A NEW METHOD FOR THE PROOF OF THEOREMS

DOINA TATAR*

Received: November 11, 1991
AMS subject classification: 68T15

Rezumat. În lucrare se prezintă un sistem formal de demonstrare prin respingere a teoremelor. Condiţia necesară şi suficientă impusă acestui sistem se bazează pe metoda lui J.Hsiang de demonstrare a teoremelor cu ajutorul sistemelor de rescriere a termenilor.

1. **Introduction.** Let T be a set of linguistic, algebraic or symbolic objects (as, for instance, first-order terms, programs) and let ~ be an equivalence relation on T.

DEFINITION [2]. A computable function $S:T \to T$ is called a canonical simplifier for the equivalence relation ~ on T iff for all $s, t \in T$:

$$S(t) \sim t$$

$$S(t) \leq t$$

(for some ordering $\leq$ on T)

$$t \sim s \to S(t) = S(s)$$

For computer algebra, the problem of constructing canonical simplifiers is basic, because of the following theorem:

THEOREM [2]. *Let T be a set of linguistic objects and ~ an equivalence relation on T. Then ~ is decidable iff there exists a canonical simplifier S for ~ .*

Let $T = T(F,V)$ be the algebra free generated by the set of variables V with the set of functions F; that is T is the minimal set of words on the alphabet $F \cup V \cup \{(,)\}$ such that:

1. $V \subseteq T$

* *University of Cluj-Napoca, Faculty of Mathematics, 3400 Cluj-Napoca, Romania*

2. If $f \in F$, $\alpha(f)$ is its arity, and if $t_1, \ldots, t_{\alpha(f)} \in T$, then $f(t_1, \ldots, t_{\alpha(f)}) \in T$

Let $E \subseteq T(F,V) \times T(F,V)$ be a set of equations. By the Birkhoff theorem (1935) s and t are semantically equal in the equational theory $E(E \models s = t)$ iff s and t are provably equal in the theory $E(E \vdash s = t)$.

Let $s \sim t$ be the equivalence relation defined by $E \vdash s = t$. Then $\sim$ is decidable iff there exists a canonical simplifier S for $\sim$.

2. Associated term rewriting system and the completion. Let E be a set of equations $E \subseteq T \times T$ and let R_E a term rewriting system (TRS) obtained such that

$\ell \to r \in R_E \Leftrightarrow \ell = r \in E$ and

$v(r) \subseteq v(\ell)$, where $v(t)$ is the set of variables in the term (object) $t \in T$. This system will be called TRS associated with E. The rewriting relation $\vec{R}_E$ has the inverse relation, transitive closure, the reflexive-symmetric-transitive closure denoted by $\overleftarrow{R}_E$, $\vec{R}_E$ and $\overleftrightarrow{R}_E$ respectively. Also, we have:

$$\tilde{E} = \overset{*}{\overleftrightarrow{R}}_E$$

For a TRS denoted R let be the following definition [3], [7], [8]:

DEFINITION. R is noetherian (R has the finite termination property) iff there is no infinite chain

$$t_1 \ \vec{R}_E \ t_2 \ \vec{R}_E \ t_3 \ \vec{R}_E \ldots$$

DEFINITION. R is confluent iff $\forall x, y, z \in T \ \exists u \in T$ such that if $x \overset{*}{\vec{R}_E} z$ and $x \overset{*}{\vec{R}_E} y$ then $z \overset{*}{\vec{R}_E} u$, $y \overset{*}{\vec{R}_E} u$.

DEFINITION. If $x \in T$, $x\downarrow \in T$, $x \overset{*}{\vec{R}_E} x\downarrow$ and it does not exist t such that $x\downarrow \vec{R} t$ then $x\downarrow$ is normal form for x in TRS R (denoted $x \downarrow R$).

If R_E which is associated with a system of equation E is noetherian and confluent (i.e. complete) then, for $\forall x \in T$, the application $S(x) = x \downarrow R_E$ is a canonical simplifier. Then ~ is decidable, and we have :

$$s \sim t \quad \text{iff} \quad s \downarrow R_E = t \downarrow R_E$$

Stated in the context of confluence, the idea of completion is straightforward:

Given a set of equations E we try to find a set of equations F such that: $\underset{E}{\sim} = \underset{F}{\sim}$ and the relation $\vec{R}_F$ is confluent.

If this set of equations do not exists, then the completion must terminate with failure or the completion is impossible.

The first completion algorithm for rewrite rules is that of Knuth-Bendix (1967). For a general formulation of this algorithm some additional notion for describing the replacement of terms in terms are needed.

DEFINITION [1],[2],[5]. Let $O(t)$ be the set of occurrences of a term t. If s, $t \in T(F,V)$ and $u \in O(t)$ then $t[u \leftarrow a]$ is the term that derives from t if the term occurring at u in t is replaced by the term s (t/u becomes s).

DEFINITION. $s \rightarrow t$ iff there is a rule $a \rightarrow b \in R_E$ (or an equation $((a,b) \in E)$, a substitution τ and an occurrence $u \in O(s)$ such that

$$s/u = \tau\ (a) \text{ and } t = s\ [u \leftarrow \tau\ (b)]$$

DEFINITION. The terms p and q form a critical pair in E iff there are equations $(a_1,b_1) \in E$ and $(a_2,b_2) \in E$, an occurrence u in $O(a_1)$ and the substitution τ_1, τ_2 such that:

1. a_1/u is not a variable
2. $\tau_1(a_1/u) = \tau_2(a_2)$
3. $p = \tau_1(a_1)\ [u \leftarrow \tau_2(b_2)]$
   $q = \tau_1(b_1)$

The algorithm Knuth-Bendix is based on the

THEOREM: *A TRS noetherian R_E is confluent iff for all critical pairs (p,q) of E: $p \downarrow R_E = q \downarrow R_E$.*

Then it suggests to augment R_E by the rule $p \downarrow R_E \rightarrow q \downarrow R_E$ or $q \downarrow R_E \rightarrow p \downarrow R_E$. This process may be iterated until, hopefully, all critical pairs have a unique normal form or it may never stops: the algorithm is at least a semidecision procedure for $\sim$.

The completion algorithm for rewrite rules (Knuth-Bendix, 1967) is therefore [2]:

I n p u t : A finite set of equations E such that $\vec{R}_E$ is noetherian.

O u t p u t : 1. A finite set of equations F such that $\overset{*}{\underset{R_E}{\leftrightarrow}} = \overset{*}{\underset{R_F}{\leftrightarrow}}$ and relation $\vec{R}_F$ (therefore system R_F) is confluent (therefore is decidable) or

2. the procedure stops with failure or

3. the procedure never stops

Algorithm [2]:

1. $F: = E$;

2. C: = set of critical pairs of F;

3. while $C \neq 0$ do

   3.1. if $(p,q) \in C$ and $(p \downarrow R_F \neq q \downarrow R_F)$ then

      3.1.1. if $p{\downarrow}R_F \to q{\downarrow}R_F$ leaves R_F noetherian then $R_F := R_F \cup \{p{\downarrow}R_F \to q{\downarrow}R_F\}$ else if $q{\downarrow}R_F \to p{\downarrow}R_F$ leaves R_F noetherian then $R_F := R_F \cup \{ q \downarrow R_F \to p \downarrow R_F \}$

      else STOP (FAILURE)

      3.1.2. $C = C \cup \{$ critical pairs in $F \cup \{(p \downarrow R_F , q \downarrow R_F)\}\}$

      3.1.3. $F = F \cup \{(p \downarrow R_F = q \downarrow R_F)\}$

   3.2. $C := C \setminus \{(p, q)\}$

4. STOP(R_F).

The above crude form of the algorithm can be refined in many ways. The sequence of critical pairs chosen by the procedure in 3.1. may have a crucial influence on the efficiency of the algorithm.

3. The J. Hsiang's completion procedure. It is well known that a formula in first-order predicate calculus is valid, iff the closed Skolemized version of its negation is false under Herbrand interpretation. Equivalentely, a formula is valid if the set of the clauses in its clausal form is insatisfiable. Hsiang [7] first suggested using a complete rewrite system in a resolution-like theorem-proving strategy.

Let $C = \{C_1,\ldots,C_n\}$ the set of clauses of a formula in first-order predicate calculus.

Let $C_1 = L_1 \vee L_2 \vee \ldots \vee L_k$ be a clause where L_j is a literal, and let H be a mapping transforming terms of a Boolean algebra into terms of a Boolean ring:

$$H(C_i) = \begin{cases} 1 & \text{if } C_i \text{ is empty clause} \\ x+1 & \text{if } C_i \text{ is } x \\ x & \text{if } C_i \text{ is } \overline{x} \\ H(L_1) * H(L_2 V \ldots V L_k) & \text{otherwise} \end{cases}$$

THEOREM (Hsiang[7]: *Given a set of clauses* $\mathfrak{C}$ *in first-order predicate Calculus,* $\mathfrak{C}$ *is inconsistent iff the system*

$$H(C_i) = 0, \; C_i \in \mathfrak{C}, \; i = 1,n$$

has not a solution.

Now, let BR be the complete TRS [7]:

$x + 0 \to 0$

$x + x \to 0$

$x * 1 \to x$

$x * 0 \to 0$

$x * x \to x$

$x * (y+z) \to x * y + x * z$

For each equation $H(C_i) = 0$ let us consider the equation $a_i = b_i$, where a_i is the biggest monomial of boolean polynomial $H(C_i)$ and let E be the system corresponding in this fashion to the system of equations:

$$H(C_i) = 0, i = \overline{1,n}$$

The TRS R_E having all the rules of the form $a_i \to b_i$ is noetherian [7]. In the TRS formed by $R_E \cup BR$ we have:

$$s \underset{H(C_i)=0}{\sim} t \Rightarrow s \underset{E}{\sim} t \Rightarrow s \overset{*}{\underset{R_E \cup BR}{\longleftrightarrow}} t$$

because $a_i = b_i$ is equivalent with $a_i + b_i = H(C_i) = 0$

A critical pair (p,q) may be added to system R_E not only in the form $p{\downarrow}R_E \rightarrow q{\downarrow}R_E$ or in the form $q{\downarrow}R_E \rightarrow p{\downarrow}R_E$, but also in the form $p'{\downarrow}R_E \rightarrow q'{\downarrow}R_E$ where p' is the biggest monomial of Boolean polynomial $P + q$. Hence, the polynomial $p + q$ is an intermediate form to study for critical pair.

Then, the previous theorem becomes:

THEOREM [7]. *A set of clauses* $\mathfrak{C}$,*in first-order predicate calculus is inconsistent iff by Knuth-Bendix completion algorithm applied to the TRS formed by* $R_E \cup BR$, *where* E *is the set of equations* $a_i{=}b_i$, $i = 1,\ldots,n$ (a_i *is the biggest monomial of* $H(C_i)$), *the critical pair* $1{\rightarrow}0$ *is obtained.* Let us observe that KB algorithm of completion is allways terminating by STOP.

4. A new method for proving a formula. Let $S = (\Sigma, F, A, R)$ be a formal system, where Σ is the alphabet for the term in a boolean ring (including + and *), F is the set of boolean polynomials, $A = \emptyset$ and R is the single deductive rule denoted "res" or $\vdash$:

$f_i,\ f_j \vdash f_k$ iff

$f_i,\ f_j,\ f_k \in F$ and there exist the monomials $\alpha,\ \beta \in F$ and the substitution τ_1 and τ_2 such that:

$$(\alpha * \tau_1(f_i)) \downarrow BR = (\beta * \tau_2(f_j) + f_k) \downarrow BR$$

where the equality is modulo associativity and commutativity.

For this formal system the following theorem is true:

THEOREM : *Given a set of clauses* $\mathfrak{C} = \{C_1,\ldots,C_n\}$ *in first-order predicate calculus,* $\mathfrak{C}$ *is inconsistent if in formal system* S:

$H(C_1), \dots, H(C_n) \vdash 1.$

The proof of theorem in propositional calculus consists of the following three propositions (the proof of theorem in predicate calculus is analogous).

PROPOSITION 1. *If f_i, $f_j \vdash f_k$ and f_i, f_j, f_k are the clause polynomials then $H^{-1}(f_i) \wedge H^{-1}(f_j) \rightarrow H^{-1}(f_k)$.*

Proof. By the assumption:

$f_i = \tilde{a}_{i_1} * \dots * \tilde{a}_{i_k}$, where

$$\tilde{a}_{i_s} = \begin{cases} a_{i_s}+1 \\ \text{or} \\ a_{i_s} \end{cases}, \quad s=\overline{1,k}$$

and $f_j = \tilde{b}_{j_1} * \dots * \tilde{b}_{j_l}$, where

$$\tilde{b}_{j_t} = \begin{cases} b_{j_t}+1 \\ \text{or} \\ b_{j_t} \end{cases}, \quad t=\overline{1,e}$$

If $\tilde{a}_u = a + 1$, $\tilde{b}_v = a$, $u \in \{i_1, \dots, i_k\}$, $v \in \{j_1, \dots, j_e\}$: by the commutativity of operation * we can write:

$$f_i = (a + 1) * \gamma$$

$$f_j = a * \gamma$$

In boolean ring the following identity is obvious:

$$\delta*(a+1)*\gamma = \delta * a * \gamma+\delta * \gamma$$

By the comparison with the relation:

$$\alpha * f_i = \beta * f_j + f_k$$

(because $\tau_1 = \tau_2 =$ the identic substitution in propositional calculus), we observe that $f_k = \delta * \gamma$, and that $H^{-1}(f_k) = H^{-1}(\delta) \vee H^{-1}(\gamma)$.

In the propositional calculus the following implications are true:

$$(a \vee a_{i_1}^{\alpha_{i_1}} \vee \ldots \vee a_{i_k}^{\alpha_{i_k}}) \wedge (\bar{a} \vee b_{j_1}^{\alpha_{j_1}} \vee \ldots \vee b_{j_e}^{\alpha_{i_e}}) \rightarrow (a_{i_1}^{\alpha_{i_1}} \vee \ldots \vee b_{j_1}^{\alpha_{i_1}})$$

where $i_s \neq u$, $j_t \neq v$,

$$\alpha_{i_s}, \ \alpha_{j_t} \in \{0,1\}, s = \overline{1,k} \quad , \quad t = \overline{1,e}$$

and

$$a_{i_s}^{\alpha_{i_s}} = \begin{cases} a_{i_s} & \text{if } \alpha_{i_s} = 1 \\ \overline{a_{i_s}} & \text{if } \alpha_{i_s} = 0 \end{cases}$$

and analogously for $b_{j_t}^{\alpha_t}$.

The above implication is therefore:

$H^{-1}(f_i) \wedge H^{-1}(f_j) \rightarrow H^{-1}(f_k)$

PROPOSITION 2. *If* $\mathfrak{C} = \{C_1, \ldots, C_n\}$ *is a set of clauses, and if:*

$H(C_1), \ldots, H(C_n) \vdash U$

U is clause polinomial, then

$$C_1 \wedge \ldots \wedge C_n \rightarrow H^{-1}(U)$$

Proof: To prove this proposition we proceed by induction after the length i of the deduction of U from $H(C_1), \ldots, H(C_n)$ in formal system S.

If $i = 0$, then exists j such that $U = H(C_j)$ and

$$H^{-1}(H(C_j)) = C_j$$

The following implication is true:

$$C_1 \wedge \ldots \wedge C_n \rightarrow C_j \ , \ j = 1, \ldots, n$$

We suppose that the proposition 2 is true for the length $\leq i - 1$ of deduction, and let $f_0, \ldots, f_m = U$ a deduction of U with the length i.

For the three last polynomials f_{m-2}, f_{m-1}, f_m in the system S there is the relation:

$$\alpha * f_{m-2} = \beta * f_{m-1} + f_m$$

Moreover, if f_m is a clause polynomial, f_{m-2} and f_{m-1} are too, and f_{m-2} and f_{m-1} are obtained by the deduction of length $\leq i - 1$.

From the induction hypothesis we have:

$$C_1 \wedge \ldots \wedge C_n \rightarrow H^{-1}(f_{m-2})$$

$$C_1 \wedge \ldots \wedge C_n \rightarrow H^{-1}(f_{m-1})$$

By the formula:

$$\vdash (A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow B \wedge C))$$

results by modus poneus:

$$\vdash C_1 \wedge \ldots \wedge C_n \rightarrow H_{-1}\ (f_{m-2}) \wedge H_{-1}\ (f_{m-1})$$

From proposition 1 we have:

$\vdash H^{-1}\ (f_{m-2}) \wedge H^{-1}\ (f_{m-1}) \rightarrow H^{-1}\ (f_m)$ and by the rule of syllogism

$$\vdash C_1 \wedge \ldots \wedge C_n \rightarrow H^{-1}\ (f_m)$$

or

$$\vdash C_1 \wedge \ldots \wedge C_1 \rightarrow H^{-1}\ (U) \text{ q.e.d.}$$

PROPOSITION 3. *If* $H(C_1),\ldots,H(C_n) \vdash 1$ *then* $\mathfrak{C} = \{C_1,\ldots,C_n\}$ *is inconsistent.*

Proof. From the proposition 2 we have:

$$\vdash C_1 \wedge \ldots \wedge C_n \rightarrow H^{-1}\ (1)$$

but H^{-1} (1) is the empty clause. q.e.d.

But the condition (x) "$H(C_1),\ldots,H(C_n) \vdash 1$ iff $\mathfrak{C} = \{C_1,\ldots,C_n\}$ is inconsistent" is also true hence the implication

"$H(C_1),\ldots,H(C_n) \vdash 1 \to \mathfrak{C} = \{C_1,\ldots,C_n\}$ is inconsistent" is true even through not all the polynomials f_i, f_j, f_k in the propositions are the clause polynomials.

Exemple: (In propositional calculul $\tau_1 = \tau_2 =$ identic substitution) $\mathfrak{C} = \{P \vee \bar{Q} \vee R, \bar{P} \vee Q \vee \bar{R}, \bar{P} \vee \bar{Q}, Q \vee P, P \vee \bar{R}\}$

$H(C_1) = PQR + QR + PQ + Q$

$H(C_2) = PQR + PR$

$H(C_3) = PQ$

$H(C_4) = QR + Q + R + 1$

$H(C_5) = PR + R$

$H(C_1),\ H(C_2) \vdash PR + PQ + RQ + Q$

(due to the fact that $PQR + PQ + RQ + Q = (PQR + PR) + (PR + PQ + QR + Q)$)

$PQ + PR + RQ + Q,\ H(C_3) \vdash PR + RQ + Q$

$PR + RQ + Q\ ,\qquad H(C_5) \vdash RQ + Q + R$

$(PR + RQ + Q = H(C_5) + QR + Q + R)$

$H(C_4),\ RQ + Q + R \vdash 1$

This set of clauses is inconsistent, and the triplet f_i, f_j, f_k is not in each step the clause polynomials (like in proposition 1).

In fact the following observation is true: if A_i is the set of all the clauses with i positive variables (nonnegative): $C_1 \in A_i$ and $C_2 \in A_j$ are two clauses, $|i-j| \geq 2$, and $H(C_1)$, $H(C_2) \vdash f_k$ then f_k is not a clause polynomial. Moreover, if $C_1 \in A_i$ and $C_2 \in A_{i+1}$ differ by a number n of variables, with $n \geq 2$, and $H(C_1)$, $H(C_2) \vdash f_k$ then f_k is not a clause polynomial.

The condition (*) results from Hsiang's theorem (§ 3) by following observations:

Let us observe that the deductive rule "res": $f_i,\ f_j \vdash f_k \Leftrightarrow \exists\ \alpha, \beta$ (monomials) such that $(\alpha * \tau(f_i))\downarrow BR = (\beta*\tau_2(f_j) + f_k)\downarrow BR$ is a special fashion to calculate a critical pair. Indeed, the biggest monomial in $\alpha*\tau_1(f_i)$ (i.e. $MP\ f_i$) and the biggest monomial in $\beta*\tau_2(f_j)$ (i.e. $MP\ f_j$) are equal and:

$(f_k)\downarrow BR = (\alpha*\tau_1(f_i) + \beta*\tau_2(f_j))\downarrow BR = (MP\ f_i + MP\ f_j + \text{REST}\ f_i + \text{REST}\ f_j\ )\ \downarrow BR = (\text{REST}\ f_i + \text{REST}\ f_j)\downarrow \text{BR}$

This is the case $\tau_1(a_1) = \tau_2(a_2)$ and $(p,q) = (\tau_1(b_1),\ \tau_2(b_2)$ is a critical pair. The intermediate form $p + q$ of critical pair (in our case f_k) is studied.

THEOREM: *The set of clauses* $\mathbb{C} = \{C_1,\dots,C_n\}$ *is inconsistent iff*

$$H(C_1),\dots,H(C_n) \vdash 1$$

Proof: If $\mathbb{C} = \{C_1,\dots,C_n\}$ is inconsistent, by Hsiang's theorem the system $H(C_i) = 0,\ i = 1,\dots,n$ has not a solution, or, equivalentaly, by completion in R_E the rule $1 \rightarrow 0$ is obtained. Therefore, a critical pair (1,0) or $(f_k,\ 0)$ is obtained. We have:

$$(f_k)\ \downarrow\ BR = 1 = (1 + P + P)\ \downarrow\ BR$$

In formal system S we can write $1 + P,\ P \vdash f_k(= 1)$ where P is a boolean polynomial.

Conversely, if $H(C_1),\dots,H(C_n)\ \vdash\ 1$ then there exists a deduction $f_0,\dots,f_k = 1$ from $H(C_1),\dots,H(C_n)$.

Therefore, there exists f_i and f_j such that $f_i,\ f_j \vdash f_k(=1)$. But f_k is a critical pair corresponding to a rule $1\rightarrow 0$, and by Hsiang's theorem $\mathbb{C}$ is inconsistent.

REFERENCES

1. Avenchaus, J.Denzinger,J. Muller: *"THEOPOGLES - An efficient Theorem Prover based on Rewrite-Techniques"*, Dep. of Comp.Sc.University of Kaiserslautern, 1990.
2. B.Buchberger, R.Loos: *"Algebraic simplification"* Computing, Suppl. 4, 11-43, 1982.
3. B.Buchberger: *"History and Basic Features of the Critical Pair Completion procedure"*, J. Symbolic Computation, 3, 3- 38,1986.
4. J.P.Delahaye: *"Outils logigues pour l'intelligence artificielle"*, Ed. Eyrolles, 1986.
5. M.Dershowits: *"Completion and its applications"*. "Resolution of Equations in Algebraic Structures", vol.*2*.
6. J.Hsiang, M.Rusinowith: *"On word problems in equational theories"* 14-th ICALP, Karlsruhe, 1987.
7. J.Hsiang: *"Refutational theorem proving using Term Rewriting System"*, Artificial Intelligence, 25, 255-300, 1985.
8. M.Jantzen: *"Confluent String Rewriting"*, EATCS, Springer Verlag, 1988.
9. J.P.Jouannaud, P.Lescanne: *"La reecriture"*, Technique et Science Inf., 6, 433-452, 1986.
10. J.Muller: *"Topics in completion Theorem Proving"*, Univ. Kaiserlautern, 1990.
11. P.Lescanne: *"Computer Experiments with REVE Term Rewriting System Generator"*, Tenth. Annual ACM Symposium on Princ. of Progr. Lang., 99-108, 1983.
12. P.Rety, C.Kirchner, H.Kirchner, P.Lescanne: *"Narower, a new-algorithm for unification and its application to logic programming"*, LNCS 202, 141-157, 1985.
13. M.Rusinowitch: *"Demonstration automatiques. Techniques de reecriture"*, Intern. Edition, Paris, 1989.
14. D.Tătar: *"Normalized rewriting systems and applications in the theory of program"*, Analele Univ. Bucureşti, nr.2, 76-80, 1989.