Matching, Unification and Complexity *

- A Preliminary Note -

Deepak Kapur and Paliath Narendran

Computer Science Branch
General Electric Company
Corporate Research and Development
Schenectady, NY 12345

Pattern matching and unification are two key primitive operations used in many inference based systems including theorem provers, term rewriting systems, computer algebra systems, deductive data bases, Prolog, logic programming systems, functional language systems, systems for analysis of specifications and program synthesis, and program verification systems. In this note, we give results recently obtained in studying the complexity of matching and unification problems for first-order terms especially when some function symbols are interpreted. For further details and proofs, the reader may wish to refer to a forthcoming paper by the authors.

Both matching and unification problems for first-order terms built solely from uninterpreted function symbols have been known to be linear in the sum of the sizes of the input terms [PW]. When function symbols have properties such as associativity, idempotency, etc., both the problems turn out to much harder (in fact, intractable in most cases).

Given a theory E (presented usually as a finite set of equations), the *matching* problem is defined to be: Given a term (pattern) p and another term (subject) s, does there exist a substitution σ for variables in p such that

$$\sigma(p) = s \textit{ in } E?$$

The *unification* problem is defined to be: Given two terms s and t, does there exist a substitution σ for variables in s and t such that

$$\sigma(s) = \sigma(t) \textit{ in } E?$$

In the following table, symbols are used to stand for theories. The associated axiom(s) with each of the symbols is given below. For example, the symbol A implies that some of the function symbols in the terms under consideration are associative.

*Work partially supported by the NSF Grant No. DCS-8408461.

$$A : f(x, f(y, z)) = f(f(x, y), z)$$

$$C : f(x, y) = f(y, x)$$

$$I : f(x, x) = x$$

$$U : f(x, 1) = x$$

$$D : f(x, g(y, z)) = g(f(x, y), f(x, z))$$

When more than one symbol is used to stand for a theory, it means that the axioms corresponding to each of the symbols are conjuncted. For example, ACI, stands for the theory in which some function symbols appearing in the theory are assumed to be associative, commutative and idempotent. AC matching is an NP complete problem even if each variable in the pattern is restricted to have only at most two occurrences. $AC1$ stands for the theory in which function symbols may be associative-commutative and terms under consideration for unification and matching have unique occurrences of each variable.

Set matching problem is defined as the problem of checking, given a set of patterns (sp) and a set of subjects (ss), whether there exists a substitution σ such that the set of terms obtained by applying σ on sp is the same as the set ss. Similarly, set unification problem is defined as the problem of checking, given two sets of terms st and ss, whether there exists a substitution σ such that the set of terms obtained after applying σ on st is the same as the set of terms obtained after applying σ on ss. Bag matching and bag unification are defined analogously except that bags of terms, instead of sets of terms, are considered, i.e., number of occurrences of a term also becomes relevant. As should be evident, set matching and set unification are special cases of ACI matching and ACI unification, respectively, whereas bag matching and bag unification are special cases of AC matching and AC unification, respectively.

Most of the results obtained by the authors are shown by reducing the matching and unification problems to NP-complete problems such as 3SAT, Mono-3SAT, one-in-three 3-SAT, etc. Showing that these problems can be done in NP has been quite easy in general except in the case of associative-commutative (and associative-commutative-unity as well as associative-commutative-idempotent) unification, where it turns out to be quite nontrivial.

As the table indicates, in most cases, both matching and unification problems turn out to be of the same order of complexity even though matching problem is a special case of the unification problem. The complexity does not seem to grow even when additional properties of function symbols are assumed in some cases.

It also appears that for linear terms (terms in which every variable appears uniquely), both matching and unification problems are easier than for nonlinear terms (for matching, only the pattern has to be linear). This perhaps suggests that one of the main sources of complexity is the nonlinearity of terms.

There is one anomaly in this table, which is with respect to associative matching and unification. As the table states, associative matching is NP-complete, whereas associative unification (solvability of word equations over free semigroups) is only known to be decidable. The only complexity result known about associative unification is that it is primitive-recursive. A better upper bound is not known.

In the table, results are also given for unification problems over finitely presented algebras. In a finitely presented algebra, the presentation consists of a finite set of generators, a finite set of relations expressed using generators and the operator symbols of the algebra. Variables are not allowed in the relations. Terms under consideration for unification are "elementary terms," i.e., they can have variables but they do not have any uninterpreted function symbols. For example, $FPAG$ is a finite presentation of abelian groups generated by a finite set of generators with a finite set of relations expressed in terms of generators and the operators of abelian groups. $FPBR$ stands for finitely presented boolean rings. $FPCSG$ stands

Table: Complexity of Matching and Unification Problems

E	Matching	Unification
Φ	linear	linear [PW]
U	NP-complete [AT]	NP-complete [AT]
I	NP-complete [KN87]	NP -complete [KN87]
C	NP-complete [BKN]	NP-complete [S]
A	NP-complete [BKN]	decidable [Makanin]
CU	NP-complete [KN87]	NP-complete [KN87]
CI	NP-hard [KN861]	NP-hard [KN861]
AU	NP-complete [KN87]	decidable [Makanin]
AI	NP-hard [KN861]	NP-hard [KN861]
AC	NP-complete [BKN], [CK]	NP-complete [KN862]
ACU	NP-complete [KN862]	NP-complete [KN862]
ACI	NP-complete [KN862]	NP-complete [KN862]
D	NP-hard [AT]	NP-hard [AT]
DU	NP-hard [AT]	NP-hard [AT]
Set	NP-complete [KN861]	NP-complete [KN861]
Bag	NP-complete [KN87]	NP-complete [KN87]
$AC1$	P [BKN]	P [KN862]
$FPCSG$	decidable [KN87]	decidable [KN87]
$FCSG$	NP-complete [KN862]	NP-complete [KN862]
$FCSGI$	P [KN87]	P [KN87]
$FCMI$	P [KN87]	P [KN87]
$FPAG$	P [KKN]	P [KKN]
FBR	NP-complete [KKN]	NP-complete [KKN]
$FPBR$	NP-hard [KKN]	NP-hard [KKN]
FPA	NP-complete [K]	NP-complete [K]
SR	NP-complete [KN87]	NP-complete [KN87]

for finitely presented commutative semigroups. *FPA* stands for arbitrary finitely presented algebras. If a finitely presented algebra does not have any relation, it is said to be freely generated. *FCSG* stands for finitely generated free commutative semigroups; *FCSGI* stands for finitely generated free commutative semigroups with idempotency; similarly, *FCMI* stands for finitely generated free commutative monoids with idempotency. *FBR* stands for finitely generated free boolean rings.

SR is a theory presented by a finite complete (canonical) term rewriting system in which for each rule, the right-hand-side is either a ground term or a subterm of the left-hand-side.

References

[AT] Arnborg, S. and Tiden, E., "Unification problems with one-sided distributivity," Proc. of the first Conference on *Rewriting Techniques and Applications*, Dijon, France, May 1985, LNCS 202, Springer Verlag, New York, 398-406.

[BKP] Benanav, D., Kapur, D., and Narendran, P.,"Complexity of matching problems," Proc. of the first Conference on *Rewriting Techniques and Applications*, Dijon, France, May 1985, LNCS 202, Springer Verlag, New York, 417-429.

[CK] Chandra, A., and Kanellakis, P., personal communication, 1985.

[GJ] Garey, M.R., and Johnson, D.S., *Computers and Intractibility: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, San Francisco, 1979.

[KKN] Kandri-Rody, A., Kapur, D., and Narendran, P., "An ideal-theoretic approach to word problems and unification problems over finitely presented commutative algebras," Proc. of the first Conference on *Rewriting Techniques and Applications*, Dijon, France, May 1985, LNCS 202, Springer Verlag, New York, 345-364.

[KN861] Kapur, D., and Narendran, P., "NP-completeness of the set unification and matching problems," Eighth International Conference on *Automated Deduction*, Oxford, England, July 1986, LNCS 230, Springer Verlag, New York, 489-495.

[KN862] Kapur, D., and Narendran, P., "NP-completeness of the associative-commutative unification and related problems," Unpublished Manuscript, Computer Science Branch, General Electric Corporate Research and Development, Schenectady, NY, Dec. 1986.

[KN87] Kapur, D., and Narendran, P., "Matching, unification and complexity," to appear.

[K] Kozen, D., "Complexity of finitely presented algebras," Report No. 76-294, Dept. of Computer Science, Cornell University, Ithaca, NY. cited in Garey and Johnson, p. 253.

[PW] Paterson, M.S., and Wegman, M.N., "Linear unification," *J. of Computer and System Sciences*, 16, 1978, 158-167.

[S] Sethi, R., cited in Garey and Johnson, p. 252.