\partial$ [mɛ[t]+z
ambition $\partial$ [bɛs tə]+n
america $\partial$ [mɛr t kə]
amid $\partial$ [mɪd]
amount $\partial$ [mɔ[w]+t
amuse $\partial$ [myu[t]+z
an $\partial n$
anchor $[\text{əŋ kər}]
and $\partial n+d$
angle $[\text{əŋ gɛr}$
animal [æn ɪ m+t] announce θ [n³w³n]+s
annual [æn y+] another θ [n³Λ ³ θ]
answer [æn s³r] ant [æn]+t
anxiety æη [z³v ³ θ tɪ] anxious [æn s³]+s
any [èn i] anxious [æn i h³w³]
anyone [èn i wΛn] anyhow [èn i θ t η]
anyway [èn i we³] anywhere [èn i wer]
apart θ [pår]+t apartment θ [pår m³n]+t
apparent θ [pèr ³ θ n]+t appear θ [p³]+r
append θ [p³n]+d appetite [æp ð ta³]+t
apple [æp ³] application [æp ln] [k³v s³]+n
apply θ [p³l³v³] appoint θ [p³v³n]+t
appointment θ [p³v³nt m³n]+t appreciate θ [pr³ ³i e³]+t
approach θ [pr³w³c] approve θ [pr³]+v
april [æ³ pr³] apt [æp]+t
arch [ár]+ɛ are ar
area [èr i θ] aren't [ár η ]+t
argue [ár gyu] argument [ár gyu m³n]+t
arise θ [r³v³]+z arm [ár]+m
armor [ár m³] army [ár mi]
around θ [r³w³n]+d arouse θ [r³w³]+z
arrange θ [r³v³n]+t arrangement θ [r³v³n m³n]+t
array θ [r³v³] arrival θ [r³v³ v³]+
arive θ [r³v³]+v art [ár]+t
article [ár t t k³] artificial [ár t t ] [f³s ³]
artist [ár t t ]+st as æz
ascend θ [s³n]+d ash [æs]
ashame θ [s³v³]+m ashore θ [s³w³r³]
ask [æs]+k asleep θ [s³l³]+p
aspect [æs pek]+t assault θ [s³l³]+t
assemble θ [s³m b+t] assembly θ [s³m bli]
assert θ [s³r]+t assign θ [s³l³]+n
assist ə [sɪʃ] + t assistance ə [sɪʃ tɔn] + s
associate ə [səʊ] + t association ə [səʊ] + n
assume ə [sʊ] + m astonish ə [stʌn] + s
at æt ate [ɛt] + t
atlantic æt [ˈæt ləntɪk] atmosphere æt [məʊ sfl] + r
attach ə [ˈtætʃ] attain ə [tætʃ] + n
attempt ə [ˈtæmp] + t attend ə [tɛn] + d
attendant ə [ˈtɛn dɑn] + t attention ə [tɛn sə] + n
attitude [ˈæt ɪ tu] + d attorney ə [tɪdn] ni
attract ə [ˈtrækt] + t attractive ə [ˈtrækt ut] + v
attribute [ˈæt rə baɪ] + t audience ə [d i ən] + s
august ə [ˈɔg ət] + st aunt ə [ən] + t
author ə [ɔr] authority ə [θɔr tɪ]
automobile [ˈɔt ə məʊbəl] [ɪ] + 1 autumn ə [ɔt ə] + m
available ə [vəʊ] ɪə əʊkt] avenue ə [əv ən]
average ə [ˈæv ər] + j avoid ə [vəʊ] + d
await ə [wæt] + t awaken ə [wæt kə] + n
aware ə [wɔr] away ə [wɔ] + j
awhile ə [wɔl] + 1 baby ə [bæ] + j
back [bæk] background ə [bæk grəʊ] + n
backward [bæk wəd] + d bad ə [bæd]
bag ə [bæg] bake ə [bæ] + k
balance ə [bæl ən] + s ball ə [bɔl]
band ə [bænd] + d bank ə [bæn] + k
banker ə [bænk ə] bar ə [bɑr]
bare ə [bɑr] bargain ə [bɑr ɡə] + n
bark ə [bɑrk] + k base ə [bə] + s
basin ə [bæs sə] + n basis ə [bæs ə] + s
basket ə [bæs kət] + t bath ə [bæθ]
bathe ə [bæθ] + ə battle ə [bæt ə]
bay ə [bæ] + i be ə [bi]
beach ə [bɪ] + ə beam ə [bɪ] + m
<table>
<thead>
<tr>
<th>Word</th>
<th>Pronunciation</th>
</tr>
</thead>
<tbody>
<tr>
<td>bean</td>
<td>[bɛn]</td>
</tr>
<tr>
<td>beard</td>
<td>[bɛrd]</td>
</tr>
<tr>
<td>beat</td>
<td>[bɛt]</td>
</tr>
<tr>
<td>beautiful</td>
<td>[bɛrθl]</td>
</tr>
<tr>
<td>became</td>
<td>[bɛm]</td>
</tr>
<tr>
<td>become</td>
<td>[bɛm]</td>
</tr>
<tr>
<td>bed</td>
<td>[bɛd]</td>
</tr>
<tr>
<td>bee</td>
<td>[bɛ]</td>
</tr>
<tr>
<td>been</td>
<td>[bɛn]</td>
</tr>
<tr>
<td>before</td>
<td>[bɛfr]</td>
</tr>
<tr>
<td>began</td>
<td>[bɛn]</td>
</tr>
<tr>
<td>begun</td>
<td>[bɛn]</td>
</tr>
<tr>
<td>beheld</td>
<td>[bɛhɛd]</td>
</tr>
<tr>
<td>behold</td>
<td>[bɛhɔd]</td>
</tr>
<tr>
<td>belief</td>
<td>[bɛli]</td>
</tr>
<tr>
<td>bell</td>
<td>[bɛl]</td>
</tr>
<tr>
<td>below</td>
<td>[bɛlu]</td>
</tr>
<tr>
<td>bend</td>
<td>[bɛnd]</td>
</tr>
<tr>
<td>benefit</td>
<td>[bɛnfi]</td>
</tr>
<tr>
<td>berry</td>
<td>[bɛr]</td>
</tr>
<tr>
<td>best</td>
<td>[bɛst]</td>
</tr>
<tr>
<td>betray</td>
<td>[bɛtri]</td>
</tr>
<tr>
<td>between</td>
<td>[bɛtwɛn]</td>
</tr>
<tr>
<td>bible</td>
<td>[bɪbəl]</td>
</tr>
<tr>
<td>big</td>
<td>[bɪg]</td>
</tr>
<tr>
<td>bind</td>
<td>[bɪnd]</td>
</tr>
<tr>
<td>birth</td>
<td>[bɜθ]</td>
</tr>
<tr>
<td>bit</td>
<td>[bɪt]</td>
</tr>
<tr>
<td>bitter</td>
<td>[bɪtər]</td>
</tr>
<tr>
<td>blank</td>
<td>[blænk]</td>
</tr>
<tr>
<td>blast</td>
<td>[blæst]</td>
</tr>
<tr>
<td>blind</td>
<td>[blænd]</td>
</tr>
<tr>
<td>bear</td>
<td>[bɛr]</td>
</tr>
<tr>
<td>beast</td>
<td>[bɛst]</td>
</tr>
<tr>
<td>beaten</td>
<td>[bɛt]</td>
</tr>
<tr>
<td>become</td>
<td>[bɛm]</td>
</tr>
<tr>
<td>bedroom</td>
<td>[bɛdru]</td>
</tr>
<tr>
<td>beef</td>
<td>[bɛf]</td>
</tr>
<tr>
<td>beer</td>
<td>[bɛr]</td>
</tr>
<tr>
<td>beg</td>
<td>[bɛg]</td>
</tr>
<tr>
<td>begin</td>
<td>[bɛg]</td>
</tr>
<tr>
<td>behave</td>
<td>[bɛvaɪ]</td>
</tr>
<tr>
<td>behind</td>
<td>[bɛnd]</td>
</tr>
<tr>
<td>being</td>
<td>[bɛŋ]</td>
</tr>
<tr>
<td>believe</td>
<td>[blɛv]</td>
</tr>
<tr>
<td>belong</td>
<td>[blɛŋ]</td>
</tr>
<tr>
<td>bench</td>
<td>[blɛŋ]</td>
</tr>
<tr>
<td>beneath</td>
<td>[blɛŋ]</td>
</tr>
<tr>
<td>bent</td>
<td>[blɛŋ]</td>
</tr>
<tr>
<td>beside</td>
<td>[blɛŋ]</td>
</tr>
<tr>
<td>bet</td>
<td>[blɛt]</td>
</tr>
<tr>
<td>better</td>
<td>[blɛtər]</td>
</tr>
<tr>
<td>beyond</td>
<td>[blɛnd]</td>
</tr>
<tr>
<td>bid</td>
<td>[blɪd]</td>
</tr>
<tr>
<td>bill</td>
<td>[blɪl]</td>
</tr>
<tr>
<td>bird</td>
<td>[blɜθ]</td>
</tr>
<tr>
<td>birthday</td>
<td>[blɜθd]</td>
</tr>
<tr>
<td>bite</td>
<td>[blɜθ]</td>
</tr>
<tr>
<td>black</td>
<td>[blæk]</td>
</tr>
<tr>
<td>blanket</td>
<td>[blæŋk]</td>
</tr>
<tr>
<td>blew</td>
<td>[blu]</td>
</tr>
<tr>
<td>block</td>
<td>[blæk]</td>
</tr>
<tr>
<td>blood</td>
<td>[blʌd]</td>
</tr>
<tr>
<td>bloom</td>
<td>[blʊm] + m</td>
</tr>
<tr>
<td>blow</td>
<td>[biəʊ]</td>
</tr>
<tr>
<td>blush</td>
<td>[blʌʃ]</td>
</tr>
<tr>
<td>boast</td>
<td>[bɔːst] + t</td>
</tr>
<tr>
<td>body</td>
<td>[bɔːdi]</td>
</tr>
<tr>
<td>bold</td>
<td>[bɔːld] + d</td>
</tr>
<tr>
<td>bond</td>
<td>[bɔːnd]</td>
</tr>
<tr>
<td>book</td>
<td>[bʊk]</td>
</tr>
<tr>
<td>border</td>
<td>[bɔːdr]</td>
</tr>
<tr>
<td>born</td>
<td>[bɔːn] + n</td>
</tr>
<tr>
<td>bosom</td>
<td>[boʊz ə] + m</td>
</tr>
<tr>
<td>boston</td>
<td>[boʊs təʊn]</td>
</tr>
<tr>
<td>bother</td>
<td>[boʊðə]</td>
</tr>
<tr>
<td>bottom</td>
<td>[bɔt ə] + m</td>
</tr>
<tr>
<td>bound</td>
<td>[bɔːnd] + d</td>
</tr>
<tr>
<td>bowl</td>
<td>[bɔːl]</td>
</tr>
<tr>
<td>boy</td>
<td>[bɔː]</td>
</tr>
<tr>
<td>brake</td>
<td>[brɛk] + k</td>
</tr>
<tr>
<td>brand</td>
<td>[brænd] + d</td>
</tr>
<tr>
<td>brave</td>
<td>[brɛv] + v</td>
</tr>
<tr>
<td>break</td>
<td>[brɛk] + k</td>
</tr>
<tr>
<td>breast</td>
<td>[brɛst] + t</td>
</tr>
<tr>
<td>breathe</td>
<td>[brɛt] + s</td>
</tr>
<tr>
<td>brick</td>
<td>[brɪk]</td>
</tr>
<tr>
<td>bridge</td>
<td>[brɪdʒ]</td>
</tr>
<tr>
<td>bright</td>
<td>[brɪt] + t</td>
</tr>
<tr>
<td>bring</td>
<td>[brɪŋ]</td>
</tr>
<tr>
<td>broke</td>
<td>[brɔʊk]</td>
</tr>
<tr>
<td>brook</td>
<td>[brʊk]</td>
</tr>
<tr>
<td>brother</td>
<td>[brʌðə]</td>
</tr>
<tr>
<td>brow</td>
<td>[brɔʊ]</td>
</tr>
<tr>
<td>Word</td>
<td>Pronunciation</td>
</tr>
<tr>
<td>-------</td>
<td>----------------</td>
</tr>
<tr>
<td>brush</td>
<td>[brʌʃ]</td>
</tr>
<tr>
<td>bud</td>
<td>[bʌd]</td>
</tr>
<tr>
<td>built</td>
<td>[bɪlt]</td>
</tr>
<tr>
<td>bull</td>
<td>[bʌl]</td>
</tr>
<tr>
<td>bunch</td>
<td>[bʌŋ] +ŋ</td>
</tr>
<tr>
<td>burden</td>
<td>[bɜːd ˈdɔːr] + n</td>
</tr>
<tr>
<td>burn</td>
<td>[bɜːn]</td>
</tr>
<tr>
<td>bury</td>
<td>[bɜːri]</td>
</tr>
<tr>
<td>bushel</td>
<td>[ˈbʊʃəl]</td>
</tr>
<tr>
<td>busy</td>
<td>[ˈbʌzi]</td>
</tr>
<tr>
<td>butter</td>
<td>[ˈbʌtər]</td>
</tr>
<tr>
<td>buy</td>
<td>[buy]</td>
</tr>
<tr>
<td>by</td>
<td>baɪ</td>
</tr>
</tbody>
</table>
Appendix VI - Sample Output

This is the C B S segments
|D|I|s|I|z|D|x|s|iY|b|iY|E|s|
onsets
|D|s|z|D|s|b|s|
peaks
|I|I|x|iY|iY|E|
codas
|D|s|z|D|s|b|s|
inflectional affix
|z|
syllables
|(D (I s))|(I z)|(D x)|(s iY)|(b iY)|(E s)|
|I s| |(D (x s))|iY|iY|
|s (I z)| |x|(s (iY b))|
|x s|(iY b)|
gc-ed syllables
|(D (I s))|(I z)|(D x)|(s iY)|(b iY)|(E s)|
|(D (x s))|iY|iY|
|(s (iY b))|
|(iY b)|
segments
| D | I | s | I | z | D | x | s | iY | b | iY | E | s |

stray syllable
| (D x) |
| (D x s)) |
| x |
| (x s) |

foot1
| (D (I s)) | (I z) |
| (s iY) | (b iY) | (E s) |
| (I s) |
| (s (iY b)) | (iY) |
| (s (i z)) |
| (iY) |
| (iY b) |

foot2
| ((D (I s)) (I z)) |
| ((s iY) (b iY)) |
| (I s) | (I z) |
| (s (iY b)) | (iY) |
| ((s (I z)) (D x)) |
| (iY (b iY)) |
| ((s (I z)) (D (x s))) |
| ((iY b) iY) |
| ((I z) (D x)) |
| ((iY (E s))) |
| ((I z) (D (x s))) |
| ((iY (E s))) |

foot3
| ((s (I z)) ((D x) (s iY))) |
| ((s (I z)) ((D x) (s (iY b)))) |
| ((I z) ((D x) (s iY))) |
| ((I z) ((D x) (s (iY b)))) |
segments
| D | I | s | I | z | D | x | s | iY | b | iY | E | s |
gc-ed word tree
| (D (I s)) | (I z) | (D x) | (s iY) | (b iY) | (E s) |
| ((D (I s)) (I z)) | (D (x s)) | iY | ((b iY) (E s)) |
  | ((I z) (D x)) | (s (iY b)) | iY |
  | ((I z) (D (x s))) | (iY b) | (iY (E s)) |
  | (((I z) (D x) (s iY))) |
  | (((I z) (D x) (s (iY b)))) |
  | ((s iY) (b iY)) |
  | ((s (iY b)) iY) |
  | ((iY (b iY)) |
  | (((iY b) (iY)) |

canonicalized onsets
| D | I | s | I | z | D | x | s | iY | b | iY | E | s |
canonicalized rhyme
| .Is | .Iz | .x | .i | .i | .Es | .xs | .ib |
canonicalized syllables
| ...Dis | ...Iz | ...Dx | ...s1 | ...d1 | ...Es |
  | ...Dsxs | ...i | ...i |
  | ...sib |
  | ...ib |
sylparts in dictionary
| ...Dis | ...Iz | ...Dx | ...s1 | ...b1 | ...Es |
  | ...Dis | ...i | ...i |
  | ...DsDs |
words in dictionary
| ...THIS | ...IS | ...THE | ...SEE | ...BEE | ...S |
  | ...THIS |
  | ...BE |
  | ...THUS |
  | ...B |
  | ...SEA |
  | ...C |
gc-ed words in dictionary
| ...THIS | ...IS | ...THE | ...SEE | ...BEE | ...S |
  | ...SEA | ...BE |
(The third choice "C" under "SEE" and "SEA" was not printed due to a bug in the printer. The same bug explains why "B" was not printed under "BEE" and "BE".)
Evening news with segments
|qiY|v|n|...G..n.|uW.|z|w|...T|
onsets
|v|n|...n|...z|w|...T|
peaks
|qiY|...|uW|...|
codas
|v|n|...G..n|...z|...T|
infectional affix
...z|
syllables
|qiY|...|(n |)|...|(n uW)|...|(w |)|
|qiY v|...|(n |G)|...|(n uW z)|...|(w |T)|
|...|...|uW|...|
|...|...|(G)|...|(uW z)|...|(T)|
gc-ed syllables
|qiY v|...|(n |G)|...|(n uW)|...|z|...|(w |T)|
|...|(n uW z)|
segments
|qiY|v|n|...G..n.|uW.|z|w|...T|
stray syllable
|...|(n |)|...|(w |)|
|...|(n |G)|...|(w |T)|
|...|...|...|
|...|...|(G)|...|(T)|
foot1
|qiY|...|(n uW)|
|qiY v|...|(n uW z)|
|uW|...|
|...|(uW z)|
foot2
|qiY v|...|(n |)|...|(n uW z)|...|(w |)|
|qiY v|...|(n |G)|...|(n uW z)|...|(w |T)|
|...|(uW z)|...|(w |T)|
|...|(uW z)|...|(w |T)|
foot3
|qiY v|...|(n |G)|...|(n uW)|
|qiY v|...|(n |G)|...|(n uW z)|
segments
qiY v n G n uW z w T
gc-ed word tree
((qiY v) (n (G)) ((n uW) z) (w (T)))

canonicalized onsets
.v [nd] [nd] [z w] [T]
.n [n] [n]

canonicalized rhyme
.i [i] [u] [u u]
.i [iv] [G uzu] [T]

canonicalized syllables
.iv [ndG ndu wT]
.nG [nu nduz nuz]
.wT

syllables in dictionary
.iv [nG nu w]
.wIT

words in dictionary
......EVENING......NEW WITH......
...KNEW....
...NEWS....

gc-ed words in dictionary
......EVENING....NEW z..WITH....
...NEWS...
Sample Output

Walter Cronkite and

segments
onsets
[.w.] [.t.] [.kH.] [.r.] [.n.] [.kH.] [.n.]
[(kH r).]

peaks
[.c.] [.R.] [.a.] [.AY.] [.O-]
codas
[.t.] [.n.] [.t>] [.n.]

inflectional affix

syllables
[(w (c t)). [.R.] ((((kH r) (a n)) (kH AY))) ((O- n)).]
[(c t).]
[(r (a n)). ((kH (AY t>))]
[(t R).]
[(a n).] [.AY.]
[(AY t>)]
gc-ed syllables
[(w (c t)). [.R.] ((((kH r) (a n)) (kH AY t>)) (O- n)).]

segments

stray syllable
[(t R).]

foot1
[(w (c t)). ]
[((kH r) (a n)) (kH AY)]
[ (O- n)].
[(c t).]
[(r (a n)). ((kH (AY t>))]
[(a n).] [.AY.]
[(AY t>)]

foot2
[((w (c t)) .R.) ((((kH r) (a n)) (kH AY))]
[((c t) .R.) ((((kH r) (a n)) (kH (AY t>))]
[[(r (a n)) (kH AY))]
[[(r (a n)) (kH (AY t>))]
[[(a n) (kH AY))]
[[(a n) (kH (AY t>))]
[[(kH (AY t>)) (O- n)]
[[(AY t>) (O- n)]

foot3
[((w (c t)) (R, ((kH r) (a n))))]
[((c t) (R, ((kH r) (a n))))]
[[(w (c t)).]
[[(kH r) (a n)) (kH AY)]
[[(c t).]
[[(r (a n)). ((kH (AY t>))]
[[(a n).] [.AY.]
[[(AY t>)]

Appendix VI
segments
[.w.|.c.|.t.|R.,|kH.|.r.|.a.|.n.|kH.|AY.|t>|θ-|n.]
gc-ed word tree
[.w.(c|t).|R.,|((kH |r| (a n))).|((kH (AY t))).|θ-|n.]
[.w.(c|t).|R.,|((kH |r| (a n)) (kH (AY t))).]
[.w.(c|t).|R.,|((kH |r| (a n)))|((kH (AY t))).|θ-|n.]
segments
[.w.|.c.|.t.|R.,|kH.|.r.|.a.|.n.|kH.|AY.|t>|θ-|n.]
canonicalized onsets
[.w.|.t.|.k.|.r.|.nd.|.k.|.nd.]
[.kr.|.n.|.n.]
canonicalized rhyme
[.ct.|.|R.|.|nd.|.|A.|.|nd.|]
[xr.|.|an.|.|At.|]
canonicalized syllables
[.wct.|.|R.|.|kran.|.|kAt.|.|nd.|]
[xr.|.|kran.|]
sylparts in dictionary
[.wct.|.|R+3|.|kran.|.|kAt.|.|nd.]
[.|n.]
words in dictionary
[.WATER.|.|.CROWNITE.|.|AND.]
[.WALTER.|.|AIR.|.|OUR.|.|HOU.]
gc-ed words in dictionary
[.WATER.|.|.CROWNITE.|.|AND.]
[.WALTER.|.|.AND.|.|.AN...|
tonight  part  two

segments
| TH | U | n | AY | qt | pH | a | r | qt | TH | uW |

onsets
| TH | n | qt | pH | r | qt | TH |

peaks
| U | AY | a | uW |
| (a r) |

codas
| n | qt | qt |

inflectional affix

syllables
| (TH U) | (n AY) | (pH ((a r) qt)) | (TH uW) |
| (TH U n) | AY | ((a r) qt) | uW |
| U | (n (AY qt)) |
| (U n) | (AY qt) |

gc-ed syllables
| (TH U) | (n (AY qt)) | (pH ((a r) qt)) | (TH uW) |
| (TH U n) | (AY qt) |

segments
| TH | U | n | AY | qt | pH | a | r | qt | TH | uW |

stray syllable

foot1
| (TH U) | (n AY) | (pH ((a r) qt)) | (TH uW) |
| (TH U n) | AY | ((a r) qt) | uW |
| U | (n (AY qt)) |
| (U n) | (AY qt) |

foot2
| ((TH U) (n AY)) | ((pH ((a r) qt)) | (TH uW) |
| ((TH U n) AY) | ((a r) qt) | (TH uW) |
| ((TH U) (n (AY qt))) |
| ((TH U n) (AY qt)) |

foot3
segments
\{tH.|U.|n.|AY.|qt.|pH.|a.|r.|qt.|tH.|uW.\}
gc-ed word tree
\{(tH U).\{(n (AY qt))\{(pH ((a r) qt))\{(tH uW)\}\}\}\{(tH U n)).\{(AY qt)\{(pH ((a r) qt))\{(tH uW)\}\}\}\}\{(tH U) (n (AY qt))\}\\{(tH (U n)) (AY qt)\}\{(n (AY qt))\{(pH ((a r) qt))\}\\{(AY qt)\{(pH ((a r) qt))\}\\}
canonicalized onsets
\{t.|\{nd.|\{t.|p.|\{r.|t.|t.|\\{n.|\}
canonicalized rhyme
\{U.|\{A.|\{...art..\}|\{u.|\{..Und.|\{At..\|\{..Un...\|\{nAt.....\|
canonicalized syllables
\{tU...\{ndAt.....\{part.....\{tu...
\{tUnd.....\{At...
\{tUn.....\|\{nAt.....\|\}
sylparts in dictionary
\{tU...\|\{..At....\{part.....\{tu...
\{tUn.....\|\{tun.....\|
words in dictionary
\{tWO+3.|\{..PART.....\{tWO+3.|\{tWO+3.|\{tUNE.....\|\{TONIGHT.....\|
gc-ed words in dictionary
\{..TONIGHT.....\{..PART.....\{tWO+3.|
Appendix VII - Glossary

In preparing an interdisciplinary document such as this, it has become impossible to give an adequate
definition of all terms in the text. In this appendix, we hope to alleviate difficulties that the reader may have
encountered by providing a usable working approximate definition. This appendix is intended to aid the
reader in unfamiliar territory. We hope that experts will not be too disturbed by gross over-simplifications.
Many of these definitions are quoted directly from *A First Dictionary of Linguistics and Phonetics* [24].

**acoustic feature/cue:** A characteristic of a speech sound when analyzed in physical terms, e.g., fundamental
frequency, amplitude, harmonic structure. Such analyses are provided by acoustic phonetics, and it is
possible to make acoustic classifications of speech sounds based upon such features, as when one classifies
vowels in terms of their formant structure. The acoustic properties of a sound which aid its identification
in speech are known as acoustic cues. In the distinctive feature theory of phonology of Jakobson & Halle,
acoustic features are the primary means of defining the binary oppositions that constitute the phonological
system of a language.

**acyclic:** Opposed to cyclic. See cyclic and directed acyclic graph.

**adjacency coding:** A term using in computer science referring to a representation of matrices. Adjacency
coding represents the non-zero elements associated with their locations. In this way, adjacency coding
optimizes storage costs for sparse matrices. See §5.2.

**affix:** Originally, the affix position was restricted to *inflectional affixes* (e.g., plural *-s*, past tense *-ed*, passive
*-en*, *-*ed, and progressive *-ing*) and *derivational affixes* (i.e., prefixes such as *un-*, suffixes such as *-ness*, and
infixes in other languages). Derivational affixes can also be split into two types: + and #. + affixes are
generally derived from Latin and # affixes tend to be historically associated with Greek and ancient
Germanic languages. These historical trends are statistically correct, but not perfect. *-ist*, for example, is
exceptional since it is a + affix, and it is yet derived from German. Many linguists would prefer to define
the difference between + and # affixes in some other way. (See Appendix III.)

In more recent treatments, the affix position has been generalized by Hayes [40, 41] to include
morpheme final segment(s) in order to obtain a more accurate account of stress assignment. These final
segments are sometimes called extrametrical. Fujimura and Lovins [35] have also generalized the affix
position in a different way in order to account for phonotactic constraints in the rhyme. They noticed that
morpheme final coronal consonants such as the /ksθs/ in *sixths* violate a number of phonotactic
constraints (e.g., length restrictions, sonority). Thus, if we assign these consonants to a separate position (i.e., the affix), we are in a better position to excuse their exceptional behavior.

affricate: The manner feature associated with [k] (e.g., church) and [t] (e.g., judge).

allo-: A prefix for referring to any noticeable variation in the form of a linguistic unit which does not affect the unit's functional identity. The formal variation noted is not linguistically distinctive, i.e., no change of meaning is involved. Examples: allographs (variants of the same grapheme, e.g., A, a, .ByteArray), allophones (variants of the same phoneme, e.g., [tʰ] and [t̪]), allomorphs (variants of the same morpheme, e.g., the plural morpheme can be realized as /s/ in caps, /z/ in logs, tensing in mice, and 0 in sheep).

allophone: A variant of a phoneme (e.g., the aspirated /t/ in Tom, the flapped /r/ in butter, an unreleased /l/ in cat and a glottalized /u/ in atlas).

alternation: The relationship that hold between alternative forms (or variants) of a linguistic unit. In phonology, for example, the related vowel qualities of such words as telegraph ~ telegraphic, receive ~ reception are sometimes described as alternants, as are the various allophones of a phoneme, and the various allomorphs of a morpheme. See allo-.

ambiguous: A term in computational linguistics for structures that have two or more interpretations. Sentences that can be assigned two or more parse trees (constituent structures) are said to be syntactically ambiguous. Words that have two or more lexical entries are said to be lexically ambiguous. There are many other types of ambiguity: semantics, thematic, referential, pragmatic, etc.

analysis-by-synthesis: A theory of speech perception which credits listeners with an internal, language-specific mechanism that responds to incoming speech by selecting certain acoustic cues, and then attempting to synthesize a replica of the input. Analysis-by-synthesis is very similar to the AI paradigm of generate-and-test.

arc: A relation between two nodes in a graph (or lattice or matrix). An arc is equivalent to an edge.

articulatory phonetics: The branch of phonetics which studies the way in which speech sounds are made (articulated) by the vocal organs.

ascii: The standard representation of letters (graphemes) in many (but not all) computer systems.
aspiration: A term in phonetics for the audible breath which may accompany a sound's articulation, as when plosive consonants are released. It is usually symbolized by a small raised [ʰ] following the main symbol. In examples such as English pin [pʰiIn], the aspiration may be felt by holding the back of the hand close to the mouth while saying the word; the contrast with bin, where there is no aspiration is noticeable. Some languages, such as Hindi, have a contrast of aspiration applying to both voiceless and voiced stops. In English, aspiration is generally controlled by the phonetic environment: voiceless stops such as /p/ aspirate in foot initial position, but not in other positions (e.g., stop, spin). Aspiration devoices a following /l, r, w, y/ in please, twice, queue, but only when the aspiration and the semi-vowel are found in the same constituent. (Recall the contrast between gray plane and grape lane; devoicing is found in the first case, but not in the second.)

assimilation: A general term in phonetics which refers to the influence exercised by one sound segment upon the articulation of another, so that the sounds become more alike, or identical. Examples: voicing assimilation (e.g., the plural morpheme agrees in voicing with the preceding phonemes; note the /s/ in cats and the /z/ in dogs), place assimilation (e.g., nasal consonants agree in place with the following consonant in words such as cent, ramp, ring). Assimilation is opposed to dissimilation.

augmented transition network (ATN): A class of parsers, popularized by Woods [125]. ATNs are a generalization of finite-state transition networks (also known as Markov chains and finite-state machines). Woods augmented finite-state networks in two ways. First, he allowed networks to recursively refer to other networks. For example, the S network could refer to the NP and VP networks, and each of these networks could recursively refer back to the S network. Woods called this machine a recursive transition network (RTN); it has the (strong) generative capacity of a context-free grammar. Woods then added registers to the machine, so that it could process "transformations" like passive and wh-movement. The resulting machine, which he called an augmented transition network, has the (strong) generative capacity of an arbitrary rewrite grammar (or a Turing Machine).

automaton: A term from computer science referring to an abstract class of machines. Examples: finite-state machines, push-down automata (PDA), linear bounded automata (LBA), Turing Machines (TM). These machines are strongly equivalent to regular grammars, context-free grammars, context-sensitive grammars and arbitrary rewrite-rule grammars, respectively.
**binary search**: A term from computer science for a particular search algorithm that splits the search space in half on each iteration. Imagine that we were trying to find someone’s number in the phone book. A binary search would split the phone book into two halves and ask which half the target number is in. If the target is in the first half, then the binary search algorithm will split the first half into two quarters and ask which quarter the target is in. In this way, the algorithm will find the target in \( \log_2 n \) steps, where \( n \) is the size of the search space (e.g., the number of numbers in the phone book).

**batch**: A term from computer science referring to a mode of programming where users submit jobs to the system. After the program has been executed, the user receives a listing of the output. Batch mode is opposed to interactive programming, where users are constantly interacting with the operating system via an interpreter. Batch mode attempts to limit the number of transactions between the user and the system. Batch mode is usually cheaper for executing working programs, whereas interactive mode tends to be more convenient for developing new code.

**bilabial**: A term in the classification of consonant sounds on the basis of their place of articulation: it refers to a sound made by the coming together of both lips. Examples are the initial sounds in *pin, bin, mat, vat, fat*.

**bleed**: See feed.

**bottom (of lattice)**: See lattice.

**bound morpheme**: A morpheme that must be bound to another morpheme and does not form a word by itself (e.g., *cran-* as in *cranberry* and *crimin-* as in *criminal*). Opposed to free morpheme.

**bug**: A computational term for an error. According to rumor, the term was derived historically when someone actually fixed (debugged) a computer by literally removing a bug from its insides.

**cache memory**: A term in computer science referring to a memory architecture. Memory would be partitioned into two types: a cache of fast, but expensive memory and a main store of slower and cheaper memory. The cache would be used to store recently accessed values on the grounds that they were likely to be accessed again in the near future.

**chart**: The data structure (or well-formed-substring-table) of a chart parser (e.g., Earley’s algorithm [30]). The chart stores a set of triples \(<i, j, c>\), where \( c \) is a terminal or non-terminal category in the grammar (otherwise known as an ATN state or a part of speech), and \( i, j \) are pointers into the input sentence denoting the left and right edges of the phrase. For example, the phrase \([NP \textit{flying planes}]\) in the input
sentence $_0$ *They* $^1$ are $^2$ *flying* $^3$ *planes* $^4$ would be represented as the triple: $\langle$NP, 2, 4$\rangle$.

**Chomsky Hierarchy**: A term in linguistics and computer science referring to the four properly nested classes of grammars/automata: finite-state, context-free, context-sensitive, arbitrary rewrite-rules.

**clobber**: A term in computer science used to refer to what side-effecting operations do to the environment. See *side-effect*. (The status of the term *clobber* is unclear; it was once considered to be slang, but more recently, it seems to be becoming part of the accepted technical jargon.

**closure**: A general term used in phonetics to refer to an articulation where the contact between active and passive articulators obstructs the air stream through the mouth and/or nose. A ‘complete closure’ exists in the case of plosives, affricates and nasal, and in the glottalic and velaric air-stream mechanisms. An ‘intermittent closure’ exists in the case of rolls, flaps and taps. A ‘partial closure’ exists in the case of laterals. Some phoneticians would include fricatives under the heading of ‘partial’ or ‘incomplete’ closure. A narrowing of the vocal tract where there is no articulatory contact is usually called a stricture.

**cluster**: A term used in the analysis of connected speech to refer to any sequence of adjacent consonants occurring initially or finally in a syllable, such as the initial /br-/ of *bread* or the final /-st/ of *best*.

**coda**: A term used in phonetics and phonology to refer to post-vocalic segments.

**code**: A slang term in computer science for a program.

**color**: A phonetic attribute of semi-vowels (especially /l/). In English, semi-vowels are generally light in onset position and dark in coda position.

**compiler**: A computational term for a machine that (partially) evaluates a program for a large class of possible inputs. It is often contrasted with an *interpreter*. (See *interpreter*).

**compile-time**: A computational term for the time when the compiler is working on a program. It is often used more generally for the state of the machine during compilation. *Compile-time* is often contrasted with *run-time*.

**constituent**: A basic term in grammatical analysis for a linguistic unit which is a component of a larger construction. In syntax, the term constituent is used interchangeably (for the most part) with the term phrase. In phonetics, a constituent is often referred to as a *cluster*, at least at the level between segments and syllables. Constituency is generally defended on two grounds. First, constituency is shown to simplify the grammar by capturing generalizations common to many “transformational” processes such as passive, wh-movement, raising, equi, question formation, etc. Secondly, constituency seems intuitively to
be a natural unit of meaning in a theory of logical form.

constraint: (1) In Artificial Intelligence, a constraint is a restriction on the search space. In some sense, it is the opposite of the statistical notion of a degree of freedom. Similar to knowledge source and level of representation. (2) In Linguistics, a constraint is a condition on the application of a transformation. (The Linguistic usage is not intended in this thesis.)

constriction: A general term used in articulatory phonetics to refer to a narrowing within the vocal tract. The different kinds and degrees of constriction are the basis of the articulatory classification of sound qualities.

core (memory): A term from computer science referring to a particular type of computer memory. Historically, core memory was constructed out of doughnut-shaped magnetic cores. Now that magnetic cores are being replaced with more modern technologies (e.g., silicon chips), the term core memory is taking on a new meaning. Core memory is real and physical (as opposed to virtual memory). In addition, core memory has relatively fast access (as opposed to disk, drum and tape and other types of secondary memory). Core memory is sometimes referred to as primary memory.

coronal: A distinctive feature for describing place of articulation. Coronal sounds are defined articulatorily as those produced with the blade of the tongue raised from its neutral position. Includes alveolar, dental, and palato-alveolar consonants: /t, d, l, n, s, z, θ, ſ/.

critical band spectrogram: See spectrogram.

cut-off frequency: A term from acoustic phonetics used to describe stop bursts and strong fricatives. Imagine what a burst looks like on a spectrogram. The cut-off frequency refers to the frequency of the bottom of the burst.

cycle: An organization of transformational rules.

dark: An impressionistic, but commonly used term for a variety of lateral sound, where the resonance is that of a back vowel of an [u] quality, as in the standard English pronunciation of /l/ [ɬ] after vowels, before consonants, and as a syllabic sound, e.g., pull, altar, bottle; it is opposed to clear /l/ or light /l/.

database: A term from computer science for a collection of data (e.g., a lexicon, a phone book). The data are composed of (key, value) pairs. The keys are used to access the values. For example, a dictionary is a database of lexical entries keyed (or indexed) on the spelling. The process or re-organizing the database on a new key is called inverting. There are many different schemes for storing, searching, and adding to a database: hash tables, binary trees, discrimination networks, etc.
debug: A computation term referring to the process of correcting a program. See bug.

dental: A place of articulation. We use it more or less interchangeably with coronal, though it should be reserved for sounds involving the teeth (e.g., /θ, ð/) and not for sounds like /t, d/ according to [24].

devolved schwa: The reduced vowel in to Tom written [ɪ]. It generally appears between two unvoiced segments. On a spectrogram, it often looks more like frication than vocalic.

diacritic: In phonetics, a mark added to a symbol to alter its value, e.g., the various accents, the signs of devoicing and nasalization.

diffuse: A distinctive feature in phonology. It describes the place of articulation for sounds formed with a relatively relaxed back (e.g., velar consonants).

diphthong: A term used in the phonetic classification of vowel sounds on the basis of their manner or articulation; it refers to a vowel where there is a single (perceptual) noticeable change in quality during a syllable, as in beer, time, loud.

directed acyclic graph (DAG): A restriction on a graph where edges are directed and acyclic. A directed edge is written with an arrow. In general, there will not be an edge (j, i) if there is a node (i, j). (In contrast, an undirected graph with always have an edge (j, i) if there is an edge (i, j)). An acyclic graph is free of cycles; that is, there is no path (sequence of edges) from a node i back to itself (excluding trivial reflexive paths).

discrimination network: A term from computer science for a particular organization of a database. See appendix I. Keys are assumed to be sequences of subkeys. The database is organized so that the lookup procedure finds an entry by discriminating on the first subkey, then the second subkey, and so on. The discrimination is implemented in different ways depending on the fanout. If there are only a few possible subkeys, then the discrimination is generally implemented with a linear search. As the fanout increases, then it makes sense to use more efficient methods (e.g., binary search, hash tables).

dissimilation: A term used in phonology to describe non-agreement constraints. Opposed to assimilation.

There aren't very many good examples of dissimilation in English. According to some, there is a place dissimilation in onset position, explaining why we don't see onsets like */tl/*, */dl/*, */pw/* and */bw/*.
(The absence of these onsets, though, is very controversial.)
distinctive(ness): A term used in Linguistics for any features of speech (or writing) which enable a contrast to be made between phonological, grammatical or semantic units. Such contrasts might also be labeled 'relevant', functional or significant. The main use of the term has been in phonology, as part of the phrase distinctive feature, where it refers to a minimal contrastive unit recognized by some linguists as a means of explaining how the sound system of languages is organized. Distinctive features may be seen either as part of the definition of phonemes, or as an alternative to the notion of the phoneme. The first of these views is found in the approach of the Prague School, where the phoneme is seen as a bundle of phonetic distinctive features: the English phoneme /p/, for example, can be seen as the result of the combination of the features of bilabiality, voicelessness, plosiveness, etc. Other phonemes will differ from /p/ in respect of at least one of these features. In distinctive feature theories of phonology, however, the phoneme is not considered to be a relevant unit of explanation: symbols such as p, b, etc. are seen simply as convenient abbreviations for particular sets of features. It is the features which are the minimal units of phonological analysis, not the phonemes. It is argued that by substituting features for phonemes in this way, generalizations can be made about the relationships between sounds in a language, which would otherwise be missed. Moreover, because features are phonetic units, it should be possible to make inter-language (e.g., historical and dialectal) and cross-language comparisons, and ultimately statements about phonological universals, more readily than by using a phonemic model of phonology.

Distinctive feature analysts claim that there are several advantages over the traditional phonetic alphabet approach to phonological description, which describes utterances as a sequence of segments. For example, it was originally suggested that a relatively small set of abstract feature oppositions (a dozen or so) would account for all the phonological contrasts made in languages: it would not then be necessary to recognize so much phonetic classificatory detail, as exists on, say, the IPA chart, where the phonological status of the segments recognized is not indicated. In fact, it has turned out that far more features are required, as new languages come to be analyzed. Another advantage, it is suggested, is that consonants and vowels can be characterized using the same set of phonetic features (unlike traditional descriptions, where the classificatory terminology for vowels — high, low, etc. — is quite different from that used for consonants — labial, palatal, etc.).

By using a system of this kind, some quite specific predictions can be made about the sound systems of languages. For example, using the Jakobson and Halle system below enables one to distinguish phonologically two degrees of front/back contrast in the consonant system and three degrees of vowel height. But what follows from this is a universal claim, that no languages permit more than these numbers
of contrasts in their phonological systems. There are empirical claims, of course, and in recent years much effort has been spent on investigating these claims and modifying the nature of the feature inventory required.

Two major statements concerning the distinctive feature approach have been made: one by Roman Jakobson and Morris Halle, in *Fundamentals of Language* (1956), the other by Noam Chomsky and Morris Halle, in *The Sound Pattern of English* (1968). The Jakobson and Halle approach set up features in pairs, defined primarily in acoustic terms (as could be detected on a spectrogram), but with some reference to articulatory criteria. Examples of their features include vocalic vs. non-vocalic, consonantal vs. non-consonantal, compact vs. diffuse, grave vs. acute, nasal vs. oral, discontinuous vs. continuant, strident vs. mellow, flat vs. sharp/plain and voiced vs. voiceless. The emphasis in this approach is firmly on the nature of the oppositions between the underlying features involved, rather than on the description of the range of phonetic realizations each feature represents. In the Chomsky and Halle approach, more attention is paid to the phonetic realizations of the underlying features recognized, and a different system of feature classification is set up. Some of the earlier features are retained (e.g., voice, consonantal, tense, continuant, nasal, strident), but many are modified, and new features added, some of which overlap with the earlier approach (e.g., sonorant vs. obstruent, delayed vs. instantaneous release, anterior vs. non-anterior, coronal vs. non-coronal, distributed vs. non-distributed, syllabic vs. non-syllabic). The application of these features to languages is not without controversy, and in recent years further suggestions have been forthcoming as to the need for additional features, such as labiality.

dynamic time warping: A recognition algorithm that attempts to match a speech utterance with a template by stretching the utterance and the template in time in order to obtain the best fit according to some distance metric. The algorithm is an instance of a broad class of algorithms called dynamic programming. (Matrix multiplication and context-free parsing are other cases of dynamic programming.)

dynamic programming: see arc.

effective procedure: A term from computer science referring to a procedure that can be computed on a Turing Machine. Note all procedures are effective. For example, it cannot be decided on a Turing Machine whether or not a procedure will halt. This problem is known as the Halting Problem.
environment: In computer science, the environment refers to variable bindings. In linguistics and phonetics, the environment refers to specific parts of an utterance (or text) near or adjacent to a unit which is the focus of attention. Features of the linguistic environment may influence the selection of a particular unit, at a given place in an utterance, and thus restrict its occurrence, or distribution. For example, in phonology, whether a consonant phoneme is lip-rounded or not may depend on the presence of a rounded vowel in its phonetic environment. Sounds are referred to as being 'conditioned' by their environment. In grammar, the term is used similarly, e.g., the occurrence of one morpheme may depend on the prior use of another in its environment, as with cran-, which occurs only in the grammatical environment of -berry. In recent years, the term context has come to be widely used in this sense.

empethesis: A term used in phonetics and phonology to refer to a type of intrusion, where an extra sound has been inserted medially in a word. The phenomenon is common both in historical change and in connected speech (e.g., incredible → inc[ə]redible).

equivalence: See generative capacity.

equivalence relation: An algebraic term for a relation that is reflexive, symmetric, and transitive. An equivalence relation is often written with the symbol “=“.

extrametrical: A constituent that is exempt from the usual metrical stress assignment rules. Hayes [40, 41] employs the mechanism heavily in order to explain the distribution of stress. In particular, extrametricality is central to Hayes’ account of English verbs and unsuffixed adjectives [41, p. 238]. These words receive final stress if they end in a string of at least two consonants (e.g., torment, usúr, robúst, ovért) or with a syllable having a long vowel (e.g., obéy, atóne, divíne, discréet), otherwise penultimate stress (e.g., astónish, dévélóp, cómmón, illícit). Hayes accounts for this distribution of stress with two assumptions: the main stress rule and consonant extrametricality. The main stress rule assigns primary stress to the rightmost branching rhyme (syllable ending in two consonants or a long vowel). Consonant extrametricality exempts the final consonant from the main stress rule. These two assumptions account for the stress facts cited above. The first class (e.g., torment) receive final stress because the final rhyme is branching (dominates two or more phonemes) even without the final extrametrical consonant. The second class (e.g., obéy) also received final stress because of the long vowel in the final syllable. The third class (e.g., dévélóp) is assigned penultimate stress because of extrametricality. Extrametricality strips away the final consonant (e.g., develop) leaving the final syllable non-branching (dominating less than two segments). Thus, main stress is assigned to the penultimate syllable which is the right most branching syllable. (Extrametricality applies slightly differently in nouns in order to account for the facts.)
feature: A term used in linguistics and phonetics to refer to any typical or noticeable property of spoken or written language. Features are classified in terms of the various levels of linguistic analysis, e.g., 'phonetic/phonological/grammatical features' or in terms of a dimension of description, e.g., 'acoustic/articulatory/auditory features'.

feed: A technical linguistic term relating to the interaction of two or more transformational rules. A transformational rule (e.g., + affixation) is said to feed another transformational rule (e.g., tri-syllabic laxing) if the first rule sets the environment up so that the second rule can apply. The term feed is opposed to the term bleed; bleeding occurs when one transformational rule disturbs the environment so that the second rule cannot apply.

finite-state grammar: A grammar that can be processed by a finite state machine (Markov network). These grammars include all non-center-embedded context-free grammars. (A center-embedded grammar has a production or series of productions such that a non-terminal X derives aXb where a and b are non-empty.) Finite-state grammars are also known as regular grammars.

flap (or tap): A phonetic sound produced by a single rapid contact between two organs of articulation (excluding vocal cord vibration). The usual occurrence of this is in the production of types of /t, d, n, r/. (We have not discussed the /r/ flaps.) Some phoneticians distinguish between flap and tap (taps are shorter), but we don't. The duration of a flap is usually between 10ms. and 30ms.

foot: A term used by some phoneticians and phonologists to describe the unit of rhythm in languages displaying isochrony, i.e., where the stressed syllables fall at approximately regular intervals throughout an utterance. It is an extension of the term used in traditional studies of metrical verse structure, where the many regular patterns of stressed/unstressed syllable sequence were given a detailed classification (e.g., 'iambic' for a two syllable sequence of unstressed syllable followed by a stressed one; 'trochaic' for a stressed + unstressed pattern (the usual in English); 'spondaic' for stressed + stressed; 'dactylic' for stressed + unstressed + unstressed). In modern treatments (e.g., [78]), the metrical foot is used to capture stress generalizations at the word level (e.g., [missi] [ssippi]) and at the phrase level (e.g., [light house] keeper vs. light [house keeper]).

formant: A term in acoustic phonetics of a particular value in the classification of vowels and vowel-like sounds, and of transitional features between vowels and adjacent sounds. A formant is a concentration of acoustic energy, reflecting the way air from the lungs vibrates in the vocal tract, as it changes shape. (The formants correspond to poles in the vocal tract transfer function.) Formants show up clearly on a
spectrogram (plot of energy as a function of time and frequency) as thick black lines. The first three formants (written \(F_1\), \(F_2\) and \(F_3\)) provide the basis of vowel description. These three formants (and especially the first two) are related to the distinctive features HIGH/MID/LOW and FRONT/BACK.

**free morpheme:** A morpheme that forms a word by itself (e.g., *berry* in *cranberry*). Opposed to bound morpheme.

**frication:** A term from phonetics for the portion of a strong fricative or stop where two organs come so close together that the air moving between them produces audible frication. Friction is clearly visible on a spectrogram as noisy dark black smudges at high frequencies.

**fricative:** A manner of articulation. Also known as 'spirant'. The term refers to sounds made when two organs come so close together that the air moving between them produces audible friction. There is no complete closure between the organs (in which case a plosive articulation would be produced): there is simply a stricture or narrowing. Examples of fricatives are: *fin* /f/, *yan* /v/, *thin* /θ/, *this* /ð/, *sin* /s/, *zoo* /z/, *ship* /ʃ/, *measure* /m/, and (according to some) *judge* /j/, *church* /ɛ/, *hoop* /h/. Fricatives can be further sub-divided in a number of ways (e.g., strong/weak, labial/coronal/palatal/velar, voiced/unvoiced). Strong fricatives /s, z, ð, j, ɛ/ are much darker on a spectrogram than weak fricatives (all the rest).

**garbage collection:** A computational term for a procedure that reclaims storage that can no longer be referenced. The procedure marks all memory cells that can be referenced by marking all cells on the stack and then recursively marking all cells that they reference and all cells that they reference and so on. The remaining cells are swept away and placed on the free list. This algorithm is also called a mark-sweep. Garbage collecting (automatic storage de-allocation) is most common in programming languages like Lisp. In other languages, such as PL/I, storage is de-allocated manually with a free command. There are a number of trade-offs between automatic and manual de-allocation of memory. Manual de-allocation tends to be a source of bugs, but automatic methods can be too slow for many applications, especially problems with real time constraints (where one cannot afford to wait while the garbage collector runs).

**gemination:** A term from phonetics for the combination of two short adjacent consonants into one long one. For example, the /sʃ/ in *gas shortage* can be geminated (in some analyses) into a single long /ʃ/.
generate-and-test: A brute force search paradigm from Artificial Intelligence. Large numbers of hypotheses are generated sequentially (generally in a straightforward unintelligent fashion). After each hypothesis is generated, it is tested to see if it matches the goal.

generative capacity: The set of structural descriptions generated by a grammar (or automaton). To grammars (or automata) are considered to be strongly equivalent if they generate the same structural descriptions. Two grammars (or automata) are considered to be weakly equivalent if they generate the same set of sentences (language).

glide: A term used in phonetics to refer to a transitional sound as the vocal organs move towards or away from an articulation (on-glide and off-glide respectively). An example is the /y/ glide (sometimes written [j]) heard in some pronunciations of words like tune (ɪˈtjuːn). Diphthongs are sometimes referred to as 'gliding vowels'.

glottis: The aperture between the vocal cords.

glottalization (glottal stop): A closure a glottis. On a spectrogram, glottalization is seen as irregular pitch pulses. In some languages, glottalization is phonemic. In English, glottalization is only phonetic. Glottalization is often found in utterances beginning with a vowel. (Intuitively, the vocal cords are a bit like automobile engines. When you first start them up on a cold day, they jerk forward at irregular intervals until they warm up.) Glottalization can also be found in certain stops such as the /t/ in atlas (əˈtæləs) and (in Cockney) in bottle (bɒtl).

graph: A term from computer science and algebra for a collection of nodes and edges. According to [1 p. 50], “a graph \( G = (V, E) \) consists of a finite, nonempty set of vertices \( V \) and a set of edges \( E \). If the edges are ordered pairs \( (v, w) \) of vertices, then the graph is said to be directed; \( v \) is called the tail and \( w \) the head of the edge \( (v, w) \).” There are many equivalent names for these terms. Nodes are also called vertices and points; edges are also called arcs, branches or relations. We use the terms lattice and matrix more or less interchangeably with graph. Technically, a lattice is a restriction on a graph; a lattice is a directed acyclic graphic (DAG) with a unique top node \( \top \) and a unique bottom \( \bot \). The top node is also known as the greatest element, the beginning and the start node. The bottom node also goes by many other names: the end, the goal, the target, the least element. (This computation usage of the term graph should not be confused with the linguist notion of a grapheme: a character or letter from the alphabet.)
Halting Problem: A term from computer science referring to a classic example of an undecidable problem.

See effective procedure.

hash table: A term from computer science for an organization of a data base for efficient retrieval. There is function (called a hashing function) that maps the space of keys down into a relative small space of numbers. These values are then used as addresses into the structure (called a hash table).

holistic recognition system: Holistic systems are opposed to compositional systems such as our own proposal.

Holistic systems tend to work by matching the input utterance against a set of templates in the lexicon (Templates are used to represent lexicon entries; a template is a speech utterance with some (but not much) data compression.) Holistic systems have enjoyed some success in small vocabulary tasks, especially for speaker dependent, isolated words. For large vocabulary tasks, it is questionable whether holistic systems are appropriate because of the space and time requirements, and because of problems associated with the collection of suitable templates.

homophone: A term used in semantic analysis to refer to words (i.e., lexemes) where have the same pronunciation, but differ in meaning (e.g., threw/through and rode/rowed.

homorganic: A general term in the phonetic classification of speech sounds, referring to sounds which are produced at the same place of articulation, such as /p, b, m/.

International Phonetic Association (IPA): An organization founded in 1886 by a group of European phoneticians (Paul Passy (1859-1940) and others) to promote the study of phonetics. In 1889 it published the International Phonetic Alphabet which, in modified and expanded form, is today the most widely used system for transcribing the sounds of language.

invariance: A principle in some approaches to phonology whereby each phoneme is seen as having a set of defining phonetic features, such that whenever a phoneme occurs the corresponding features will also occur. Along with the conditions of linearity and biuniqueness, the invariance principle establishes a view of phonemic analysis which has been criticized by generative phonologists, as part of a general attack on taxonomic phonemics.

interactive: Opposed to batch. See batch.
**Glossary**

**interpreter**: A computational term for a machine that evaluates programs for a particular input. The term is often contrasted with a **compiler**, which partially evaluates a program for a class of possible inputs. As a general rule of thumb, there is an interpretation/compilation trade-off. Interpretation is preferable for "throw away" programs that are only going to be executed a few times; compilation is more efficient for programs that will be run very often.

**juncture**: A term used in phonology to refer to the phonetic boundary features which may demarcate grammatical units such as morpheme, word or clause. The most obvious junctural feature is silence, but in connected speech this feature is not as common as the use of various modifications to the beginnings and endings of grammatical units.

**key**: A term in computer science and in particular, data base management. A key is the handle which the look-up procedure uses when accessing the data base. For example, the spelling of a word could be used as a key for accessing a dictionary.

**Knowledge Source (KS)**: A term from Artificial Intelligence (especially common at CMU and Stanford) to refer to a set of restrictions imposed by a particular domain of application. Typical knowledge sources in the speech area include: phonology, phonetics, syntax, semantics and pragmatics. The term is (more or less) equivalent to **constraint** (common in Artificial Intelligence at MIT) and **level of representation** (common in linguistics).

**labial**: A general term in the phonetic classification of speech sounds on the basis of their place of articulation: it refers to active use of one's lip(s). Examples of labial sounds are /f, v, p, b, m/.

**labial transitions**: A term from acoustic phonetics for the spectral transitions that are often associated with labial sounds. Labial sounds tend to pull $F_2$ down in adjacent sounds because it takes some time to move the lips from the protruded position into some other position. In the mean time, with the lips protruded, the vocal cavity tends to be longer than normal, causing a drop in resonant frequencies (formants).

**lambda binding**: A term in computer science for the acceptable from of communication between functions (as opposed to side-effects). In many programming languages, lambda binding is the default binding mechanism employed when a function is invoked. The dummy variables in the function are said to be lambda bound to the arguments in the function call. The term lambda binding is derived from Church's lambda calculus [18]. The term lambda binding is also used in linguistic analyses of logical form.
lateral(-ly): A term used in the phonetic classification of consonant sounds on the basis of their manner of articulation: it refers to any sound where the air escapes around one or both sides of a closure made in the mouth, as in the various types of /l/ sound. Lateral sounds may be voiced as in lady, pool or voiceless as in play, where the /l/ have been devoiced due to the influence of the preceding voiceless consonant.

lattice: A term from computer science and mathematics for a restricted form of a graph. (See the definition of graph, above.) In speech recognition, the term lattice takes on a more restricted meaning in the phrase segmentation lattice. A segmentation lattice is a lattice (or more precisely a directed acyclic graph since the starting and end points need not be unique and the time segments need not be well-connected) of speech segments (e.g., phoneme/phonetic events). Each edge represents a speech segment and each node represents a point in time (corresponding to an endpoint of a segment).

lax: One of the features of sound set up by Jakobson and Halle in their distinctive feature theory of phonology, to handle variations in manner of articulation. Lax sounds are those produces with less muscular effort and movement, and which are relatively short and indistinct, compared to tense sounds. Examples of lax vowels are bit /i/ and bet /e/. Compare these with tense vowels such as beet /i/, bike /e/. Compare these with tense vowels such as beet /i/, bike /e'/.

level: (1) A general term in Linguists to refer to a major dimension of structural organization. The most widely recognized ‘levels of analysis’ are phonology, grammar and semantics, but these can be decomposed in many frameworks to distinguish phonetics from phonology, morphology and syntax from grammar, and so forth. (2) In generative linguistics, level is used to refer to the different types of representation encountered within the derivation of a sentence (e.g., deep and surface structure ‘levels of representation’). (3) The different structural layers within a linguistic hierarchy are often referred to as ‘levels’ (e.g., within a grammar one might talk of the levels of sentence, clause, phrase, word and morpheme.

linearity: A term used in linguistics to describe the characteristic representation of language as a unidimensional sequence of elements or rules. In phonology, linearity is an organizational principle whereby each occurrence of a phoneme is associated with a specific sequence of phones. The principle has been criticized by generative phonologists, as part of a general attack on taxonomic phonemics.
linear search: A search algorithm that looks at each element sequentially until it finds the goal. This algorithm requires time proportional to the size of the search space. That is, search time is a linear function of the size of the space.

light: A term of phonetics for distinguishing among various allophones of liquids, especially /l/. The /l/ is light, as opposed to the /l/ in all which is dark.

liquid: A term used by some phoneticians in the classification of speech sounds, referring collectively to allophones of /l/ and /r/.

manner (of articulation): One of the main parameters in the phonetic or phonological classification of speech sounds, referring to the kind of articulatory process used in a sound’s production. The distinction between consonant and vowel is usually made in terms of manner of articulation. Within consonants, several articulatory types are recognized, based on the type of closure, e.g., plosive (stop), affricate, nasal, lateral, semi-vowel, diphthong, pure vowel.

memoization: A computational term for a procedure which remembers (or memoizes) its behavior on previous inputs. In this way, the memoizing procedure can make use of previous experience, if it should receive the same inputs on a subsequent invocation, it can simply return the result it produced before and avoid potentially expensive recomputation.

methathesis: A term used in linguistics to refer to an alteration in the sequence of elements in a sentence — usually of sounds, but sometimes of syllables, words, or other units. ‘Metathese’ are well-recognized in historical language studies (e.g., Old English brid becoming bird), but they can also be seen in performance errors and phonological rules.

morpheme: The minimum distinctive unit of grammar. Its original motivation was as an alternative to the notion of the word, which had proved to be difficult to work with when comparing languages. There are several standard types of morphemes: bound vs. free (free morphemes are words in their own right; bound morphemes are not), affixes vs. roots, inflectional (e.g., -s, -ed) vs. derivational. There are two major classes of derivational morphemes: ‘+’ morphemes (e.g., -er, -ity) and ‘#’ morphemes (e.g., -ness, -ment). In general, ‘+’ morphemes are associated with stress shifts (e.g., stupid → stupidity), vowel shifts (e.g., divine → divinity); ‘#’ morphemes do not induce stress shifts (e.g., stupid → stupid-ness) or vowel shifts. There is a loose historical correlation between ‘+’ morphemes and Latin, and between ‘#’ morphemes and Greek and German.
Unfortunately, the morpheme is not well defined. Do words like cranberry and raspberry consist of one morpheme or two? Cran- and rasp- don’t seem to mean anything, and yet, there is a strong motivation that these words should be decomposed in order to capture the generalization that they are both berries.

**morphology:** The branch of grammar which studies the structure or forms of words, primarily through the use of the morpheme construct.

**murmur:** See nasal murmur.

**narrow band spectrogram:** See spectrogram.

**nasal:** A term used in the phonetic classification of speech sounds on the basis of manner of articulation: it refers to speech sounds produced while the soft palate is lowered to allow an audible escape of air through the nose. (The feature [+nasal] is in opposition to [+oral].) Both consonants and vowels may be articulated in this way. Nasal consonants (e.g., /n, m, ñ/ as in ran, ram, rang) occur when there is a complete closure in the mouth, and all the air thus escapes through the nose. English has no distinct nasal vowels, but nasalization is often heard in English vowels in certain contexts such as band.

**nasalization:** Nasal co-articulation with an oral vowel as in band. On a spectrogram, nasalization generally obscures the first formant of the vowel. This is probably an artifact of the way that spectrograms are computed. In fact, nasalization adds an extra pole and zero in the general vicinity of the first formant. Since spectrograms are generally computed with inadequate frequency resolution in this region, the combination of the first formant, the nasal formant, and the nasal zero are blurred together into a mess. This mess has become the cue for nasalization. There is hope that the display can be improved with the critical band spectrograms [67]. These techniques improve the frequency resolution in exchange for time resolution at low frequencies. At high frequencies, critical band analysis achieve better time resolution in exchange for frequency resolution. It is claimed that these trade-offs can be supported in terms of the psycho-acoustics of the ear.

**nasal murmur:** A term from acoustic-phonetics for the closure portion of a released nasal. On a spectrogram, the nasal murmur has a distinctive plateau shape (like an extremely strong voice bar).

---

193. We have been informed that cranberries derive their name from Crane’s Beach, a famous beach near Cape Anne.
neutralization: A term used in phonology and phonetics to describe what happens when the distinction between two phonemes is lost in a particular environment. For example, in English, the contrast between aspirated (voiceless) and unaspirated (voiced) plosives is normally crucial, e.g., tip vs. dip, but this contrast is lost, or 'neutralized', when the plosive is preceded by /s/, as in stop, skin, speech, and as a result, there are no pairs of words in the language of the type skin vs. *sgin. From a phonetic point of view, the explanation lies in the phonetic change which happens to /k/ in the position: the /k/ lacks aspiration, and comes to be physically indistinguishable from /g/. Flapping is another example of a neutralization process. The normally distinctive different between /v/ and /d/ is (essentially) lost in words pairs like writer/ rider, for those speakers of American English who flap these medial stops.

node: In linguistics, node is equivalent to constituent. In computation, a node is part of a graph.

obstruent: A term in the phonetic classification of speech sounds to refer to sounds involving a constriction which impedes the flow of air through the nose or mouth, as in plosives, fricatives, affricates and nasals.

In distinctive feature theory, the term obstruent is used in opposition to sonorant.

on/off-glide: Terms used in phonetics to refer to the auditory effect of articulatory movement at points of transition between sounds. An off-glide is a movement which occurs at the end of diphthongs; an on-glide is a movement into a vowel as in the word news.

onset: A term used in phonetics and phonology to refer to the opening segment of a linguistic unit (e.g., a syllable, a tone unit) or to the articulatory movement which initiates a speech sound (e.g., the closure phase of a stop).

open class item: The open class has a potentially infinite membership. Open class words (e.g., nouns, verbs, adjectives) are in opposition to closed class (function) words (e.g., in, on, and) whose membership is small and finite. The distinction is not quite as clear-cut as it seems.

open syllable: A term used in the two-way classification of syllable structure, referring to a syllable which ends in a vowel, as opposed to the closed syllable, which ends in a consonant.

oral: not nasal

palate: The arched bony structure which forms the roof of the mouth, and which is much used used for the articulation of speech sounds. The delimitation and classification of the palatal area is controversial. In phonetics, the palate is restricted to the hard palate (immediately behind the alveolar ridge); the soft palate or 'velum' (the mobile part of the palate farther back toward the uvula) is reserved for velar or
uvular sounds.

**palatal**: A term used in the phonetic classification of speech sounds on the basis of their place of articulation: it refers to a sound made when the front of the tongue forms a constriction with the hard palate.

**palatalization**: A phonological/phonetic process which moves the place of articulation toward the hard palate. For example, there is a phonological rule of palatalization that accounts for the alternation of *t* before *i* in *president* → *presidential*. There is also a phonetic rule of palatalization that accounts for the alternation of *t* before *y* in *met you* → *mɛtʃoʊ*. Palatal sounds are often transcribed with a hache (e.g., /ɛ/, /ʃ/, /ʃ/, /tʃ/) or with a superscript ‘y’ as in [kʰ]y.

**parsing**: This term refers to the task of labeling the grammatical elements of sentences, e.g., subject, predicate, past tense, noun, verb. The difficulty of this task depends (in a certain sense) on the generative capacity of the grammar. In the case of context-free grammars, this problem is known to be reducible (in a formal computational sense) to the problem of matrix multiplication. Some grammars are easier (e.g., finite-state grammars), and other grammars are harder (e.g., context-sensitive grammars). There are grammars which cannot be parsed (in a formal computational sense), e.g., arbitrary rewrite-rule grammars.

**path**: A sequence of arcs in a graph.

**peak**: The vocalic portion (constituent) of a syllable. In most syllables, the peak contains just a single vowel, but in some cases it might also include a liquid or glide. For example, in words like *interest* and *difference*, we might include the /r/ in the peak.

**phone**: A term used in phonetics to refer to the smallest perceptible discrete segment (in time) of sound in a stream of speech. From the viewpoint of segmental phonology, phones are the physical realizations of phonemes; phonetic variants of a phoneme are referred to as allophones.

**phoneme**: The minimal unit in the sound system of languages according to traditional phonological theories. Phonemes were originally introduced in order to reduce the range of possible meaningful contrasts. The phonetic specifications of phones, it was realized, contain far more detail than language makes use of. The phoneme allowed linguists to group similar phones together in the same underlying unit.

**phonetics**: The description, classification and transcription of speech. Three branches are traditionally recognized: articulatory phonetics, acoustic phonetics and auditory phonetics. These sub-areas focus respectively on the generation of speech sounds from the vocal tract, the transmission of speech sounds from the vocal tract to the ear, and the reception of speech sounds at the ear. As a matter of convention,
phonetic transcriptions are written inside square brackets (e.g., [hl[/lt]]) whereas phonemic transcriptions are delimited by slashes (e.g., /hl/lt/).

phonology: A branch of linguistics which studies the distinctive sound systems of languages. Of the wide range of phonetic possibilities, only a relatively small number are used distinctively in any one language. Phonology is divided into many sub-areas. In particular, there are segmental and suprasegmental phonology. Segmental phonology analyzes speech into discrete segments such as phonemes. Suprasegmental phonology looks at properties that extend over multiple segments such as prosody and intonation. There are also the synchronic and diachronic sub-areas: the former studies patterns of sound change across history whereas the latter looks at alternations within a single period of history.

phonotactics: The study of the distribution ("tactic behavior") of phonemes in language. In English, sequences like /nd/ are not found at the beginning of a word.

phrase-structure (PS) grammar: The non-transformational base of a transformational grammar. Phrase-structure grammars include context-free grammars, context-sensitive grammars and finite-state grammars.

place of articulation: The class of phonological/phonetic features referring to the source of a sound in the vocal tract. Examples include labial, labio-dental, dental, alveolar, palatal, velar, uvular, pharyngeal, glottal, and more. In this thesis, we have been using a four valued system: labial /p, b, f, v, m/, velar /k, g, η/, dental /s, z, t, d, l, n/, and palatal /ʃ, ʒ, ʃ, j/.

plosive: see stop

postlexical rules: A term from Lexical Phonology [86] for rules which apply after the lexicon (e.g., flapping, retroflex). These rules correspond closely with the phonetic classification of allophonic rules.

precedence: A formal computation device for expressing ordering dependencies among operations. The computational phrases "x takes precedence over y" and "y yields precedence to x" can be translated roughly into the linguistic phrases "x applies before y" or "y applies after x". We prefer the computation terminology because we do not wish to imply that the processing model has to work in the specified order. All that is required is that it produce the same results that would have been produced had it proceeded in the specified order.
prosody: A term from suprasegmental phonetics used to refer collectively to variations in pitch, loudness, tempo, rhythm, stress and intonation.

quality: A term used in phonetics and phonology to refer to the characteristic resonances, abstractions of formants. The term is generally opposed to quantity (or length).

quantity: A term used in phonetics and phonology to refer to the duration (or length) of a constituent in abstract units. In metrical phonology, quantity is assumed to be related to the branching-ness of a node. Thus, a branching coda as in verb *wurpp* is considered to be heavier than the non-branching coda in *common*. This difference is manifested in the assignment of stress.

recognition: (1) In contrast to parsing, recognition need not assign a phrase-structure to grammatical sentences; it merely distinguishes grammatical sentences from ungrammatical ones. (2) In contrast to speech understanding, recognition does not (necessarily) involve the use of higher level constraints (e.g., syntax, semantics, pragmatics). A speech understanding device reacts appropriately to auditory input; a recognition device merely decodes the auditory input into some more convenient format (e.g., a text file).

reduced vowel: A term used in phonology and phonetics referring to a vowel which is very short and central (articulators in their resting state) as the result of being unstressed. Examples: *of* the *telegram*

release: A term used in phonetics to refer to the type of movement made by the vocal organs away from a point of articulation, particularly with the reference to plosives (stops). Release is often distinguished from aspiration. Voiced stops, in English, are often released but rarely aspirated. Release is generally associated with stops, but can also be applied to other manners of articulation such as nasals.

retroflex: A term used in the phonetic classification of consonant sounds on the basis of the place of articulation: it refers to /r/-like sounds, sounds made when the tip of the tongue is curled back in the direction of the front part of the hard palate — in other words, just behind the alveolar ridge. The degree of retroflexion varies considerably between sounds, contexts and dialects. Retroflexed vowels are sometimes called 'r-colored'.

rhyme: A term from phonology and phonetics used to describe the end of a syllable (i.e., the vowel and post-vocalic consonants), which are assumed to form a constituent. The name is derived from poetry, where a rhyme is a relation among the ends of two syllables.
rounding: A term used in the classification of lip position in phonetics, referring to the visual appearance of the lips when uttering a rounded sound such as the vowel in *round*. Rounded vowels are often transcribed with a 'W' off-glide such as /oʊ/.  

run-time: A computational term for the time when a program is being interpreted. It is often contrasted with compile-time.  

schwa, shwa: The name for a neutral reduced vowel (transcribed as /ə/). On a spectrogram, schwas are generally relatively easy to label correctly because they are extremely short. Schwas are typically 10-30 ms long, whereas most other vowels are typically 100 or 200 ms in duration. In some treatments, a distinction is made between a back/neutral schwa /ə/ and a raised or front schwa /i/.  

search: A term from computer science referring to the problem of finding a target (or goal) in a (large) space of elements.  

segment: A term used in phonetics, speech research and linguistics to refer to any discrete unit (interval of time) that can be identified in a stream of speech. In phonetics, a segment is sometimes considered to be synonymous with 'phone'.  

semi-vowel: A term used in the classification of consonant sounds on the basis of their manner of articulation: it refers to a sound functioning as a consonant but lacking the phonetic characteristics normally associated with consonants (such as friction or closure). Examples: /w/ (wet) and /y/ (yet).  

side-effect: A computational term for a non-standard method of communication. Examples of side-effecting operations are: assignment, array store, rplaca/rplacd, setf, transformation. Side-effects are generally considered poor programming style. Among certain extreme purists, there has been a movement to ban side-effects from programming languages altogether, and replace them with the only acceptable form of communication (i.e., lambda binding). Side-effecting operations are sometimes said to clobber the environment.  

silence: A time interval of relatively little acoustic energy across all frequencies. Silence is generally associated with the closure portion of stop (the time between the constriction and the release).  

sonorant: A distinctive feature of phonology. It refers to the manner of articulation for sounds produced with relatively free air flow: vowels, liquids, nasals and laterals. It is opposed to obstruent, which refers to sounds where there is a stricture impeding air flow as in plosives, fricatives and affricates.
sonority (hierarchy): A controversial ordering relation used to define the syllable structure of utterances. Syllables are assumed to be more sonorous at the peak and degrade toward less sonorant sounds at the margin.

sparse matrix: A term in computer science referring to a matrix with only a few non-zero elements. There are relatively algorithms for performing matrix operations (e.g., addition, multiplication, transitive closure) efficiently for sparse matrices.

spectrogram: A plot of acoustic energy as a function of time (horizontal axis) and frequency (vertical axis). There is a common distinction between wide band spectrograms and narrow band spectrograms, depending on the size of the window used in the filtering. Narrow band spectrograms have better resolution in the frequency dimension; wide band spectrograms have better resolution in the time dimension. Critical band analysis is a promising compromise between the two.

speech recognition: Decoding a speech utterance into a text file. See recognition.

speech synthesis: Transforming a text file into a speech utterance.

speech understanding: Reacting appropriately to speech input.

stop: A term used in the phonetic classification of speech sounds on the basis of their manner of articulation: it refers to any sound which is produced by a complete closure in the vocal tract, e.g., plosives /p, t, k, b, d, g, ŋ, j/ and nasals. (We generally use the terms stop and plosives as synonymous.)

stop burst: The burst of energy often found at the end of a stop when the stop is released.

stop gap: The silence period before the release of a stop.

stress: A term used in phonetic transcription to refer to the degree of force used in producing a syllable. Stress is often correlated with observable parameters such as length, energy, and so forth, but the relationship is not clearly understood. Stress is represented differently in various systems.

(287) "The multivalued notation utilized in SPE (a representation reminiscent of some of the structuralist approaches, e.g., Trager & Smith 1951, A. Hill 1958), recognizes at least four degrees of stress. (For words in isolation there are only three degrees, since [2 stress] is reserved for the phrasal level.) The [1 stress] of SPE is equivalent to Kenyon & Knott's primary, whereas the latter's secondary corresponds to the SPE [3 stress] and [4 stress], e.g., artificíality vs. artificiality. Where the two systems differ most radically is in the SPE treatment of Kenyon & Knott's (and my) NON-REDUCED vowels in UNACCENTED syllables. In SPE, the
majority of these vowels are considered stressed: thus stressed syllables may be contiguous ... [For example, *Daytona* is 310 in SPE and WSW in Shane's notation. In contrast, *Dakota* is 010 in SPE and WSW in Shane's notation. Share accounts for SPE's 3/1 contrast with a distinction between reduced and non-reduced Ws. The first syllable in *Daytona* is a non-reduced W; the first syllable in *Dakota* is a reduced W.] Because of this difference (and even possible ambiguity) in regard to what may constitute a stressed syllable, I prefer, following Kenyon & Knott, to use the terms 'accented' and 'unaccented'." [108, p. 563]

We have used at least four different notational schemes in this thesis: (1) Shane's S and W system, (2) SPE's numbering system, (3) Kenyon & Knott's accents, and (4) theory-neutral accents (e.g., *récord*).

**strong fricative:** An acoustic phonetic term for a fricative that is high in acoustic energy (e.g., /s, z, ñ, þ, ð, þ/). Opposed to weak fricatives (e.g., /f, v, θ, ð/).

**suprasegmental:** A term used in phonetics and phonology to refer to an effect over more than one sound segment (e.g., pitch, stress, juncture, intonation). We use the term in a slightly non-standard way to refer to any process that extends over multiple segments (e.g., processes that relate to clusters, onsets, peaks, codas, rhymes, syllables, feet, etc.).

**syllable:** A unit of pronunciation typically larger than a single sound and smaller than a word. It is not very easy to provide a precise definition of the syllable. From a phonetic point of view, one can attempt to define the syllable in terms of parameters such as energy or chest pulses. From a phonotactic point of view, one can attempt a definition in terms of co-occurrence relationships among phonemes. We have attempted to define the syllable as the unit which makes for the simplest formulation of allophonic and phonotactic constraints.

**syllable nucleus:** Equivalent to syllable peak.

**template:** Templates are common in holistic recognition systems such as dynamic time warping [44]. A template is a speech utterance with some (but not much) data compression.

**tense vowel:** A vowel that is not lax and not reduced. Tense vowels are (generally) diphthongs (end in an off-glide, e.g., /y, w/).

**top (of lattice):** See lattice.
transcription: A method of writing down speech sounds in a systematic and (hopefully) consistent way. Two main kinds of transcription are recognized: phonetic and phonemic. Phonetic transcriptions are written inside square brackets; phonemic transcriptions appear between slashes. Phonemic transcriptions are generally simpler and less detailed. Phonetic transcriptions would include a number of diacritics to describe aspiration, nasalization, glottalization, and other properties which are not distinctive in English. Thus, *pen* would be transcribed phonemically as /pen/ and phonetically as [pʰɛn]. Phonetic transcriptions vary on a continuum from narrow (very detailed) to broad (almost phonemic).

transformation: A formal linguistic operation which enables two levels of structural representation to be placed in correspondence. The input to a transformational rule is called the structural description (SD) and produces as output a structural change (SC). Transformational grammar is strongly associated with the work of Noam Chomsky at MIT.

transitive closure: A term from computer science for an operation performed on graphs. The transitive closure produces an graph whose edges correspond to paths (sequences of edges) in the input lattice. There are well-known transitive closure algorithms that have the same time complexity as matrix multiplication.

Turing: A term from philosophy and computer science for an automaton (abstract machine) that can compute any effective procedure. In formal treatments, a Turing Machine is defined in architectural terms (e.g., a Turing machine is a machine with an infinitely long tape and a single head that reads the current value on the tape of 1's and 0's and performs one of a limited set of operations depending on the value on the tape). The reader is referred to [43: chapter 7] for a precise definition. Suffice it to say that a Turing Machine is extremely powerful; it is strongly equivalent to an arbitrary rewrite grammar, the most powerful class of grammars in the Chomsky Hierarchy (i.e., regular grammars → context-free grammars → context-sensitive grammars → arbitrary rewrite grammars). However, there are limits to the power of a Turing Machine. For example, it cannot be decided on any Turing Machine whether or not a particular procedure will halt. Such procedures are said to be undecidable.

unreleased stop: An allophone of a stop, generally found in syllable final position. It is characterized by its lack of a release.
**uvelar:** A term used in phonetics for consonant sounds produced against the uvula (rare in English).

**variant cue:** A cue such as aspiration and nasalization that depends on context. Variant cues are generally considered to be phonological, because they are not distinctive (almost by definition). (The concept of variant is fundamental to the notion of allo-.)

**velar:** A term used in the phonetics classification of consonant sounds on the basis of their place of articulation: it refers to a sound made by the back of the tongue against the soft palate (or velum). Examples in English are /k/, /g/, and /ŋ/ (the -ng- sound in sing).

**velar pinch:** A term from acoustic phonetics for the spectral transitions that are often associated with velar sounds. Velar sounds tend to pull F₂ and F₃ together into a distinctive “pinching” shape.

**virtual memory:** A term in computer science for a particular type of computer memory. In many computer systems, it is possible for the machine to reference many more memory locations than there really are. Rather than restricting users to work within the limits of physical memory (which tend to be rather unpredictable depending on machine load), many computer systems implement a paging scheme whereby the system can swap pages of user memory between core and disk. When a user references a virtual memory location, the system consults the page map to find out where that page of virtual memory resides. If the memory is already in core, the virtual reference is simply translated into a fetch into the appropriate location in core. If not, the system signals a page fault, which causes the page to be swapped into core from disk. In many systems, a page fault is 2000 times slower than a physical memory fetch.

**voice:** A fundamental term used in the phonetic classification of speech sounds, referring to the auditory result of the vibration of the vocal cords. Examples of voiced sounds are: /b, d, g, z, v, ʃ, ʒ/ as opposed to /p, t, k, s, f, θ, ç/. A sound which is normally voiced (such as /l/), but which can be unvoiced in a particular environment such as /pl/ in please is called **devoiced.**

**voice bar:** A term used in acoustic-phonetics to refer to a bar of low frequency energy that is often found during the closure portion of a voiced stop.

**voice onset time (VOT):** A term used in acoustic-phonetics referring to the duration between the release of a stop and the start of voicing. In general, voiced stops /b, d, g/ have shorter VOTs than unvoiced stops /p, t, k/. In addition, VOT is generally shorter in foot-internal position than in foot-initial position. VOT is also related to place of articulation, though the effects are (generally) smaller in magnitude.
**weak fricative**: An acoustic phonetic term for a fricative that is weak in acoustic energy (e.g., /v, f, θ, θ/). The term is contrasted with strong fricative.

**well-formed-substring-table**: See chart.

**wide band spectrogram**: See spectrogram.