ISSN 0105-8517

Literal Resolution: A Simple Proof of Resolution Completeness

Guo Qiang Zhang

DAIMI PB – 277
April 1989

AARHUS UNIVERSITY
COMPUTER SCIENCE DEPARTMENT
Ny Munkegade 116 – DK 8000 Aarhus C – DENMARK
Telephone: + 45 6 12 71 88 *Telex: 64767 aausci dk*

Literal Resolution: A Simple Proof of Resolution Completeness

Guo Qiang Zhang
Department of Computer Science
Aarhus University
Ny Munkegade 116, 8000 Aarhus, Denmark

Literal resolution (LR) is a new resolution strategy for propositional calculus. Each step of LR involves a literal A. At an A-step of LR, the old clause set S is replaced by a new clause set S' consisting of all the resolvents from S which involve A, together with those clauses of S which do not contain A or its negation. LR repeatedly formulates new clause sets in this way and future resolution does not need any old clause. LR is sound and complete. It is conceptually simple and easy to understand, and it provides an intuitive and straightforward proof for the completeness of the propositional version of Robinson resolution.

1. Introduction

The resolution principle (Robinson resolution) discovered by Robinson [Ro65] is a cornerstone for mechanical theorem proving and problem solving. Robinson resolution has become so popular a theory that it has gone into text books for undergraduate teaching in computer science. This does not imply, however, that the presentation of the theory has left no room for improvements. In fact, it is the purpose of this note to give a proof for the completeness of Robinson resolution for propositional calculus based on a very simple idea of ***literal resolution***, which is different from known resolution strategies such as linear resolution [Lo70], unit preference resolution [WoRo64], semantics trees [KoHa69], etc.

The completeness of Robinson resolution relies on the expressiveness of Herbrand's universe, the compactness of propositional calculus and a key fact that the propositional version of Robinson resolution is complete.

This version of Robinson resolution is described as follows. For any finite clause set T of atomic propositions (i.e. propositional variables), define

$$\mathcal{R}(T) = T \cup \{ C \mid C \text{ is a resolvent of two clauses in } T \}.$$

Now, given a clause set S, let

$$\mathcal{R}^0(S) = S$$

$$\mathcal{R}^{i+1}(S) = \mathcal{R}(\mathcal{R}^i(S)) \text{ for each } i \geq 0$$

and let $\mathcal{R}^*(S) = \bigcup_{i\geq 0} \mathcal{R}^i(S)$. Because there can only be finitely many resolvents, $\mathcal{R}^*(S) = \mathcal{R}^n(S)$ for some n. The proof of the completeness of the resolution goes by showing that S is logically equivalent to $\mathcal{R}^n(S)$ so the inconsistency of S will be detected by finding a $\Box$ (the empty clause) in $\mathcal{R}^n(S)$. But the strategy is not very intuitive and the proof is not so straightforward.

Literal resolution (LR) uses the idea of a transition system. Each step of LR can be seen as a transition labelled with a literal A. At an A-transition of LR, written $S \xrightarrow{A} S'$, there is at least one resolution from S involving A and the old clause set S is replaced by a new clause set S' consisting of all the resolvents from S which involve A, together with those clauses of S which do not contain A or its negation. LR repeatedly formulates new clause sets in this way, resulting in a sequence of transitions like

$$S_0 \xrightarrow{A_0} S_1 \xrightarrow{A_1} S_2 \xrightarrow{A_2} \cdots S_k \xrightarrow{A_k} S_{k+1} \xrightarrow{A_{k+1}} \cdots.$$

Future resolution does not need any old clause. In other words, when $S \xrightarrow{A} S'$, the satisfiability of S is the same as that of S'. Note this is weaker than requiring S and S' to be logically equivalent. In fact from $S \xrightarrow{A} S'$ we can not deduce that S and S' are logically equivalent (Example 1).

LR is sound and complete. It is sound in the sense that whenever there is a clause set in the transition which contains an empty clause, the original clause set must be inconsistent; It is complete because if the original clause set is inconsistent, we will be able to detect it by deriving an empty clause from the transition. Since literal resolution is sound and complete, it is as powerful as Robinson resolution. Literal resolution enjoys, however, the following advantages.

In terms of the number of transitions, LR clearly terminates in m steps, where m is the number of propositional variables in the input clause set S (Note this does not imply that LR is polynomial as the size of the clause set may grow exponentially. In fact it is unlikely that there is a polynomial time algorithm for any resolution because the 3-satisfiability problem is well-known to be NP-complete).

LR abandons old clause set. After an A-transition like $S \xrightarrow{A} S'$, clauses in S which contain A or $\bar{A}$ do not appear in S', and S will never be needed in the future, compared with Robinson resolution where all old clause sets remain active (i.e. they may be needed in the future) and the number of clauses never decrease.

More heuristic information is provided by LR. For LR, there are no two different transitions with the same label A. This means we have done, once and for all, all resolutions which involve A. For Robinson resolution, it does not say when we do not need resolutions which involve a particular literal.

Based on the above three, we argue that LR is conceptually simpler and easier to understand, resulting in a more intuitive and straightforward proof of the completeness of the propositional version of Robinson resolution.

2. Literal Resolution

Let $\mathcal{A} = \{ \top, A_0, A_1, \cdots \}$ be a countable set of propositional variables (or atomic propositions) where $\top$ stands for 'true'. We assume reader's familiarity with propositional calculus and follow standard notations. So $\wedge$, $\vee$, $\neg$, etc., are propositional connectives. A ***literal*** is a member of $\mathcal{A} \cup \bar{\mathcal{A}}$, where $\bar{\mathcal{A}} = \{ \bot, \neg A_0, \neg A_1, \cdots \}$ with $\bot = \neg\top$, standing for 'false'. Let A, B range over literals. Write $\bar{A}$ for the negation of A, which is always in $\mathcal{A} \cup \bar{\mathcal{A}}$ by the axiom $\neg\neg A \Leftrightarrow A$. A ***clause*** is a finite subset of literals. Let C range over clauses. The ***empty*** clause is written as $\Box$.

A ***truth assignment*** Φ is a function from $\mathcal{A}$ to $\{ \bot, \top \}$. Truth assignment extends to all literals by assigning $\neg\Phi(A)$ to $\Phi(\neg A)$. We write $\Phi[\bot/A]$ for the truth assignment which is the same as Φ except that at A it is assigned $\bot$ and similarly $\Phi[\top/A]$ is a truth assignment which is the same as Φ except that at A it is assigned $\top$. A clause set S is ***valid*** at Φ, written $\Phi \models S$, if for any clause C in S, $\top \in \{ \Phi(A) \mid A \in C \}$. Otherwise we say S is ***invalid*** at Φ. S is ***satisfiable*** if it has a valid truth assignment. Otherwise S is ***unsatisfiable.*** An unsatisfiable clause is also called ***inconsistent.*** S is ***equivalent*** to S' if they have the same set of valid truth assignments.

Each proposition has a conjunctive normal form, which is written as a clause set. Robinson introduced a single rule for clause sets called ***resolution rule***, written as

$$\frac{C_0 \cup \{ A \} \qquad C_1 \cup \{ \bar{A} \}}{C_0 \cup C_1},$$

where C_0 and C_1 are clauses and A is a literal, with $\bar{A}$ standing for its negation. $C_0 \cup C_1$ is called the ***resolvent*** of $C_0 \cup \{ A \}$ and $C_1 \cup \{ \bar{A} \}$. To be more precise we call an application of the resolution rule an A-***resolution.*** when the literals involved are A and $\bar{A}$.

Definition $S \xrightarrow{A} S'$ is a ***literal transition*** from a clause set S to a clause set S' labelled with literal A if

1. there exist C, C' in S such that $A \in C$, $\bar{A} \in C'$;

2. $S' = \{C \mid C \text{ is a resolvent of an } A\text{-resolution of two clauses in } S\}$

   $\cup \{C' \mid C' \in S \text{ and } \{A, \bar{A}\} \cap C' = \emptyset\}$

To be definite, we always label a literal transition $S \xrightarrow{A} S'$ with $A \in \mathcal{A}$. Call $S \xrightarrow{A} S'$ an A-transition. Sometimes we omit the label and write $S \longrightarrow S'$, to mean that there is a literal transition from S to S'.

Example 1

$$\{ \{ A, B \}, \{ \neg B \} \} \xrightarrow{B} \{ \{ A \} \}$$

is a B-transition.

By ***literal resolution*** (LR) we mean the following algorithm based on literal transitions.

Literal Resolution

Step 1. Input a finite clause set S_0.

Step 2. For any literal A, any clause C of S_0, if A and $\bar{A}$ both appear in C, replace that clause by $\{\top\}$.

Step 3. If $\square \in S_0$ or there is no resolvent from S_0, then stop.

Step 4. Choose a literal A such that for some C, $C' \in S_0$, $A \in C$, $\bar{A} \in C'$; Replace S_0 by S, where S is specified by the A-transition $S_0 \xrightarrow{A} S$; Go to step 2.

After the algorithm stops, if $\square$ is in the current clause set then the input clause is unsatisfiable; otherwise it is satisfiable.

Example 2 For clause set

$$\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}\},$$

we have the following sequence of literal transitions:

$$\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}\}$$

$$\xrightarrow{C} \{\{A, B\}, \{A, \neg B\}, \{\neg A\}\}$$

$$\xrightarrow{B} \{\{A\}, \{\neg A\}\}$$

$$\xrightarrow{A} \{\square\}.$$

So

$$\{\{A, B, \neg C\}, \{A, B, C\}, \{A, \neg B\}, \{\neg A\}\}$$

is unsatisfiable.

Literal resolution is a resolution strategy which *replaces the old clause set by a new clause set.* The key idea is that each step performs resolutions related to a *single literal.* Note that if at each step we only perform *one* resolution, or *all* possible resolution, the corresponding algorithms are not correct as shown by the following two examples.

Example 3 Performing an A-resolution on

$$\{\,\{A,\,B\},\ \{A,\,\neg B\},\ \{\neg A\}\,\}$$

we can get

$$\{\,\{B\},\ \{A,\,\neg B\}\,\}.$$

This clause set is satisfiable while the original one is not.

Example 4 Performing *all* possible resolutions on

$$\{\,\{A,\,B\},\ \{\neg B\},\ \{\neg A\}\,\}$$

we get

$$\{\,\{B\},\ \{A\}\,\}.$$

We also failed in detecting the inconsistency of the original clause set.

Literal resolution works for the two examples as demonstrated below. For Example 3 we have

$$\{\,\{A,\,B\},\ \{A,\,\neg B\},\ \{\neg A\}\,\} \xrightarrow{A} \{\,\{B\},\ \{\neg B\}\,\}$$

$$\xrightarrow{B} \{\,\square\,\},$$

while for Example 4,

$$\{\,\{A,\,B\},\ \{\neg B\},\ \{\neg A\}\,\} \xrightarrow{A} \{\,\{B\},\ \{\neg B\}\,\}$$

$$\xrightarrow{B} \{\,\square\,\}.$$

3. Soundness and Completeness

In this section we prove that literal resolution is sound and complete. LR is ***sound*** in the sense that for any input clause set S_0, whenever $\square$ is in the current clause set after literal resolution terminates, S_0 is unsatisfiable. It is *complete* in the sense that if the input clause S_0 is unsatisfiable, then $\square$ must be in the current clause set when literal resolution stops.

Note that for Robinson resolution, $\mathcal{R}^*(S)$ is equivalent to S. This is not a necessity for resolution to work. In fact literal resolution does not have this property, i.e., when $S \longrightarrow S'$, S and S' need not be equivalent as Example 1 shows.

The following lemma is the key for completeness. We write $S \uplus S'$ for the disjoint union of two clause sets S, S'.

Lemma Let S be a finite clause set and

$$S \stackrel{A}{\longrightarrow} S'.$$

Then the satisfiability of S' implies the satisfiability of S.

Proof By mathematical induction on n, the number of clauses in S, we show that

$$\forall S, S', \Phi. (S \stackrel{A}{\longrightarrow} S' \;\&\; \Phi \models S') \Longrightarrow (\Phi[\bot/A] \models S \text{ or } \Phi[\top/A] \models S).$$

Base case ($n = 2$). Assume $S \stackrel{A}{\longrightarrow} S'$, $\Phi \models S'$ and $\mid S \mid= 2$. By the definition given for $\stackrel{A}{\longrightarrow}$, in this case S must be of the form

$$\{ C \cup \{ A \}, C' \cup \{ \neg A \} \}$$

and S' of the form $\{ \{ C \cup C' \} \}$. Since $\Phi \models S'$, either $\Phi \models \{ \{ C \} \}$, which implies

$$\Phi[\bot/A] \models \{ C \cup \{ A \}, C' \cup \{ \neg A \} \};$$

or $\Phi \models \{ \{ C' \} \}$, which implies

$$\Phi[\top/A] \models \{ C \cup \{ A \}, C' \cup \{ \neg A \} \}.$$

Induction step. Suppose the conclusion is true for all S with k or smaller number of clauses. Suppose $S \stackrel{A}{\longrightarrow} S'$ and $\Phi \models S'$, where $S = S_{\neg A} \uplus S_A \uplus R \uplus \{ C \}$. Here the size of $S_{\neg A} \uplus S_A \uplus R$ is not more than k, $S_{\neg A}$ consists of all the clauses of $S \setminus \{ C \}$ which contain the literal $\neg A$, and S_A consists of all the clauses of $S \setminus \{ C \}$ which contain the literal A. If $A \in C$, then either (i) $\Phi \models \{ C \setminus \{ A \} \}$ or (ii) $\Phi \models \{ C' \setminus \{ \neg A \} \mid C' \in S_{\neg A} \}$, since $\Phi \models S'$ and $W \subseteq S'$, where

$$W = \{ (C' \setminus \{ \neg A \}) \cup (C \setminus \{ A \}) \mid C' \in S_{\neg A} \}.$$

For case (i), if $S_A = \emptyset$ then clearly $\Phi[\perp/A] \models S$; Otherwise, if $S_A \neq \emptyset$, then we have

$$S_{\neg A} \uplus S_A \uplus R \xrightarrow{A} S' \setminus W$$

and

$$\Phi \models S' \setminus W.$$

By induction hypothesis,

$$\Phi[\perp/A] \models S_{\neg A} \uplus S_A \uplus R$$

or

$$\Phi[\top/A] \models S_{\neg A} \uplus S_A \uplus R.$$

This implies

$$\Phi[\perp/A] \models S_{\neg A} \uplus S_A \uplus R \uplus \{ C \}$$

or

$$\Phi[\top/A] \models S_{\neg A} \uplus S_A \uplus R \uplus \{ C \}$$

since $\Phi \models \{ C \setminus \{ A \} \}$.

For case (ii), it is easy to see that

$$\Phi[\top/A] \models S_{\neg A} \uplus S_A \uplus R \uplus \{ C \}$$

because $S_{\neg A}$ consists of all the clauses in S which contain $\neg A$, and they are already evaluated to $\top$ under Φ without considering the truth assignment for A.

The proof for the case $\neg A \in C$ works by symmetry.

We are left to check the case $\{ A, \neg A \} \cap C = \emptyset$. By definition, $C \in S'$ and

$$S_{\neg A} \uplus S_A \uplus R \xrightarrow{A} S' \setminus \{ C \}.$$

Clearly $\Phi \models (S' \setminus \{ C \})$. By induction hypothesis, $\Phi[\perp/A] \models S_{\neg A} \uplus S_A \uplus R$ or $\Phi[\top/A] \models S_{\neg A} \uplus S_A \uplus R$. Therefore $\Phi[\perp/A] \models S_{\neg A} \uplus S_A \uplus R \uplus \{ C \}$ or $\Phi[\top/A] \models S_{\neg A} \uplus S_A \uplus R \uplus \{ C \}$ since $\{ C \} \in S'$. **QED**

Now we can prove

Theorem (The soundness and completeness of LR) S_0 is unsatisfiable iff there are literals $A_0, A_1, \cdots A_n$ in S_0 such that

$$S_0 \xrightarrow{A_0} S_1 \xrightarrow{A_1} S_2 \xrightarrow{A_2} \cdots S_{n-1} \xrightarrow{A_{n-1}} S_n$$

and $\Box \in S_n$, where whenever a literal and its negation appear in some clause at the same time, we always replace that clause by $\{\top\}$.

Proof $(\Leftarrow)$: Suppose S is satisfiable and $S \xrightarrow{A} S'$. Then clearly S' is satisfiable by the soundness of the resolution rule. Thus if $\Box \in S_n$, S_0 must be unsatisfiable.

$(\Rightarrow)$: Suppose S_0 is unsatisfiable. Suppose

$$S_0 \xrightarrow{A_0} S_1 \xrightarrow{A_1} S_2 \xrightarrow{A_2} \cdots S_{n-1} \xrightarrow{A_{n-1}} S_n,$$

$\Box \notin S_n$, but S_n has no further transition. Then S_n is satisfiable because it has the valid truth assignment Φ_0 given below. For each $C \in S_n$, $C \neq \emptyset$. Let Φ_0 be an assignment which assigns $\top$ to A if $A \in \bigcup S_n$; assings $\bot$ to A if $\neg A \in \bigcup S_n$. Clearly $\Phi_0 \models S_n$ because $A \in \bigcup S_n$ implies $\neg A \notin \bigcup S_n$ and $\neg A \in \bigcup S_n$ implies $A \notin \bigcup S_n$.

By the lemma given above, we deduce that S_0 is satisfiable, which is a contradiction. This means we must have $\Box \in S_n$. **QED**

It is clear that for a clause set S, the satisfiability of S can be decided in no more than m steps of transition, where m is the number of propositional variables in S.

Using the idea of LR we propose the following restricted form of Robinson resolution: For any (finite) clause set S of atomic propositions, define

$$\mathcal{L}(S) = \bigcup_{i \geq 0} S_i,$$

where $S_0 = S$ and S_i's form an LR transition sequence:

$$S_0 \xrightarrow{A_0} S_1 \xrightarrow{A_1} S_2 \xrightarrow{A_2} \cdots S_k \xrightarrow{A_k} S_{k+1} \xrightarrow{A_{k+1}} \cdots.$$

Obviously it is always the case that

$$\mathcal{L}(S) \subseteq \mathcal{R}^*(S).$$

From the completeness of LR we immediately get

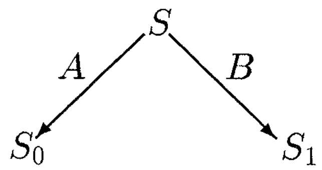
Corollary The propositional version of Robinson resolution is sound and complete.

In most cases the size of $\mathcal{L}(S)$ can be much smaller than that of $\mathcal{R}^*(S)$. Take Example 2, for example. LR produced 6 new clauses while brute force Robinson resolution produces the following 11 new clauses.

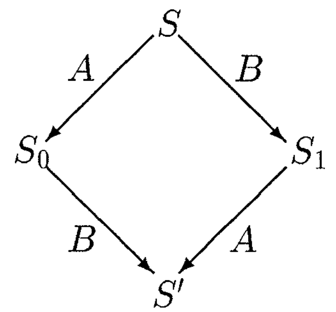
$$\{A, B\}, \{B, \neg C\}, \{B, C\}, \{\neg B\}, \{A, \neg C\}, \{A, C\},$$
$$\{A\}, \{B\}, \{C\}, \{\neg C\}, \Box.$$

4. Conclusion

We have presented a resolution strategy called literal resolution and we have shown that literal resolution is sound and complete. The basic idea of literal resolution is to perform all resolutions related to a particular literal in each step, and to replace the old clause set by a new clause set consisting of the literal-related resolvents, and clauses which can not involve in any resolution related to the literal. From the proof of the soundness and completeness theorem we can see that the order of literals chosen at each step does not affect whether we can find a $\Box$ eventually. This is not to say, however, that if there are two transitions from S such as



then there exists a clause set S' such that



(this is the so called Church-Rosser property) as Example 5 shows.

Example 5 Consider the clause set

$$\{\{A, \neg B\}, \{\neg A, B\}, \{B\}\}.$$

We have

$$\{\{A, \neg B\}, \{\neg A, B\}, \{B\}\} \xrightarrow{A} \{\{\top\}, \{B\}\}$$

and

$$\{\,\{\,A, \neg B\,\},\ \{\,\neg A, B\,\},\ \{\,B\,\}\,\} \xrightarrow{B} \{\,\{\,\top\,\},\ \{\,A\,\}\,\}.$$

But there are no transitions for $\{\,\{\,\top\,\},\ \{\,B\,\}\,\}$ and $\{\,\{\,\top\,\},\ \{\,A\,\}\,\}$ to get to the same clause set.

This leaves us some room to choose a better transition sequence to reduce the time complexity. In general it is suggested that we choose a literal which results in as small the number of resolvents as possible. In particular, if $C \cup \{\,A\,\}$, $C' \cup \{\,\neg A\,\}$ are the only two clauses in S which have A or $\neg A$ present, then a transition $S \xrightarrow{A} S'$ will make the size of S' less than that of S.

It is clear that if there are no more than k literals in S then LR will terminate with a time complexity of $O(n^{2^k})$, where n is the size of S. It should be interesting to give an analysis of the average time complexity of LR and to compare it with other resolution strategies.

References

[Ch73] Chang, C.L. and Lee, R.C.T., ***Symbolic Logic and Mechanical Theorem Proving***, Academic Press, New York, 1973

[DaPu60] Davis, M., Putnam, H., A Computing Procedure for Quantification Theory, JACM 7, 1960

[KoHa69] Kowalski, R.A., Hayes, P. Semantic Trees in Automatic Theorem Proving, ***Machine Intelligence*** Vol 4, (B. Meltzer and D. Michie, eds), American Elsevier, New York, 1969

[Ko79] Kowalski, R.A., ***Logic for Problem Solving***, Elsevier North Holland, New York, 1979

[LePa81] Lewis, H.R. and Papadimitriou, C.H, ***Elements of the Theory of Computation,*** Prentice-Hall, Englewood Cliffs NJ, 1981

[Lo70] Loveland, D.W., A Linear Format for Resolution, Proc. IRIA Symp. Automatic Demonstration, Springer-Verlag, New York, 1970

[Ro65] Robinson, J.A., A Machine-oriented Logic Based on the Resolution Principle, JACM, 12, (1) 1965

[WoRo64] Wos, L., Carson D.F. and Robinson, J.A. The Unit Preference Strategy in Theorem Proving, AFIPS Conference Proceedings, 1964