MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Cambridge, Massachusetts

PROJECT MAC

ARTIFICIAL INTELLIGENCE PROJECT
MEMO 91

MEMORANDUM MAC-M-285
November, 1965

A Useful Algebraic Property of Robinson's Unification Algorithm

Timothy P. Hart

This memo presupposes some acquaintance with "A Machine Oriented Logic Based on the Resolution Principle", J.A. Robinson, JACM Jan.'65. The reader unfamiliar with this paper should be able to get a general idea of the theorem if he knows that σ_A is a post operator indicating a minimal set of substitutions (most general substitution) necessary to transform all elements of the set of formulae, A, into the same element (to "unify" A), so that when σ_A exists $A\sigma_A$ is a set with one element (a "unit").

Example:

$$A = \{f(x),\ y,\ f(g(u)),\ f(g(z))\}$$
$$\sigma_A = \{g(u)/x,\ f(g(u))/y,\ u/z\}$$
$$A\sigma_A = \{f(g(u))\}$$

Another most general unifier of A is $\{\ g(z)/x,\ f(g(z))/y,\ z/u\ \}$.

Theorem:

$\sigma_{B\sigma_A}$ exists iff $\sigma_{A\sigma_B}$ exists. Both $\sigma_A \sigma_{B\sigma_A}$ and $\sigma_B \sigma_{A\sigma_B}$ unify the set A and at the same time unify the set B.

Proof:

ρ is a most general simultaneous unifier of A and B if $A\rho$ is a unit, $B\rho$ is a unit, and if $A\theta$ and $B\theta$ are units then there is a substitution μ such that $\theta = \rho\mu$.

By the Unification Theorem if θ simultaneously unifies A and B it can be written as $\sigma_A\lambda$ for some λ because it unifies A. Since θ also unifies B, $B\theta = B\sigma_A\lambda$ will be a unit, so that λ must unify $B\sigma_A$. $\sigma_{B\sigma_A}$ is a most general unifier of $B\sigma_A$ so there is some μ such that any unifier of $B\sigma_A$ may be written $\sigma_{B\sigma_A}\mu$. Therefore θ may be written as $\sigma_A\sigma_{B\sigma_A}\mu$ and $\sigma_A\sigma_{B\sigma_A}$ is a most general simultaneous unifier of A and B. It follows then that there exists a simultaneous unifier of A and B iff $\sigma_A\sigma_{B\sigma_A}$ exists iff σ_A and $\sigma_{B\sigma_A}$ exist. By symmetrical reasoning $\sigma_B\sigma_{A\sigma_B}$ is also a most general simultaneous unifier of A and B, and since $\sigma_{B\gamma}$ exists implies σ_B exists, $\sigma_{A\sigma_B}$ exists iff $\sigma_{B\sigma_A}$ exists.

Example:

$$A = \{x,\ f(a,y),\ f(u,y)\}$$

$$B = \{y,\ g(u)\}$$

$$\sigma_A = \{f(a,y)/x,\ a/u\}$$

$$\sigma_B = \{g(u)/y\}$$

$$B\sigma_B = \{x,\ f(a,g(u)),\ f(u,g(u))\}$$

$$B\sigma_A = \{y,\ g(a)\}$$

$$\sigma_A\sigma_B = \{f(a,g(a))/x,\ a/u\}$$

$$\sigma_B\sigma_A = \{g(a)/y\}$$

$$\sigma_A\sigma_B\sigma_A = \sigma_B\sigma_A\sigma_B = \{g(a)/y,\ f(a,g(a))/x,\ a/u\}$$

Example:

$$A = \{f(x,y),\ f(a,b)\}$$

$$B = \{x,\ y\}$$

$$\sigma_A = \{a/x,\ b/y\}$$

$$\sigma_B = \{y/x\}$$

$$A\sigma_B = \{f(y,y),\ f(a,b)\}$$

$$B\sigma_A = \{a,\ b\}$$

Neither $\sigma_A\sigma_B$ nor $\sigma_B\sigma_A$ exists.

Application

This theorem shows that certain proofs in Robinson's resolution system are equivalent. Thus improvements can be made to search procedures which test equivalent proofs.

Let $A = \{A_1, A_2\}$ and $B = \{B_1, B_2\}$ such that σ_A and σ_B exist, and let $\{A_1\}$, $\{\sim A_2 \; B_1\}$, and $\{\sim B_2\}$ be clauses (sets of literals) with no variables common between the sets so that $\{A_1\} \sigma_B = \{A_1\}$ and $\{B_2\} \sigma_A = \{B_2\}$. (This is the result of a preliminary step in resolution.)

Then the proof (where $\vee$ indicates resolution)

$$\{A_1\} \quad \{\sim A_2, B_1\}$$

$$\{B_1\} \sigma_A \qquad \{\sim B_2\} \sigma_A$$

exists, i.e. $\sigma_B \sigma_A$ exists, if and only if the proof

$$\{\sim A_2, B_1\} \quad \{\sim B_2\}$$

$$\{A_1\} \sigma_B \qquad \{\sim A_2\} \sigma_B$$

exists, since this proof exists only if $\sigma_A \sigma_B$ exists.

By an extension of this reasoning it can be seen that any such rearrangement of a proof tree corresponding to a rearrangement of the order of unification leads to an equivalent proof. In particular there are $\frac{(2m - 2)!}{m!\,(m - 1)!}$ (asymptotically 4^m)

ways of developing a proof without factoring involving m-2 2-literal, terminal clauses, and 2 1-literal, terminal clauses. e.g. for m=4 there are 5 equivalent structures: