# Automated theorem-proving for the theories of partial and total ordering

J. R. Slagle* and L. M. Norton†

*Heuristics Laboratory, Division of Computer Research and Technology, National Institutes of Health, Department of Health, Education and Welfare, Bethesda, Maryland 20014, USA*

**To make further progress, resolution principle programs need to make better inferences and to make them faster. Previous papers of the authors have presented a fairly general approach for taking some advantage of the structure of special theories, for example, the theories of equality, partial ordering and sets, and described experiments with a program based on this approach. The object of the approach is to replace some or all of the axioms of the given theory by (refutation) complete, valid, efficient (in time) inference rules.**

**In this paper, the approach is used to develop an improved procedure for 'building in' partial ordering and a procedure for total ordering. These results may be stated roughly as follows.**

**1. If the five (not all independent) partial ordering axioms for $\{=, \leq, <\}$ are replaced by the irreflexivity rule $r_i$ and the transitivity rule $r_t$ (for $<$), by an expansion rule, and by an extension to hyper-resolution, then refutation completeness is preserved.**

**2. If only the connectivity axiom, $\{x < y \vee y \leq x\}$, is retained from the five total ordering axioms for $\{=, \leq, <\}$ and if the other four are replaced by $r_i$, $r_t$, and an antisymmetry rule, refutation completeness is preserved.**

**A program using total ordering inference rules is then described. Differences between the rules as presented in the theoretical development and as implemented in the program are noted. The paper concludes with a discussion of the program's successful performance on a large collection of problems taken from published papers.**

(Received April 1973)

## 1. Introduction

The purposes of programming a computer to prove theorems concern Artificial Intelligence (Slagle, 1971), deduction, mathematics, and mathematical logic. See (Slagle, 1972) for a discussion. The resolution principle (Robinson, 1965b) is an inference rule used in automated theorem-proving. L. Wos *et al.* (1964; 1965), J. Allen and D. Luckham (1970), and others have written proof-finding programs embodying the resolution principle. Although quite general, these programs have been so slow that they have proved only a few theorems of any interest.

As a step in coping with this problem, a previous paper (Slagle, 1972) presented a fairly general approach which, when given the axioms of some special theory, often yields complete, valid, efficient (in time) rules corresponding to some or all of the given axioms. In that paper, the approach was applied to several theories, including the theories of equality, partial ordering, and sets. The theories of equality and partial ordering were built into a computer program, and the experimental results were favourable (Slagle and Norton, 1973). In the present paper, the approach is applied to partial ordering again and to total ordering. For partial ordering, the result obtained is Theorem 1, which is an improvement over Theorem 7 in Slagle (1972). Theorem 1 can be extended to set theory, etc. just as Theorem 7 in Slagle (1972) was.

A new application of the approach is to 'build in' the theory of total ordering. Total ordering is important because it is so frequently found in other important theories, for example, number theory. The result obtained (Theorem 2) turned out to be considerably simpler than we had expected beforehand. This result suggested modifications and additions to make the partial ordering program of Slagle and Norton (1973), into a total ordering program. The program departs from the theory in several significant ways, illustrating issues which arise at the time of implementation. The new program has performed successfully on a large collection of problems involving partial and total ordering.

Much of the theoretical groundwork for this paper has been presented in Slagle (1972), and the total ordering program is similar in many respects to the partial ordering program reported in Slagle and Norton (1973). It would require a large amount of repetition of the material in Slagle (1972) and Slagle and Norton (1973) to make this paper self-contained. We therefore refer freely to these earlier papers, and assume familiarity on the reader's part with them.

**Table 1** is a key to symbols used in this paper. The table is binding in the sense that, for example, when we use $n$, we implicitly mean a nonnegative integer even though this is not said explicitly. 'Rule' and 'variable' are abbreviations for 'inference rule' and 'individual variable'.

## 2. Brief review of hyper-resolution and paramodulation

We start with a vocabulary of (individual) variables, function symbols, and predicate symbols. Terms, atoms, and literals are introduced next. (See Robinson, 1965*b*; Robinson, 1967; Slagle, 1971 for a full description.) A *clause* is a finite disjunction of zero or more literals. When convenient, we regard a clause as the set of its literals. To facilitate matters, we take similar liberties with the nomenclature later, but what is meant will always be clear from the context. An *mgu* (most general unifier) $\mu$ for a set of expressions is a substitution with the property that, for any two members $e_1$ and $e_2$ of the set, $e_1\mu = e_2\mu$, and there is no more general substitution with this property‡. See Slagle (1972) for precise definitions of deduction, subsumption deduction, factoring, equality axioms, etc. It is assumed that factoring accompanies every rule used in the present paper. A *refutation* is a deduction of the empty (contradictory) clause, denoted by *false*. In what follows, we assume the presence of a literal ordering $L$ (Slagle, 1972) but seldom mention $L$ explicitly. Thus, we shall be implicitly dealing with factored $L$-electrons, $L - r_h$ ($L$-hyper-resolution), $L - r_p$ ($L$-paramodulation), $L - r_i$ ($L$-irreflexivity rule), etc.

*Definitions*

A clause $E' = \{a \vee D\}$ is a *factored electron* of an *electron*

*Computer Science Laboratory, Communications Sciences Division, Naval Research Laboratory, Washington, D.C. 20375, USA.

†Data Management Branch, DCRT, NIH.

‡As in this definition, the symbol = with surrounding spaces will be used for the equality (identity) of two expressions. Without surrounding spaces, it will be used as the equality predicate symbol in an atom.

**Table 1 Key to symbols**

| *Symbol* | *Meaning(s)* |
|---|---|
| $a$ | atom, positive literal, antisymmetry |
| $d$ | deduction |
| $e$ | expression, expansion |
| $f$ | function |
| $h$ | hyper-resolution |
| $i$ | irreflexivity, index for integer |
| $k$ | literal |
| $n$ | nonnegative integer |
| $p$ | paramodulation, positive integer |
| $q$ | integer |
| $r$ | (inference) rule |
| $s$ | term |
| $t$ | term, transitivity |
| $u$ | term |
| $x$ | (individual) variable |
| $y$ | variable |
| $z$ | variable |
| $A$ | function |
| $B$ | set of binary predicates |
| $C$ | clause |
| $D$ | clause, disjunction of literals |
| $E$ | electron, clause, disjunction of literals |
| $L$ | literal ordering |
| $N$ | nucleus |
| $P$ | set of predicate symbols |
| $Q$ | countable set of clauses |
| $R$ | set of rules |
| $S$ | countable set of clauses |
| $T$ | countable set of clauses |
| $\theta$ | substitution |
| $\mu$ | most general unifier |
| $\sigma$ | most general simultaneous unifier |
| $\vee$ | inclusive or |
| $\sim$ | not |
| $\leq$ | less than or equal |
| $<$ | less than |
| $\subseteq$ | contained in or equal |
| $\in$ | is a member of |
| $\cup$ | union |
| iff | if and only if |
| $\vdash$ | yields |

$E = \{D_1 \vee D_2\}$ where $D_1$ is a disjunction of one or more positive literals and where $D_2$ is a disjunction of zero or more positive literals, if there is an mgu (most general unifier) $\mu$ for $D_1$ such that

(*a*) $a = D_1\mu$,
(*b*) $D = E\mu - a$.

The literal $a$ is the *distinguished literal* (Kowalski and Hayes, 1969) of $E'$. In this paper, the use of $\{a \vee D\}$ implies that $a$ is not one of the literals in $D$. If $a$ might be in $D$, we shall use $\{a\} \cup D$. A similar convention is used throughout this paper.

*Definitions*
A clause $N' = \{\sim a_1 \vee \ldots \vee \sim a_p \vee D'\}$ is a *factored nucleus* of a *nucleus* $N = \{D_1 \vee \ldots \vee D_p \vee D\}$ where $D$ is a disjunction of zero or more positive literals if there is an mgsu (most general simultaneous unifier) $\sigma$ of $D_1, \ldots, D_p$ such that $D' = D\sigma$ and for $i = 1, \ldots, p$, $\sim a_i = D_i\sigma$. The literals $\sim a_1, \ldots, \sim a_p$ are the *distinguished literals* of $N'$. Note that a nucleus is a factored nucleus of itself.

The literals shown first in clauses are the distinguished literals. We assume that no variable in one clause occurs in another. We shall state our results in terms of hyper-resolution (Robinson, 1965*a*), though they can be extended to renamable clash (or even semantic) resolution.

*Definitions*
*Hyper-resolution* with respect to $S$ is the following rule, denoted by $r_h$. From the (nucleus $N \in S$ with) factored nucleus $N' = \{k_1 \vee \ldots \vee k_p \vee D\}$ and from the (electrons $E_i$ with) factored electrons $E_i' = \{a_i \vee D_i\}$ where

(*a*) there is an mgsu $\sigma$ such that $\sim a_i\sigma = k_i\sigma$ simultaneously for all $i$,
(*b*) no $a_i\sigma$ or $\sim a_i\sigma$ occurs in $N'\sigma$ except as $k_i\sigma$ (This and corresponding conditions are *clash conditions*.),
(*c*) no $a_i\sigma$ occurs in $E_1'\sigma, \ldots, E_p'\sigma$ except in $E_i'\sigma$ itself as $a_i\sigma$ (This is the *electron clash condition*, which holds for all rules in this paper and therefore will not be repeated for subsequent rules.),

infer the *hyper-resolvent* (*resolvent*) $D\sigma \cup D_1\sigma \cup \ldots \cup D_p\sigma$.

*Definitions*
A *hyper-deduction* is a deduction in which hyper-resolution is the only inference rule used. A *hyper-refutation* is a hyper-deduction of *false*.

*Definition*
The *set of* $\{=\}$*-reflexive axioms* for $S$ consists of $\{x=x\}$ and the functionally reflexive axioms for $S$.

We shall write $e[t]$ to indicate that a term $t$ occurs in a particular position in $e$. Later in the same discussion, $e[s]$ will mean the result obtained by replacing that particular occurrence of $t$ in $e$ by $s$.

In this paper, unlike Slagle (1972), 'two-way' paramodulation, as defined below, is used. The following definition is an extension of that given by G. Robinson and L. Wos (1969). Note that both premises for paramodulation are factored electrons and therefore consist of only positive literals.

*Definitions*
*Paramodulation*, denoted by $r_p$, is the following rule. From the factored electrons $\{s=t \text{ (or } t=s) \vee D_1\}$ and $\{a[u] \vee D_2\}$ where

(*a*) there is an mgu $\mu$ such that $s\mu = u\mu$,
(*b*) $s\mu$ is not $t\mu$,
(*c*) $a[t]\mu$ is not $(s=t)\mu$ (clash condition),

infer the *paramodulant* $\{a[t]\mu\} \cup D_1\mu \cup D_2\mu$.

## 3. Partial ordering and total ordering

In this section, we state the partial and total ordering axioms we shall use, two lemmas, the definition of a $P$-ordering, and two rules common to the partial and total ordering completeness theorems. The proofs of the lemmas are straightforward and are not given. Lemma A allows us to dispense with three axioms (and therefore some rule complications) used in Slagle (1972). Lemma B simply shows that every total ordering as given by our axiomatisation, is a partial ordering. For partial ordering, we used the symbols $\subseteq$ and $\subset$ in Slagle (1972), since we subsequently discussed set theory. We shall use $\leq$ and $<$ here, since we shall subsequently discuss the theory of total ordering.

*Definitions*
The *set of partial ordering axioms for* $\{=, \leq, <\}$ with respect to $S$ consists of $\{x \leq x\}$ (*reflexivity axiom*), $\{\sim(x < x)\}$ (*irreflexivity axiom*), $\{\sim(x < y) \vee \sim(y < z) \vee x < z\}$ (*transitivity axiom*), $\{\sim(x \leq y) \vee x < y \vee x=y\}$ (*expansion axiom*), $\{\sim(x < y) \vee x \leq y\}$ (*less axiom*), and the equality axioms for both $S$ and the above axioms.

*Definitions*
The *set of total ordering axioms for* $\{=, \leq, <\}$ with respect to $S$ consists of $\{x \leq x\}$, $\{\sim(x < x)\}$, $\{\sim(x < y) \vee \sim(y < z) \vee x < z\}$, $\{\sim(x \leq y) \vee \sim(y \leq x) \vee x = y\}$ (*antisymmetry axiom*), $\{x < y \vee y \leq x\}$ (*connectivity axiom*), and the equality axioms for both $S$ and the above axioms.

*Lemma A:*
*The conjunction of the transitivity axiom, the expansion axiom, the less axiom, and the equality axioms for the above axioms has the following valid consequences:*

(*a*) $\{\sim(x < y) \vee \sim(y \leq z) \vee x < z\}$
(*b*) $\{\sim(x \leq y) \vee \sim(y < z) \vee x < z\}$
(*c*) $\{\sim(x \leq y) \vee \sim(y \leq z) \vee x \leq z\}$

*Lemma B:*
(*a*) *The expansion and less axioms are valid consequences of the conjunction of the irreflexivity, transitivity, antisymmetry, and connectivity axioms.*

(*b*) *Therefore, every partial ordering axiom for* $\{=, \leq, <\}$ *is a valid consequence of the set of total ordering axioms for* $\{=, \leq, <\}$.

*Definition*
Let $P$ be a set of predicate symbols. A *P-ordering* is a literal ordering (Slagle, 1972) in which every member of $P$ has equal rank.

*Definition*
The *irreflexivity rule* (for $<$), denoted by $r_i$, is the following. From the factored electron $E' = \{s < t \vee D\}$ where

(*a*) $E'$ is not the connectivity axiom,
(*b*) there is an mgu $\mu$ such that $s\mu = t\mu$,

infer $D\mu$.

*Definition*
The *transitivity rule* (for $<$), denoted by $r_t$, is the following. From the factored electrons $\{s < t_1 \vee D_1\}$ and $\{t_2 < u \vee D_2\}$ where there is an mgu $\mu$ such that

(*a*) $t_1\mu = t_2\mu = t$,
(*b*) neither $s\mu$ nor $u\mu$ is $t$,

infer $\{s\mu < u\mu\} \cup D_1\mu \cup D_2\mu$.

## 4. Refutation completeness for the theory of partial ordering

For partial ordering, we extend the definitions of hyper-resolution, factoring, and subsumption, and we define the expansion rule. We define an irreflexive deduction for the completeness of total ordering and an extended irreflexive deduction for the completeness of partial ordering. Our completeness theorem (Theorem 1) for the theory of partial ordering states that, under appropriate conditions, there is an extended irreflexive refutation.

*Definition*
*Extended hyper-resolution*, denoted by $r_{eh}$, yields everything yielded by $r_h$ and, in addition, the distinguished literal $s_1 < t_1$ or $s_1 = t_1$ or $t_1 = s_1$ in a factored electron may be unified with a distinguished literal $\sim(s_2 \leq t_2)$ in a factored nucleus just as if the distinguished literal in the factored electron were $s_1 \leq t_1$.

*Definition*
The *expansion rule*, denoted by $r_e$, is the following. From the factored electron $\{s \leq t \vee D\}$, infer $\{s < t \vee s = t\} \cup D$.

For the deduction by $\{r_{eh}, r_p, r_i, r_t, r_e\}$ mentioned in Lemma C and Theorem 1, factoring is extended so that the predicate symbols $<$ and $\leq$ as well as $=$ and $\leq$ may be combined in the obvious way. For example, a positive literal containing $<$ may be unified with a positive literal containing $\leq$ just as if the $<$ were $\leq$. Similarly, a negative literal containing $\leq$ may be unified with a negative literal containing $<$ just as if the $\leq$ were $<$.

*Definition*
An (*extended*) *irreflexive deduction d* is a deduction having the following property. If $E'$ is a factored electron in $d$ and if $E''$ is the instance of $E'$ used as a premise in an inference in $d$,

(*a*) $\{x \leq x\}$ does not subsume $E''$,
(*b*) if $\{x = x\}$ subsumes $E''$, then either $E'$ is $\{x = x\}$ and the inference is an (extended) hyper-resolution or $E'$ is a functionally reflexive axiom and the inference is a paramodulation into $E'$.

*Definition*
*C less-equal-subsumes E* if some clause obtained from $C$ by replacing zero or more literals $s_i < t_i$, $s_i = t_i$, and $t_i = s_i$ by the corresponding literals $s_i \leq t_i$ subsumes $E$.

The proof of the following lemma is similar in spirit though simpler than the lemma preceding Theorem 7 in Slagle (1972). Theorem 1 is proved from Lemma C in much the same way as Theorem 7 is proved in Slagle (1972). Note that, unlike in Slagle (1972), the reflexivity axiom $\{x \leq x\}$ is not in $T$ in Lemma C and Theorem 1. Recall that an electron consists of one or more positive literals.

*Lemma C:*
*Let B be* $\{=, \leq, <\}$. *Let T be the set of* $\{=\}$*-reflexive axioms for S. Let Q be the set of partial ordering axioms for B. Let L be a B-ordering. If there is a subsumption L-hyper-deduction from* $S \cup Q$ *of an electron* (*or the empty clause*) *E, there is an extended irreflexive deduction from* $S \cup T$ *by*

$$R = \{L - r_{eh}, L - r_p, L - r_i, L - r_t, L - r_e\}$$

*of some clause C which less-equal-subsumes E.*

*Theorem* 1:
*Let B be* $\{=, \leq, <\}$. *Let T be the set of* $\{=\}$*-reflexive axioms for S. Let Q be the set of partial ordering axioms for B. Let L be a B-ordering. S is Q-unsatisfiable iff there is an extended irreflexive refutation from* $S \cup T$ *by* $\{L - r_{eh}, L - r_p, L - r_i, L - r_t, L - r_e\}$.

## 5. Refutation completeness for the theory of total ordering

For total ordering, we state the antisymmetry rule. Next we extend the definitions of factoring and subsumption in a different way than we did for partial ordering. We then prove our main result, Theorem 2. This completeness theorem for the theory of total ordering states that, under appropriate conditions, there is an irreflexive refutation.

*Definition*
The *antisymmetry rule*, denoted by $r_a$, is the following. From the factored electrons $\{s_1 \leq t_1 \vee D_1\}$ and $\{t_2 \leq s_2 \vee D_2\}$ where there is an mgsu $\sigma$ such that $s_1\sigma = s_2\sigma = s$ and $t_1\sigma = t_2\sigma = t$, infer $\{s = t\} \cup D_1\sigma \cup D_2\sigma$.

For the irreflexive deduction mentioned in Lemma D and Theorem 2, factoring is extended so that the predicate symbols $=$ and $\leq$ may be combined in the obvious way. For example, a positive literal containing $=$ may be unified with a positive literal containing $\leq$ just as if the $=$ were $\leq$. Unlike the partial ordering case, the predicate symbols $<$ and $\leq$ need never be combined, roughly because the less axiom is not in the set of total ordering axioms. *C equal-subsumes E* if some clause obtained from $C$ by replacing zero or more literals $s_i = t_i$ and $t_i = s_i$ by the corresponding literals $s_i \leq t_i$ subsumes $E$. $S$ is a *set of clauses without negative inequalities* if no literal of the form $\sim(s < t)$ nor of the form $\sim(s \leq t)$ occurs in $S$. Note that any set of clauses can be transformed into a logically equivalent (in the presence of the set of total ordering axioms) set of clauses without negative inequalities. To do this, the connectivity axiom is used to transform $\sim(s \leq t)$ into $t < s$ and $\sim(s < t)$ into $t \leq s$. We now prove the following lemma, whose proof is similar to that of Lemma C. The proof of Theorem 2 then follows from Lemma D in much the same way as Theorem 1 is proved from Lemma C.

*Lemma D:*
*Let $T$ consist of the connectivity axiom and the $\{=\}$-reflexive axioms for a set $S$ of clauses without negative inequalities. Let $Q$ be the set of total ordering axioms for $\{=, \leq, <\}$. Let $L$ be a $\{=, \leq\}$-ordering. If there is a subsumption $L$-hyper-deduction from $S \cup Q$ of an electron (or the empty clause) $E$, there is an irreflexive deduction from $S \cup T$ by $R = \{L - r_h, L - r_p, L - r_i, L - r_t, L - r_a\}$ of some clause $C$ which equal-subsumes $E$.*

*Proof:*
The proof is by induction on the depth $n$ of the given deduction. The case when $n$ is zero is trivial. The only point worth noting is that, if the given deduction consists of $\{x \leq x\}$, the irreflexive deduction $d$ consists of $\{x=x\}$. Assume that the lemma is true for $n = 0, \ldots, q$. Let the deduction have depth $q + 1$. Let the final inference be $E_1, \ldots, E_p, N \vdash E$ where $E'_1, \ldots, E'_p$ are the corresponding factored electrons. Let $E'_i = \{k_i \vee D_i\}$. If $E'_i\sigma$ is the instance used in the inference and if $\{x=x\}$ or $\{x \leq x\}$ subsumes $D_i\sigma$, which is contained in $E$, we let $d$ consist of $\{x=x\}$, which equal-subsumes $E$. (The cases when $\{x=x\}$ or $\{x \leq x\}$ subsumes $k_i\sigma$ are handled later.)

Except in the above case, we use the inductive hypothesis. For each $i = 1, \ldots, p$, there is an irreflexive deduction from $S \cup T$ by $R$ of some clause $C_i$ which equal-subsumes $E_i$. If some $C_i$ equal-subsumes $E$, we are done. If there is no such $C_i$, there is at least one literal in each $C_i$ corresponding to the distinguished literal in $E'_i$. Hence, for each $i$ there is a factored electron $C'_i$ corresponding to $E'_i$ such that $C'_i$ equal-subsumes $E'_i$ and such that the distinguished literal in $C'_i$ is the only literal in $C'_i$ corresponding to the distinguished literal in $E'_i$. Here, as elsewhere, the $\{=, \leq\}$-ordering $L$ presents no difficulties and so will not be mentioned again in this proof. The remainder of the proof is divided into cases depending on the nucleus $N$. The clash and the electron clash conditions are easily verified in all cases.

*Case* 1:
If $N$ is $\{\sim(x < x)\}$, then $N'$ is $N$. The factored electron is $E'_1 = \{s < t \vee D\}$. The resolvent $E$ is $D\sigma$. If $E'_1$ were the connectivity axiom, $E$ would have been subsumed by $\{x \leq x\}$. If $E'_1$ is not the connectivity axiom, applying $r_i$ to $C'_1$ yields a clause which equal-subsumes $E$.

*Case* 2:
If $N$ is $\{\sim(x < y) \vee \sim(y < z) \vee x < z\}$, the treatment is similar to that of transitivity of $\leq$ in Theorem 6 in Slagle (1972).

*Case* 3:
If $N$ is $\{\sim(x \leq y) \vee \sim(y \leq x) \vee x=y\}$, then $N'$ is either $N$ or $\{\sim(x \leq x\} \vee x=x\}$. The latter case is impossible. The factored electrons are $E'_1 = \{s_1 \leq t_1 \vee D_1\}$ and $E'_2 = \{t_2 \leq s_2 \vee D_2\}$. The resolvent $E$ is $\{s=t\} \cup D_1\sigma \cup D_2\sigma$ where $s_1\sigma = s_2\sigma = s$ and $t_1\sigma = t_2\sigma = t$. Note that $s \neq t$, since otherwise $\{x=x\}$ would have subsumed $E$. If the predicate symbol in the distinguished literal in $C'_1$ or $C'_2$ is $=$, then $C'_1$ or $C'_2$ respectively would have equal-subsumed $E$. Hence, both predicate symbols are $\leq$. Applying the antisymmetry rule $r_a$ to $C'_1$ and $C'_2$ yields a clause which equal-subsumes $E$.

*Case* 4:
If $N$ is $\{x_1 \neq x_0 \vee \sim(x_1 \leq x_2) \vee x_0 \leq x_2\}$, then $N'$ is $N$. The factored electrons are $E'_1 = \{s_1=s_0 \vee D_1\}$ and $E'_2 = \{t_1 \leq t_2 \vee D_2\}$. The resolvent $E$ is $\{s_0\sigma \leq t_2\sigma\} \cup D_1\sigma \cup D_2\sigma$. If $s_1\sigma = s_0\sigma$ or if $t_1\sigma = t_2\sigma$, then $C'_2$ or $C'_1$ respectively would have equal-subsumed $E$. Whether $t_1 \leq t_2$ or $t_1=t_2$ is an instance of the distinguished literal in $C'_2$, a paramodulant of $C'_1$ into $C'_2$ equal-subsumes $E$. The arguments given in the previous two sentences provide special treatment for the predicate symbol $\leq$, which cannot be treated like other predicate symbols as in Case 6 below.

*Case* 5:
If $N$ is $\{x_2 \neq x_0 \vee \sim(x_1 \leq x_2) \vee x_1 \leq x_0\}$, the treatment is similar to that in the previous case.

*Case* 6:
If $N$ is a member of $Q$ and has not been treated in a previous case, $N$ is an equality axiom, and the treatment is similar to that in Theorem 4 in Slagle (1972).

*Case* 7:
If $N$ is in $S$, the treatment is similar to that in Theorem 4 in Slagle (1972). This completes the proof.

*Theorem* 2:
*Let $T$ consist of the connectivity axiom and the $\{=\}$-reflexive axioms for a set $S$ of clauses without negative inequalities. Let $Q$ be the set of total ordering axioms for $\{=, \leq, <\}$. Let $L$ be a $\{=, \leq\}$-ordering. $S$ is $Q$-unsatisfiable iff there is an irreflexive refutation from $S \cup T$ by $\{L - r_h, L - r_p, L - r_i, L - r_t, L - r_a\}$.*

*Proof:*
Let $R$ be $\{L - r_h, L - r_p, L - r_i, L - r_t, L - r_a\}$.
A. Suppose there is an irreflexive refutation from $S \cup T$ by $R$. Since every member of $R$ is $Q$-valid, $S \cup T \cup Q$ is unsatisfiable. Since every member of $T$ is in $Q$ or is an instance of $\{x=x\}$, which is in $Q$, $S \cup Q$ is unsatisfiable; that is, $S$ is $Q$-unsatisfiable.
B. Suppose that $S$ is $Q$-unsatisfiable. By Theorem 2 in Slagle (1972), there is a subsumption $L$-hyper-refutation from $S \cup Q$. Hence, by Lemma D, there is an irreflexive deduction from $S \cup T$ by $R$ of some clause $C$ which equal-subsumes the empty clause, *false*. Since *false* is the only clause which equal-subsumes *false*, this deduction is a refutation.

## 6. A total ordering computer program

In Slagle and Norton (1973) we described a computer program which uses a modified version of the partial ordering rules described in Slagle (1972). The reader familiar with Slagle and Norton (1973) will observe that the program represents a modification of this paper's partial ordering rules as well. The program was used to prove a set of 32 problems adapted from Hoare (1971). Foley and Hoare, in (1971), continued the analysis of the Quicksort algorithm begun in Hoare (1971), and we were able to convert the additional analysis more or less directly into twelve new problems, five of which require the hypothesis of total ordering. Thus we obtained a set of 44 problems on which we could test the performance of a total ordering program.

The new program was created directly from the old one, rather than developed from the theoretical basis for dealing with total ordering by inference rules presented in this paper. The changes necessary to the old program turned out to be minimal. We simply replaced the treatment of the expansion axiom with a restricted use of the connectivity axiom. Recall that the partial ordering program had an expansion rule allowing the inference of $\{t_1 < t_2 \vee t_1=t_2 \vee E\}$ from a clause $\{t_1 \leq t_2 \vee E\}$ (if $t_1 \neq t_2$). The new clause was allowed to participate *only* in resolution steps (on the literal $t_1 < t_2$, since all steps were

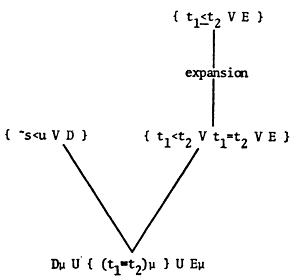


Fig. 1*a*

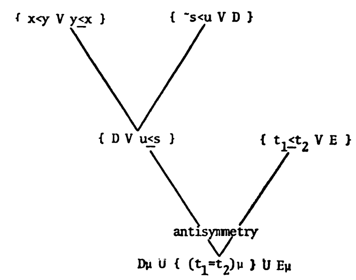


Fig. 1*b*

restricted to first literals of clauses). The total ordering program dispenses with the expansion rule. Instead, the connectivity axiom $\{x < y \vee y \leq x\}$ is automatically included in the set of input clauses for a problem, but it is restricted to participation in resolution steps only, in the same way that clauses inferred by the expansion rule were restricted in the old program. Thus the connectivity axiom does not ever become a premise for the transitivity rule or the antisymmetry rule, which in our program are combined into a single rule (which also carries out transitivity for both $<$ and $\leq$; i.e. embodies the clauses of Lemma A).

In view of Lemma B it should not be surprising that the total ordering program dispenses with expansion. Unfortunately, the discrepancies between theory and implementation make us unable to prove that the program is as 'strong' as the old one. That is, we would like to demonstrate that any inference made by the old program which involved the expansion rule could be replaced by a chain of inferences using only rules in the present program. However, consider the following 'heuristic justification'.

Any use of expansion by the old program was restricted to be of the form shown in **Fig. 1**(*a*). Here we assume $s < u$ is unifiable with $t_1 < t_2$ by $\mu$. $D$ can be assumed to contain only positive literals. In **Fig. 1**(*b*) we show a derivation of the same result from the same input clauses, using the connectivity axiom and the antisymmetry rule. The program cannot quite perform this chain of inferences, due to the restriction of the first literal strategy. Thus our argument is only suggestive, and we must fall back upon the observation that the first literal strategy, though incomplete, has never presented any difficulties for our purposes; i.e. all of the theorems we have been considering have proofs compatible with this restriction.

We added one heuristic to the total ordering program. The user may specify function symbols which may not be nested. At the same time that each literal of a new clause is checked to see that it does not exceed the literal length bound, it is also checked to see that it does not contain any forbidden nestings.

We now contrast our program with the theoretical development presented in the first part of this paper. We will mention only the major differences. Finer details can be inferred from Slagle and Norton (1973) and the above description.

The program implements hyper-resolution by sequences of $P1$-deductions. It augments this by allowing IMPRES (IMPlication RESolution) steps, which permit inferences *via* resolution from input clauses of the form $\{\sim a_1 \vee \ldots \vee \sim a_n \vee a_{n+1}\}$ and arbitrary unit nuclei which unify with $a_{n+1}$. If any of the literals $\sim a_i\sigma$ in the resolvent are complements of existing electrons, they are immediately deleted from the resolvent. IMPRES steps add an element of 'working backward' to the 'working forward' character of proofs produced using hyper-resolution.

Paramodulation is used in the program without the presence of functionally reflexive axioms. This necessitates paramodulation into nuclei as well as into electrons. However, the program prohibits paramodulation into free variables. This last point eliminates the justification for the irreflexivity rule. We can simply use the axiom $\{\sim x < x\}$ in all problems.

Other differences particular to total ordering are the already mentioned rule combining transitivity and antisymmetry, and the use of the connectivity axiom only in resolution steps. We include the axiom $\{x \leq x\}$ in all problems when using the program, and use a literal evaluation scheme based on implicit subsumptions and resolutions with $\{x = x\}$, $\{x \leq x\}$, and $\{\sim x < x\}$. When formulating problems we do not convert to clauses without negative inequalities. We did, however, alter the formulation of the axioms assumed for the 'plus one' and 'minus one' functions.

The program uses general subsumption to discard newly generated clauses which are subsumed by existing clauses. This is more restrictive than the property of being an irreflexive deduction. Finally, and most important, the program uses the first literal strategy for all rules (except IMPRES), even though it is known to be incomplete.

## 7. Experimental results

As an example of the program's performance, we present the proof obtained (in 13 seconds) for the final lemma corresponding to the analysis of Foley and Hoare (1971). This may be axiomatised as:

1. $\{j < i\}$
2. $\{m \leq p\}$
3. $\{p \leq q\}$
4. $\{q \leq n\}$
5. $\{\sim m \leq x \vee \sim x < i \vee \sim j < y \vee \sim y \leq n \vee A(x) \leq A(y)\}$
6. $\{\sim m \leq x \vee \sim x \leq y \vee \sim y \leq j \vee A(x) \leq A(y)\}$
7. $\{\sim i \leq x \vee \sim x \leq y \vee \sim y \leq n \vee A(x) \leq A(y)\}$
8. $\{\sim A(p) \leq A(q)\}$

(The notation reflects that of Foley and Hoare (1971).)
In addition, we have available $\{x = x\}$, $\{x \leq x\}$, and $\{x < y \vee y \leq x\}$, the connectivity axiom. The proof begins with three IMPRES steps:

| | |
|---|---|
| 9. $\{\sim q \leq j\}$ | 6 IMPRES 8, followed by 2 and 3 |
| 10. $\{\sim i \leq p\}$ | 7 IMPRES 8, followed by 3 and 4 |
| 11. $\{\sim p < i \vee \sim j < q\}$ | 5 IMPRES 8, followed by 2 and 4 |

The connectivity axiom is then used twice to obtain

| | |
|---|---|
| 12. $\{\sim j < q \vee i \leq p\}$ | (the intermediate $P1$-resolvent) |
| 13. $\{i \leq p \vee q \leq j\}$ | |

Clause 13 resolves with clause 10, and the result with clause 9 to complete the proof.

It is reasonable to expect that a total ordering program would take longer to prove partial ordering problems, since the introduction of connectivity allows extra inferences. In particular, the number of these extra inferences will differ for our program depending on which literal is placed first in the connectivity axiom. For a large and diverse collection of problems, the difference may not be as pronounced, but for our problem set the choice was clear. In fact, the new program took significantly longer than the partial ordering program on only one of the 39 partial ordering problems when we entered the connectivity axiom as $\{x < y \vee y \leq x\}$. This problem was one of the new ones adapted from Foley and Hoare (1971), and the increase was from 28 to 91 seconds of cpu time. For the other problems, the times were comparable; in particular, the original 32 problems of Slagle and Norton (1973) were proved in approximately the same amount of time as before. Of course the new program could also prove the 5 problems requiring total ordering. Each of two of these required as much effort as the previous hardest problem (around 6 minutes of cpu time to generate proofs using 27 clauses derived from 15 input clauses), and also for these we needed the new heuristic restricting function nesting. The entire set of 44 problems when proved using the total ordering program required about 35 minutes of cpu time.

It should be emphasised that the total ordering inference rules as used in the program are not merely equivalent to the use of the corresponding chains of resolutions and paramodulations involving the corresponding total ordering axioms. The absence of the axioms prevents their appearance in a proliferation of other, unneeded, resolution and paramodulation inferences. In addition, the intermediate clauses of the chains of inferences are not present to participate in yet more excess inferences. The theoretical development in the first part of this paper reveals that only a fraction of the resolution and paramodulation steps involving some of the total ordering axioms are necessary to preserve completeness. The rules have the effect of 'compiling' the restrictions thus allowed into valid procedures which are efficient in time.

In Slagle and Norton (1973) we related how it was necessary to treat partial ordering by rules. Using partial ordering axioms, only the simplest of the examples could be proved, given a time limit of 400 seconds per problem. This was true even though we used all the heuristics which were not dependent on the partial ordering rules, in particular the first literal strategy (but not the IMPRES mechanism).

For the case of total ordering, we tried to use the partial ordering program as described in Slagle and Norton (1973), simply including the connectivity axiom with the statement of each problem. Thus we were using partial ordering rules, retaining all heuristics, but treating the extension to total ordering by axiom. The example at the beginning of this section, which is the simplest of the five total ordering problems, could not be proved in 400 seconds. What happened was that the connectivity axiom participated in transitivity steps involving axioms 1 through 4, and the clauses produced by these inferences participated in further unnecessary transitivity steps, etc. Once the connectivity axiom was restricted to participate only in resolution steps, i.e. once we used the total ordering rule, the situation was brought under control, and the proof produced in 19 seconds. Deleting the expansion rule, once we observed that it apparently was superfluous, further reduced the time to 13 seconds.

**References**

ALLEN, J., and LUCKHAM, D. (1970). An Interactive Theorem-Proving Program, *Machine Intelligence* 5, ed. by B. Meltzer and D. Michie, American Elsevier, New York, pp. 321-336.

FOLEY, M., and HOARE, C. A. R. (1971). Proof of a Recursive Program: Quicksort, *The Computer Journal*, Vol. 14, No. 3, pp. 391-395.

HOARE, C. A. R. (1971). Proof of a Program: FIND, *CACM*, Vol. 14, pp. 39-45.

KOWALSKI, R., and HAYES, P. J. (1969). Semantic Trees in Automatic Theorem-Proving. *Machine Intelligence* 4, ed. by B. Meltzer and D. Michie, American Elsevier, New York, pp. 87-101.

ROBINSON, G., and WOS, L. (1969). Paramodulation and Theorem-Proving in First Order Theories with Equality, *Machine Intelligence* 4, ed. by B. Meltzer and D. Michie, American Elsevier, New York, pp. 135-150.

ROBINSON, J. A. (1965*a*). Automatic Deduction with Hyper-Resolution. *Int. J. of Comp. Math.*, Vol. 1, pp. 227-234.

ROBINSON, J. A. (1965*b*). A Machine Oriented Logic Based on the Resolution Principle. *JACM*, Vol. 12, pp. 23-41.

ROBINSON, J. A. (1967). A Review of Automatic Theorem-Proving, *Proc. of the Symposia in Applied Mathematics*, Vol. 19, Providence, R.I., American Mathematical Society, pp. 1-18.

SLAGLE, J. (1971). *Artificial Intelligence: The Heuristic Programming Approach*, McGraw-Hill Book Company, New York.

SLAGLE, J. (1972). Automatic Theorem Proving with Built-in Theories Including Equality, Partial Ordering and Sets, *JACM*, Vol. 19, pp. 120-135.

SLAGLE, J., and NORTON, L. (1973). Experiments with an Automatic Theorem Prover Having Partial Ordering Rules, *CACM*, Vol. 16, pp. 682-688.

WOS, L., CARSON, D., and ROBINSON, G. (1964). The Unit Preference Strategy in Theorem Proving, *Proc. of the* 1964 *Fall Joint Computer Conference*, pp. 615-621.

WOS, L., ROBINSON, G., and CARSON, D. (1965). Efficiency and Completeness of the Set of Support Strategy in Theorem Proving, *JACM*, Vol. 12, pp. 536-541.

# Book review

*An Introduction to Digital Logic*, by A. Potton, 1973; 144 pages. (*Macmillan*, £2·40.)

At first sight this looks to be a very useful introductory book, especially since a glance through the contents reveals chapters on combinatorial logic, boolean algebra, Karnaugh maps, bistable systems, counters and registers, binary arithmetic operations and practical logic design considerations. The book is aimed at readers with no previous logic design or electronics experience.

Although such basic material is presented, the criticisms are that for a novice, the presentation appears dry, with a number of terms undefined and a few references to transistors which add nothing to the reader's understanding. An introductory chapter outlining the goals of logic design and the history and variety of logic elements would help, since it is not until Chapter 8 that real-life problems are introduced which give a feel for the subject. Further, a glossary should have been included in a book such as this, which is aimed at physicists, electrical engineers and computer scientists. Finally, the sections on transistor logic in the early chapters should have been put in an appendix together with a more complete description of their operation, since they added little to the understanding of principles.

Because there are a number of gaps in the subjects covered, (Gray code and race hazards are only mentioned in passing), I feel this book would only be useful as an adjunct to a lecture course and would not be suitable for self-study by novices.

S. R. WILBUR (London)