AUTOMATED HTPOTHESIS GENERATION USING EXTENDED INDUCTIVE RESOLUTION

Charles 0. Morgan
Department of Philosophy
University of Alberta
Edmonton, Alberta
Canada

Abstract

We report on a method of automated hypothesis generation, called f-resolution, which is derived from deductive resolution techniques. The method is inductive in character, in the sense that given input statement E, it generates hypotheses H, such that E is a deductive consequence of E. The method is extended by a generalized unification algorithm which introduces appropriate identity assumptions needed to unify a pair of literals. The f-resolution technique is shown to embody a version of Ockham's raror as a pruning heuristic. Some promising experimental results are also presented.

In [5] we discussed a general method for transforrairg deductive consequence generators into inductive consequence generators. In this paper we wish to report on a general mechanised inductive method, to be called f-resolution, which was derived from normal resolution procedures. The f-reeolution procedure was not designed as a special purpose routine only applicable to certain problems (like the routine described in [2]), but rather it is applicable to any problem in any context suitably descrlbable in the syntax of first-order predicate calculus with identity. The potential range of applicability of the method is thus extremely broad. Our routine appears to deal with a more general class of hypotheses than previously reported efforts in this area; in particular, see[1], [9], [10], [11], and [12]

We assume familiarity with the usual terminology and theory of resolution. For any expression E, we represent the set of clauses obtained from the clause form of E by C(E). For a set C of clauses, we represent the set of all pairwise resolvents of members of C by R(C). We then define $R0(C) = C$ and $R^{n+1}(C) = R(R^n(C)) \cup R^n(C)$.

For our purposes, we will consider resolution as a consequence generator. As such, resolution ie not complete. That is, for any expression E, there are expressions E' such that E entails E' but for nc n is it the case that $C(E') \subset Rn(c(E))$. For just one type of example, consider the case in which E' contains predicates which do not occur in E, and note that resolution introduces *no* new predicates.

The basic f-resolution principle is very similar to the basic resolution principle. The principle appears to be, in essential respects, the same as the basic inverse method of [3], apparently developed by Maslov as early as 1964. However, our development was independent of Maslov's. work, and our routine is used inductively rather than deductively. Ac with the deductive case, before the principle can be applied to a given expression, the expression muet first be transformed into a certain normal form. For f-resolution, input expressions are first transformed into prenex disjunctive normal form, using the standard equivalences:

(Q1) ... (Cii)(E1 V ... V Qa)

where the Qi are quantifiers, each over a different variable, and each Ei is a conjunction of literals. We must now remove the quantifiers to obtain the quantifier-free disjunctive normal form. To avoid introduction of more complex Skolera functions than necessary, we next employ the standard quantifier distribution rules to obtain an equivalent expression such that the scope of each quantifier contains as few cf the Ei as possible. We then remove all universal quantifiers by replacing them with appropriate Skolem functions. (That is, if a universal quantifier does not occur within the scope of any existential quantifier, we replace the variable of the universal quantifier, wherever it occure in the Ei, by a new constant (O-place function). If a universal quantifier occurs within the scope of existential nuantifiers over variables x1, . . . , xp, then we replace the variable of the universal quantifier wherever it occurs in the Ei by *f(x1,* ..., xp), where f is some new p-place function.) We then drop all quantifiers. The resulting expression is called the f-clauoe form; a conjunction of literals is called an f-clause. If E is any exprepsion, then we represent the set of f-clauses obtained from E by fC(E). Note that in the f-clause form, free variablee represent existential quantification. Note alco that if the input is to be several expressions, At, ..., Aq, then we must obtain the f-clauses from their conjunction, A1 & ... & Aq.

We assume familiarity with the standard terminology regarding substitutions and with the unification algorithm. As usual, if C is a set of expressions and X is a set of substitutions, by CX we mean the set of expressions obtained by performing every substitution in X on every member of C.

The basic f-resolution principle allows the inference of one f-clause from a given pair of f-clauses. Cur first step is to rename the variables in one of the f-clauses so that the pair has no variables in common. Let E1 and E2 be the two resulting f-clauses. Let *C1* and *C2* be the sets of literals occurring in E1 and E2, respectively. Further, suppose there are subsets C1' and C2' of C1 and C2:

$$C1' = \{E1_1, \ldots, E1_n\}$$

$$C2' = \{\neg E2_1, \ldots, \neg E2_m\}$$

such that $C1' \cup \{E2_1, \ldots, E2_m\}$ is unifiable by a most general unifier λ. Then an f-resolvent of £1 and E2 is the f-clause which results from conjoining the literals in the set:

(*) $(C1 \sim C1')\lambda \cup (C2 \sim C2')\lambda$

where of course " ~ " is set theoretic subtraction. Note that we must include the *empty clause,* designated by EMP, as a possible f-resolvent. For our purposes, EMP is taken to be <u>true</u> in erery interpretation,

For any expression E, there are only a finite number of f-resolventa of pairs in fC(E). We designate the set of f-resolvents of pairs of f-clauses in set C by fR(C). We then define fHP(C) = C and $fR^{n+1}(C) = fR(fR^{n}(C)) \cup fR^{n}(C)$. A detailed proof of the following useful theorem may be found in [6].

<u>Theorem</u> 1: Let E be any closed first-order expression. Then E is valid if and only if for some n, EMP € fRn(fC(E)).

For any two expressions, E1 and E2, if E2 is false in every interpretation which falsifies E1, then we write E1 |-F E2; if E2 is true in every interpretation which satisfiea E1, we write E1 *I- E2.* Clearly we have E1 II-F E2 if and only if E2 II- E1. So if an expression E2 is generated from expression E1 by falsehood preserving rules, we know that E2 entails E1 in the usual sense. The important thing about f-resolution is that it-is falsehood preserving. A detailed proof of the following theorem may be found in [8].

<u>Theorom 2</u>: Let E1 and E2 be any two closed first-order expressions. If for some n, $fC(E2) \subseteq fR^{n}(fC(E1))$, then E2 $\Vdash$ E1 (i.e., E1 $\Vdash_F$ E2).

Thus we can use f-resolution to generate hypotheses from which our input may be deduced (see [7] for a general discussion of objections to this approach). However, ae an hypothesis generator, f-recolution is incomplete in the sense that for any expression E1, there are expressions E2 such that E1 |I-F E2 but for no n is it the case that fC(E2) c fRn(fC(E1)). Again simply note that f-resolution introduces no new predicates. So f-resolution automatically incorporates some form of pruning.

It is useful to characterize the pruning heuristic in more intuitive terms. We may think of a literal as a statement of a basic (or atomic) fact about the subject matter under consideration. Note firtt that when we substitute a term, which occurs in one literal, for a free variable in some other literal, we are in an intuitive sense reducing the number of unnamed entities which might possibly be required to satisfy the literals; <u>intuitively</u>, the m*ore* existential quantifiers in the prenex form of an expression, the more entities which might <u>possibly</u> be required to satisfy the expression. But note secondly that the only substitutions that are made are those which allow the elimination of a pair of literals. So the intuitive heuristic seems to be:

(H.1) Reduce the nunher of unnamed entities where ouch a reduction will allow a reduction in the number of basic facts.

Thus we can see that basic f-resolution incorporates a version of Ockham's razor.

Let us consider an extremely simple example: Either Smokey is a black animal or there is an animal that is not black. In symbols we have:

(a.1) (As & Bs) v (∃y)(Ay & ¬ By)

We obtain the following f-clauses:

(a.2) As & Bs

(a.3) Ay & ¬ By

By basic f-resolution we obtain:

(a.4) As

And (a.4) simply says: Smokey is an animal. By Theorem 2, we know that (a.4) entails (a.1). Note that we have reduced the number of unnamed entities as well as the number of basic facts in arriving at the hypothesis.

The heuristic (H.1) is actually a bit broader than the performance of f-resolution. Suppose we receive the following report from a detective at the scene of the crime: Someone killed a person named John and someone killed a person named Bill. We might represent our information symbolically by:

(b.1) Pj & (∃x)(Px & Kxj) & Pb & (∃y)(Py & Kyb)

From (b.1) we obtain only one f-clause:

(b.2) Pj & Px & Kxj & Pb & Py & Kyb

Since we have only one f-clause, basic f-resolution is not applicable. But heuristic (H.1) would direct us to the following hypotheses, among others:

(b.3) Pj & Kjj & Pb & Py & Kyb

(John committed suicide and someone killed Bill.)

(b.4) Pj & Kjj & Pb & Kbb

(John and Bill both committed suicide.)

(b.5) Pj & Pb & Kbj & Kjb

(John and Bill killed each other.)

(b.6) Pj & Kbj & Pb & Kbb

(Bill killed John and committed suicide.)

(b.7) Pj & Px & Kxj & Pb & Kxb

(The same person killed both John and Bill.)

Such examples and the statement of the heuristic suggest adding another rule of inference; the rule will be called linear contraction, or LC for short. The rule allows us to infer one f-clause from one other f-clause. Let C bo the set of literals in some given f-clause, and suppose there is a subset of C, say $C' = [E1, \ldots, En]$, such that there is a most general unifier λ of C'; that is, $E1\lambda = \ldots = En\lambda$. Then LC allows us to infer the f-clause formed by conjoining the members of $C\lambda$.

The rule LC would allow the generation of (b.3 - 7) from (b.,2), as well as some additional hypotheses. But note that the example indicates the sensitivity of LC to the amount of detail in the representation. If the predicate for personhood were eliminated from (b.2) we would have:

(b.8) Kxj & Kyb

Since instantiating x for y (or y for x) would not allow us to reduce the number of basic facts, even LC would not allow any inference from (b.8). The fact that all the individuals in (b.1) are <u>persons</u> means that instantiation reduces the number of persons. Consequently, in any proposed application, we must be careful to include all information concerning the individuals named or quantified.

It is very simple to establish that LC is falsehood preserving. Since free variables represent existential quantification, substitution of a term, t, for a *free* variable simply involves one of three possibilities: (i) replacing a pair of existential quantifiers by a single existential quantifier, in case t is a variable} or (ii) replacing an existential quantifier by a universal quantifier, in case t is composed of Skolem functione; or (ill) replacing an existential quantifier by some interpreted constant, in case t is an interpreted constant. But then we just note that: (i) $(\exists x)(E1(x) \& E2(x))$ entails $(\exists x)E1(x) \& (\exists y)E2(y)$; (ii) $(x)E(x)$ entails $(\exists x)E(x)$; and (iii) $E(a)$ entails $(\exists x)E(x)$. These considerations, combined with the fact that E & E is equivalent to E, establish the desired result.

Both basic f-resolution and LC strive to reduce the number of atomic facte. However, since neither deals explicitly with equality, both share the implicit assumption that constants represent distinct entities. If we could sometimes assume that certain terms designate the same object, wo might be able to further reduce the number of atomic facts. For example, in (b.8) if we could assume J = b, then we could arrive at Kxj via LC. These considerations led us to develop a slightly different version of the unification algorithm, called equality assumption introduction (EAI), which introduces just those equality assumptions which will allow a reduction in the number of basic facts. We will give a brief statement of a version of the EAI algorithm suited to the unification of pairs of literals, E1 and K2. We assume E1 and E2 are either both negated or both unnegated and that both employ the same predicate. In the following algorithm, k is just a counter, λ will correspond to *e* most general unifier, and eq will be a set of required equality assumptions. By (t/t'), we mean the substitution of t for t'; we use $\lambda \circ \lambda'$ to designate the usual composition of substitution sets; we use $\emptyset$ for the empty set.

1. Set $k = 0$, $E1_0 = E1$, $E2_0 = E2$, $\lambda_0 = \emptyset$, and $eq_0 = \emptyset$.
2. If $E1_k = E2_k$, then set $\lambda = \lambda_k$ and $eq = eq_k$ and stop.
3. Find the (ordered) disagreement set $(t1_k, t2_k)$ of $E1_k$ and $E2_k$.
4. If $t1_k$ is a variable not in $t2_k$, then go to step 8.
5. If $t2_k$ is a variable not in $t1_k$, then go to step 10.
6. If $t1_k$ is shorter than, or of the same length as, $t2_k$, then set $\lambda_{k+1} = \lambda_k$ and $eq_{k+1} = eq_k(t1_k/t2_k) \cup \{t1_k = t2_k\}$ and go to step 11.
7. Set $\lambda_{k+1} = \lambda_k$ and $eq_{k+1} = eq_k(t2_k/t1_k) \cup \{t2_k = t1_k\}$ and go to step 9.
8. Set $\lambda_{k+1} = \lambda_k \circ \{(t2_k/t1_k)\}$ and $eq_{k+1} = eq_k(t2_k/t1_k)$.
9. Set $E2_{k+1} = E2_k(t2_k/t1_k)$ and $E1_{k+1} = E1_k(t2_k/t1_k)$ and go to step 12.
10. Set $\lambda_{k+1} = \lambda_k \circ \{(t1_k/t2_k)\}$ and $eq_{k+1} = eq_k(t1_k/t2_k)$.
11. Set $E1_{k+1} = E1_k(t1_k/t2_k)$ and $E2_{k+1} = E2_k(t1_k/t2_k)$.
12. Set $k = k + 1$ and go to step 2.

This modified unification algorithm ie Just an extension of the usual one. If two clauses are unifiable, then our algorithm returns s most general unifier (in the usual sense) and an empty eq. Our algorithm differs from the normal unification plgorithm at those points where the normal algorithm terminates unsuccessfully. The normal algorithm hilts with no unifier if (i) the disagreement set contains no variable, or if (ii) one term is contained in the other. In such cases, EAI introduces (via the set eq) the explicit assumption that the two terms are eoual and rubatitutes the shortes for the longer. The reason for substituting the shorter for the longer is to guarantee that the substitution process will terminate.

If we employ EAI, then basic f-resclution and LC must be slightly altered so that the resultant f-clauses contain the contents of eq. The f-resolvent is obtained by conjoining the literals in the following set (compare with (*), above):

(**) $(C1 \sim C1')\lambda \cup (C2 \sim C2')\lambda \cup eq$

For LC, the resultant clause *is* the conjunction of the members of Cλ U eq (instead of Just, Cλ) In both cases, of course, the λ is the substitution set obtained from EAI.

Basic f-resolution and LC when extended by 2AI still generate only a finite number of f-clauses from any given finite set. For a finite set C of f-clauses, we mean by efR(C) tne set of all f-clauses which result from the application of basic f-resolution and LC, both extended by SAI, As usual, we define efR°(C) = C and efftrrM(C) = $efR^m(C)$ U efR(efIP(C)). For any set C of f-clauses, by EQ(C) we mean the set of equality axioms of the language of C. It is then possible to prove the following:

Theorem 3: Let E1 and E2 be any two closed expressions in a first-order language with identity. If for some n, $fC(E2) \subseteq efR^n(fC(E1))$, then $\{E2\} \cup EQ(fC(E1)) \Vdash E1$.

In other worus, our methods, extended by EAI, will generate from the f-clauses of an input expression E1, the f-clauses of expressions E2 such that given the usual equality axioms, £1 ie entailed by E2.

The intuitions embodied in the EAI extension are slightly more complex than those embodied in heuristic (H.1) in two ways. First, the extended method proposes a reduction in the number of named entities, providing such a reduction will reduce the number of atomic facts. And secondly, the extended method proposes restrictionf on certain functional values, again providing that such restrictions will reduce the number of atomic facts.

Two rather distinct circumstances for the application of f-resolution may be identified. In the first case, no background assumptions arc presuppoeori. The input to our procedure is then Just the set of f-clauses obtained from the expression of the state of affairs for which we would like an hypothesis.

Let us consider an example. Suppose we wish to find an hypothesis to account for the following expression:

(c.1) $(x)((\exists y)(Rxy \;\&\; \neg Py) \vee (z)(\neg Rxz \vee (w)(\neg Rzw \vee Pw)))$

From (c.1) we obtain the following f-clauses:

(c.2) $Ray \;\&\; \neg Py$

(c.3) $\neg Rab$

(c.4) $\neg Rbc$

(c.5) Pc

Our method yields the following results:

(c.6)	$\neg Pb$	from (c.2) and (c.3)
(c.7)	$a = b \;\&\; \neg Pc$	from (c.2) and (c.4)
(c.8)	Rac	from (c.2) and (c.5)
(c.9)	$b = c$	from (c.3) and (c.8)
(c.10)	$a = b$	from (c.4) and (c.8)
(c.11)	$b = c$	from (c.5) and (c.6)
(c.12)	$a = b$	from (c.5) and (c.7)

To obtain the desired hypothesis, we must trarslate back into normal first-order notation, Put the question is, which f-clauses should be included? Py Theorem 3, we could choose any single f-clause, or Any non-empty subnet. Various heuristics could be used to make the selection, but we will mention only one here.

Suppose we are after hypotheses which account for a disjunction, say P V Q, and suppose we know that S entails P and R entails 0. Then we could use S, or we could use R, or we could use the disjunction S V R. One reason for preferring the disjunctive hypothesis is that it is deductively weaker than the alternatives; it therefore places fewer reptrictions on the range of possibilities and is thus in some intuitive sense more likely to be true. To make this idea a little more precise, we introduce a definition. We will say that one f-clause A covers another f-clause B just in case either (i) A is identical to 6, or (ii) A results from applying extended LC to some f-clause C, where C covers B, or (iii) A results from applying extended basic f-resolution to f-clauses C and D, where B is covered by at least one of C and D. We may then formalize our heuristic by saying that in the selection of f-clauses for hypotheses, we should include f-clauses sufficient to cover every f-clause in the original input. Thus in our example, we should not use Just (c.4), or Just (c.6), or just (e.12). In fact, there is no single f-clause which covers all of (c.2 - 5). But we could use (c.11) and (c.12) to obtain:

(c.13) $(x)(y)(z)(x = y \vee y = z)$

Or we could use (c.3), (c.4), and (c.8) to obtain:

(c.14) $(x)(y)(z)(\neg Rxy \vee \neg Ryz \vee Rxz)$

Or we could use (c.3) and (c.12) to obtain:

(c.15) $(x)(y)(\neg Rxy \vee x = y)$

Each of these hypotheses entails (c.1). There are various other possibilities as well, but the range of possibilities is substantially smaller than that which we would have to consider if no heuristic were employed.

The second circumstance in which we might wish to use f-resolution is the more usual one. Frequently we wish to obtain an hypotheiss to account for A, given the presupposition B. That is, we want some C ouch that:

(d.1) C & B ⊩ A

But (d.1) is equivalent to:

(d.2) C ⊩ ¬ B ∨ A

And (d.2) is equivalent to:

(d.3) ¬ B ∨ A ⊩$_F$ C

From this example we can see that the input to our f-resolution procedure should be the f-clauses in A along with the f-clausos in the negation of our presupposition. If we have more than one presupposition, then we must take the negation of the conjunction of all our presuppositions. Our input presuppositions can be any statements at ell, whether factual or logical. We could for example have the axioms for identity theory as our presuppofiitione.

For simplicity, let us consider a standard syllogistic example. Suppose we would like an hypothesis which would entail "All dogs are hairy", given the assumption that "All dogs are mammals". Our assumption may be formalized as:

(e.1) (x)(¬ Dx ∨ Mx)

The negation of (e.1) is just:

(e.2) (∃x)(Dx & ¬ Mx)

From (e.2) we obtain the f-clause:

(e.3) Dx & ¬ Mx

The statement we want to be able to derive may be formalized as:

(e.4) (x)(¬ Dx ∨ Hx)

And (e.4) gives us the following f-clauses:

(e.5) ¬ Da

(e.6) Ha

From (e.3) and (e.5) we obtain:

(e.7) ¬ Ma

No other applications of our rules are possible. So we get the following hypotheses, among others:

(e.8) (x)Hx from (e.6)

(Everything is hairy.)

(e.9) (x) ¬ Mx from (e.7)

(There are no mammals.)

(e.10) (x)(¬ Mx ∨ Hx) from (e.6) and (e.7)

(All mammals are hairy.)

Note that (e.10) is the only hypothesis obtained in accord with the covering heuristic discussed above.

We should say a few more words about the covering heuristic when used with antecedently given information. Suppose we are assuming B1 and B2 as antecedently given information and we are seeking an hypothesis to account for A. Our input would then be the f-clauses from ¬ B1 ∨ ¬ B2 ∨ A. Suppose further that in deriving our hypotheses, we make no use of any f-clause from ¬ B1. Then the hypotheses given by the covering heuristic would be of the form ¬ B1 ∨ C. But note that B1 & (¬ B1 ∨ C) is equivalent to B1 & C. So there would be no need to include the f-clauses from ¬ B1 in our hypotheses under these circumstances. The same considerations would apply to conjunctive components of B1 and B2 (i.e., to disjunctive components of ¬ B1 and ¬ B2). Consequently, the covering heuristic should only be applied to that portion of the input which does not arise from antecedently given presuppositions. In this simple example, the covering heuristic should be applied only to A and not to ¬ B1 and not to ¬ B2. In the syllogistic example above, the covering heuristic should be applied only to (e.5) and (e.6) and not to (e.3).

Experimentally, we can rerort that for any syllogism whose validity does not assume existential import, given the conclusion and one premise, our method with the covering heuristic will yield Just the required additional premise. Further, for those syllogisms which do require existential import, our method generates the hypothesis that there is something in the appropriate subject class from the input of the desired conclusion and the oremises as assume—tions.

The most extensive experimental test of our methods has beer made in the ares of non-standard logics. In particular we have dealt in detail with many systems of modal logic and many systems, of relevance logic. The characteristic semantics for such systems depends on the determination of certain restrictions to be placed on given semantic relations. Using our methods we have been able to generate from a semantic translation of the axioms And inference rules the known semantic requirements for every case investigated—about ?0 systems have been examined at this time (see [4] and [8]). Further, in these cases, we can employ a slightly modified version of the covering heuristic and prove that the hypotheses gererated by our methods will yield the chcracteristic semantics if any first-order conditions will. These results are particularly pleasing, as our methods were designed from a general point of view, with no thought of this application.

We would like to see more extensive testing of our methods 1n more practical, less formal areas. In any case, it should be borne in minc' that the method carnot be used to gercrate new predicates (i.e., new "concepts"). The hypotheses generated ?re always in terms of the input predicates, with the possible exception of equality.

We would finally like to point out that EAI could be used to supplement norrial deductive resolution. The resultant clause would Ve formed in the usual way along with the disjuretion of the negations of the equality aesumptione in eq. By employing simple lexical ordering principles in formulating equality statements, no special axioue for equality would be required. (Such an application would be similar to the method described in [13].) It seems intuitively that EAI might substantially speed up proofs involving equality by introducing Just the appropriate equality statements needed for obtaining resolvents. However, at this time we have not conducted an experimental examination of EAI in the deductive context.

References

[1] Brown, J. S. "Steps toward automatic theory formation," Third International Joint Conference on Artificial Intelligence (1973) Advance Papers of the Conference. Stanford Research Institute, Menlo Park, California, pp. 121-129.

[2] Buchanan, B., Sutherland, G., and Feigenbaum, £• A. "Heuristic dendral: a program for generating explanatory hypotheses in organic chemistry," Machine Intelligence 4, E. Meltzer and D. Michie, eds. Edinburgh University Press (1969), pp. 103-133.

[3] Haslov, S. Ju. "Proof-search strategies for methods of the resolution type," Machine Intelligence 6, B. Meltzer and D. Michie, eds. Edinburgh University Press (1971), pp. 77-90.

[4] Morgan, C. G. "Automated generation of semantic hypotheses for modal logics," Society for Exact Philosophy, annual meeting, 1974, fort! coming.

[5] Morgan, C. G. "Hypothesis generation by machine," Artificial Intelligence, vol. 2 (1971), pp. 179-187.

[61 Morgan, C. G. Inductive Resolution. M.Sc. thesis. Department of Computing Science, University of Alberts, 1972.

[7] Morgan, C. G. "On the algorithmic generation of hypotheses," Scientia. vol. 108 (1973), PP. 585-598.

[8] Morgan, C. G. "Semantic discovery for relevance logics," International Conference or Relevance Logics, 1974, forthcoming.

[9] Plotkin, G. D. "A note on inductive generalization," Machine Intelligence 5. B. Meltzer and D.~Michie eds. Edinburgh Univeraity Press (1969), pp. 153-163.

[10] Plotkin, G. r. "A further note on inductive generalization," Machine Intelligence 6, B. Meltzer and 0. Michie, eds. Edinburgh University Press (1971), pp. 101-124.

[11] Pople, H. E., Jr. "On the mechanization of abductive logic," Third International Joint Conference on Artificial Intelligence (1973) Advance Papers of the Conference. Stanford Research Institute, Menlo Park, California, pp. 147-152.

[12] Popplestone, R. J. "An experiment in automatic induction," Machine Intelligence 5, B. Meltzer and D. Michie, eds. Edinburgh Eniversitv Press (1969), pp. 203-215.

[13] Sibert, E. E. "A machine oriented logic incorporating the equality relation," Machine Intelligence 4, E. Meltzer and P. Michie, ede. Edinburgh University Press (1969), pp. 103-133.

This research was supported by a grant from the Killam Foundation, administered by the Canada Council.