UNIFICATION AS A COMPLEXITY MEASURE
FOR LOGIC PROGRAMMING

by

A. Itai and J.A. Makowsky

Technical Report #301
November 1983
UNIFICATION AS A COMPLEXITY MEASURE
FOR LOGIC PROGRAMMING

A. Itai

J.A. Makowsky

Department of Computer Science,
Technion - Israel Institute of Technology,
Haifa, Israel

Abstract

Unification complexity of Horn clause programs is introduced and its complexity is investigated for various classes of universal Horn formulas. A faithful simulation theorem is proved which associates with every \( k \)-tape Turing machine a Horn clause program requiring exactly as many unification steps as the Turing machine. From this it follows that Horn clause programs are computationally complete even in the case of Bi-Horn (=Krom) formulas; and that the unification complexity of Horn clause programs is not recursively bounded. The faithful simulation theorem is also used to give a new interpretation to hierarchy theorems in the context of logic programming.

Keywords: Logic programming, Hierarchy Theorems, Horn formulas.
1. INTRODUCTION

In logic programming, as for example in the programming language PROLOG, a program is a set of clauses. PROLOG and other Horn Clause Logic Programming Languages restrict their formulas to Horn clauses, i.e. to clauses of the form

\[ A_1 \land \cdots \land A_m \rightarrow B \]

where \( A_i \) and \( B \) are either atomic or the literals true or false.

The computation of the program consists of finding an assignment which satisfies the clauses. Two basic operations used are unification (of terms and clauses) to find assignments and resolution of clauses to show that a set of thus obtained clauses is unsatisfiable.

A straightforward implementation of resolution for Horn clauses requires \( O(n^3) \) time. However, resolution can be shown by other (more efficient) methods. Indeed, we suggest a linear uniform-time RAM algorithm. This linear algorithm may be of independent interest. It was folklore knowledge that \( O(n^2) \), or possibly even \( O(n \log n) \), algorithms exist, but, to the best of our knowledge, the existence of a linear algorithm is new.

As for the other operation, Paterson and Wegman [PW] have suggested a linear uniform-time RAM algorithm for unification. Since each unification is usually followed by a satisfiability step \( U \), the number of unification steps, is a natural complexity measure for logic programs. Furthermore, since both unification and satisfiability have linear RAM algorithms the RAM uniform time-complexity is bounded by \( nU \).

Shapiro [Sh] and Lingas [Li] discussed the number of nodes and depth of the resolution tree. Their approach sheds light on the complexity of resolution rather than on the inherent complexity of proving satisfiability of Horn clauses.
Our main result, the Faithful Simulation Theorem, shows that we can simulate Turing machines by Horn clause logic programs such that each step of the Turing machine corresponds exactly to a single unification step in the execution of the logic program. The emphasis here is on the execution time since Tarnlund [T] has already shown that the simulation can be done in polynomial time. In the other direction, logic programs can be simulated by Turing machines in polynomial-time, though we did not compute the exact polynomial $P_{\text{Horn}}(n)$. It should be noted here, that a corresponding Faithful Simulation Theorem also holds for many other programming languages, such as PASCAL. It is surprising that in the case of Horn clause logic programs the number of unification steps corresponds so naturally to the number of steps of the Turing machine computation.

There are several applications of the Faithful Simulation Theorem: First we show that in general the number of unifications needed in a Horn clause program is not recursively bounded. Note that for this result a Polynomial Simulation Theorem suffices. Next we apply recent hierarchy theorem (Paul 1978 [P], Fürer 1983 [FÜ], Moran 1981 [Mo]) to construct Horn clause programs the number of unification of which is nearly optimal in the following sense:

For every function $f$ of a large class (the fully time constructible functions) there is a Horn clause logic program $\varphi_f$ which requires $f$ unification steps and no other Horn clause logic program accepts the same ground clauses in time $g(n)=o(P_{\text{Horn}}(f(n)))$.

For this result the Faithful Simulation Theorem is needed in its full strength.

We also give a similar theorem involving oracles. In the context of Horn clause programs it seems rather natural to consider oracles, since they correspond to sets of ground clauses which can be viewed as previously obtained data. Here we can even omit the polynomial $P_{\text{Horn}}$ and construct the Horn clause program uniformly for any r.e. class of fully time constructible functions.
Finally we look at the number of unification required for function-free Horn clause programs. In this case, unification reduces to instantiation. Using our linear algorithm for unsatisfiability of propositional Horn clauses and an exponential lower bound for the consequence problem for function-free Horn clauses [CLM] we construct function-free Horn clause programs which require an exponential number of unification steps.

In detail the paper is organized as follows: In section 2 we prove the Faithful Simulation Theorem. In section 3 we give the applications to lower bounds on unification complexity. In section 4 we give the linear algorithm for the satisfiability problem for propositional Horn clauses and for quantifier-free Horn clauses without function symbols. In section 5 give lower bounds for the unification complexity for function-free Horn clause programs. Finally, in section 6 we discuss some possible improvements and open problems.

It is assumed that the reader is familiar with the basics of automated theorem proving as presented in [CL], [LP] or [Ma].

We wish to thank O.Goldreich, who had presented to us an $O(n \log n)$ algorithm for the satisfiability problem of propositional Horn clauses, as a solution to a problem posed in a logic course. It was this algorithm which arouse our interest in the subject.

2. FAITHFUL SIMULATION

Let $T$ be any first order theory with a recursive set of axioms, and $\varphi$ a formula over $T$. $T$ refutes $\varphi$ if no model of $T$ satisfies $\varphi$. A Horn formula is bi-Horn if it contains at most two literals. $T$ is bi-Horn if all its axioms are bi-Horn formulas.
2.1 The Construction

Let $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ be a deterministic $k$-tape TM with alphabet $\Sigma$, state space $Q$, transition function $\delta$, start state $q_0$, and accepting states $F$. We also assume that there are no transitions from an accepting state. We shall construct a bi-Horn first order theory $T_M$, the terms of which encode the tapes of $M$, and atomic formulas consist of a $(2k+1)$-ary relation $R$. The axioms of $T_M$ encode the transition function $\delta$.

In order to encode the tape $a_1 \cdots a_n \in \Sigma^*$ with the head at position $i$, we shall create two terms, one representing $a_1 \cdots a_{i-1}$ and the other $a_i \cdots a_n$. Let $c$ be a binary function. Then $a_1 \cdots a_{i-1}$ is encoded as $c(a_{i-1}, c(a_{i-2}, \ldots, c(a_1, \phi) \ldots)$; whereas, $a_i \cdots a_n$ is encoded as $c(a_i, c(a_{i+1}, \ldots, c(a_n, \$) \ldots)$, where $\phi$ and $\$ are new symbols (i.e. $\phi; \$ $\Sigma$). We shall avoid this cumbersome notation and write $a_{i-1}a_{i-2} \cdots a_i \phi$ and $a_i a_{i+1} \cdots a_n \$ for short. (Note that the tape left of the head is reversed.)

For notational convenience we shall assume $k = 1$ then the atomic formula $R(q, a_{i-1} \cdots a_i \phi, a_i \cdots a_n \$) encodes the configuration in which the control is in stage $q$, the tape contains $a_i \cdots a_n$ and the head is at position $i$.

Since $M$ is deterministic, the transition is uniquely determined by the current state $q$, the letter seen by the head, ($a = a_i$). If $\delta(q, a) = (q', a', 1)$ (the head moves right) $(a_{i-1} \cdots a_i \phi, a_{i+1} \cdots a_n \$) becomes $(a' a_{i-1} \cdots a_i \phi; a_{i+1} \cdots a_n \$) such a transition is reflected by the axiom:

$$R(q, x, ay) \rightarrow R(q', ax, y).$$

If $\delta(q, a) = (q', a', -1)$ (the head moves left), $(ba_{i-2} \cdots a_i \phi, a' a_{i+1} \cdots a_n \$) becomes $(a_{i-2} \cdots a_i \phi, b a' a_{i+1} \cdots a_n \$), and the transition is reflected by the axioms:

$$R(q, bx, ay) \rightarrow R(q', x, ba'y)$$

Thus, when the head moves to the right we have one axiom and when it moves left $|\Sigma|$ axioms.
Finally, we add the axioms:

\[ \neg R(q, x, y) \text{ for all } q \in F. \]

Note that the number of axioms of \( T_M \) is \( O(|\Sigma| \cdot |M|) \), where \( |M| \) is the number of bits required to represent \( \delta \).

In order to simplify notation let \( \Sigma \) the alphabet of all Turing machines discussed in this paper and \( \Sigma \cap \{\emptyset, \$\} = \emptyset \). Consequently, the set of constants of the theory \( T_M \) is \( \Sigma \cup \{\emptyset, \$\} \). Let

\[ \varphi_a = R(q, \emptyset, \alpha \$). \]

Note that \( \varphi_a \) is independent of \( M \).

2.2 Unification Complexity

Let \( t_1, \ldots, t_n \) be constant terms, then \( \varphi(t_1, \ldots, t_n) \) is a ground instance of \( \varphi \). Herbrand theorem asserts that \( T \) refutes \( \varphi \) iff there exist ground instances \( \varphi_1, \ldots, \varphi_k \) of \( \varphi \) such that \( \bigwedge_{i=1}^k \varphi_i \) is unsatisfiable over \( T \). This reduces the original problem to that of propositional calculus. The clauses are found by the unification algorithm. The Unification length of \( \varphi \), \( U_T(\varphi) \), is the minimum \( k \) for which \( k \) such ground instances exist \( (U_T(\varphi) = \infty \text{ if } \varphi \text{ is satisfiable}) \). The unification algorithm is used to find instances \( \psi_1, \psi_2, \ldots, \psi_k \). However, these instances may contain variables. The Completeness Theorem for the unification methods asserts that \( T \) refutes \( \varphi \) iff \( \bigwedge_{i=1}^k \psi_i \) is unsatisfiable in the propositional calculus. In Section 4 a linear algorithm will be shown for the satisfiability of Horn formulas in the propositional calculus.

\( T \models \varphi \), \( \varphi \) is a consequence of \( T \) if \( \varphi \) is true in every model of \( T \). \( T \) is decidable in time \( t \) if there exists a TM which can solve the consequence problem \( T \models \varphi \) in time \( \leq t(|\varphi|) \), where \( |\varphi| \) is the length (in bits) of \( \varphi \).
Theorem 1 (Faithful simulation): Let $M$ be a $k$-tape deterministic Turing machine then $M$ accepts $\alpha \in \Sigma^*$ iff $T_M$ refutes $\varphi_\alpha$. Moreover, if $\alpha$ is accepted, then the number of steps required is $U_T(\varphi_\alpha) - 1$.

Proof: If $R(q, \beta_1, \beta_2) \rightarrow R(q', \beta_1', \beta_2')$ is the result of the unification of $R(q, \beta_1, \beta_2)$ with an axiom, then $R(q, \beta_1, \beta_2)$ encodes a configuration and $R(q', \beta_1', \beta_2')$ encodes the unique configuration which follows it in one step. (There is only one possible transition since $M$ is deterministic.)

Assume there exists a refutation of $\varphi_\alpha$. At each unification step (except the last one) there is a ground clause $R(q, \beta_1, \beta_2)$ encoding a configuration which is unified with one of the axioms to yield the ground clause

$$R(q, \beta_1, \beta_2) \rightarrow R(q', \beta_1', \beta_2').$$

Resolution now yields the new ground clause $R(q', \beta_1', \beta_2')$ in a single step.

In the last step of the proof the new ground clause is the empty one. Thus each unification, except the last one, uniquely corresponds to a transition in the accepting computation.

The following corollary to Theorem 1 improves the results of Tarlund [T]:

Corollary: Bi-Horn formulas are computationally complete.

Let $Space(M, \alpha)$ be the space required by $M$ to accept $\alpha$, Clause-Length($T, \varphi, r$) be the length of the longest clause used in the refutation of $\varphi$ over $T$, and

$$\text{Clause-Length}(T, \varphi) = \min_r \{\text{Clause-Length}(T, \varphi, r)\},$$

where the minimum is taken over all refutations $r$.

Theorem 2 (Faithful simulation of space): Using the notation of Theorem 1 for all $\alpha \in \Sigma^*$ accepted by $M$

$$Space(M, \alpha) = \phi(\text{Clause-Length}(T_M, \varphi_\alpha)).$$

I.e. there exist constants $c_1, c_2 > 0$ such that
Proof: Since the ground clauses correspond to configurations, their lengths are proportional to the sum of the length of all the words on the tapes and the length of the largest ground clause is proportional to the space requirements.

3. APPLICATIONS OF THE MAIN CONSTRUCTION

Using the Theorems 1. and 2. together with various hierarchy theorems we can get additional results:

A function \( f \) is recursively bounded if there exists a recursive function \( g \) such that \( f(n) \leq g(n) \).

For an undecidable theory \( T \) \( U_T(\varphi) \) (as a function of \( \varphi \)) is not recursive (since \( U_T(\varphi) = \infty \) iff \( \varphi \) is satisfiable).

To use Herbrand's theorem to show that a formula \( \varphi \) is unsatisfiable one should find \( k \) ground instances \( \varphi_1, \ldots, \varphi_k \), such that \( \bigwedge_{i=1}^{k} \varphi_i \) is unsatisfiable. If there were a recursive bound on the total length \( \sum_{i=1}^{k} |\varphi_i| \) then the problem would be decidable. Thus, \( \sum_{i=1}^{k} |\varphi_i| \) is not recursively bounded. The next theorem shows that even \( k \), the number of ground instances, cannot be recursively bounded.

Theorem 3 Let \( F \) be the theory which contains only the axioms of first order logic, then \( U_F(\psi) \) is not recursively bounded.

Proof: Given a DTM \( M \) and \( \alpha \in \Sigma^* \) there is no recursive function \( g(|M|,|\alpha|) \) such that the time required for \( M \) to accept or reject \( \alpha \) is bounded by \( g \).

Use the main construction to construct \( T_M \). \( T_M \) has a finite set of axioms, let \( \xi_M \) be the conjunction of these axioms.
$T_M /\not= \emptyset$ if $x /\not= \xi M \rightarrow \emptyset \varphi$. (1)

If $M$ accepts $\alpha$, $\varphi_{aM}$ is unsatisfiable in any model of $T_M$. Thus, in every model of $T_M$ $\neg \varphi_{aM}$ is satisfied, i.e. $T_M /\not= \neg \varphi_{aM}$. Using (1) $M$ accepts $\alpha$ iff $\not= \xi M \rightarrow \neg \varphi_{aM}$.

Or $M$ accepts $\alpha$ iff $\neg \neg (\xi_M \rightarrow \neg \varphi_{aM})$ is unsatisfiable. Let $\psi = \xi M \land \neg \varphi_{aM}$. Since $\psi$ is equivalent to $\neg (\xi_M \rightarrow \neg \psi_{aM}), \psi$ is unsatisfiable iff $M$ accepts $\alpha$.

Any unification-resolution proof of $\psi$ over $F$ is a unification-resolution proof of $\varphi_{aM}$ over $T_M$. Thus

$$U_F(\psi) = U_{T_M}(\varphi_{aM}) = 1 + \text{time}(M; \alpha).$$

$U_F$ is not recursively bounded since $\text{time}$ (the number of steps required by $M$ to accept $\alpha$) is not.

A function $f(n)$ is fully time constructible if there exists a $TM$ that for each input of length $n$ requires exactly $f(n)$ steps.

**Theorem 4** Let $f$ be a fully time constructible functions then there exists a theory $T_f$, the unification complexity of which is $f$, but $T_f /\not= \emptyset$ cannot be decided by any deterministic $TM$ with time complexity $g(|\varphi|)$, where $g(n) = o(f(n))$.

The proof follows immediately from Theorem 1 and the following theorem by Fürer:

**Theorem (Fürer):** Let $f(n) > n$ be a fully time constructible functions then there exists a language which is accepted in time $f(n)$ but not accepted in time $g(n) = o(f(n))$.

**Remark 1** Using space hierarchy results [P] and Theorem 2, one can prove that for every space constructible function $f(n) \Rightarrow n$ there exists a theory $T_f$, for which the length of the longest term in the unification proof of $T_f /\not= \emptyset$ is $f(|\varphi|)$ and $T_f /\not= \emptyset$ cannot be decided by any TM of space complexity $g(n) = o(f(n))$. 


A proof system is essentially a NDTM. Any NDTM of time complexity $t(n) \rightarrow n$ can be simulated by a DTM of time $a^t(n)$ (for some $d>1$). Let $f$ be a fully time constructible functions then there exists a theory $T_f$ whose unification complexity is $f$, and the time complexity of any proof system is $\log_d f$.

Moreover, if $f$ grows fast enough, i.e. $f(n) > I(\log_d n)^*$, then $\log_d f(n) \rightarrow f(n/d)$, then the complexity of any proof is essentially $f$.

Theorem 5 Let $\tau$ be a r.e. set of fully time constructible functions then there exists a theory $\Delta$ such that for each $f(n) \in \tau$ there exist a formula $\psi$ such that

- $U_\Delta(\psi \rightarrow \varphi) \leq f(|\varphi|)$
- Any NDTM $M$ with oracle $\Delta$, which solves the decision problem $\Delta \models \psi \rightarrow \varphi$ requires at least $f(|\varphi| - 3)$ time.

The theorem follows immediately from Theorem 1 and the following theorem.

Theorem (Moran) Let $\tau$ be a r.e. class of fully time constructible functions. Then there exists an oracle set $E$ such that for each $t(n) \in \tau$ there exists a language $L_t$ such that $L$ can be recognized by a DTM with oracle $E$ in time $t(n)+2$, but by no NDTM with oracle $E$ in time $t(n)$.

4. A LINEAR ALGORITHM

We now turn to the linear algorithm to check satisfiability of a set of Horn formulas. A basic Horn formula over the propositional variables $\{P_0, \ldots, P_n\}$ is a disjunction of literals of which at most one is positive. We wish to check satisfiability of a set $\varphi$ of basic Horn formulas. A quadratic algorithm has been known though we have not been able to trace its originator. Here we present

$I(m)$ is the iterated exponential, i.e. $I(0)=1$ and $I(m+1)=2^{I(m)}$
a linear algorithm.

Any basic Horn formula can be written in one of the following forms:

\[ H_1 : Q_1 A \cdots \wedge Q_m \rightarrow R_i \]

\[ H_2 : Q_j \]

where each \( Q \) is some \( P_h \) and \( R_i \) is either some \( P_j \) or the constant false, which will be denoted \( Q_0 \).

A true variable is a variable which must be assigned the value true in any satisfying assignment. All variables appearing in \( H_2 \)-formulas are true. The algorithm replaces the original \( \varphi \) by equivalent sets. The new sets are derived by either deleting a basic Horn formula whose right-hand-side is a true variable, or by deleting a true variable from a the left-hand-side of an \( H_1 \)-formula. If the entire left-hand-side is deleted then the variable on the right-hand-side becomes a true variable. \( \varphi \) is unsatisfiable if \( Q_0 \) becomes true. The algorithm terminates successfully when no true variable appears in any \( H_1 \)-formula. Then \( \varphi \) can be satisfied by assigning the value false to all variables appearing in \( H_1 \)-formulas.

To implement the above algorithm we should be able to do the following operations:

(i) Find the next true variable to be processed.

(ii) Delete all occurrences of a true variable.

(iii) Delete an \( H_1 \)-formula the right-hand-side of which is a true variable.

To implement (i) we shall maintain a queue of all unprocessed true variables, such that adding and removing a single variable takes constant time.

The right-hand-side of the \( H_1 \)-formulas are represented by a vector. As for the left-hand-side, we first sort the left-hand-side of each \( H_1 \)-formula by index of the variables. Using bucket-sort the entire reordering requires \( O(|\varphi|) \) time. Now the left-hand-side of the \( H_1 \)-formulas can be represented by a sparse matrix.
whose rows represent formulas and whose columns represent variables. The matrix consists of nodes, one for each occurrence of a variable, with pointers to the next variable in the formula and to the next occurrence of the variable. Pointers are maintained to the first variable of each formula and to the first and last occurrence of each variable. (This last pointer is required only to create the structure in linear time and can be discarded later.)

The number of nodes is linear in the length of \( \varphi \). Since each node can be deleted at most once, and the deletion of a node requires constant time, the entire algorithm is linear.

This proves the following:

**Theorem 6** There is a linear decision procedure for the satisfiability of propositional Horn formulas.

Let \( P=\{p_0, p_1, p_2, \ldots \} \) be the set of propositional variables, \( A=\{a_0, a_1, a_2, \ldots \} \) the set of atomic formulas over some countable similarity type and \( \varphi \) some quantifier-free formula over the similarity type. If the set of atomic formulas of \( \varphi \) is \( A_\varphi \), then \( \varphi \) is logically equivalent to \( \varphi^P \) obtained from \( \varphi \) by replacing each atomic formula \( a_j \) by the propositional variable \( p_j \).

However, this transformation is not satisfactory from a complexity point of view, since the maximum index of the variables might be very large (consider, for example, the case where the space depends on the maximum index). To remedy this problem, the atomic formulas \( A_\varphi \) are replaced by the propositional variables \( \{p_0, p_1, \ldots, p_r\} \), such that \( p_j \) replaces the \( j \)-th formula in the lexicographic ordering of \( A_\varphi \). To find the lexicographic index of the \( a_j \)'s we construct a binary tree corresponding to their binary representation. The lexicographic index of the \( a_j \)'s can be found by traversing the tree in infix order. The entire transformation involves two passes over \( \varphi \) and one tree traversal, thus \( O(|\varphi|) \) time. Note also that if \( a_j \) is renamed \( p_k \) then \( |p_k| \leq |a_j| \). Thus \( |\varphi^P| \leq |\varphi| \).
Theorem 7 There is a linear decision procedure for the satisfiability of quantifier-free Horn formulas with no function symbols.

5. THE UNIFICATION COMPLEXITY OF FUNCTION-FREE HORN FORMULAS

The main construction depends heavily on the simulation of concatenation by a binary function symbol. The same effects could be achieved by allowing constants and existential quantifiers.

A theory with neither function symbols nor existential quantifiers is decidable [DG]. Regarding its complexity we have the following theorem.

Theorem (CLM) There is a constant \( d > 1 \) such that the consequence problem, \( T \not\vdash \varphi \), for universal Horn formula with relation symbols and constants (but no function symbols) cannot be solved in time \( d^{|\varphi|} \) (the result holds even for atomic \( \varphi \)).

Using this theorem and Theorem 7 we can show:

Theorem 8 There is a constant \( d > 1 \) and a theory \( T \) such that \( U_T(\varphi) > d^{|\varphi|} \), with \( \varphi \) as in Theorem CLM.

In the proof of Theorem CLM and hence for Theorem 8 relation symbols of unbounded arity are used. In fact, in the formula of length \( O(n) \) relation symbols of arity \( O(n \log n) \) are used. If one allows relation symbols of arity \( O(n) \), Theorem 8 can be proved directly in the following way (as suggested by H. Lewis): Write a formula of arity \( O(n) \) with an \( n \)-ary relation symbol \( P \) and two constants 0, 1 which says:

\[ P(0,0, \ldots, 0) \text{ is true} \]
\[ P(1,1, \ldots, 1) \text{ is false} \]

and if \( P(y_1, \ldots, y_n) \) is true and \( y_1 \cdots y_n \) is the binary notation for \( k \) and \( z_1 \cdots z_n \) is the binary notation for \( k+1 \) then \( P(z_1, \ldots, z_n) \) is true. This can be done by a
formula of length $O(n)$ using coding tricks, also H. Lewis has shown that $\Omega(n^n)$
ground instances are needed to show its inconsistency.

6. IMPROVEMENTS, OPEN PROBLEMS AND COMMENTS

In both proofs of Theorem 8 the arity of the relation symbols is not bounded.
However, it is preferable to keep the arity of the relation symbols bounded. As in
descriptive geometry, where every $n$-ary relation can be represented faithfully
by $n-1$ binary relations (the various projections on the planes), allowing addi-
tional constant symbols one ternary relation symbol suffices. The length of the
formula obtained is quadratic in the arity $n$.

Hence we have the following problems:

Problem 1 Is $d^n$ a lower bound for $HS(\varphi)$ if $\varphi$ contains only one fixed relation sym-

Problem 2 Is $d^n$ a lower bound for $HS(\varphi)$ if both the number of constant symbols,
the number and arity of the relation symbols are bounded?

For a class $\Sigma$ of formulas which is decidable in $f(|\varphi|)$ time our examples
always imply that $H(\varphi)$ is bounded $O(f)$.

Problem 3 Is there a class $\Sigma$ of formulas which is decidable in $f(|\varphi|)$ time but
$H(\varphi)$ is not bounded by $O(f)$?

Note that M. Wietlisbach [Wie] has shown that there are formulas in which
both the Herbrand size and the resolution part are exponential, provided the
resolutions are regular (Galil, Tseitin, cf. [Ga]).

The evaluation problem is checking the validity of a formula of the following
form:

$$Q_1x_1Q_2x_2\cdots Q_nx_n\varphi(x_1,x_2,\ldots,x_n),$$

where $Q_i$ are alternately existentially and universal quantifiers, $x_i$'s are proposi-
tional variables and $\varphi$ is quantifier free propositional formula. Apsvall Plass and
Tarjan [APT] showed a linear algorithm for the case where $\varphi$ is in conjunctive normal form with at most 2 literals per clause. We pose the following open problem:

**Problem 4** What is the complexity of the evaluation problem where $\varphi$ is a propositional Horn formula.

References


