TERM REWRITING SYSTEMS AND COMPLETION THEOREMS PROVING: A SHORT SURVEY

Doina TĂTAR*

Received: February 15, 1993
AMS Subject Classification: 68T15

REZUMAT. - Sisteme de rescriere a termenilor şi demonstrarea teoremelor prin algoritmi de completare. În acest articol sunt prezentate principalele rezultate privind problema cuvântului pentru o teorie ecuaţională, tratată ca sistem de rescriere a termenilor (TRS), inclusiv o versiune sintetică a algoritmului de completare Knuth-Bendix.

Abstract. In this paper we survey the main results concerning equations and the methods for reasoning about them like term rewriting systems (TRS). This TRS are used to reduce expressions to canonical form, if this form is unique. A simplified version of Knuth-Bendix completion algorithm is presented.

Like most surveys, ours does not contain any new results, but it gives an idea on the application of TRS to theorems proving. This paper is justified by the interest of this subject and it presents the most important things in the completion idea. Since the pioneering paper (Knuth and Bendix, 1970), which indroduced the algorithm Knuth-Bendix, and the influential papers (Huet and Oppen, 1980), there has been a great deal of research in this field. For an excellent survey see (Avenhaus, Madlener, 1990).

This paper is organized as follows: Chapter 1 presents

* *Univ. "Babeş-Bolyai" of Cluj-Napoca, Dept. of Computer Science 3400 Cluj-Napoca, România*

equational systems and TRS, Chapter 2 the "critical-pair" idea and the "critical-pair" completion algorithm, and Chapter 3 some of the examples.

1. Introduction.

DEFINITION: An equational theory (F, V, E) consists of

- a set F of function symbols or operators (with the same sort, for simplicity).
- a set V of variables. Let $T(F,V)$ be the set of terms built from F and V.
- a set E of pair of equations, $s=t$, $s,t \in T(F,V)$.

The set of equations E defines a syntactical equality relation $=_E$ on $T(F,V)$, usualy defined by the concept of "replacing equals by equals". One has also a semantical (logic) definition in equational theory E denoted by: $E \models s=t$.

The theorem of Birkhoff (1935) assures that both notions coincide: $t_1 =_E t_2 \leftrightarrow E \models t_1=t_2$.

A fundamental problem is the "validity problem" or "word problem", which is undecidable in general:

"Give $s,t \in T(F,V)$, does $s =_E t$?"

Obviously, the undecidability (more precisely, the semi-decidability) of the "word problem" is transfered on the approach by TRS. But this approach is, on the our opinion, more algorithmicaly.

DEFINITION: A TRS R is a set of rules:

$R = \{l \to r \mid l, r \in T(F,V)$ every variable occuring in term r also occurs in term $l\}$.

It defines a rewrite relation $\underset{R}{\rightarrow}$:

DEFINITION: $s \underset{R}{\rightarrow} t$ iff there is a rule $l \rightarrow r \in R$, an occurence p in s, such that:

$$s/p=\sigma(l), \quad t=s[p \leftarrow \sigma(r)].$$

for some substitution σ .

The relation $\underset{R}{\rightarrow}$ is compatible with the term structure in $T(F,V)$ (i.e. $s \rightarrow t$ implies $t_1[p \leftarrow s] \underset{R}{\rightarrow} t_1[p \leftarrow t]$) and with the substitutions (i.e. $s \rightarrow t$ implies $\sigma(s) \underset{R}{\rightarrow} \sigma(t)$ for each σ).

We denote by $\overset{*}{\underset{R}{\rightarrow}}$, $\overset{*}{\underset{R}{\leftrightarrow}}$ the reflexive - transitive and reflexive - transitive - symmetric closure of $\underset{R}{\rightarrow}$.

DEFINITION: The transfering of "word problem" to a TRS is:

"Given an equational theory E, compute an R such that $s \underset{E}{=} t$ is equivalent to $s \overset{*}{\underset{R}{\leftrightarrow}} t$ ".

The problem of compute R is a "completion procedure because R is constructed step by step by collecting new rules in R, which is in the same time simplified as much as possible.

Let $t \downarrow R$ (or $t\downarrow$) normal form of t, that is such term with the properties:

1) $t \overset{*}{\underset{R}{\rightarrow}} t\downarrow$

2) $t\downarrow$ is irreducible.

If R has the properties that every term has a unique normal form, then:

$$s \overset{*}{\underset{R}{\leftrightarrow}} t \quad iff \quad s\downarrow = t\downarrow.$$

because $s \overset{*}{\underset{R}{\leftrightarrow}} t$ is $s \overset{*}{\underset{R}{\rightarrow}} s\downarrow = t\downarrow \overset{*}{\underset{R}{\leftarrow}} t$)

Fact: If in R every term t has a unique normal form then

$$s =_E t \text{ if } s{\downarrow} \equiv t{\downarrow}$$

DEFINITION: A term rewriting system R with the property that every term has a unique normal form is *convergent* and it has the properties:

a) R is terminating (or Noetherian) that is it allows no infinite sequences:

$$t_1 \rightarrow_R t_2 \rightarrow_R t_3 \rightarrow \ldots$$

(such that every term t has at least a normal form $t{\downarrow}$)

b) R is confluent, that is: $t_1 \overset{*}{\rightarrow}_R t_2$, $t_1 \overset{*}{\rightarrow}_R t_3$ implies that does exist u such that $t_2 \overset{*}{\rightarrow}_R u$, $t_3 \overset{*}{\rightarrow}_R u$ (every term t has at most a normal form $t{\downarrow}$).

The properties 1 and 2 are, unfortunately, undecidable (Dershowitz, 1987). However, there are useful tools to prove termination, the most important ones are **reduction orderings** on $T(F,V)$. An ordering > on $T(F,V)$ is a reduction ordering if > is well founded, is compatible with the term structure and is compatible with substitutions.

THEOREM: *A TRS R is terminating iff there is a reduction ordering > such that $\ell > r$ for every rule $\ell \rightarrow r$ in R.*

Due to a result of Newman (Newman, 1972) the confluence is, for terminating TRS, equivalent to the weaker property of local confluence, which is: $t_1 \rightarrow_R t_2$ *and* $t_1 \rightarrow_R t_2$, implies that does exist u such that $t_2 \overset{*}{\rightarrow}_R u$, $t_3 \overset{*}{\rightarrow}_R u$. Hence, a terminating R is confluent iff it is locally confluent.

2. Critical pair completion.

DEFINITION: Let $\ell_1 \to r_1$ and $\ell_2 \to r_2$ be two rules in R. By renaming of variables we may assume that they do not share common variables. If σ_1 and σ_2 are two substitutions, such that:

$$\sigma_1(\ell_1) = \sigma_2 \ell_2)$$

then $(\sigma_1(r_1), \sigma_2(r_2))$ is a critical pair for R. Let CP(R) be the set of all critical pairs for R as equations. "Critical Pair Lemma" (Knuth and Bendix, 1970) say that:

For any TRS R, if $t_1 \to_R t_2$ *and* $t_1 \to_R t_3$ then either does exist a term u such that $t_2 \overset{*}{\to}_R u$, $t_3 \overset{*}{\to}_R u$ (if R is locally confluent) or $t_1 =_{CP(R)} t_2$.

Clearly, if for every $(\alpha_1, \alpha_2) \in$CP(R) we have $\alpha_1 \overset{*}{\to}_R \alpha_2$ *or* $\alpha_2 \overset{*}{\to}_R \alpha_1$, then R is locally confluent. This think can be tested. The completion procedure do this. The simplest form of a completion procedure is:

INPUT: A set E of equations, an reduction ordering >.

OUTPUT: a) A TRS R_E convergent, such that

$$=_E = \overset{*}{\leftrightarrow}_{R_E}$$

b) FAILURE.

c) The algorithm run forever.

The Completion Algorithm:

$R = \emptyset$.

If every equations in E can be oriented

then

$R := \{\ell \to r \mid \ell > r., \ell = r \in E\}$

else

"FAILURE". STOP.

```
while          CP(R) ≠ ø do
        (t1,t2):= an element in CP(R)
        It calculates t1↓ and t2↓.
        If t1↓ ≠ t2↓
             then
             If neither t1↓ > t2↓ nor t2↓ > t1↓
                              then
                              "FAILURE" STOP
                              else
                              CP(R)=CP(R)\{(t1,t2)}.
                              R = R ∪ {t1↓ → t2↓}.
                                         or
                              R = R ∪ {t2↓ → t1↓}
             else
                   CP(R)=CP(R)\{t1, t2)}.
   STOP.
```

Some observations are immediatelly:

a) If CP(R) is transformed in ø, then the procedure stop succesfully.

b) The procedure may stop by "FAILURE" if R is not terminating.

c) The procedure may run forever, if CP(R) can be not transformed in ø.

(CP(R) grows and degrows in a step).

3. Theorem proving examples. 1) Let E be given by $E=\{e*x = x,\ i(x)*x=e,\ x*(y*z) = (x*y)*z\}$, hence E are the axioms for a group.

If the ordering is $i > * > e$, then R is, at beginning:

$$R\begin{cases} r_1:\ e*x \to x \\ r_2:\ i(x)*x \to e \\ r_3:\ (x*y)*z \to n*(y*z) \end{cases}$$

From r_2 and r_3 we obtain:

$$r_4: i(x)*(x*y) \to y.$$

because:

$$(i(x)*x)*y \underset{r_2}{\to} e*y \underset{r_1}{\to} y$$

$$(i(x)*x)*y \underset{r_3}{\to} i(x)*(x*y)$$

Hence, $(i(x)*(x*y),y)$ is a critical pair and r_4 is the new correspondent rule. At the end, we obtain the following TRS convergent:
$R\ \{r_1,\ldots r_{10}\}$, where r_1, r_2, r_3, r_4, are the previously, and:

r_5: $i(e) \to e$

r_6: $x*e \to x$

r_7: $i(i(x)) \to x$

r_8: $x*i(x) \to e$

r_9: $x*(i(x)*y) \to y$

r_{10}: $i(x*y) \to i(y) * i(x)$.

Thus, for example we have:

$i(i(x*y)*e) * i(y*y)=i(y*i(x))$

$i(i(x*y)*e) * i(y*y) \underset{r_{10}}{\to} (i(e) * i(i(x*y)) * i(y*y) \underset{r_5,r_7}{\to}$
$(e*(x*y)) * i(y*y) \underset{r_1}{\to} (x*y) * i(y*y) \underset{r_{10}}{\to} ((x*y) * i(y)) * i(y)$
$(x*y) * (i(y) * i(y)) \underset{r_3}{\to} x * (y*(i(y) * i(y)) \underset{r_9}{\to} x * i(y)$

For second member:

$$i(y*i(x)) \underset{r_{10}}{\cdot} i(i(x)) * i(y) \underset{r_7}{\cdot} x*i(y)$$

Thus theorem $i(i(x*y)*e)*i(y*y)=i(y*i(x))$ is proved, as each member have the same normal form $x * i(y)$.

In present, several very strong prover have been developed, such as REVE (Lescanne, 1984), RRL (Kapur et al. 1986) and the Large prover (Garland and Guttag 1989). A catalogue of theorems proved is given in (Hermann 1991). Some experiments with a completion theorem prover was made in (U. Martin and M. Lai, 1989). Another way for using TRS in theorem proving is that suggested by Hsiang (Hsiang 1985, Tătar 1992): The TRS denoted BA for Boolean Algebra. This is a rewrite-based method for first-order predicate calculus.

At the end of this short survey let list the best paper concerned TRS and Theorem proving.

REFERENCES

1. Avenhaus, J., Madlener, K. : *Term rewriting and Equational Reasoning*, Formal Techniques in A.I., A Sourcebook, R.B.Banerji (ed) 1990.
2. Avenhaus, J. : *On the Descriptive Power of TRS*, J. of Symbolic Computation 2, 1986, pp. 109-122.
3. Avenhaus, J., Denzinger, J., Muller, J. : *THEOPOGLES anefficient Th. Prov. based on Rewr. Technique*, Tech. Rap.,Univ. of Kaiserslautern, 1990.
4. Bachmair, L., Dershowitz, N., Hsiang, J. : *Orderings for equational proofs*, 1st LICS (1986), pp. 346-357.
5. Bellegarde, F., Lescanne, P. : *Transformation orderings*, 12th CAAP, 1987, LNCS 249, pp. 69-80.
6. Ben Cherifa, A., Lescanne, P. : *Termination of TRS by Polynomial Interpretations and its Implementation*, Science of Comp. Programming, 9, 1987.
7. Buchberger, B. : *History and Basic Features of the Critical-Pair/Completion Procedure*, J. Symbolic Comp.3, 1987, pp.3-38.
8. Buchberger, B., Loos, R. : *Algebraic Simplification*, in Computer Algebra-Symb. and Alg. Comp., 1982, pp.11-43.
9. Buchberger, B. : *A critical-pair/completion algorithm in reduction rings*, Technical Rap, CAMP-LINZ, 1983.
10. Bundgen, R. : *Buchberger's Algorithm: The Term Rewriter's point of view*, ICALP'92, to appear.
11. Dershowitz, N. : *Completion and its Applications*, Coll. on the

"Resolution of Eq. in Alg. Structures", vol.2, 1989.
12. Dershowitz, N. : *Termination*, J. Symbolic Comp, 3(1987), pp.69-116.
13. Dershowitz, N., Hsiang, N. : *Refutational Theorem Proving with Oriented Equations*, Coll. on the Res. of Eq. in Alg Struct., 1987.
14. Goguen, J. : *Proving and rewriting,* 2nd International Conf. on Alg. and Logic Programming, oct. 1990.
15. Hermann, M., Kirchner, C., Kirchner, H. : *Implementations of TRS*, Computer Journal 34, 1991, pp. 20-33.
16. Hsiang, J. : *Two results in term rewriting theorem proving*, LNCS 202, pp. 301-324.
17. Hsiang, J. : *Rewrite method for Theorem proving in First Order Theory with Equality*, J. Symb. Comp. 1-2, vol. 3, 1987.
18. Hsiang, J., Rusinowitch, M. : *On word problems in equational theories*, LNCS 267, pp. 54-71.
19. Huet, G., Oppen, D.C. : *Equationa and Rewrite Rules: A Survey*, in "Formal languages: theory, perspective and open problems" (ed. R. Book), 1980.
20. Jantzen. M. : *Confluent string rewriting,* EATCS monographs 11, 1988.
21. Jouannaud, J.P., Lescanne, P. : *Rewriting Systems*, Techhnology and Science of Informatics, 1987, pp. 181-199.
22. Kaplan, S. : *Conditional Rewrite Rules*, Th. Comp. Sci., 33, 1984, pp. 175-193.
23. Knuth, D.E., Bendix, P.P. : *Simple Word Problqms in Universal Algebras*, Comp. Prob. in Abstract Algebra, (ed. J. Leech), 1970.
24. Lescanne, P. : *Term Rewriting Systems and Algebra*, LNCS, 170, 1984.
25. Lescanne, P. : *Computer Experiments with the REVE TRS Generator*, Proc. 10 ACM POPL, Austin, 1983, pp: 99-108.
26. Martin, U., Lai, M. : *Some experiments with a completion theorem prover*, J. Symb. Comp., 13, 1992, pp. 81-100.
27. Muller, J. : *Topics in Completion Theorem Proving*, Technical Rap., Univ. Kaiserslautern, 1990.
28. Newman, M.H.A. : *On theories with a combinatorial Definition of Equivalence*, Ann. of Math., *43*, 1942, pp. 223-243.
29. Rusinowitch, M. : *Path of Subterms Ordering and Recursive Decomposition Ordering Revisited*, J. Symb. Comp., 1987, vol.3, pp. 117-133.
30. Rusinowitch, M. : *Demonstration automatique. Techniques de réécriture*, Inter. Edition, Paris, 1989.
31. Tătar, D. : *A new method for the proof of theorems*, Studia Univ. "Babeş-Bolyai", 1991, pp.
32. Tătar, D.: *Critical pair as a deduction rule,* Proceedings of 3th. Coll. on Logic Lang., Math. Linguisitc, Braşov, mai, 1991.
33. Tătar, D.:*Normalized TRS and applications in the theory of programs*, Analele Univ. Buc., nr.*2*, 1989, pp. 76-80.
34. Winkler, F., Buchberger,B.: *A criterion for eliminating unnecessary reductions in the K-B algorithm* , 983, Coll. Alg. Combin. and Logic, Györ.