A SEMANTICALLY GUIDED DEDUCTIVE SYSTEM FOR AUTOMATIC THEOREM-PROVING

by

Raymond Reiter

Department of Computer Science
University of British Columbia
Vancouver, B.C., Canada

Abstract

This paper presents a semantic and deductive formal system for automatic theorem-proving. The system has, as its deductive component, a form of natural deduction. Its semantic component relies on an underlying representation of a model. This model is invoked to prune subgoals generated by the deductive component, whenever such subgoals test false in the model. In addition, the model is used to suggest inferences to be made at the deductive level. Conversely, the current state of the proof suggests changes to be made to the model, e.g. when a construction is required as in geometry.

The system is seen to possess a very smooth and transparent interface between its semantics and deductive syntax. These semantic and syntactic subsystems interact continuously during the search for a proof, each suggesting to the other how next to proceed. Particularly appealing is the naturalness of the system from a human point of view.

1. Introduction

The past dozen years or so have witnessed a great deal of programming energy devoted to mechanizing first order logic. Several proof procedures have been proposed and implemented with varying degrees of success. Among these are systems of natural deduction[13], Herbrand search procedures[4], and resolution[11].

It quickly became apparent that these proof procedures alone were impractical on any interesting mathematical theory. One approach toward alleviating these difficulties was to develop completeness preserving refinements of the rules of inference. Essentially, these are suitably restricted rules, often depending upon the syntactic structure of the formulae, which generate a narrower (hut usually deeper) search tree. Virtually all of the results obtained along these lines are for resolution systems. Examples are resolution with merging[1], linear resolution', A-ordering[12] etc. plus a whole host of combined strategies. Experimental evidence[14] indicates that this approach alone fails on even mildly serious theorems.

Virtually everyone is now agreed that knowledge about the problem domain must be used in the logic. The question is how. There seem to be two approaches.

1.1 Semantics as Domain Dependent Heuristics

In this approach semantic Information is embedded, in the logic, as suitable domain dependent heuristics which depend, for their effect, upon the syntactic form of the current formulae and which therefore act like new rules of inference. No representation of the problem domain itself is present. Semantics is conveyed through some fixed set of heuristic procedures representing that knowledge of the problem domain which is believed to be significant in guiding the search for proofs. This semantic Information is incorporated into the system by augmenting its purely syntactic rules - the theorem-prover continues to be syntax-driven.

Examples of this approach may be found for analysis , set theory2, and equality9. It is reasonably clear that such domain dependent heuristics will be essential components of any theorem-prover capable of doing real mathematics. For example, a number theorist will require procedures for solving equations and formula manipulation. We shall argue that much more than this Is required.

One difficulty with this approach is the need to anticipate, at the coding stage, all. of that knowledge about the problem domain which could be of assistance in discovering proofs. This knowledge is then embedded in the logic in the form of suitable heuristics which govern the generation of successor nodes in the proof tree. Unfortunately, such heuristics are rarely independent, but interact in highly complex ways. If, later, some new heuristic is discovered, this can lead to a major overhaul of the program. There is, under this approach, a very real danger of an overproliferation of special, mutually interacting heuristics with an attendant loss of system extensibility.

It can also be argued that domain dependent heuristics capture a weak notion of semantics in the sense that they affect the proof tree only under certain prespecified conditions. Insofar as a formula enters into such a condition it may he said to have meaning so that the. corresponding heuristic decision has a semantic basis. But there is no concept of the meaning of an arbitrary formula, and hence no provision for decisions based upon general semantic considerations. In particular then, if a node of the proof tree has no associated heuristic, no semantic decision can be made about its most plausible successors.

Worse still is the lack of any kind of reasonable control over dead-end searches. If the application of an heuristic or rule of inference leads the proof astray, there is no provision for using knowledge about the problem domain to detect this. Those techniques which are currently used, such as setting parameters for maximal clause length or depth of function-nesting are clearly ad hoc, and independent of the domain. This difficulty with blind alleys is compounded in the presence of a large number of axioms and theorems which might be irrelevent to the proof being sought. Such formulae are guaranteed to lead to dead-end searches. There is no way that a serious theorem-proving system can avoid having to deal with this situation. To our knowledge, no current theorem-prover, all of which are based on refinements and/or domain dependent heuristics, Is capable of coping with this problem.

1.2 Semantics as the Representation of Models

The main thrust of this paper is the following: Instead of relying exclusively upon domain dependent heuristics which represent fixed, *a. priori* knowledge about the problem domain, represent the problem domain itself, i.e. present to the theorem-prover a model of the axiomatic system involved. In addition, what is needed is a set of procedures for extracting information about the model when required by the theorem-prover, together with a flexible, general interface between such a semantic subsystem and the purely syntactic logical system.

The distinction, therefore, between this approach and that based on domain dependent heuristics is that the latter explicitly represents *that* semantic information which is believed a priori to be relevant, whereas the former implicitly represents all of the information available in the model which is capable of being extracted by the available procedures.

The idea of using models for theorem-proving is by no means new. In the late 1950's Gelernter and his co-workers5,6 developed a system for plane geometry whose success was due primarily to its use of geometric diagrams. Despite this early success the use of models has not been widely adopted in theorem-proving circles, with the sole exception of work by Slagle12 in the context of resolution.

The present paper represents a generalization of the work of Gelernter. The resulting system has, as a deductive component, a form of natural deduction as opposed to resolution. Its underlying semantic subsystem relies upon some representation of a model, and is invoked to prune dead-end searches, and to make inferences at the deductive level. The deductive level in turn, is used to dynamically modify the underlying model, as the proof unfolds. In addition to its naturalness, the system provides considerable control over the search for a proof.

2. Models and the Interpretation of Formulae

Assume given some quantifier-free first order theory with variables $x_1, x_2, \ldots$, predicate symbols P, Q, R, ..., and function symbols f, g, h, ... The notions of term, well formed formula (wff), atom, and literal have their usual definitions [4].

An interpretation I for this theory is specified by a non empty set D (the domain) together with

1. For each n-ary predicate symbol P, an associated function $P_I : D^n \to \{0,1\}$.
2. For each n-ary function symbol f, an associated function $f_I : D^n \to D$.

The interpretation I is finite provided D is finite. For $a_1, \ldots, a_n \in D$, define $[x_i]_I (a_1,\ldots,a_n) = a_i$, $1 \le i \le n$.

If $t^{(1)},\ldots,t^{(m)}$ are terms, f an m-ary function symbol, and $x_1,\ldots,x_r$ are all of the variables occurring in these terms, then for $1 \le r \le n$, define

$$[f(t^{(1)},\ldots,t^{(m)})]_I (a_1,\ldots,a_n) = f_I([t^{(1)}]_I(a_1,\ldots,a_n),\ldots,[t^{(m)}]_I (a_1,\ldots,a_n))$$

If P is an m-ary predicate symbol, define

$$[P\, t^{(1)},\ldots,t^{(m)}]_I (a_1,\ldots,a_n) = P_I[t^{(1)}]_I(a_1,\ldots,a_n),\ldots,[t^{(m)}]_I (a_1,\ldots,a_n)$$

The extension of these ideas to non-atomic wffs is made in the usual way. Thus

$[\overline{W}]_I (a_1,\ldots,a_n) = 1$ if $[W]_I (a_1,\ldots,a_n) = 0$

$= 0$ if $[W]_I (a_1,\ldots,a_n) = 1$

$[W \wedge W']_I (a_1,\ldots,a_n) = 1$ if $[W]_I (a_1,\ldots,a_n) = 1$ and $[W']_I (a_1,\ldots,a_n) = 1$

$= 0$ otherwise

etc.

Our aim is to define the notion "Wff W is true in I". Before doing so, it must be decided whether the variables of W are to be interpreted as existentially quantified, or universally quantified. Later, both notions will be required. Therefore, define

$I \models_E W$ iff there exist $a_1,\ldots,a_n \in D$ such that $[W]_I(a_1,\ldots,a_n) = 1$. Otherwise $I \not\models_E W$.

$I \models_U W$ iff for all $a_1,\ldots,a_n \in D$, $W_I(a_1,\ldots,a_n) = 1$. Otherwise $I \not\models_U W$

I is a model for W iff $I \models_U W$. I is a model for a set of wffs iff I is a model for each wff in the set. Reference 10 contains a more thorough discussion of models, in particular, the problem of representing infinite models.

2.1 Wffs in Skolem Form

Most axiomatic theories in mathematics are formulated in a first order theory with quantifiers, rather than in a quantifier-free form. The usual procedure in automatic theorem-proving is to first eliminate all quantifiers by introducing Skolem functions[11], and we propose to do the same. However, because the deductive system of this paper deals with natural deduction, the system establishes validity instead of unsatisfiability. Hence, a dual process of Skolemization is used, i.e. universally quantified variables are replaced by Skolem functions, rather than existentially quantified variables, as is the case with unsatisfiability[11]. For example, the wff

$$(x)[(u)[\, Pu,x \wedge \exists y((z)Qy,z,u \supset Rx,y)] \supset \exists w Sx,w$$

has Skolem form

$$Pu,a \wedge (Qf(u),g(u),u \supset Ra,f(u)) \supset Sa,w$$

where a, f and g are Skolem functions.

Notice that in an interpretation I Skolem functions are treated as function symbols, i.e. if f is an n-place Skolem function, then I assigns to f a function $f_I : D^n \to D$. For many purposes such an interpretation of Skolem functions is not as general as one would like. For example, consider the theorem "In a group, if $x^2 = e$ for all group elements x, then the group is commutative". Formally, in Skolem form, the theorem is

$$\text{Group axioms} \wedge x^2 = e \supset ab = ba$$

where a and b are Skolem constants. An appropriate model M would be a finite group with domain D for which $d^2 = e$ for each element $d \in D$. In addition, a and b would each be bound to fixed elements d_a and d_b of D. Then to test the truth value in M of a wff W(a,b), one determines the truth value of $W(d_a,d_b)$. Evidently there is a loss in generality here, since intuitively W(a,b) is true in M iff for all $d_1,d_2 \in D$, $W(d_1,d_2)$ is true, i.e. a and b should independently range over the domain D rather than being bound to fixed elements of D. Unfortunately, there appear to be technical problems associated with this more general point of view, so for the purposes of

this paper, we shall hold to the definition of model given above. For a more complete discussion of these problems, see reference 10. Notice, however, that the formal system L of Section 3 is not dependent on any particular definition of a model, although of course a more general notion of semantics will yield a correspondingly more powerful system.

3. The Semantic and Deductive System L

The system L has twelve rules of inference. These are applied, in order, to any current subgoal. If any rule fails, the next rule is attempted. If all fail on a current subgoal, NIL is returned. The top-level subgoal has the form $H_1,H_2,\ldots,H_n \vdash W$ where the wffs $H_1,H_2,\ldots,H_n$ are the axioms, previously proved theorems, definitions, and special hypotheses of the current theorem W. These wffs, as well as A, B and C are all Skolemized. s and t are terms. ψ, ψ_1 and ψ_2 are (possibly null) sequences of wffs. In an expression of the form $\psi \vdash A$ the set of wffs in the sequence ψ is called the antecedent. Unless otherwise stated, M is a model of the antecedent of the current subgoal in the rules of inference below. The square brackets enclosing a wff in Rules 6 and 8 indicate that the free variables of the wff are given an existential interpretation[8] i.e. a model M of the antecendent in $\psi_1,[A],\psi_2 \vdash B$ has the property $M \models_E A$ rather than $M \models_U A$.

Although for clarity we have not explicitly provided for it in the rules, it is understood that any wff or subwff of the form $\neg(A \vee B)$ shall be replaced by $\overline{A} \wedge \overline{B}$, $\neg(A \wedge B)$ by $\overline{A} \vee \overline{B}$, and $\overline{\overline{A}}$ by A.

	Current Subgoal	Next Subgoal/ Value Returned
1.	$\psi \vdash A$ If σ is a substitution such that $M \models_E A\sigma$, and if $\psi \vdash A\sigma$ returns σ_1	$\sigma\sigma_1$
2.	$\psi, A \wedge B \vdash C$	$\psi, A, B \vdash C$
3.	$\psi_1, A, \psi_2 \vdash B$ (i) If A and B have most general unifier σ_1	σ_1
	(ii) If σ_1 is a substitution which pairwise unifies all terms of A and B with the exception of correseponding terms $t_1,\ldots,t_r$ and $s_1,\ldots,s_r$, and if $M \models_E \bigwedge_{i=1}^{r}(t_i = s_i)\sigma_1$, and $\psi_1, A, \psi_2 \vdash \bigwedge_{i=1}^{r}(t_i = s_i)\sigma_1$ returns σ_2	$\sigma_1\sigma_2$
4.	$\psi \vdash A \wedge B$ If $\psi \vdash A$ returns σ_1, $M \models_E B\sigma_1$, and $\psi \vdash B\sigma_1$ returns σ_2	$\sigma_1\sigma_2$
5.	$\psi \vdash A \vee B$ If $M \models_E A$	$\overline{A}, \psi \vdash B$
	If $M \models_E B$	$\overline{B}, \psi \vdash A$
	Otherwise	$\psi \vdash A$ or $\psi \vdash B$
6.	$\psi, A \supset B \vdash C$ If $A \supset B, \psi, [B] \vdash C$ returns σ_1, $M \models_E A\sigma_1$ and $A \supset B, \psi \vdash A\sigma_1$ returns σ_2	$\sigma_1\sigma_2$
7.	$\psi \vdash B \supset C$	$\psi, B \vdash C$
8.	$\psi, A \vee B \vdash C$ If $A \vee B, \psi, [A] \vdash C$ returns σ_1, $M \models_E B\sigma_1$, and $A \vee B, \psi, [B\sigma_1] \vdash C$ returns σ_2	$\sigma_1\sigma_2$
9.	$\psi \vdash \overline{A}$	$\psi, A \vdash$ NIL
10.	$\psi_1, s=t, \psi_2 \vdash A$	$\psi_1', \psi_2' \vdash A'$ where ψ_1', ψ_2', A' are obtained from ψ_1, ψ_2, A respectively by substituting t for s.
11.	$\psi \vdash s = t$ If σ_1 is a most general unifier of s and t	σ_1
12.	$\psi, A \vdash B$	$A, \psi \vdash B$

3.1 Remarks

1. The purely syntactic rules of the system L have been patterned after a similar deductive system due to Bledsoe, Boyer, and Henneman3 and represent a slight reformulation and generalization of their rules. Essentially what is new here is the integration of semantics into their purely deductive syntax. In fact, it can be shown10 that the system L is a generalization of the syntactic and semantic formal system underlying the work of Gelernter and his co-workers on the. Geometry Machine5,6.

The deductive syntax of the system L stands in contrast to the usual resolution based theorem-proving techniques11, and ismuch closer to various systems of natural deduction, for example those used by Wangl3. L's deductive component also bears a resemblance to the system of Nevins8. In particular, Rules 4 and 8 correspond to his case analysis rules. We are encouraged by this since Nevins' theorem-prover appears to be the most successful by far of any current general purpose system.

L is incomplete. For example, the tautology $((p \supset q) \wedge (\overline{p} \supset q)) \supset q$ is not provable.

2. The substitution σ of Rule 1 allows for "good guesses" to be made by the theorem-prover, based upon observations made in the model, as to what the occurring free variables actually represent. Thus, if $M \models_E W(x_1,\ldots,x_n)$ and, moreover, if there exist unique objects $a_1,\ldots,a_n$ in the domain of M such that $W(a_1,\ldots,a_n)$ is true in M and if these $a_1,\ldots,a_n$ can be interpreted in the syntax as terms $t_1,\ldots,t_n$ involving only function symbols and Skolem functions, then it would be an excellent guess to attempt, as the current syntactic goal, $\vdash W(t_1,\ldots,t_n)$ rather than $\vdash W(x_1,\ldots,x_n)$. Clearly, the first goal will, in general, be much easier to prove than the second. Even when the a_i are not unique, there may be additional semantic information available with which to make a plausible guess, or several guesses which are pursued in parallel. This facility for "σ-guessing" seems to

us to be central to human proof discovery, and represents a powerful use for models in automatic theorem-proving.

3. Suppose Rule 4 applies, and succeeds on its first AND-subgoal by returning σ_1. If an appropriate model M can be found such that $M \models_E B\sigma_1$, or if $\psi \vdash B\sigma_1$, returns NIL, then we must back up to the first AND-subgoal, attempt to have it succeed by returning a different substitution σ_1', then try to establish the second AND-subgoal with σ_1'. Without the use of M, this corresponds to the back-up procedure used by Bledsoe et al. A similar technique is used with Rules 6 and 8. This need for back-up is a serious computational limitation of Bledsoe's rules of inference; it also occurs in Nevins' system and, in disguised form, in resolution based systems of deduction.

In effect therefore, the system L provides for the use of counterexamples to prune the search tree. If the test $M \models_E B\sigma_1$ fails, M is a counterexample to the subgoal $\psi \vdash B\sigma_1$ so there is no sense in pursuing the second AND-subgoal. The model is being used as a "semantic seive" for trapping our incorrect substitutions σ_1. Moreover, an incorrect σ_1 is recognized before embarking on the second AND-subgoal $\psi \vdash B\sigma_1$, rather than as a consequence of syntactic failure on that subgoal.

The statement of Rule 4 as well as the previous discussion suggest that no use is made of the wff B, in pursuing the subgoal $\psi \vdash A$. In fact, B can (and should) be used to monitor the proof of A, as follows. Assume, for the sake of discussion, that A and B have one free variable, x, in common. Suppose that, during an attempted proof of A, x is instantiated by the term t. At this point, make the semantic test $M \models_E B(t)$. If successful, proceed with the proof of A. Otherwise, A's proof has obviously gone astray and must be redirected. Thus, rather than patiently waiting for A to deliver a (possibly wrong) σ_1, the wff B should be continuously semantically monitoring the proof of A, thereby minimizing the risk of receiving an incorrect σ_1. Moreover, if $\psi \vdash A$ returns σ_1, and if $A\sigma_1$ is not fully instantiated, then $A\sigma_1$ can be used, in the same way, to semantically monitor the proof of $\psi \vdash B\sigma_1$. Similar remarks apply to Rules 6 and 8. We believe that this kind of parallel processing of dependent subgoals will considerably alleviate the problem of back-up encountered by purely syntactic theorem-provers.

it. It Is not intended that the model M necessarily remain fixed during the course of a proof. Thus, each application of one of the rules may invoke a different model, where appropriate. For example, in geometry,constructions may be suggested by the proof so far, in which case the current model will be the initial model augmented by suitable new points and line segments. Since substitution Instances can often be interpreted as constructively asserting the existence of new objects, every such substitution encountered during the course of the proof thus far can be used to augment the initial model by these new objects, in order to yield the current model. Example 4 below illustrates how the system L suggests, in a very natural way, the necessary changes to be made to an initial model, and how the model evolves during the course of the proof.

Similarly, a case analysis might invoke a different model for each case. Thus an application of Rule 8 will typically call on two models, one for the "A case" and one for the B.

5. We feel that the system L possesses a very smooth and natural Interface between its semantics and deductive syntax. In particular, as suggested in remarks 3, and 4. above, these semantic and syntactic subsystems interact continuously during the search for a proof, each suggesting to the other how next to proceed. There is a naturalness with which systems of natural deduction admit a semantic component with the result that a great deal of control is gained over the search for a proof, It is precisely for this reason that we argue in favour of their use in automatic theorem-proving, in opposition to the usual resolution-based systems, which appear to lack any kind of reasonable control over dead-end searches.

4. Examples

We give a number of examples of the use of the system L. For economy, explicit references to ψ will often be omitted, in which case only those wffs relevant to the proof, or none at all, will be indicated to the left of the symbol $\vdash$.

Example 1

We first give an example of a proof in propositional logic. The top level goal is

$\vdash ((A \supset B) \supset (C \supset B)) \wedge (A \supset B) \wedge (D \supset B) \wedge C \supset D \vee B$

One application of Rule 7 followed by three of Rule 2 leads to

$(A \supset B) \supset (C \supset B), A \supset B, D \supset B, C \vdash D \vee B$

Rule 5 applies; with $M = \{A,B,C,\bar{D}\}$ one of its OR-subgoals is semantically eliminated leaving

1. $\bar{D}, (A \supset B) \supset (C \supset B), A \supset B, D \supset B, C \vdash B$

Rule 12 applies.

11. $C, \bar{D}, (A \supset B) \supset (C \supset B), A \supset B, D \supset B \vdash B$

Rule 6 applies, but with $M = \{A,B,C,\bar{D}\}$ its second AND-subgoal fails. Rule 12 applies.

111. $D \supset B, C, \bar{D}, (A \supset B) \supset (C \supset B), A \supset B \vdash B$

Rule 6 applies, but with $M = \{\bar{A},B,C,\bar{D}\}$ its second AND-subgoal fails. Rule 12 applies.

1111. $A \supset B, D \supset B, C, \bar{D}, (A \supset B) \supset (C \supset B) \vdash B$

Rule 6 applies.

11111. $(A \supset B) \supset (C \supset B), A \supset B, D \supset B, C, \bar{D}, C \supset B \vdash B$

and

11112. $(A \supset B) \supset (C \supset B), A \supset B, D \supset B, C, \bar{D} \vdash A \supset B$

Both succeed, the first by Rules 6 and 3, the second by Rule 3.

Notice that the semantic proof uses two models, and that these are dynamically determined, as the proof unfolds.

Example 2

This example illustrates how a diagram in geometry rejects an application of Rule 6. The theorem is "If triangle ABC has equal base angles, then AB = AC". Assume that one of the axioms present is

α: Square xyzw $\supset$ xy = yz

The theorem is

$\Delta ABC, \angle ABC = \angle ACB, \psi \vdash AB = AC$

where ψ is a sequence of all of the axioms. Several applications of Rule 12 will eventually yield the subgoal

$\alpha \vdash AB = AC$

Rule 6 applies yielding the second AND-subgoal

)— Square ABCw

which is clearly false in any right angle free diagram of an isosceles triangle, so that this application of Rule 6 is rejected. Notice that if we had been unfortunate enough to have chosen, as a diagram of an isosceles triangle, one in which BAC was a right angle, there would have been no justification in rejecting the application of Rule 6 - a satisfying point w would exist.

Example 3

This is a geometry example which illustrates the use of semantics in rejecting a subgoal generated by Rule 4. The theorem states "For every triangle there is a point equidistant from its three vertices". Assume that, among the axioms present is

α : midpt(u,v)u = midpt(u,v)v

(Each line segment uv has a midpoint midpt(u,v))

The theorem is

$\Delta ABC,\ \psi \vdash xB = xC \wedge xA = xC$

where ψ is a sequence of all of the axioms.

Rule 4 yields the first AND-subgoal

1. $\Delta ABC,\ \psi \vdash xB = xC$

Since α is embedded in ψ, Rule 3 applies yielding the substitution midpt(B,C)|x. This backs up to the application of Rule 4, yielding the second AND-subgoal

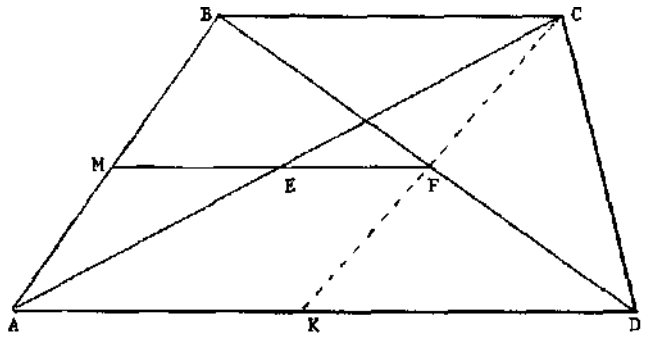
$\Delta ABC,\ \psi \vdash midpt(B,C)A = midpt(B,C)C$

which is false in any "random" diagram of a triangle ABC. Hence, a different proof of 1. is attempted.

Example 4

This example is drawn from Gelernter et al . It is of interest because its proof requires a subtle construction. We sketch a portion of the proof, using the system L, which illustrates how L forces the necessary construction, as the proof unfolds.

The theorem states "If ABCD is a trapezoid with BC||AD, and if the line joining the midpoint E of AC to the midpoint F of BD meets AB in M, then MA-MB," The Initial model (without the point K and dotted line segment CFK) is:



The crux of the proof is to prove EF|JAD since then, in triangle BAD, FB-FD and MF||AD whence MB-MA. To prove EF |AD, the line segment CF must be drawn, and extended to meet AD in K. (The Geometry Machine was unable to discover this construction, and had to be given this hint before it found a proof). Then it must be established that FC=FK whence, since in triangle ACK EC=EA and FC=FK, EF||AK. We shall show how the subgoal EF||AD leads to the subgoal FC=FK, in the process forcing the construction of the line segment CF and its extension to K.

Assume the presence (among others) of the two axioms

α : xy||uv $\wedge$ Coll uvw $\supset$ xy||uw

β : Δxyz $\wedge$ Coll xuy $\wedge$ Coll xwz $\wedge$ ux=uy $\wedge$ wx=wz $\supset$ uw||yz

Assuming that EF||AD is to be proved, the current subgoal is

$\beta, \alpha \vdash$ EF||AD

Rule 6 applies.

1. β, xy||uw $\vdash$ EF||AD

Rule 3 yields σ_1 = {E|x,F|y, A|u, D|w}. Since $M \models_E$ EF||Av $\wedge$ Coll AvD (any v on AD will do), proceed with

2. $\beta \vdash$ EF||Av $\wedge$ Coll AvD

Rule 4 applies. The first AND-subgoal is

21. $\beta \vdash$ EF||Av

Rule 6 applies.

211. uw||yz $\vdash$ EF||Av

Rule 3 yields σ_1 = {E|u, F|w, A|y, v|z}.

Now $M \models_E \Delta xAv \wedge Coll xEA \wedge Coll xFv \wedge Ex=EA \wedge Fx=Fv$

Indeed, in the diagram there are unique satisfying points x and v, namely x=C and v=K. (Notice that K will be determined from the model, as the term intersection (A,D,C,F).) Hence, take as the σ of Rule 1 σ = {C|x,K|v}. This yields a new model M' containing the new point K and new line segment CFK. Moreover, backing up the substitution K|v to the second AND-subgoal derived from 2., we see that $M' \models_E$ Coll AKD. Thus, we confidently proceed with the second AND-subgoal of Rule 6.

212. $\vdash \Delta CAF \wedge$ Coll CEA $\wedge$ Coll CFK $\wedge$ EC=EA $\wedge$ FC=FK

The first four literals in this conjunct are easily established, leaving the subgoal $\vdash$ FC = FK which is as far as we wanted to carry the proof. There still remains to be established the second AND-subgoal of Rule 4 at 2. This yields, backing up the substitution K|v

22. $\vdash$ Coll AKD

which is easily proved.

Notice the use, at 211, of the substitution σ. This is a good example of the use of a model in making "premature" instances of variables, and illustrates why Rule 1 makes provision via σ for such "good guesses". Notice also how, at 211., a higher level parallel subgoal (namely the second AND-subgoal derived from 2.) was used to semantically monitor the substitution K|v via the test $M' \models_E$ Coll AKD. Finally the example illustrates how the system L encourages a model to change during the course of a proof.

Example 5

The theorem states "If S is a non-empty subset of a group such that $xy^{-1} \varepsilon S$ whenever x and $y \varepsilon S$, then $x^{-1} \varepsilon S$ whenever $x \varepsilon S$".

ex=x, xe=x, xI(x)=e, I(x)x=e,Sb, Sx $\wedge$ Sy $\wedge$ xI(y) = z $\supset$ Sz $\vdash$SI(b)

Rule 6 applies and yields as its second AND-subgoal

1. $\vdash Sx \wedge Sy \wedge xI(y) = I(b)$

which is semantically true.

Rule 4 applies, yielding as its first AND-subgoal

11. $\vdash Sx$

which succeeds with $b|x$. This backs up to the second AND-subgoal of Rule 4

12. $\vdash Sy \wedge bI(y) = I(b)$

This subgoal may or may not be semantically true, depending upon the model being used. Let us assume this failure is not detected and proceed. It is clear that a second application of Rule 4 will yield $b|y$ from its first AND-subgoal, leaving as its second AND-subgoal

$\vdash bI(b) = I(b)$

which is false in any model in which b is not assigned the group identity. This failure backs up to 1. This time, take as the first AND-subgoal of Rule 4

11. $\vdash xI(y) = I(b)$

which succeeds with $e|x$, $b|y$. The second AND-subgoal is

12. $\vdash Se \wedge Sb$

which is semantically true and easily proved.

Notice that a clever theorem-prover would have observed that, in M, $\{e|x, b|y\}$ and $\{I(b)|x, e|y\}$ are the only two obvious "solutions" to 1. Hence, it would have returned the two "guesses" $\sigma = \{e|x, b|y\}$ and $\sigma' = \{I(b)|x, e|y\}$ in its treatment of 1. σ' leads to the subgoal

$\vdash SI(b) \wedge Se \wedge I(b)I(e) = I(b)$

which is rejected because it is subsumed by the top level goal. σ leads to the subgoal

1. $\vdash Se \wedge Sb \wedge eI(b) = I(b)$

which has a simple, back-up-free proof.

Acknowledgements

This research was supported by National Research Council of Canada grant A-7642. I have greatly profited by conversations with John Seeley Brown about this work.

References

1. Andrews, P.B. Resolution with merging. J.ACM, 15 (1968), 367-381.
2. Bledsoe, W.W. Splitting and reduction heuristics in automatic theorem proving. Artificial Intelligence, 2 (1971), 55-77.
3. Bledsoe, W.W., Boyer, R.S. and Henneman, W.H. Computer proofs of limit theorems. Artificial Intelligence, 3, (1972), 27-60.
4. Davis, M. and Putnam, H. A computing procedure for quantification theory. J.ACM, 7 (1960), 201-215.
5. Gelernter, H. Realization of a geometry theorem-proving machine. Computers and Thought (eds, Feigenbaum, E.A, and Feldman, J.) New York: McGraw-Hill (1963), 134-152.
6. Gelernter, H., Hansen, J.R., and Loveland, D.W. Empirical explorations of the geometry theorem machine. Computers and Thought (eds, Feigenbaum, E.A. and Feldman, J.) New York: McGraw-Hill (1963) 153-163.
7. Loveland, D.W. A linear format for resolution. Symposium on Automatic Demonstration (ed. Laudet, M.) New York: Springer-Verlag (1970).
8. Kevins, A.J. A human oriented logic for automatic theorem proving, MIT A.I, Laboratory Memo No.268 (Oct.1972),
9. Norton, L.M. Experiments with a heuristic theorem-proving program for predicate calculus which equality. Artificial Intelligence, 2 (1971), 261-284.
10. Reiter, R. The use of models in automatic theorem proving. University of British Columbia, Dept. of Computer Science Technical Report 72-09 (Sept. 1972).
11. Robinson, J.A. A machine oriented logic based on the resolution principle, J.ACM, 12 (1965), 23-41.
12. Slagle, J.R. Automatic theorem proving with renamable and semantic resolution. J.ACM 14 (1967), 687-697.
13. Wang, H. Toward mechanical mathematics. IBM J. Res. Dev. 4 (1960), 2-22.
14. Woe, L., Robinson, G.A., and Carson, D.F. The concept of demodulation in theorem proving. J.ACM, 14 (1967), 698-709.