

## On the Existence of Polynomial Time Algorithms for Interpolation Problems in Propositional Logic

E. DAHLHAUS, A. ISRAELI, and J. A. MAKOWSKY\*

### *1 Introduction*

**1.1 The interpolation problem** Let  $\phi$  and  $\psi$  be propositional formulas with  $\phi(\bar{x}, \bar{y}), \psi(\bar{y}, \bar{z})$  where  $\bar{x}, \bar{y}, \bar{z}$  are disjoint lists of propositional variables. In other words, the only variables  $\phi$  and  $\psi$  have in common are the variables of  $\bar{y}$ . Let us assume further that  $\phi \wedge \psi$  is unsatisfiable. The question is whether there is a propositional formula  $\theta(\bar{y})$ , built only from the variables shared by  $\phi$  and  $\psi$ , such that

- (i)  $\phi \rightarrow \theta$  and
- (ii)  $\theta \rightarrow \neg\psi$

are valid.

The interpolation theorem states that this is always the case. We call such a  $\theta$  the Craig interpolant (in the sequel shortly interpolant) for  $\phi$  and  $\psi$ . (Note that we deviate from traditional usage in that one usually calls this the interpolant for  $\phi$  and  $\neg\psi$ ). The interpolation theorem was first stated and proved for first-order logic by Craig [9]. For a discussion of the first-order case without equality the reader is referred to [32]. The statement for propositional logic follows from Craig's theorem trivially. One can also give direct proofs, e.g. by looking at the Boolean function defined by  $\Theta = \exists \bar{x}\phi(\bar{x}, \bar{y})$ . By functional completeness of propositional logic there is a formula  $\theta(\bar{y})$  which represents  $\Theta$ . It is easily seen that  $\theta$  satisfies (i) and (ii). There is a vast literature about various

---

\*The results of this paper were obtained in 1984. We would like to thank G. Buntrock, A. Itai, and two anonymous referees for their extremely valuable comments.

interpolation theorems for many kinds of propositional and higher-order logics, such as propositional temporal logics, intuitionistic logics and modal logics, and first-order logics extended by generalized quantifiers or infinitary operations. For an extensive bibliography the reader is referred to [22] and [5]. Complexity issues connected with the interpolation theorem were first studied by Kreisel and Friedman for the first-order case and Mundici for the propositional case [10],[24].

The purpose of this paper is to study the complexity of various interpolation problems in possibly quantified propositional logic. We continue here the investigations started by Mundici in [24]–[27] and relate the interpolation problems to the complexity hierarchies between deterministic logarithmic space (**L**) and deterministic polynomial space (**PSpace**).

**1.2 The quantified interpolation problem** We now introduce the interpolation problems for quantified Boolean formulas, i.e. formulas of the form  $Q_1x_1, \dots, Q_nx_n\phi$ . Here  $Q_i$  is  $\exists$  or  $\forall$  and  $\phi$  is any quantifier-free Boolean formula. For any (quantified) Boolean formula  $\phi$ ,  $\exists(\forall)x\phi$  is interpreted as “for some  $x \in \{0,1\}$   $\phi$  is true” (respectively “for all  $x \in \{0,1\}$   $\phi$  is true”). Let  $\Sigma(0) = \Pi(0)$  be the class of quantifier-free Boolean formulas. We define  $\Sigma(k+1)$  as the set of all formulas of the form  $\exists x_1\exists x_2 \dots \exists x_n\phi$  with  $\phi \in \Pi(k)$  and  $\Pi(k+1)$  by looking at formulas of the form  $\forall x_1\forall x_2 \dots \forall x_n\phi$  with  $\phi \in \Sigma(k)$ . Finally  $\Sigma^*$  is the union of all  $\Sigma(k)$ . Note that  $\Sigma^*$  are all quantified Boolean formulas. The  $\Sigma(k)$  interpolation problem,  $\Sigma(k)$ -INT is defined as follows: Given two formulas  $\phi(\bar{x}, \bar{y}), \psi(\bar{y}, \bar{z}) \in \Sigma(k)$  with disjoint vectors of free variables  $\bar{x}, \bar{y}, \bar{z}$  such that their conjunction is not satisfiable, find a quantifier-free propositional formula,  $\theta(\bar{y})$ , with its free variables all among  $\bar{y}$ , such that both

- (i)  $\phi \rightarrow \theta$  and
- (ii)  $\theta \rightarrow \neg\psi$

are valid.

$\theta(\bar{y})$  is called an *interpolant* for  $\phi(\bar{x}, \bar{y}), \psi(\bar{y}, \bar{z})$ . The  $\Pi(k)$  interpolation problem,  $\Pi(k)$ -INT and  $\Sigma^*$ -INT are defined in the obvious way. Analogously we define the  $\Sigma(k)$  and  $\Pi(k)$  satisfiability problem, which we denote by  $\Sigma(k)$ -SAT and  $\Pi(k)$ -SAT respectively.

By a similar argument as above for the quantifier-free case there is always a quantifier-free interpolant for the  $\Sigma^*$  interpolation problem. Consider two  $\Sigma(k+1)$ -formulas  $\phi = \exists \bar{v}\phi_1(\bar{v}, \bar{x}, \bar{y})$  and  $\psi = \exists \bar{w}\psi_1(\bar{w}, \bar{y}, \bar{z})$ , where  $\phi_1$  and  $\psi_1$  are  $\Pi(k)$ -formulas. Then the interpolation of  $\phi(\bar{x}, \bar{y})$  and  $\psi(\bar{y}, \bar{z})$  and of  $\phi_1(\bar{v}, \bar{x}, \bar{y})$  and  $\psi_1(\bar{w}, \bar{y}, \bar{z})$  are trivially equivalent problems. That means that  $\Sigma(k+1)$ -INT and  $\Pi(k)$ -INT are polynomially equivalent problems.

There are various ways of measuring the complexity of the interpolation problem: We can look at the length (or some other size) of the shortest interpolant or at the complexity of the most efficient computation of an interpolant, which then possibly is bigger than the shortest interpolant. Formulas can be represented as trees or as directed acyclic graphs (DAGs). The leaves or the bottom vertices (vertices of indegree 0) are labeled by variables and all other vertices are labeled by  $\wedge$ ,  $\vee$ , or  $\neg$ . Any vertex labeled by  $\neg$  has an indegree of 1. Any vertex labeled by  $\wedge$  or  $\vee$  has an indegree of two. We consider only DAGs

with exactly one top vertex (outdegree is 0). Then the interpretation of a formula in DAG description is the canonical interpretation of the top vertex. Note that formulas in DAG form are nothing else than switching circuits. We also call formulas in DAG form *reduced formulas*. We say that  $\Sigma(k)$ -INT can be computed in polynomial time ( $\Sigma(k)$ -INT can be computed in logarithmic space) and we write, by abuse of notation,  $\Sigma(k)$ -INT<sub>red</sub> ∈ **P**, ( $\Sigma(k)$ -INT<sub>red</sub> ∈ **L**), if there is a polynomial time (logarithmic space) algorithm which gives us an interpolant possibly in DAG-representation. Note that if the input implication is not a tautology, then for an algorithm computing INT<sub>red</sub> or INT no condition is imposed on the output. We denote  $\Sigma(0)$ -INT<sub>red</sub> also by INT<sub>red</sub>. We omit the index in INT<sub>red</sub>, if we require that the interpolant be given in tree representation. For the case that the input implication  $\phi \rightarrow \psi$  also has a DAG-representation we introduce new variables for each DAG-vertex. But then we have an equivalent interpolation problem such that the input implication is in tree form. That means INT<sub>red</sub> is not harder if the input implication is in DAG-form. Note that a polynomial time algorithm always produces an output whose size is polynomial in the input. Therefore, in order to show that  $\Sigma(k)$ -INT<sub>red</sub> ∉ **P**, it suffices to give an exponential lower bound for the shortest interpolant in DAG-representation. On the other hand, if there is a polynomial  $P_k$  which gives a bound on the shortest  $\Sigma(k)$ -interpolant then  $\Sigma(k)$ -INT ∈ **PSpace**. This follows from the fact that **PSpace** = **NPSpace** and that  $\Sigma^*$ -SAT is in **PSpace** [23],[11].

**Conjecture** *There is an exponential lower bound for the shortest  $\Sigma^*$ -interpolant in DAG-representation.*

The conjecture, actually, implies that **P** ≠ **PSpace**. This will be clear from the remark above and Theorem 8 at the end of Section 2.

**1.3 The complexity hierarchies** The complexity classes lying between **L** and **PSpace** which are usually considered are ordered by known inclusions:

$$\mathbf{L} \subset \mathbf{NL} \subset \mathbf{P} \subset \mathbf{NP} = \Sigma^1 \mathbf{P} \subset \Sigma^k \mathbf{P} \subset \Sigma^* \mathbf{P} \subset \mathbf{PSpace}.$$

Here **NL** is nondeterministic logarithmic space, **P** is polynomial time, **NP** is nondeterministic polynomial time, and  $\Sigma^* \mathbf{P} = \bigcup_k \Sigma^k \mathbf{P}$  is the polynomial time hierarchy. The latter is sometimes called the Stockmeyer hierarchy. It is known that **L** ≠ **PSpace** and therefore that **L** ≠ **P** or **P** ≠ **PSpace** but it is not known which of the inequalities hold [11]. We denote the class of problems, whose complement is in **NL** or **NP** by **CoNL** or **CoNP** respectively. By a recent result of Immerman [13] it was shown that **NL** = **CoNL**. The problem of the complement closure of **NP** remains open.

**1.4 Summary** The paper is organized as follows: In Section 2 we present the general theorems and all our results concerning the complexity of  $\Sigma(k)$ -INT<sub>red</sub>. We begin with an easy remark on separation from which we can deduce several corollaries.

In Section 3 we study the complexity of *HORN-INT*, the interpolation problem for quantifier-free propositional Horn formulas, i.e. formulas in conjunctive normal form whose disjuncts contain at most one nonnegated variable.

We present also a polynomial time algorithm for the interpolation problem for (unquantified) propositional Horn formulas. The main result here is Theorem 10, which asserts that *HORN-INT* is computable in polynomial time.

In Section 4 we study the complexity of *KROM-INT*, the interpolation problem for quantifier-free propositional Krom formulas, i.e. formulas in conjunctive normal form whose disjuncts contain at most two literals. Our main result here is a nondeterministic logspace algorithm for the interpolation and quantifier elimination problem for quantified propositional Krom formulas (Theorem 17). This last result is based on Tarjan's linear time algorithm for strongly connected components of a graph and its application to the satisfiability problem for quantified Krom formulas in [4].

Interpolation for Krom formulas was first studied by Krom in [20]. Complexity issues arising from Horn and Krom formulas were first studied by Börger in [6] and Aanderaa in [1]. The reader is also referred to [2],[3].

In the light of the hierarchies between **P** and **PSpace** our results in these sections can be summarized in Table 1. In the table each line gives three equivalent statements and the corresponding number of the statement.

Table 1.

	Interpolation	Satisfiability	
<b>P = PSpace</b>	$\Sigma^*-INT_{red} \in \mathbf{P}$	$\Sigma^*-SAT \in \mathbf{P}$	Theorem 8
<b>P = NP</b>	$\Sigma(k)-INT_{red} \in \mathbf{P}$ $\Sigma(2)-INT_{red} \in \mathbf{P}$ <b>NP = CoNP</b> and $INT_{red} \in \mathbf{P}$	$SAT \in \mathbf{P}$	Theorems 6 and 7
<b>L = P</b>	$HORN-INT \in \mathbf{L}$	$HORN-SAT \in \mathbf{L}$	Corollary 11
<b>L = NL</b>	$KROM-INT \in \mathbf{L}$	$KROM-SAT \in \mathbf{L}$	Theorem 14
<b>L = NL</b>	$\Sigma^*-KROM-INT \in \mathbf{L}$	$\Sigma^*-SAT \in \mathbf{L}$	Corollary 19

## 2 Interpolation problems and the polynomial time hierarchy

**2.1 A general remark on separation** Let **C** be a (not necessarily deterministic) complexity class like **L**, **NL**, **P**, **NP**, **PSpace**; let **D**  $\subset$  **C** be a deterministic complexity class like **L**, **P**, or **PSpace**, and let  $\Phi$  be a class of quantified propositional formulas such that the following hold:

- (i)  $\Phi$ -SAT is **C**-complete via **D**-reductions, i.e. for each  $A$  recognizable in **C** we can find a function  $f \in \mathbf{D}$  into  $\Phi$ , such that  $f(x) \in \Phi$ -SAT iff  $x \in A$
- (ii)  $\Phi$ - $INT_{red}$  is in **D**.

Then each disjoint pair of sets  $A_1, A_2 \in \mathbf{C}$  can be separated by a  $B \in \mathbf{D}$  (i.e. there is  $B \in \mathbf{D}$  such that  $A_1 \subset B$  and  $A_2 \cap B = \emptyset$ ).

*Proof:* Let  $A_1, A_2 \subset \{0,1\}^*$  be in  $\mathbf{C}$  and  $A_1 \cap A_2 = \emptyset$ . By (i) there are functions  $f: \{0,1\}^* \rightarrow \Phi$  and  $g: \{0,1\}^* \rightarrow \Phi$  such that for every  $a \in \{0,1\}^*$

- (iii)  $f(a)$  is satisfiable iff  $a \in A_1$  and
- (iv)  $g(a)$  is satisfiable iff  $a \in A_2$ .

Furthermore, w.l.o.g. we can assume that

- (v)  $f(a)$  and  $g(a)$  have no variables in common.

$f(a)$  and  $g(a)$  are inconsistent. Since the only formulas with no free variables are **true**, **false**, the interpolant for the formulas  $f(a), g(a)$  is **true** or **false** and can be calculated in  $\mathbf{D}$ . Next we define  $B \subset \{0,1\}^*$  by  $a \in B$  iff the interpolant for  $f(a), g(a)$  is **true**. It is easy to check that  $B \in \mathbf{D}$  and  $B$  separates  $A_1, A_2$ .

We get the following immediate consequence using the fact that the satisfiability problem for unquantified propositional formulas is  $\mathbf{NP}$ -complete:

**Corollary 1** *Assume INT is polynomial time computable. Then*

- (i) *for every two disjoint sets  $A_1, A_2 \in \mathbf{NP}$  there is  $B \in \mathbf{P}$  separating  $A_1, A_2$  and*
- (ii)  $\mathbf{P} = \mathbf{NP} \cap \mathbf{CoNP}$ .

The next corollary makes use of the fact that *HORN-SAT* (*KROM-SAT*) is  $\mathbf{P}$ -complete ( $\mathbf{CoNL}$ -complete) via log space reductions (compare also Theorems 12 and 15, presented later).

**Corollary 2**

- (i) *Assume HORN-INT is computable in logarithmic space. Then  $\mathbf{L} = \mathbf{P}$ .*
- (ii) *Assume KROM-INT is computable in logarithmic space. Then:*
  - (a) *for every two disjoint sets  $A_1, A_2 \in \mathbf{CoNL}$  there is  $B \in \mathbf{L}$  separating  $A_1, A_2$ .*
  - (b)  $\mathbf{L} = \mathbf{NL} \cap \mathbf{CoNL} = \mathbf{NL}$ .

Remark: Corollaries 1 and 2 can be strengthened to (disjoint)  $INT_{red}$ ,  $HORN-INT_{red}$ , and  $KROM-INT_{red}$ .

**2.2 Consequences of  $\mathbf{P} = \mathbf{NP}$**  In [27] the following result is proved:

**Theorem 3** (Mundici) *At least one of the following statements is true:*

- (i)  $\mathbf{P} = \mathbf{NP}$ ;
- (ii)  $\mathbf{NP} \neq \mathbf{CoNP}$ ;
- (iii) *INT is not polynomial time computable.*

Now we present a converse of Corollary 1 and extensions thereof to the polynomial time hierarchy. It is also an improvement of the above result of [27]. For this purpose we need a special case of a result of Fischer and Pippenger, as stated in [28].

**Proposition 4** (Fischer and Pippenger) *For every polynomial time computable function  $f$  there is a uniform sequence of Boolean circuits  $C_n$  of size polynomial in  $n$  such that  $C_n$  computes  $f$  restricted to input length  $n$ .*

Recall that uniform here means that the  $C_n$ 's can be computed from  $n$  in  $SPACE(O(\log n))$ .

Recall that Boolean circuits are just a notational variant of DAG-formulas. Thus we get the following

**Proposition 5** *Let  $\Phi$  be a class of (possibly) quantified propositional formulas such that  $\Phi$ -SAT is in  $\mathbf{P}$ . Then  $\Phi$ -INT<sub>red</sub> can be computed in polynomial time.*

*Proof:* Let  $\phi(\bar{x}, \bar{y}), \psi(\bar{y}, \bar{z})$  be an unsatisfiable pair of formulas. Clearly  $\phi_1 = \exists \bar{x} \phi(\bar{x}, \bar{y})$  is a Boolean function equivalent to an interpolant. Given  $\bar{y}$  and  $\exists \bar{x} \phi(\bar{x}, \bar{y})$  we can evaluate  $\phi_1$  in  $\mathbf{NP}$  and, by our assumption, also in  $\mathbf{P}$ . By Proposition 4 we can find a uniform sequence of Boolean circuits  $C_n$  evaluating  $\phi_1$  with input  $\bar{y}$  and  $\phi$ . Now let  $k = l(\phi) + l(\bar{y})$ , where  $l(x)$  denotes the length of an expression  $x$ . Consider now  $C_k$ . From  $C_k$ , fixing the binary code of  $\phi$ , we obtain the Boolean circuit representing the requested interpolant.

**Theorem 6**  $\mathbf{P} = \mathbf{NP}$  iff  $\mathbf{NP} = \mathbf{CoNP}$  and INT<sub>red</sub> is computable in polynomial time.

*Proof:* Assume  $\mathbf{P} = \mathbf{NP}$ . Trivially  $\mathbf{NP} = \mathbf{CoNP}$ . The second claim follows from Proposition 5. The other direction follows from Corollary 1.

Under the assumption that  $\mathbf{NP} = \mathbf{CoNP}$  this theorem means that INT<sub>red</sub> is not harder (or even simpler) to compute than SAT. If we allow existentially quantified propositional formulas, i.e. we look at  $\Sigma(k)$ -INT<sub>red</sub> for  $k \geq 2$ , the assumption  $\mathbf{NP} = \mathbf{CoNP}$  can be dropped. Note, however, that INT<sub>red</sub> and  $\Sigma(1)$ -INT<sub>red</sub> are the same problems.

**Theorem 7** *The following are equivalent:*

- (i)  $\mathbf{P} = \mathbf{NP}$
- (ii) For every  $k \geq 2$  there is a polynomial algorithm for  $\Sigma(k)$ -INT<sub>red</sub>
- (iii) There is  $k \geq 2$  and a polynomial algorithm for  $\Sigma(k)$ -INT<sub>red</sub>
- (iv) There is a polynomial algorithm for  $\Sigma(2)$ -INT<sub>red</sub>.

*Proof:* (i)  $\rightarrow$  (ii): Proposition 5. (ii)  $\rightarrow$  (iii) and (iii)  $\rightarrow$  (iv): Trivial. (iv)  $\rightarrow$  (i): Assume (iv). We prove that SAT  $\in \mathbf{P}$ . Let  $\phi(\bar{x})$  be given. Clearly,  $\exists \bar{x} \phi(\bar{x}), \forall \bar{x} \neg \phi(\bar{x})$  is unsatisfiable and the interpolant is either **true** or **false** and can be computed in polynomial time. Now  $\phi(\bar{x})$  is satisfiable iff the interpolant is **true**.

For unbounded alternations of quantifiers we actually have a stronger result:

**Theorem 8**  $\Sigma^*$ -INT is polynomial time computable iff  $\mathbf{P} = \mathbf{PSPACE}$ .

*Proof:* if  $\mathbf{P} = \mathbf{PSPACE}$  we use again Proposition 5. For the other direction we reduce the evaluation problem to  $\Sigma^*$ -INT similarly to the previous proof.

### 3 Interpolation for Horn formulas

**3.1 A polynomial time algorithm** Recall that a *Horn formula*  $\phi$  is a propositional formula in conjunctive normal form such that every clause contains either *exactly one positive literal* or *exactly one occurrence of the propositional constant false*. A clause is called a *singleton* if it is a positive literal or it is identical to **false**. A *proper Horn clause* is a Horn clause which is not a singleton.

We write proper Horn clauses as  $x_1 \wedge \dots \wedge x_{n-1} \rightarrow x_n$  rather than  $\neg x_1 \vee \dots \vee \neg x_{n-1} \vee x_n$ . By our definition a negative literal is a Horn formula since it can be written as  $x \rightarrow \text{false}$ .

Recall further that *resolution* is a proof procedure which allows us to deduce from two clauses  $x \vee L_1 \vee \dots \vee L_k$  and  $\neg x \vee L'_1 \vee \dots \vee L'_n$  the clause  $L_1 \vee \dots \vee L_k \vee L'_1 \vee \dots \vee L'_n$ . *Unit resolution* is a proof procedure which is like resolution but with the additional requirement that either  $k = 0$  or  $n = 0$ . The following is easy and was first observed in [12]:

**Theorem 9** *A set of Horn clauses  $\Sigma$  is unsatisfiable iff there is a unit resolution proof of the empty clause.*

Remark: *HORN-SAT* is easily seen to be in **P**. In [14],[15] a linear time algorithm for *HORN-SAT* is presented.

For Horn formulas of the form  $a_1 \wedge \dots \wedge a_n \rightarrow a$  we have two rules which arise from the unit resolution, where we try to derive **false** instead of the empty clause:

- (1)  $a_1 \wedge \dots \wedge a_n \rightarrow a$  ( $a$  may be **false**),  $a_i$  is transformed into  $a_1 \wedge \dots \wedge a_{i-1} \wedge a_{i+1} \wedge \dots \wedge a_n \rightarrow a$
- (2)  $a_1 \wedge \dots \wedge a_n \rightarrow a$ ,  $a \rightarrow \text{false}$  is transformed into  $a_1 \wedge \dots \wedge a_n \rightarrow \text{false}$ .

It is easily seen that for the test of satisfiability we need only the first rule. We can replace it by a *parallel unit resolution rule*:  $a_1, \dots, a_n, a_1 \wedge \dots \wedge a_n \rightarrow a$  is transformed into  $a$ .

The *Horn Interpolation Problem (HORN-INT)* consists of finding for every unsatisfiable pair of Horn formulas  $\phi(\bar{x}, \bar{y}), \psi(\bar{y}, \bar{z})$  an interpolant  $\theta(\bar{y})$ . The *Disjoint Horn Interpolation Problem (DHORN-INT)* is the Horn Interpolation Problem under the additional assumption that there are no variables shared by  $\phi$  and  $\psi$ .

In the following we use unit resolution to compute an interpolant.

**Theorem 10** *There is a P-time algorithm which computes for every unsatisfiable pair of Horn formulas  $\phi(\bar{x}, \bar{y}), \psi(\bar{y}, \bar{z})$  an interpolant  $\theta(\bar{y})$  which is again a Horn formula.*

*Proof:* We first define a **P**-time algorithm *A1* which checks satisfiability of sets of Horn clauses  $\Sigma$  and, additionally, in case  $\Sigma$  is unsatisfiable, generates an unsatisfiable subset of clauses  $R \subset \Sigma$  in which each variable occurs at most once on the right hand side of the arrow.

*Definition of A1:* At the beginning we put  $S_0 =$  set of singletons of  $\Sigma, R_0 = N_0 = S_0$ , and  $var =$  set of variables and constants (the only constant is **false**) which appear in  $\Sigma$ .

$$N_{i+1} = \{y \in var: y \notin S_i, y \text{ arises from singletons and some formula } \psi \in \Sigma \text{ by parallel unit resolution}\}.$$

$$S_{i+1} = S_i \cup N_{i+1}.$$

For each  $y \in N_{i+1}$  we choose a formula  $\psi = h_y \in \Sigma$ , as in the definition of  $N_{i+1}$ .

$$R_{i+1} = R_i \cup \{h_y: y \in N_i\}.$$

If  $N_i = \emptyset$  then put  $S = S_i$  and  $R = R_i$  and stop.  $\Sigma$  is satisfiable iff **false**  $\in S$ .

*Correctness of A1:* Note that each  $S_i$ , and therefore also  $S$ , is a set of singletons. For each  $y \in N_i$  we take a proper Horn clause  $h_y \in \Sigma$  such that  $y$  is a parallel unit resolvent of some formula of  $S_i$  using  $h_y$ , therefore, by the completeness of parallel unit resolution for Horn clauses,  $\Sigma$  is satisfiable, provided **false**  $\notin S_i$ . Otherwise,  $\Sigma$  is unsatisfiable and, by our construction,  $R \subset \Sigma$  contains all the clauses needed to obtain unsatisfiability. It is easy to see from the construction of  $R$  that each variable occurs at most once on the right side of the arrow.

*Complexity of A1:* Let  $\Sigma$  have at most  $k$  different variables and contain at most  $h$  many proper Horn clauses. Since both the  $S_i$  and  $R_i$  grow monotonically, there are at most  $k$  iterations. Each iteration needs at most  $h$  steps and the formulas in  $\Sigma$  remain unchanged. Therefore this algorithm can be performed in **P**-time.

An algorithm *A2*, computing the interpolant for an unsatisfiable pair of Horn formulas  $\phi$  and  $\psi$ , can be defined as follows: Now, assume  $\Sigma_\phi, \Sigma_\psi$  are the sets of Horn clauses corresponding to  $\phi(\bar{x}, \bar{y})$  and  $\psi(\bar{y}, \bar{z})$  respectively. Put  $\Sigma = \Sigma_\phi \cup \Sigma_\psi$ .  $\Sigma$  is unsatisfiable. Apply *A1* to compute  $R$  and  $S$ . Clearly  $R$  is unsatisfiable. To compute the interpolant put  $\Sigma_0 = \Sigma_\phi \cap R$ .

W.l.o.g. let  $\{x_1, x_2, \dots, x_m\}$  be the set of all the singletons of  $S$  which do not occur as variables in  $\Sigma_\psi$ . Let  $g(y)$  be the left side of the implication  $h_y$ . If  $x_{i+1}$  appears as a nonnegated singleton we obtain  $\Sigma_{i+1}$  by dropping  $x_{i+1}$  from each left side of any clause where it appears. Put

$$\Sigma_{i+1} = \{h[x_{i+1}/g(x_{i+1})] : h \in \Sigma_i\}.$$

Here  $h[u/v]$  is the expression which is generated by  $h$  replacing  $u$  by  $v$  and deleting redundant literals. We replace the variable  $y = x_{i+1}$  by the left side of  $h_y$  in each  $h \in \Sigma_i$ .

Clearly, each  $\Sigma_i$  and therefore  $\Sigma_m$  is a consequence of  $\Sigma_\phi, \Sigma_m \cup \Sigma_\psi$  is unsatisfiable and each variable occurring in  $\Sigma_m$  occurs both in  $\Sigma_\phi$  and  $\Sigma_\psi$ . Therefore  $\Sigma_m$  is the required interpolant. The fact that *A2* can be performed in polynomial time is immediate.

Recall that *DHORN-INT* is the disjoint Horn interpolation problem. Combining Theorem 10 and Corollary 2(i) we get the following result:

**Corollary 11** *The following are equivalent:*

- (i) **L = P**
- (ii) *DHORN-INT*  $\in$  **L**
- (iii) *HORN-INT* can be solved in *Logspace*.

This shows that for unquantified Horn formulas the satisfiability and the interpolation problem are both of the same degree of difficulty. It is an open problem, if the same holds for quantified Boolean Horn formulas. Schaefer [30] claimed that the quantified Horn satisfiability problem is in **P** (see also [11], Problem [LO6]), but he never published a proof of it. (Recently Schaefer [31] sent us a preprint describing a polynomial time algorithm for the quantified Horn satisfiability problem.) Karpinski, Kleine-Büning, and Schmitt independently developed a polynomial time algorithm for the quantified Horn satisfiability problem [19].

**4 Interpolation for Krom formulas**

**4.1 Consequences of  $L = NL$**  Recall that Krom formulas are propositional formulas in conjunctive normal form where each disjunct contains at most two literals. We collect the known results of [18] and [4] in the following theorem:

**Theorem 12**

- (i) *KROM-SAT* is **CoNL**-complete via log space reductions [18]
- (ii)  $\Sigma^*$ -*KROM-SAT* can be solved in linear time [4].

We first analyze the complexity of *KROM-INT*.

**Proposition 13** Given two Krom-formulas  $\phi(\bar{x}, \bar{y})$  and  $\psi(\bar{y}, \bar{z})$ , let  $\theta(\bar{y})$  be the conjunction of all Krom clauses, which are consequences from  $\phi(\bar{x}, \bar{y})$ . Then  $\theta(\bar{y})$  is an interpolant of  $\phi(\bar{x}, \bar{y})$  and  $\psi(\bar{y}, \bar{z})$ . Moreover, to be a clause of  $\theta(\bar{y})$  can be checked in **NL** and  $l(\theta) = O(l(\bar{y})^2)$ .

*Proof:* Clearly,  $l(\theta) = O(l(\bar{y})^2)$ . By the known variable elimination algorithm of Davis and Putnam the inconsistency of  $\phi(\bar{x}, \bar{y}) \wedge \psi(\bar{y}, \bar{z})$  can be determined. We can at first eliminate the variables of  $\bar{x}$ . Then we get a formula  $\theta(\bar{y}) \wedge \psi(\bar{y}, \bar{z})$ . This formula remains a contradiction. But  $\theta(\bar{y})$  is the conjunction of all Krom clauses which are consequences from  $\phi(\bar{x}, \bar{y})$  and do not contain variables from  $\bar{x}$ . Clearly each clause of  $\theta(\bar{y})$  can be recognized in **NL**.

**Theorem 14**  $L = NL$  iff *KROM-INT*  $\in L$ .

*Proof:* Assume  $L = NL$ . Then by Proposition 13, *KROM-INT*  $\in L$ . Suppose *KROM-INT*  $\in L$ . We know by [13] that **NL** = **CoNL**. By Corollary 2(ii)  $L = NL$ .

**4.2 Eliminating quantifiers from Krom formulas** Our next result is a reinterpretation of the proof of Theorem 12(ii). In [4] the evaluation problem of Krom formulas is first reduced to a graph problem: W.l.o.g. all clauses of a Krom formula have exactly two literals. We can always write  $L \vee L$  instead of  $L$  for any literal  $L$ . Let  $K$  be a set of Krom clauses whose variables are exactly  $\bar{x}$ . The graph  $\Gamma(K)$  is then defined as follows: Let  $V(K)$ , the vertices, be the set of *literals* (variables and negated variables) of  $\bar{x}$ . Denote the negation of a literal  $L$  by  $\neg L$  ( $\neg\neg L = L$ ).

Let  $E(K)$  be the set of directed edges  $(v_1, v_2)$  such that the Krom clause  $v_1 \rightarrow v_2 := \neg v_1 \vee v_2 \in K$ . For a Krom formula  $\phi$  with  $K_\phi$  as its clause set we set  $\Gamma(\phi) = \Gamma(K_\phi)$ . Two vertices  $v_1, v_2$  of  $\Gamma(K)$  are in the *same strongly connected component* if there are directed paths from  $v_1$  to  $v_2$  and vice versa.

Let  $Q^\# = Q_1 x_1 Q_2 x_2 \dots Q_n x_n$  be a quantifier prefix and  $\phi$  be a Krom formula and  $K_\phi$  the set of Krom clauses from  $\phi$ . We call a vertex  $x_i$  ( $\neg x_i$ ) in  $\Gamma(K_\phi)$  *existential (universal)* if  $Q_i$  is existential (universal) or  $x_i$  is a free variable. In [4], Theorem 2, the following is proved:

**Lemma 15** The quantified Krom formula  $Q^\# \bar{x}\phi(\bar{x}, \bar{y})$  is satisfiable iff none of the following three conditions holds.

(i) Some existential vertex  $x_i$  is in the same strongly connected component as its complement  $\neg x_i$

(ii) Some universal vertex  $v_i$  is in the same strongly connected component as an existential vertex  $v_j$  such that  $j < i$  where  $v_i(v_j)$  is either  $x_i(x_j)$  or  $\neg x_i(\neg x_j)$

(iii) There is a path from a universal vertex  $u$  to another universal vertex  $v$ .

**Lemma 16** *Let  $\phi(x_1, x_2, \dots, x_n)$  be a quantified Krom formula with free variables  $x_1, x_2, \dots, x_n$  and let  $\phi_0$  be the quantifier free part of  $\phi$ . Let  $\alpha \in 2^{\{x_1, x_2, \dots, x_n\}}$  be an assignment of truth values for the free variables. Then  $\phi[\alpha] = 1$  iff*

(i)  $\phi(x_1, x_2, \dots, x_n)$  is satisfiable

(ii) there is no literal  $L$  whose variable is free in  $\phi$  and there is no literal  $U$  whose variable is universally quantified in  $\phi$  such that  $L[\alpha] = 0$  and  $\Gamma(\phi_0)$  has a directed path from  $U$  to  $L$  and

(iii) there are no literals  $L, M$  whose variables are free in  $\phi$  such that  $L[\alpha] = 1$ ,  $M[\alpha] = 0$ , and  $\Gamma(\phi_0)$  has a directed path from  $L$  to  $M$ .

*Proof:* Assume  $\phi[\alpha] = 1$ . Then (i)–(iii) are easily checked. Now assume  $\phi[\alpha] = 0$ , (ii), and (iii). We proceed to show that  $\phi(x_1, x_2, \dots, x_n)$  is not satisfiable. We define two auxiliary quantifier-free Krom formulas  $\psi_1, \psi_2$ , which define the values of  $\alpha$  as follows:

$$\begin{aligned}\psi_1 &= \bigwedge \{ \neg x_i \rightarrow x_i : \alpha(x_i) = 1 \} \\ \psi_2 &= \bigwedge \{ x_i \rightarrow \neg x_i : \alpha(x_i) = 0 \}.\end{aligned}$$

Clearly, we have:

*Claim 1:*  $\phi[\alpha] = 1$  iff  $\phi \wedge \psi_1 \wedge \psi_2$  is satisfiable.

Therefore, by our assumption,  $\phi_1 = \phi \wedge \psi_1 \wedge \psi_2$  is not satisfiable. Using Lemma 15 one can check that  $\phi(x_1, x_2, \dots, x_n)$  is not satisfiable: We use Lemma 15 and assume that  $\phi_1$  satisfies one of the conditions (i)–(iii) of Lemma 15.

Assume  $\phi_1$  satisfies condition (i) of Lemma 15. That means that  $\Gamma(\phi_1)$  contains a directed cycle  $C$  which touches existential vertices  $y$  and  $\neg y$ . We assume that  $\phi$  is satisfiable. But then the cycle  $C$  contains at least one directed edge of the form  $\neg L \rightarrow L$ . Let  $\neg L \rightarrow L$  and  $\neg M \rightarrow M$  be two consecuting directed edges of this form on  $C$ . Then there is a directed path from  $L$  to  $\neg M$  in  $\Gamma(\phi)$  although  $L[\alpha] = 1$  and  $\neg M[\alpha] = 0$ . This is a contradiction of condition (iii) of this lemma.

Assume now  $\phi_1$  satisfies condition (ii) of Lemma 15. Then we again have a cycle  $C$  touching a universal vertex  $v_i$  and an existential vertex  $v_j (j < i)$ . This cycle contains at least one directed edge of the form  $\neg L \rightarrow L$  (of  $\Gamma(\psi_1) \cup \Gamma(\psi_2)$ ). By the same argument as in the case of condition (i) of Lemma 15 we get a contradiction of condition (iii) of this lemma, if we assume that  $\phi$  is satisfiable.

Now we assume that  $\phi_1$  satisfies condition (iii) of Lemma 15. That means that there is a path from a universal vertex to another universal vertex (in  $\Gamma(\phi_1)$ ). If we assume that  $\phi$  is satisfiable, then at least one directed edge of this path is in  $K(\psi_1) \cup K(\psi_2)$ . That means that this edge is of the form  $L \rightarrow \neg L$ . We may assume that this is the first directed edge of the path. But then we have a directed path from a universal vertex to a literal  $L$ , such that  $L[\alpha] = 0$ . That is a contradiction of condition (ii) of this lemma.

**Theorem 17** *There is a log space algorithm which transforms an arbitrarily quantified Krom formula  $\phi(\bar{x})$  into an equivalent  $\Sigma(1)$ -Krom formula  $\theta(\bar{x})$ .*

*Proof:* Let  $\phi(\bar{x})$  be given with quantifier-free part  $\phi_0$ . Lemma 16 gives us a characterization of all the assignments  $\alpha$  which make  $\phi$  true. For each part of this characterization (i)–(iii) we shall construct in log space existential Krom formulas  $\sigma_i$ ,  $i = 1, 2, 3, 4$  such that

- (a)  $\sigma_1$  is without free variables and  $\sigma_1$  is true iff  $\phi(\bar{x})$  is satisfiable;
- (b)  $\sigma_2(\bar{x})$  is equivalent to: there is no literal  $L$  whose variable is free in  $\phi$  and there is no literal  $U$  whose variable is universally quantified in  $\phi$  such that  $L[\alpha] = 0$ , such that there is a directed path from  $U$  to  $L$  in  $\Gamma(\phi_0)$ .
- (c)  $\sigma_3(\bar{x})$  is equivalent to: there are no literals  $L, M$  whose variables are free in  $\phi$  such that  $L[\alpha] = 1$ ,  $M[\alpha] = 0$  and there is a directed path from  $L$  to  $M$  in  $\Gamma(\phi_0)$ .

$\theta(\bar{x})$  is the conjunction of  $\sigma_1, \dots, \sigma_3$ . Our main tool is the coding of graph accessibility problems by Krom formulas, implicitly described in [18], which we spell out as a separate lemma:

**Lemma 18**

(i) *Let  $\Gamma$  be a directed graph on  $n$  vertices  $v_1, \dots, v_n$ . Then there is a quantifier free Krom formula  $\phi_\Gamma(x_1, \dots, x_n)$  such that for any two vertices  $v_i, v_j$  we have: there is a directed path from  $v_i$  to  $v_j$  iff  $\exists \bar{x}_{ij} \phi(x_i/1, x_j/0)$  is false. (Here  $\bar{x}_{ij}$  is the vector of variables with  $x_i, x_j$  missing and the remaining variables are evaluated as indicated.)*

(ii) *The formula  $\phi_\Gamma$  is of size at most  $n^2$  and can be computed in log space of  $n$ .*

*Proof:* For every edge  $(v_i, v_j)$  we form the clause  $x_i \rightarrow x_j$  and set  $\phi_\Gamma$  to be the conjunction of all these clauses. The rest is verified easily.

*The construction of  $\sigma_1$ :* An immediate consequence of Lemma 15 is that the nonsatisfiability problem of  $\phi(\bar{x})$  is in **NL**. Therefore by [18] the satisfiability of  $\phi(\bar{x})$  can be reduced to the satisfiability of a nonquantified Krom formula  $\phi'$  in logspace. Let  $\sigma_1 = \exists \bar{y} \phi'$ .

*The construction of  $\sigma_2$ :* For every literal  $L$  of a free variable of  $\phi$  define a graph  $\Gamma_{L,2}$  as follows. We take a copy of the graph  $\Gamma(\phi)$  and add two new vertices  $v_L$  and  $\bar{v}_L$ . We add a new arc  $(L, v_L)$  and new arcs  $(\bar{v}_L, u)$  for each universal vertex  $u$ . Clearly there is no path from any universal literal to  $L$  in  $\Gamma(\phi)$  iff there is no path from  $\bar{v}_L$  to  $v_L$  in  $\Gamma_{L,2}$ . Let  $\sigma'_{L,2}(x_{1,L}, \dots, x_{n,L})$  be the formula which arises from  $\Gamma_{L,2}$  by Lemma 18. Let  $x_{1,L}$  be the variable corresponding to  $\bar{v}_L$  and  $x_{2,L}$  be the variable corresponding to  $v_L$ . Define  $\sigma_{L,2} = \sigma'_{L,2} \wedge x_{1,L} \wedge (x_{2,L} \leftrightarrow L)$ . By Lemma 18  $\sigma_{L,2}$  is satisfiable iff there is no path in  $\Gamma(\phi)$  from any universal literal to  $L$ , or  $L$  is set true. Let  $\sigma' = \bigwedge \{ \sigma_{L,2} : L \text{ literal of a free variable of } \phi \}$ . Let  $\sigma_2 = \exists \bar{y} \sigma'_2$  where  $\bar{y}$  ranges over all variables which are not variables of any literal  $L$  appearing free in  $\phi$ . Clearly  $\sigma_2$  satisfies the statement (b).

*The construction of  $\sigma_3$ :* Let  $\sigma'_3$  be the formula which arises from  $\Gamma(\phi)$  by Lemma 18. Let  $\sigma_3 = \exists \bar{y} \sigma'_3$ , where  $\bar{y}$  ranges over all variables of  $\sigma'_3$  not arising in  $\bar{x}$ . By the same argument as in Lemma 18 we can make the following claim:

$\sigma_3(\bar{x})$  is true iff there is no path of  $\Gamma(\phi_0)$  from a literal  $L$ , such that  $L[\alpha] = 1$ , to another literal  $M$ , such that  $M[\alpha] = 0$ , where  $L$  and  $M$  are literals over variables of  $\bar{x}$ .

The claim can be proved in a similar way as Lemma 18.

Clearly, these constructions can be carried out in log space.

**Corollary 19**  $\mathbf{L} = \mathbf{NL}$  iff and  $\Sigma^*$ -KROM-INT  $\in \mathbf{L}$ .

*Proof:* One direction follows from Corollary 2(ii) and the other direction from Theorem 14 and Theorem 17.

## REFERENCES

- [1] Anderaa, S. O., "On the decision problem for formulas in which all disjunctions are binary," pp. 1–19 in *Proceedings of the 2nd Scandinavian Logic Symposium*, ed. J. E. Fenstad, North-Holland, Amsterdam, 1971.
- [2] Anderaa, S. O. and E. Börger, "The Horn complexity of Boolean functions and Cook's problem," pp. 231–256 in *Proceedings of the 5th Scandinavian Logic Symposium*, ed. F. V. Jensen et al., Aalborg University Press, Aalborg, 1970.
- [3] Anderaa, S. O. and E. Börger, "The equivalence of Horn and network complexity for Boolean functions," *Acta Informatica*, vol. 15 (1981), pp. 303–307.
- [4] Apsvall, B., M. F. Plass, and R. E. Tarjan, "A linear time algorithm for testing the truth of certain quantified Boolean formulas," *Information Processing Letter*, vol. 8.3 (1979), pp. 121–123.
- [5] Barwise, K. J. and S. Feferman, *Model Theoretic Logics*, Springer-Verlag, New York, 1984.
- [6] Börger, E., *Reduktionstypen in Krom- und Hornformeln*, Dissertation, Universität Münster, 1971 (see also "Beitrag zur Reduktion des Entscheidungsproblems auf Klassen von Hornformeln mit kurzen Alternationen," *Archiv für mathematische Logik*, vol. 16 (1974), pp. 67–84).
- [7] Chandra, A. and L. Stockmeyer, "Alternation," pp. 98–108 in *Proceedings of the 17th Annual Symposium on Foundations of Computer Science (IEEE)*, 1976.
- [8] Cook, S. A., "An observation in time-storage trade off," *Journal of Computer and Systems Sciences*, vol. 9 (1974), pp. 308–316.
- [9] Craig, W., *A Theorem about First Order Functional Calculus with Identity and Two Applications*, Doctoral Dissertation, Harvard University, 1951.
- [10] Friedman, H., "The complexity of explicit definitions," *Advances in Mathematics*, vol. 20 (1976), pp. 18–29.
- [11] Garey, M. R. and D. S. Johnson, *Computers and Intractability*, W. H. Freeman and Co., San Francisco, 1979.
- [12] Henschen, L. and L. Wos, "Unit refutations and Horn sets," *Journal of the ACM*, vol. 21.4 (1974), pp. 590–605.
- [13] Immerman, N., *Nondeterministic Space is Closed under Complement*, TR 552, Department of Computer Science, Yale University, 1987. (see also "Proceedings structure in complexity theory," 3rd Annual Conference (1988), pp. 112–115.)
- [14] Itai, A. and J. A. Makowsky, *On the Complexity of Herbrand's Theorem*, Department of Computer Science, Technion–Israel Institute of Technology, 1982.
- [15] Itai, A. and J. A. Makowsky, "Unification as a complexity measure for logic programming," *The Journal of Logic Programming*, vol. 4 (1987), pp. 105–117.

- [16] Jones, N. D., "Space bounded reducibility among combinatorial problems," *Journal of Computer and System Sciences*, vol. 11 (1975), pp. 68-85.
- [17] Jones, N. D. and W. T. Laaser, "Complete problems for deterministic polynomial time," *Theoretical Computer Science*, vol. 3 (1977), pp. 105-117.
- [18] Jones, N. D., Y. E. Lien, and W. T. Laaser, "New problems complete for non-deterministic log space," *Mathematical Systems Theory*, vol. 10 (1976), pp. 1-17.
- [19] Karpinski, M., H. Kleine-Büning, and P. Schmitt, "The computational complexity of quantified Horn clauses," Department of Computer Science, University of Bonn, Report No. 8520.
- [20] Krom, M. R., "Some interpolation theorems for first order formulas in which all disjunctions are binary," *Logique et Analyse*, vol. 43 (1968), pp. 403-412.
- [21] Ladner, R. E., "The circuit value problem is log space complete for P," *SIGACT News*, vol. 7.1 (1975), pp. 18-20.
- [22] McRobbie, M. A., "Interpolation theorems: A bibliography," Department of Philosophy, University of Melbourne, preprint, 1982.
- [23] Meyer, A. B. and L. J. Stockmeyer, "Word problems requiring exponential time: Preliminary report," pp. 1-9 in *Proceedings of the 5th ACM-STOC*, 1973.
- [24] Mundici, D., "Craig's interpolation theorem in computation theory," *Atti della Accademia nazionale dei lincei*, vol. 70.1 (1981), pp. 6-11.
- [25] Mundici, D., "Complexity of Craig's interpolation," *Fundamenta Informaticae*, vol. 3-4 (1982), pp. 261-278.
- [26] Mundici, D., "A lower bound for the complexity of Craig's interpolant in sentential logic," *Archiv für mathematische Logik*, vol. 23 (1983), pp. 27-36.
- [27] Mundici, D., "NP and Craig's interpolation theorem," in *Logic Colloquium '82*, ed. G. Lolli, G. Longo, and A. Marcja, North Holland, Amsterdam, 1984.
- [28] Ruzzo, W. L., "On uniform circuit complexity," *Journal of Computer and Systems Sciences*, vol. 22 (1981), pp. 365-383.
- [29] Savage, J. E., *The Complexity of Computing*, Wiley, New York, 1976.
- [30] Schaefer, T. J., "The complexity of satisfiability problems," pp. 216-226 in *Proceedings of the 10th ACM Symposium on Theory of Computing*, 1978.
- [31] Schaefer, T. J., "Deciding quantified Horn sentences and polynomial time," preprint.
- [32] Smullyan, R. M., "First order logic," *Ergebnisse der Mathematik*, vol. 34, Springer-Verlag, Berlin, 1968.
- [33] Stockmeyer, L., "Classifying the computational complexity of problems" (survey paper), *The Journal of Symbolic Logic*, vol. 52 (1987), pp. 1-43.

*E. Dahlhaus*  
 Department of Computer Science  
 University of Bonn  
 Bonn, Federal Republic of Germany

*J. A. Makowsky*  
 Department of Computer Science  
 Technion-Israel Institute of Technology  
 Haifa, Israel

*A. Israeli*  
 Department of Electrical Engineering  
 Technion-Israel Institute of Technology  
 Haifa, Israel

*and*  
 Department of Mathematics  
 University of Lausanne  
 Lausanne-Dorigny, Switzerland