数理解析研究所講究録
第322巻 1978年 154-177

154

入力導出を階層化した導出について

山崎 進, 村木一至, 薮矢哲司, 堂下修司 (京大工学部)

# On resolution layered with input resolutions

Susumu YAMASAKI, Kazunori MURAKI, Tetsuji YOSHIYA, and Shuji DOSHITA

( Faculty of Engineering, Kyoto University )

## 1. INTRODUCTION

The first-order logic is one of the most important classes in symbolic logic from both theoretical and practical points of view. Mechanical theorem proving in the logic was founded on resolution principle by J.A.Robinson.

There are many refinements of resolution, by which we can check completely and effectively the unsatisfiability of given sets of clauses, for example, semantic resolution, lock resolution, linear resolution and so on. ( A clause is a disjunction of literals. A set of clauses corresponds to a formula in Skolem standard form. ) Also efficient refinements of resolution, namely,unit resolution and input resolution, have been proposed, although we cannot check the unsatisfiability of all the unsatisfiable sets of clauses by them. In them completeness is traded for efficiency. An input resolution is a resolution in which one of the two parent clauses is a given clause. A unit resolution is a resolution in which a resolvent is obtained by using at least one unit parent clause, or a unit factor of a parent clause.

The problem to find out whether or not $NP=P$ is very impor-

tant but is left unsolved. As the unsatisfiability problem for given propositional formulas in conjunctive normal form (CNF) is *NP-complete*, *NP=P* holds if it can be decided in deterministic polynomial time by a (complete) resolution whether or not the given propositional formula in CNF is unsatisfiable. Unit resolution has such a characteristic as follows: The problem of determining whether there exists a refutation (proof) of unit resolution from the given propositional formula in CNF is *P-complete*.

In this paper we will introduce a resolution layered with input resolutions (a Restricted Linear resolution, in short RL resolution) which retains the characteristic of unit resolution and by which we can check a refutation from more sets of clauses than unit resolution.

## 2. DEFINITIONS AND FUNDAMENTAL PROPERTIES

Definitions and fundamental properties about the first-order logic are given according to [3]. Definitions for a 'term', an 'atom', a 'literal', and a 'well-formed formula (formula, wff)' are used as usual.

Definition 2.1 An interpretation of a wff $G$ in the first-order logic consists of a nonempty domain $D$, and an assignment of values to each constant, function symbol, and predicate symbol occuring in $G$ as follows: (1) To each constant, we assign an element in $D$. (2) To each n-place function symbol, we assign a mapping from $D^n$ to $D$. (3) To each n-place predicate symbol, we assign a mapping from $D^n$ to $\{T,F\}$.

For the interpretation of a wff over a domain $D$, the evaluation of the wff follows the usual rules.

Definition 2.2 A wff $G$ is satisfiable if and only if there exists an interpretation $I$ such that $G$ is evaluated to $T$ in $I$. A wff $G$ is unsatisfiable if and only if there exists no interpretation in which $G$ is evaluated to $T$.

Definition 2.3 A wff $G$ is said to be in a prenex normal form if and only if the wff $G$ is in the form of $(Q_1x_1)\ldots(Q_nx_n)(M)$ where every $(Q_ix_i)$, $i=1,\ldots,n$, is either $(\forall x_i)$ or $(\exists x_i)$, and $M$ is a formula containing no quantifiers. $(Q_1x_1)\ldots(Q_nx_n)$ is called the prefix and $M$ is called the matrix of the wff $G$.

Every wff can be transformed into the prenex normal form, where the matrix is in a conjunctive normal form.

Definition 2.4 (Skolem standard form) Let a wff $G$ be already in a prenex normal form $(Q_1x_1)\ldots(Q_nx_n)M$, where $M$ is in a conjunctive normal form. Suppose $Q_r$ is an existential quantifier in the prefix for $i=1,\ldots,n$. If in the prefix no universal quantifier appears before $Q_r$, we choose a new constant $c$ different from other constants occuring in $M$, replace all $x_r$ appearing in $M$ by $c$, and delete $(Q_rx_r)$ from the prefix. If $Q_{s_1},\ldots,Q_{s_m}$ are all the universal quantifiers appearing before $Q_r$, $1\leq s_1<s_2<\ldots<s_m<r$, we choose a new m-place function symbol $f$ different from other functon symbols, replace all $x_r$ in $M$ by $f(x_{s_1},x_{s_2},\ldots,x_{x_m})$, and delete $(Q_rx_r)$ from the prefix. After the above process is applied to all the existential quantifiers in the prefix, the last

formula we obtain is a Skolem standard form of the wff $G$. The constants and functions used to replace the existential variables are called Skolem functions.

Definition 2.5 A clause is a disjunction of literals. We shall regard a set of literals as synonymous with a clause. A one-literal clause is called a unit clause. A clause containing no literals is called the empty clause and denoted as $\square$, which is always false for any interpretation.

A set $S$ of clauses is regarded as a conjunction of all clauses in $S$, where every variable in $S$ is considered governed by universal quantifiers.

Theorem 2.1 [3] Let $S$ be a set of clauses that represents a standard form of a wff $G$. Then $G$ is unsatisfiable if and only if $S$ is unsatisfiable.

Definition 2.6 A substitution is a finite set of the form $\{t_1/v_1, \ldots, t_n/v_n\}$, where every $v_i$ is a variable, every $t_i$ is a term different from $v_i$, and no two elements in the set have the same variable after the stroke symbol. The substitution that consists of no elements is called the empty substitution and is denoted $\varepsilon$.

Let $\theta=\{t_1/v_1,\ldots,t_n/v_n\}$ be a substitution and $E$ be an expression of a wff. Then $E\theta$ is an expression obtained from $E$ by replacing simultaneously each occurrence of the variable $v_i$, $1\leq i\leq n$, in $E$ by the term $t_i$. $E\theta$ is called an instance of $E$.

The composition of substitutions are defined in [3]. We adopt it.

Definition 2.7 A substitution $\theta$ is a unifier for a set $\{E_1,\ldots,E_k\}$ if and only if $E_1\theta=E_2\theta=\ldots=E_k\theta$. The set $\{E_1,\ldots,E_k\}$ is said to be unifiable if there is a unifier for it.

Definition 2.8 A unifier $\sigma$ for a set $\{E_1,\ldots,E_k\}$ of expressions is a most general unifier if and only if for each unifier $\theta$ for the set there is a substitution $\lambda$ such that $\theta=\sigma\lambda$.

If $W$ is a finite nonempty unifiable set of expressions, then there is an algorithm to obtain a most general unifier for $W$.

Definition 2.9 If two or more literals (with the same sign) of a clause $C$ have a most general unifier $\sigma$, then $C\sigma$ is called a factor of $C$. If $C\sigma$ is a unit clause, it is called a unit factor of $C$.

Definition 2.10 Let $C_1$ and $C_2$ be two clauses (called parent clauses) with no variables in common. Let $L_1$ and $L_2$ be two literals in $C_1$ and $C_2$, respectively. If $L_1$ and $\sim L_2$ have a most general unifier $\sigma$, then the clause $(C_1\sigma-L_1\sigma)\ (C_2\sigma-L_2\sigma)$ is called a binary resolvent of $C_1$ and $C_2$. The literals $L_1$ and $L_2$ are called the literals resolved upon.

Definition 2.11 A resolvent of (parent) clauses $C_1$ and $C_2$ is one of the following binary resolvents: (1) a binary resolvent of $C_1$ and $C_2$, (2) a binary resolvent of $C_1$ and a factor of $C_2$, (3) a binary resolvent of a factor of $C_1$ and $C_2$, (4) a binary resolvent of a factor of $C_1$ and a factor of $C_2$.

Definition 2.12 Given a set $S$ of clauses, a (resolution) deduction of $C$ from $S$ is a finite sequence $C_1,C_2,\ldots,C_k$ of clauses such that each $C_i$ is either a clause in $S$ or a resolvent of clauses

preceding $C_i$, and $C_k=C$. A deduction of $\square$ from $S$ is called a refutation, or a proof of $S$.

Theorem 2.2 [3] A set $S$ of clauses is unsatisfiable if and only if there is a deduction of the empty clause $\square$ from $S$.

Definition 2.13 A clause $C$ subsumes a clause $D$ if and only if there is a substitution $\sigma$ such that $C\sigma \subset D$. $D$ is called a subsumed clause.

Definition 2.14 If $A$ is an atom, then the set $\{A, \sim A\}$ is called a complementary pair. A clause is called tautology if it contains a complementary pair.

Definition 2.15 A refinement of resolution to guarantee that the empty clause can always be derived from an unsatisfiable set of clauses is called complete.

In this paper we treat an incomplete resolution and assume without any mention that any resolvent cannot be a tautology in the resolition.

## 3. FUNDAMENTAL PROPERTIES ABOUT UNIT AND INPUT RESOLUTIONS

Definition 3.1 [4] An input resolution is a resolution in which one of the two parent clauses is an input clause. An input deduction is a deduction in which every resolution is an input resolution. An input refutaton is an input deduction of $\square$.
If there exists an input refutation from $S$, then $S$ is said input refutatable.

Definition 3.2 [4] A unit resolution is a resolution in which a resolvent is obtained by using at least one unit parent clause,

or a unit factor of a parent clause. A unit deduction is a deduction in which every resolution is a unit resolution. A unit refutation is a unit deduction of □. If there exists a unit refutation from $S$, then $S$ is said unit refutatable.

Theorem 3.1 [4] There is a unit refutation from a set $S$ of clauses if and only if there is an input refutation from $S$.

Definition 3.3 The set of clauses $Suc(C)$ is defined for a clause $C$ as follows: $Suc^0(C)=C$, $Suc^n(C)=\{D\sigma \mid D\sigma$ is a unit clause, $D \in Suc^{n-1}(C)\} \cup \{Res(D,C_1) \mid C_1$ is a unit clause, $D \in Suc^{n-1}(C)\}$, $Suc(C)=\bigcup_{n=0}^{\infty} Suc^n(C)$.

The next theorems can be easily obtained.

Theorem 3.2 Let $S$ be a set of clauses $\{S_1,\ldots,S_m\}$. If there is a unit or input refutation from $S$ and $S_i'$ subsumes $S_i$, then there is a unit or input refutation from the set of clauses $\{S_1,\ldots, S_{i-1},S_i',S_{i+1},\ldots,S_m\}$.

Theorem 3.3 Let $C_1$, $C_2$,.., and $C_k$ be unit clauses. Let $Res(C_0, C_1)=B_1$, $Res(B_{i-1},C_i)=B_i$ for $i=2,\ldots,k$, and $B_k=\square$. If $C_1'$, $C_2'$,..., and $C_k'$ are any permutation of $C_1$, $C_2$,..., and $C_k$, $Res(C_0,C_1')=B_1'$, and $Res(B_{i-1}',C_i')=B_i'$ for $i=2,\ldots,k$, then $B_k'$ should be □.

A procedure is shown to provide an input refutation from the given set of clauses, based on the property described in the proof of the equivalence of unit and input resolutions in [4]. The procedure is a little different from that in TPU[4]. Theorem 3.2 guarantees that even if the clause subsumed by another clause is deleted from the unit or input refutatable set or its re-

solvents, there exists a unit or an input refutation from the remaining set. It is concluded from Theorem 3.3 that the unit clause having been resolved against all the other clauses is not necessary in the following procedure. The procedure is continued as long as the number of generated resolvents does not exceed the fixed limit or the functon-depth which the parent unit clause has is not deeper than the fixed one.

Procedure 1 ( A procedure to provide an input refutation )

The following notations are used in the procedure.

$S$: the given set of clauses. Let $S=\{S_1, S_2, \ldots, S_m\}$.

$S_0$: the set of unit clauses. $S_0$ is a queue. Initially $S_0$ is the set of unit clauses in $S$ and unit factors from $S$ (The order is arbitrary).

$S_0'$: a queue of clauses corresponding to $S_0$, that is, corresponding to a unit clause $C$ in $S_0$, there is at least one element $S_i$ $S_0'$ such that $C$ is in $Suc(S_i)$.

$N_1$: allowable number of $\sum_{i=1}^{m} \#W(S_i)$ for the procedure to be continued, where $W(S_i)$ is a subset of $Suc(S_i)$ and is set to $S_i$ at first (# denotes the cardinality).

$N_2$: allowable number of $\sum_{i=1}^{m} \#W(S_i)$ for the procedure to be continued before the function-depth test will be given to unit clauses.

$N_3$: allowable function-depth that a unit clause may have.

$Pd$: pushdown stack.

The procedure is as follows. $N_1$, $N_2$, and $N_3$ are set to fixed numbers.

Step 1 Set $k=\sum_{i=1}^{m}\#W(S_i)$.

Step 2 If $k$ is greater than $N_1$, terminate: any refutation is not found. Otherwise, go to the next step.

Step 3 If $S_0$ is empty, terminate: any refutation does not exist. Otherwise, go to the next step.

Step 4 Set the top clause of $S_0$ to $C_x$.

Step 5 Push down on $Pd$ the clause $S_i \in S_0'$ such that $C_x = Suc(S_i)$.

Step 6 Set $l=0$.

Step 7 Set $j=1$.

Step 8 If $j$ is greater than $m$, go to the next step. Otherwise, go to Step 12.

Step 9 If $l$ is $0$, go to the next step. Otherwise, go to Step 11.

Step 10 Pop up $S_i$ from $Pd$.

Step 11 Set $S_0=S_0-C_x$. Go to Step 2.

Step 12 Set $W(S_j)=W(S_j)-\{C \mid C\in W(S_j),\ C_x \text{ subsumes } C\}$.

Step 13 Set $k=k-\#\{C \mid C\in W(S_j),\ C_x \text{ subsumes } C\}$.

Step 14 If $k$ is greater than $N_2$, go to the next step. Otherwise, go to Step 16.

Step 15 If the function depth in the clause in $C_x$ is greater than $N_3$, go to Step 18. Otherwise, go to the next step.

Step 16 Compute $\{Res(C_x,C) \mid C\in W(S_j)\}$.

Step 17 If $\{Res(C_x,C) \mid C\in W(S_j)\}$ is empty, go to Step 18. Otherwise, go to Step 19.

Step 18 Set $j=j+1$. Go to Step 8.

Step 19 If $\square$ is in $\{Res(C_x, C) \mid C \in W(S_j)\}$, go to the next step. Otherwise, go to Step 22.

Step 20 Push down $S_i$ on $Pd$.

Step 21 Obtain an input refutation using clauses in the order of $Pd$ ( Clauses which cannot be parent clauses are to be skipped ).

Step 22 Set $l=1$.

Step 23 Set $W(S_j)=W(S_j) \cup \{Res(C_x, C) \mid C \in W(S_j)\}$.

Step 24 Execute Procedure (1.1) (subsumption test).

Step 25 Set $k=k+\#\{Res(C_x, C) \mid C \in W(S_j)\} - \#\{$Clauses deleted by the subsumption test$\}$.

Step 26 Execute Procedure (1.2). Go to Step 18.

Procedure (1.1): For given $j$, delete from $W(S_j)$ the clause subsumed by another clause in $W(S_j)$.

Procedure (1.2): For given $j$, select unit clauses in $\{Res(C_x, C) \mid C \in W(S_j)\} - \{$Clauses deleted by Procedure (1.1)$\}$ and unit factors from the set. Next add those unit clauses and unit factors to $S_0$, and add $S_j$ to $S_0'$, $S_j$ being corresponded to the clauses added to $S_0$.

The next theorem is obtained from the proof of the equivalence of unit and input resolutions.

Theorem 3.4 Given an input refutatable set of clauses, there exists an input refutation where the top clause is the one on the top of $Pd$ in Procedure 1, and only clauses on $Pd$ are used.

Definition 3.4 For an input refutatable set $S$ of clauses we denote as $I(S)$ the set of clauses which are used in an input refutation which has the clause on the top of $Pd$ in Procedure 1.

Example 3.1 Let $S=\{p_1, \sim p_1 \vee p_2, \sim p_1 \vee \sim p_2 \vee p_3, \sim p_1 \vee p_4, \sim p_1 \vee \sim p_2 \vee \sim p_3\}$. $S$ is input refutatable. $\sim p_1 \vee p_4$ is not on $Pd$, and $I(S)=\{p_1, \sim p_1 \vee p_2, \sim p_1 \vee \sim p_2 \vee p_3, \sim p_1 \vee \sim p_2 \vee \sim p_3\}$.

Theorem 3.5 Let a set $S$ of clauses be input refutatable. For any clause $C_0$ in $I(S)$, there exists an input refutation with top clause $C_0$.

## 4. RESOLUTION LAYERED WITH INPUT RESOLUTIONS

We introduce a restricted linear resolution (in short, RL resolution), which will be got as a layered resolution with input resolutions.

Definition 4.1 Given a set $S$ of clauses and a clause $C_0$ in $S$, an RL (resolution) deduction of $S_n$ with top clause $C_0$ is a deduction of the linear form of resolutions in which

(a) For $i=0,1,\ldots,n-1$, $C_{i+1}$ is a resolvent of $C_i$ (called a center clause) and $B_{i+1}$ (called a side clause), and each $B_{i+1}$ is either in $S$, or in the set $MC$ as defined in (b). $C_n$ is in $MC$.

(b) (1) $C_0$ is in $MC$. (2) If $C_i$ is in $MC$ and $j$ (greater than $i$) is the least integer satisfying the following condition, then $C_j$ is also in $MC$. $C_j$ is said to be adjacent to $C_i$.

(Condition) *(a)* $C_i=C_j \vee C$, where $C$ is a unit clause ( *(a)'* $C_i \supseteq C_j$ in propositional logic). *(b)* The literals resolved upon in $C_i$, $C_{i+1},\ldots,C_{j-1}$ or in those factors and the literals subsuming

those literals cannot be contained in $C_j$. *(c) (1)* For the greatest $k$ $(i+1 \leq k \leq j-1)$ such that $C_i$ is used as $B_k$, $C_i$-$C_j$ cannot be subsumed by the literals resolved upon in $B_{k+1}, \ldots, B_j$ or in those factors. *(2)* For each $l$ from $i+1$ to $j-1$, the following condition holds; for the greatest $k$ $(i+1 \leq k \leq j-1)$ such that $B_l$ is used as $B_k$, $B_l$-$C_j$ cannot be subsumed by the literals resolved upon in $B_{k+1}$, $\ldots, B_j$ or in those factors.

An RL deduction of $\square$ is called an RL refutation.

From the definition the next property is easily derived.

Theorem 4.1 If $C_j$ in $MC$ is adjacent to $C_i$ in $MC$ in an RL deduction, then the literals in $C_i \wedge C_j$ cannot be the literals resolved upon and the terms of those literals cannot be operated by any unification in the deduction of $C_j$.

Example 4.1 let $S=\{p \vee q, p \vee \sim q, \sim p \vee q, \sim p \vee \sim q\}$. There is not a unit or input refutation from $S$ because no unit clause is in $S$. An RL refutation from $S$ is as follows: Let $C_0=p \vee q$, $C_1=q$, $C_2=p$, $C_3=\sim q$, and $C_4=\square$ be center clauses, and let $B_1=\sim p \vee q$, $B_2=p \vee \sim q$, $B_3=\sim p \vee \sim q$, and $B_4=q$ be side clauses, where $C_{i+1}$ is a resolvent of $C_i$ and $B_i$ for $i=0,1,2,3$. This deduction is an RL refutation, where $C_0$, $C_1$ and $C_4$ are in $MC$.

Theorem 4.2 Let $C_j$ be adjacent to $C_i$ in an RL deduction, and assume that for $k=i+1, \ldots, j$, $C_k$ is a resolvent of $C_{k-1}$ and $B_k$. Then there is an input refutation from $\{C_i', B_{i+1}', \ldots, B_j'\}$ with top clause $C_i'$ corresponding to the RL deduction of $C_j$, in which $B_{i+1}', \ldots,$ and $B_j'$ are side clauses in that order, where

$C_i'=C_i-C_j$ is a unit clause and for $k=i+1,\ldots,j$, $B_k'=B_k-C_j$. Also $\{C_i',B_{i+1}',\ldots,B_j'\}$ is equivalent to $I(\{C_i',B_{i+1}',\ldots,B_j'\})$.

Following the above theorem, we conclude that if $C_j$ is adjacent to $C_i$ in an RL deduction , then the other $C_k(k<i)$ in $MC$ can not be used as a side clause in the deduction of $C_j$ with top clause $C_i$.

The following point is important to get an RL refutation. Let $C_j$ be adjacent to $C_i$ in an RL refutation. Then the literals, which can be factored into the literals in $C_j$ may be added to the center clause from some side clause. Even if the factoring is executed as long as it does not unify the terms in the other literals than those to be factored, we can obtain the RL deduction of $C_j$. Thus we use the above factoring in obtaining an Rl refutation.

Theorem 4.3 Let $S$ be an input refutatable set. Then there is an RL refutation with any unit clause in $I(S)$ as top clause.

Theorem 4.4 Assume a linear deduction of $C_j$ with top clause $C_i$ such that (a) for $k=i,i+1,\ldots,j-1$, $C_{k+1}$ is a resolvent of $C_k$ and $B_{k+1}$ , and $B_{k+1}$ is an input clause or $C_i$, (b) $C_i=C_j \vee C$ ($C$ is a unit clause), (c) the literals resolved upon in $C_i,C_{i+1},\ldots,C_{j-2}$, $C_{j-1}$ or in those factors are not in $C_j$, (d) the terms of $C_j$ contained in $C_i,\ldots,C_{j-1}$ are not operated by any unification, (e) $\{C,B_k' \ (i+1\leq k\leq j)\} = I(\{C,B_k' \ (i+1\leq k\leq j)\})$ for $B_k'=B_k-C_j$ $(i+1 \leq k\leq j)$ . Then it is an RL deduction.

Based on the above properties, a procedure to check an RL refutation is shown. An RL deduction of a clause $C_j$ with a clause $C_i$ as top clause can be obtained according to Procedure 1.

Procedure 2 ( A procedure to test the existence of an RL refutation )

Let $S=\{S_1,S_2,\ldots,S_m\}$ be the given set of clauses, where the clauses in $S$ are arranged in the order of the number of literals in them. Let the number of literals in $S_i$ be $f(i)$ for $i=1,2,\ldots,m$. The continuation of the procedure is prescribed by that of Procedure (2.3) based on Procedure 1. The procedure is as follows.

Step 1 Set $i=1$.

Step 2 If $i$ is greter than $m$, terminate: no RL refutation exists. Otherwise, go to the next step.

Step 3 Execute Procedure (2.1) to obtain $C_i$.

Step 4 If $C_i$ is empty, go to the next step. Otherwise go to Step 6.

Step 5 Set $i=i+1$. Go to Step 2.

Step 6 Select an element (a set of literals) from $C_i$ and set it to $c_i$.

Step 7 Execute Procedure (2.2). Set $c_i$ to $\{L_j^i\}$ $(1\leq j\leq f(i))$.

Step 8 Set $j=1$.

Step 9 If $j$ is greater than $f(i)$, terminate: an RL refutation exists. Otherwise, go to the next step.

Step 10 Execute Procedure (2.3) (Test the existence of an RL deduction).

Step 11 If there exists an RL deduction in Step 10, go to the next step. Otherwise, go to Step 13.

Step 12 Set $j=j+1$. Go to Step 9.

Step 13 Set $C_i=C_i-c_i$. Go to Step 4.

Procedure (2.1): Get the family of all permutations of $S_i$ with literals.

Procedure (2.2): Set the literal in $c_i$ to $L_j^i$ in the order for $j=1,\ldots,f(i)$.

Procedure (2.3): Test the existence of an RL deduction of $L_1^i \vee L_2^i \vee \ldots \vee L_{j-1}^i$ with top clause $L_1^i \vee L_2^i \vee \ldots \vee L_j^i$; test the existence of an input refutation described in Theorem 4.2 where the terms of $L_1^i$, $L_2^i$,..., and $L_{j-1}^i$ are not operated by any unifications, according to Procedure 1. The remaining clauses, from which $L_1^i \vee L_2^i \vee \ldots \vee L_{j-1}^i$ is removed and which are resolved in the input refutation, are in the set as defined in Definition 3.4.

In Definition 4.1 for propositional logic, $(a)'$ $C_i \supset C_j$ is prefered to $(a)$ $C_i=C_j \vee C$ where $C$ is a unit clause. For this definition the next properties are obtained.

Definition 4.2 Let $C_1$ and $C_2$ be clauses in propositional logic such that $C_1=C_2 \vee L$ for some literal $L$. Then $C_2$ is said to have the relation $R$ with $C_1$.

Theorem 4.5 For any input refutatable set $S$ of clauses in propositional logic, there exists an RL refutation from $S$ with any clause in $I(S)$ as top clause where any clause in $MC$ has the relation $R$ with its adjacent one.

Theorem 4.6 Let $S$ be a propositional set of clauses from which there exists an RL refutation. Then there exists an RL refutation where any clause in $MC$ has the relation $R$ with its adjacent one.

In the resolutions for propositional logic, no unification for the terms is necessary. Thus the procedure to get an RL refutation from the propositional set of clauses is easier. It is given as an algorithm below.

Procedure 3 ( An algorithm to test the existence of an RL refutation from the given propositional set of clauses )

Let $S=\{S_1,S_2,\ldots,S_m\}$, and $S_i=\bigcup_{j=1}^{f(i)} L_j^i$ for $i=1,\ldots,m$. The algorithm is as follows.

Step 1 Set $i=1$.

Step 2 Set $l=0$.

Step 3 Set $T=\phi$.

Step 4 If $i$ is greater than $m$, terminate: no RL refutations exist. Otherwise, go to the next step.

Step 5 Set $j=1$.

Step 6 If $j$ is greater than $f(i)$, go to Step 13. Otherwise, go to the next step.

Step 7 If $L_j^i$ is in $T$, go to the next. Otherwise, go to Step 9.

Step 8 Set $j=j+1$. Go to Step 6.

Step 9 Execute Procedure (3.1) (Test the existence of an RL deduction).

Step 10 If there exists an RL deduction , go to the next step. Otherwise, go to Step 8.

Step 11 Set $L_j$ in $T$.

Step 12 Set $l=1$. Go to Step 8.

Step 13 If $l=1$, go to Step 15. Otherwise, go to the next step.

Step 14 Set $i=i+1$. Go to Step 3.

Step 15 If $T=S_i$, terminate: an RL refutation exists. Otherwise, go to the next step.

Step 16 Set $l=0$. Go to Step 5.

Procedure (3.1): Test the existence of an RL deduction of $T$ with $T^{\vee}L_j^i$ as top clause; test the existence of an input refutation described in Theorem 4.2 according to Procedure 1. The remaining clauses, from which $T$ is removed and which are resolved in the input refutation, are in the set as defined in Definition 3.4.

An RL deduction is a linear deduction layered with input resolutions. A deduction which is not linear is defined below as an extension of an RL deduction.

Definition 4.3 If $C$ is a unit clause or the empty clause, an RL deduction of $C$ from $\{S_1,S_2,\ldots,S_m\}$ is defined as $RL(S_1,S_2,\ldots,S_m:C)$. An RL1 deduction of $C$ from $\{S_1,S_2,\ldots,S_m\}$ for $C$ being a unit clause or the empty clause is defined as follows. It is denoted as $RL1(S_1,S_2,\ldots,S_m:C)$.

(a) $RL(S_1,S_2,\ldots,S_m:C)$ is also $RL1(S_1,S_2,\ldots,S_m:C)$.

(b) If there hold $RL1(S_1,S_2,\ldots,S_m:L_i)$ and $RL1(S_1,S_2,\ldots,S_{i-1},L_i,S_{i+1},\ldots,S_m:C)$ where $L_i$ is a unit clause, then there holds $RL1(S_1,S_2,\ldots,S_m:C)$.

(c) The deduction defined with (a) and (b) recursively is an RL1 deduction.

An RL1 deduction of $\square$ is called an RL1 refutation.

As Procedure 2 and 3 can be applied to obtain RL deductions of unit clauses, a procedure to determine whether there exists an RL1 refutation from the given set of clauses can be constructed.

Procedure 4 ( A procedure to test the existence of an RL1 refutation )

Let $S=\{S_1,S_2,\ldots,S_m\}$ be the given set of clauses, where $S_i=L_1{}^i \vee L_2{}^i \vee \ldots \vee L_{f(i)}{}^i$ for $i=1,2,\ldots,m$. Consider the sets of clauses $S_{i0}$ corresponding to $S_i$ for $i=1,2,\ldots,m$. Initially set $S_{i0}=\phi$ for each $i$. The procedure is as follows. Its continuation is prescribed by those of Procedure (4.1), (4.2) and (4.3).

Step 1 Set $i=1$.

Step 2 Set $l=0$.

Step 3 Set $S'=S$.

Step 4 If $i$ is greater than $m$, go to the next step. Otherwise, go to Step 8.

Step 5 If $l=0$, terminate: no RL1 refutations exist. Otherwise, go to the next step.

Step 6 Set $i=1$.

Step 7 Set $l=0$.

Step 8 Set $j=1$.

Step 9 If $L_j^i$ is not in $S_{i0}$, go to Step 12. Otherwise, go to the next step.

Step 10 Execute Procedure (4.1) (Test the existence of an input refutation).

Step 11 If there exists an input refutation, terminate: an RL1 refutation exists. Otherwise, go to the next step.

Step 12 Set $j=j+1$.

Step 13 If $j$ is greater than $f(i)$, go to the next step. Otherwise, go to Step 9.

Step 14 If $S_i=S_{i0}$ as a set of literals, go to the next step. Otherwise, go to Step 16.

Step 15 Set $i=i+1$. Go to Step 4.

Step 16 Execute Procedure (4.2) (Test the existence of an RL refutation).

Step 17 If there exists an RL refutation, terminate: an RL1 refutation exists. Otherwise, go to the next step.

Step 18 Set $j=1$.

Step 19 If $L_j^i$ is in $S_{i0}$, go to Step 25. Otherwise, go to the next step.

Step 20 Execute Procedure (4.3) (Test whether there exists an RL deduction of $L_j^i$).

Step 21 If there exists an RL deduction of $L_j^i$, go to the next step. Otherwise, go to Step 25.

Step 22 Set $L_j^i$ to be in $S_{i0}$.

Step 23 Set $l=1$.

Step 24 Set $S'=S' \cup L_j^i$.

Step 25 Set $j=j+1$.

Step 26 If $j$ is greater than $f(i)$, go to Step 15. Otherwise, go to Step 19.

Procedure (4.1): For given $i$ and $j$, determine whether there exists an input refutation from $S'$ with $L_j^i$ as top clause, according to Procedure 1.

Procedure (4.2): For given $i$, determine whether there exists an RL refutation from $S'$ with $S_i$ as top clause, according to Procedure 2 or 3.

Procedure (4.3): For given $i$ and $j$, determine whether there exists an RL deduction of $L_j^i$ from $S'$ with $S_i$ as top clause, applying Procedure 2 or 3. If $S_i$ is a unit clause, we consider that there is an RL deduction with $S_i$ as top clause.

The next examples show incompleteness of RL and RL1 resolutons.

Example 4.2 Let $S=\{p_0 \vee p_1 \vee p_2, \sim p_1 \vee p_2, p_1 \vee \sim p_2, \sim p_1 \vee \sim p_2, \sim p_0 \vee p_3 \vee p_4, \sim p_3 \vee p_4, \sim p_3 \vee \sim p_4, p_3 \vee \sim p_4\}$. There exists no RL refutation from $S$. As there exist RL deductions of $p_0$ and $\sim p_0$, there exists an RL1 refutation from $S$.

Example 4.3 Let $S=\{p_0 \vee p_1 \vee p_2 \vee p_3, \sim p_2 \vee p_3, p_2 \vee \sim p_3, \sim p_2 \vee \sim p_3, \sim p_0 \vee p_1 \vee p_4 p_5, \sim p_4 \vee p_5, p_4 \vee \sim p_5, \sim p_4 \vee \sim p_5, p_0 \vee \sim p_1 \vee p_6 \vee p_7, \sim p_6 \vee p_7, p_6 \vee \sim p_7, \sim p_6 \vee \sim p_7, \sim p_0 \vee \sim p_1 \vee p_8 \vee p_9, \sim p_8 \vee p_9, p_8 \vee \sim p_9, \sim p_8 \vee \sim p_9\}$. There exists no RL1 refutation

from $S$, although $S$ is unsatisfiable.

## 5. RL(RL1) REFUTATION FROM PROPOSITIONAL FORMULAS

It is shown that the problem to determine whether there exists an RL(RL1) refutation from the given propositional formula (in the set of clauses or in conjunctive normal form) is *P-complete*. In this section by the formula we mean the propositional formula in the set of clauses or in conjunctive normal form. The decidability problem is regarded with the language in the same sense.

Definition 5.1 [8] Let $L, L' \subseteq \Sigma^*$. $L \overset{\leq}{p} L'$ if and only if there exists a function $g$ such that (a) $x \in L$ if and only if $g(x) \in L'$ for $\forall x \in \Sigma^*$, (b) $g$ is computable in space complexity $log(|x|)$ for the input length $|x|$ of $x$.

Definition 5.2 [8] Let $P$ be the class of languages accepted by deterministic Turing machines in polynomial time (complexity). $L$ is said to be *P-comlete* if and only if (a) $L \in P$, (b) $L' \overset{\leq}{p} L$ for any $L' \in P$.

Theorem 5.1 [8] The problem to determine whether there exists a unit refutation from the formula of length $n$ is decidable in time complexity $O(n^2)$ and in space complexity $O(n)$, and is *P-complete*.

According to Theorem 3.1 and Procedure 1, we can easily obtain that Theorem 5.1 also holds for the input refutation.

We can obtain the following theorem according to Procedure 3.

Theorem 5.2 It is of time complexity $O(n^4)$ and of space complexity $O(n)$ to determine whether there exists an RL refutation from

the formula $S$ where $S=\{S_1,S_2,\ldots,S_m\}$, $S_i=\bigvee_{j=1}^{f(i)} L_j^i$, and $\sum_{i=1}^{m} f(i)=n$.

We can obtain the following theorem according to Procedure 3 and 4.

Theorem 5.3 It is of time complexity $O(n^6)$ and of space complexity $O(n)$ to determine whether there exists an RL1 refutation from the formula $S$ where $S=\{S_1,S_2,\ldots,S_m\}$, $S_i=\bigvee_{j=1}^{f(i)} L_j^i$, and $\sum_{i=1}^{m} f(i)=n$.

For the formula $S=\{S_1,\ldots,S_m\}$, consider the transformation $TF1$: $S\to S'=\{S_1\vee\lambda,\ldots,S_m\vee\lambda,\sim\lambda\vee\lambda_1\vee\lambda_2,\sim\lambda_1\vee\lambda_2,\lambda_1\vee\sim\lambda_2,\sim\lambda_1\vee\sim\lambda_2$ where $\lambda$, $\lambda_1$ and $\lambda_2$ are atoms not appearing in $S$. Then we can obtain the next theorem.

Theorem 5.4 "The problem to determine whether there exists a unit refutation from the formula" $\leq_p$ "The problem to determine whether there exists an RL refutation from the formula".

For the formula $S=\{S_1,\ldots,S_m\}$, consider the transformation $TF2$: $S\to S'=\{S_1\vee\lambda,\ldots,S_m\vee\lambda,\sim\lambda\vee\lambda_0\vee\lambda_1\vee\lambda_2,\sim\lambda_1\vee\lambda_2,\lambda_1\vee\sim\lambda_2,\sim\lambda_1\vee\sim\lambda_2,\sim\lambda\vee\sim\lambda_0\vee\lambda_3\vee\lambda_4,\sim\lambda_3\vee\lambda_4,\lambda_3\vee\sim\lambda_4,\sim\lambda_3\vee\sim\lambda_4$ where $\lambda$, $\lambda_1$, $\lambda_2$, $\lambda_3$ and $\lambda_4$ are atoms not appearing in $S$. Then we can obtain the next theorem.

Theorem 4.5 "The problem to determine whether there exists an RL refutation from the formula" $\leq_p$ "The problem to determine whether there exists an RL1 refutation from the formula".

From Theorem 5.1,.5.2, 5.3, 5.4, and 5.5, the next theorem is derived.

Theorem 5.6 The problem to determine whether there exists an RL or RL1 refutation from the given formula is *P-complete*.

## 6. CONCLUDING REMARKS

In this paper we defined the RL resolution layered with input resolutions which is more powerful than unit and input resolutions. Next we showed that it has the characteristic of unit (input) resolution: the problem to determine whether there exists an RL refutation from the given propositional set of clauses is *P-complete*. Thus the RL resolution is an extension of unit (input) resolution and prescribes one class of resolutions.

It is left as a problem to examine whether there exists another resolution which has the characteristic of unit and RL resolutions.

## REFERENCES

[1] A.V.Aho, J.E.Hopcroft and J.D.Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley Publishing Company, 1974.

[2] P.B.Andrews, Resolution with merging, JACM, 15, p367-381, 1968.

[3] C.L.Chang and R.C.Lee, Symbolic Logic and Mechanical Theorem Proving, Academic Press, 1974.

[4] C.L.Chang, The unit proof and the input proof in theorem proving, JACM, 17, p698-707, 1970.

[5] S.A.Cook, The complexity of theorem proving procedures, Proc. of 3rd Annual ACM Symposium on Theory of Computing, p151-158, 1971.

[6] L.Henschen and L.Wos, Unit refutations and horn sets, JACM, 21, p590-605, 1974.

[7] L.J.Henschen, Semantic resolution for horn sets, IEEE Trans. on Computers, C-25, p816-822, 1976.

[8] J.D.Jones and W.T.Laaser, Complete problem for deterministic polynomial time, Proc. of 6th Annual ACM Symposium on Theory Computing, p40-46, 1974.

[9] D.Kuehner, Some special purpose resolution systems, in Machine Intelligence 7, p117-128, 1972.

[10] G.E.Peterson, Theorem proving with lemmas, JACM, 23, p573-581, 1976.