MAX-PLANCK-INSTITUT FÜR INFORMATIK

Automated Complexity Analysis Based on Ordered Resolution

David Basin
Harald Ganzinger

MPI–I–95–2–006 November 1995

INFORMATIK

Im Stadtwald
D 66123 Saarbrücken
Germany

Authors' Addresses

David Basin, Harald Ganzinger
Max-Planck-Institut für Informatik
Im Stadtwald
D-66123 Saarbrücken, Germany
`{basin|hg}@mpi-sb.mpg.de`
`http://www.mpi-sb.mpg.de/~{basin|hg}/`

Acknowledgements

This research was supported in part by the ESPRIT Basic Research Working Groups 6028 (Construction of Computational Logics) and 6112 (COMPASS).

Abstract

We define *order locality* to be a property of clauses relative to a term ordering. This property is a generalization of the subformula property for proofs where terms arising in proofs are bounded, under the given ordering, by terms appearing in the goal clause. We show that when a clause set is order local, then the complexity of its ground entailment problem is a function of its structure (e.g., full versus Horn clauses), and the ordering used. We prove that, in many cases, order locality is equivalent to a clause set being saturated under ordered resolution. This provides a means of using standard resolution theorem provers for testing order locality and transforming non-local clause sets into local ones. We have used the Saturate system to automatically establish complexity bounds for a number of nontrivial entailment problems relative to complexity classes which include Polynomial and Exponential Time and co-NP.

Keywords

First-Order Theories, Descriptive Complexity, Complexity Analysis, Ordered Resolution, Automated Theorem Proving, Saturation

Contents

1 Introduction

Our research is motivated by the work of McAllester and Givan [10, 5] who gave procedures for recognizing sets of Horn clauses which generate polynomial time decidable entailment relations. McAllester's procedures test if Horn clauses have a property, which he calls locality, which roughly corresponds to a kind of subformula property for Horn clause proofs; when a clause set has this property, then there are a bounded number of terms (dependent only on the query) which need appear in proofs and this is used to solve ground queries in polynomial time.

In this paper, we both generalize locality and provide new techniques for proving that clause sets are local. Specifically, we provide a reformulation, called order locality, which is a property of clauses, parameterized by an ordering on terms. This strictly generalizes McAllester's notion, both as it is parameterized by orderings (McAllester's definition is relative to a fixed ordering, the subterm ordering), and as it is defined for full, as opposed to Horn, clauses. We show that order locality is closely related and in many cases is equivalent to a clause set being saturated up to redundancy under ordered resolution. This provides both an alternative characterization of when a clause set is local and yields a usable procedure for testing locality; moreover, when a clause set is not local saturation may be used to transform it into a local presentation of the same theory. This procedure has been implemented in the Saturate system [4] and has been applied to give machine checked proofs of the complexity of a number of nontrivial entailment relations and algorithms which are specified as inference problems. For example, we use Saturate to automatically verify that congruence closure, the theory of tree embeddings, and the theory of nonstrict partial orderings are polynomial time decidable, that propositional logic is in both co-NP and Exptime, and that the first-order theory of total orderings is in co-NP.

Our paper is organized as follows. In section 2 we define our notion of locality, *order locality*, and show that it generalizes McAllester's original definition. In section 3 we review saturation based theorem proving and in section 4 we show that saturation suffices to establish locality of a clause set. Unfortunately, while saturation implies locality the converse sometimes fails. In section 5 we show that for Horn clauses, under a more restrictive semantics, the converse "essentially" (in a sense that will be made clear then) holds. In section 6 we show how a complexity measure may be associated with an ordering so that when a clause set N is local relative to that ordering, then we may associate a corresponding complexity class to the entailment problem for N. In section 7 we show a form of the converse: when the entailment problem for N belongs to certain complexity classes, then there are corresponding orderings under which N is local. After, we move on to consider locality in practice. In section 8 we show how to construct orders useful for saturation and in section 9 we provide a number of examples which have been carried out with the Saturate system (including those listed above). In section 10 we give an extended example, of a theoretical nature, and show locality for a Horn theory of the propositional sequent calculus and consequences thereof. Finally, in section 11 we compare the use of Saturate to the algorithm proposed by McAllester and Givan, and in section 12 we draw conclusions.

2 Preliminaries

2.1 Entailment and Proofs

A *(full) clause* is a formula of the form $\Gamma \to \Delta$, where Γ and Δ are multisets of atoms; Γ represents negative literals and Δ positive literals. A clause is *Horn* when Δ is a singleton or empty. We will write Γ_1, Γ_2 to indicate the union of multisets and usually omit brackets, e.g., writing Γ, A or A, Γ for the union of $\{A\}$ and Γ or writing $A_1, \ldots, A_m \to B_1, \ldots, B_n$ for $\{A_1, \ldots, A_m\} \to \{B_1, \ldots B_n\}$. If C is a clause, by $\neg C$ we denote the set of unit Horn clauses $\neg L$, with L a literal in C. We say that a term t *occurs* in an atom A if t is of the form $p(\ldots, t, \ldots)$. t occurs in a clause if it occurs in an atom of the clause.

Given a set of clauses N, the *entailment problem* for N is to decide if $N \models C$, for ground clauses C, called *queries*. A *(refutational) proof* of $N \models \Gamma \to \Delta$ is a tree in which the root is the empty clause, the leafs are ground instances of N or unit clauses $\neg L$, with L a literal in $\Gamma \to \Delta$, and where each non-leaf node is the result of either a resolution inference from its two successors or a factoring inference from its successor. In the Horn clause case we will also consider *direct proofs* in which, for A a ground atom, a proof of $N \models A$ is an unordered tree, whose nodes are ground atoms, such that A is the root, and B is a node with children $B_1, \ldots, B_n$ if and only if $B_1, \ldots, B_n \to B$ is a ground instance of N. For both notions of proof, we say that a term t *occurs* in a proof if t is a subterm of an atom that occurs in the proof tree.

Let Υ be a set of ground terms and C a clause $\Gamma \to \Delta$. We say that N *entails* C *with respect to set* Υ, and write $N \models_\Upsilon C$, if C is already entailed by the set of those ground instances of N in which all terms are subterms of terms in Υ. When Υ contains precisely all ground terms in N and C, we write $N \models\!\!\!\!\dashv C$ for $N \models_\Upsilon C$. Using $\cdot\!\!>$ to denote the strict subterm ordering, $N \models_\Upsilon C$ may be reformulated by requiring the existence of a proof of C from N such that for each term s in the proof there exists a term t in Υ such that $t \cdot\!\!\geq s$.

2.2 Locality

McAllester's work is based on Horn clauses[1] and he defines a set of Horn clauses N to be *local* if for every ground Horn clause C we have $N \models C$ if and only if $N \models\!\!\!\!\dashv C$.

Locality is a desirable property of Horn clause sets because when N is local, then $N \models C$ precisely when $N' \models C$ for a ground Horn set N' which is polynomial in the size of C. Namely, N' comprises those ground substitution instances of N where all terms in N' are subterms in the "bounding set" Υ. The entailment problem for N' can be efficiently solved:

Lemma 1 ([3]) *For all sets of ground Horn clauses N and ground Horn clauses C, $N \models C$ is decidable in time linear in the sum of the sizes of N and C.*

Hence, when N is local, its ground entailment problem is tractable.

Lemma 2 ([10]) *If N is a local set of Horn clauses, then for all ground Horn queries C, $N \models C$ is polynomial time decidable in the size of C.*

[1] In [5] rules are Horn clauses and in [10] they are Horn clauses extended by a sorting discipline. Note that McAllester considers direct instead of refutational proofs for Horn clauses but for consistent Horn theories this distinction is inessential.

2.3 Order Locality

We generalize the definition of locality by allowing arbitrary term orderings, possibly different from the subterm ordering, and allowing full clauses for our clause sets N and our queries C. Let $\succ$ be a well-founded, partial ordering on terms. We say that a set N of clauses *entails a clause C with respect to* $\succ$, and write $N \models_{\preceq} C$, if and only if there is a proof of $N \models C$ from those ground instances of N in which each term is smaller or equal with respect to $\succeq$ than some term in C. Now we call a set of clauses N *local with respect to the ordering* $\succ$ if whenever $N \models C$ for a ground clause C, then $N \models_{\preceq} C$. To distinguish this notion of locality from McAllester's we will also refer to it as *order locality* in contrast to McAllester's definition which we will refer to as *subterm locality*.

In the case of Horn clause sets and queries, when N contains no ground terms, subterm locality is a special case of order locality corresponding to the subterm ordering $\cdot\!\!>$. When N contains ground terms these notions are not quite compatible since subterm locality allows ground terms in N (and their subterms) to participate in a proof of a Horn clause $\Gamma \rightarrow A$. However, for any N we can define a transformation that maps a query $\Gamma \rightarrow A$ to another query $\Gamma' \rightarrow A$ where $N \models\!\!| \; \Gamma \rightarrow A$ if and only if $N \models_{\leq} \Gamma' \rightarrow A$. Namely, when N contains the k ground terms $t_1, \ldots, t_k$, then we conservatively extend our theory with a new uninterpreted predicate symbol p of arity k and $\Gamma' = p(t_1, \ldots t_k), \Gamma$. Since the transformation of Γ to Γ' always introduces the same fixed set of ground terms, it does not change the complexity of determining entailment based on $\models_{\leq}$; hence order locality can establish that entailment is polynomial time decidable for any Horn set which is subterm local.

3 Ordered Resolution

We define the concept of saturation up to redundancy of a set of clauses by ordered resolution. Hence we have to define ordered resolution and what it means for a clause or an inference to be redundant. The definitions in this section are with respect to any well-founded, total ordering $\succ$ on ground *atoms*. We will call such an ordering *compatible* with a total ordering $\succ'$ on ground terms, if $\max_i(t_i) \succ' \max_i(s_j)$ implies that $p(t_1 \ldots t_k) \succ q(s_1 \ldots s_n)$, for any two predicate symbols p and q. Note that if our signature of predicate symbols just consists of one element which, in addition, is unary, then total orderings on ground terms can be identified with compatible orderings on ground atoms. We will later discuss methods that can be used to generate compatible atom orderings.

3.1 Clause Ordering

Given an atom ordering $\succ$, we will call an atom A [strictly] maximal with respect to a multiset of atoms Γ, if for any B in Γ we have $B \not\succ A$ [$B \not\succeq A$]. If the ordering is defined on ground atoms, we lift it to non-ground atoms A and B by: $A \succ B$ if $A\sigma \succ B\sigma$ for every ground substitution σ. We also need to explain how to extend $\succ$ to clauses. To this end we identify a positive literal A with the singleton multiset $\{A\}$, a negative literal $\neg A$ with the multiset $\{A, A\}$, and a clause with the multiset of its literals. Then, for clauses C and C', we define $C \succ C'$ if and only if C $(\succ_{ms})_{ms}$ C' in the twofold multiset extension of $\succ$. Clearly, if $\succ$ is total [well-founded] on atoms, so is its extension to clauses. Given a ground clause C and a set of clauses N, by N_C we denote the set of ground instances D of N such that $C \succ D$ in the clause ordering. When the ordering is not clear from the context, we will also use the notation $N_C^{\succ}$. Observe that when $\succ$ is total on ground terms, any term that occurs in a clause in N_C must be smaller or equal to the maximal term in C.

3.2 Ordered Resolution and Redundancy

There are a variety of formalizations of refutation-complete calculi for ordered resolution. The rules we present are related to those given for first-order clauses with equality in [2]. In addition to an atom ordering $\succ$, they take a selection function for negative literals as an additional parameter. A *selection function* S assigns to each ground clause a possibly empty set of occurrences of negative literals. If C is a clause, the literal occurrences in $S(C)$ are called selected. $S(C) = \emptyset$ indicates that no literal is selected. We extend S to non-ground clauses D by taking $S(D)$ to be the union of the $S(C)$, for all ground instances C of D.

An inference by *ordered resolution* between clauses takes the form

$$\frac{\Gamma_1 \to A, \Delta_1 \qquad B, \Gamma_2 \to \Delta_2}{\Gamma_1\sigma, \Gamma_2\sigma \to \Delta_1\sigma, \Delta_2\sigma}$$

where σ is the mgu of A and B such that (i) $A\sigma$ is strictly maximal with respect to $\Gamma_1\sigma$,$\Delta_1\sigma$, (ii) no literal is selected in the first premise, (iii) $A\sigma$ either is selected in the second premise, or else is maximal with respect to $\Gamma_2\sigma$,$\Delta_2\sigma$.

An inference by *ordered factoring* takes the form

$$\frac{\Gamma \to A, B, \Delta}{\Gamma\sigma \to A\sigma, \Delta\sigma}$$

if σ is the mgu of A and B such that (i) $A\sigma$ is strictly maximal with respect to $\Gamma\sigma$, (ii) $A\sigma$ is maximal with respect to $\Delta\sigma$, and (iii) no literal is selected in the premise.

Note that inferences by ordered resolution in which the first premise contains a selected literal are excluded, as are inferences by ordered factoring from clauses with selected literals.

A ground clause C is called *redundant in* N (with respect to $\succ$) if it is entailed by the instances of N which are smaller than C, i.e., if $N_C \models C$.

Let $\Gamma_1, \Gamma_2 \to \Delta_1, \Delta_2$ be the conclusion of an inference by ordered resolution from ground clauses $C = \Gamma_1 \to A, \Delta_1$ and $D = A, \Gamma_2 \to \Delta_2$. We call the inference *redundant in* N (with respect to $\succ$) if (i) one of the premises is redundant in N, or else if (ii) $N_D \models \Gamma_1, \Gamma_2 \to \Delta_1, \Delta_2$, that is, the conclusion is entailed by instances of N smaller than the second premise. Note that since $\succ$ is total on ground atoms the second premise of an ordered resolution inference from ground clauses is the maximal clause among the two premises and the conclusion.

We call a ground inference by ordered factoring *redundant in* N if the conclusion follows from instances in N which are smaller with respect to $\succ$ than its premise. A non-ground inference is called *redundant* if all its ground instances are redundant. Finally we call a set of clauses *saturated up to redundancy* (by ordered resolution with respect to $\succ$), if any inference by ordered resolution from premises in N is redundant in N.

As an inference is redundant whenever the conclusion is an instance of N, we can saturate N by systematically adding all conclusions of non-redundant inferences.

Given a total and well-founded ordering $\succ$ on ground terms, we call N *saturated up to redundancy* (by ordered resolution with respect to $\succ$) if N is saturated with respect to (i) some total and well-founded ordering on ground atoms that is compatible with $\succ$ and (ii) some selection function S. If the ordering $\succ$ on ground terms is partial we call N saturated up to redundancy with respect to $\succ$ if N is saturated with respect to every extension of $\succ$ that is well-founded and total on ground terms.

Saturation up to redundancy is a complete method of detecting inconsistency with respect to any $\succ$:

Theorem 1 ([1]) *Let N be a set of clauses that is saturated under ordered resolution with respect to $\succ$ up to redundancy. Then N is inconsistent if and only if N contains the empty clause.*

Proofs of entailment for clauses with respect to saturated sets N of clauses can be given in an N-*linear* form.

Theorem 2 ([2]) *Let N be saturated and let C be a clause. C follows from N if and only if there is a proof, by ordered resolution, of the empty clause from a (finite) set of ground instances in $N \cup \neg C$ such that for each inference at least one of the premises is not an instance of N.*

This theorem follows from the fact that in saturated sets N any inference from premises both in N are redundant, and that redundancy of clauses and inferences is preserved by adding clauses. Redundancy is also preserved by deleting redundant clauses that may arise during a theorem proving process.

4 Saturation and Locality

4.1 Saturation implies Locality

We start by showing that saturation implies locality for total term orderings $\succ$, provided that the selection function S is admissible in the following sense: S is called *admissible* (with respect to $\succ$) if for any ground clause C a literal is in $S(C)$ only if it contains the maximal term of C. From now on we assume that any selection function that we use for saturation is admissible.

Lemma 3 *Let $\succ$ be a total and well-founded ordering on ground terms. Moreover, let N be saturated with respect to $\succ$ and let C be a ground clause. Then $N \models C$ if and only if $N \models_{\preceq} C$.*

Proof. The right to left direction is immediate. In the left to right direction we may assume $N \models C$ and that N is saturated up to redundancy with respect to an atom ordering $\succ$ that is compatible with the given term ordering and an admissible selection function. By Theorem 2 there exists an N-linear proof of the empty clause by ordered resolution from a (finite) set of ground instances in $N \cup \neg C$. We proceed by induction on the height of an inference within the proof where we show that all terms involved in that inference are less than or equal (under $\succeq$) to terms in C.

For the base case, consider any instance $\Gamma \to A, \Delta$ of N that is a leaf in the proof and is the first premise in an inference by ordered resolution of depth 1. By N-linearity, the second premise must be a unit clause $A \to$ from $\neg C$. As the inference is ordered and $\succ$ is total, $A \succ B$, for any B in Γ or Δ. Alternatively $A, \Gamma \to \Delta$ is an instance of N that is the second premise in an inference by ordered resolution of depth 1, then the first premise must be a unit clause $\to A$ from $\neg C$. Since A is either maximal or selected, and then contains the maximal term, we have that $max_{t \in A}(t) \succeq max_{s \in B}(s)$ for any B in Γ, Δ. Note that we cannot have an inference by ordered factoring at depth 1 since all clauses in $\neg C$ are unit. Hence the conclusions of all inferences at depth 1 satisfy the required property.

Now consider an ordered resolution inference of depth > 1 from premises D and D'. Consider the case in which both D and D' are non-leaf nodes in the proof. Applying the induction hypothesis, the terms in both D and D' are smaller or equal to some term in C. As the conclusion D'' of the inference is smaller than the second premise, its terms are again bounded by the terms in C. If only one, D say, of the two premises is an inner node, the other, D' is an instance of N or a clause from $\neg C$. Applying the induction hypothesis to D, the reasoning is essentially the same as the reasoning in the base case of the induction,

Application of ordered factoring at depth > 1 does not create new atoms, and by the induction hypothesis all the old atoms are bounded by atoms in C. $\square$

In practice, the requirement that $\succ$ be total may be too strong and it suffices to consider all total well-founded extensions of partial orders.

Theorem 3 *Let N be a set of clauses and let $\succ$ be a well-founded partial ordering on terms. Then N is local with respect to $\succ$ if N is saturated up to redundancy with respect to $\succ$.*

Proof. Let N be saturated up to redundancy with respect to $\succ$. Then, by definition, N is saturated up to redundancy with respect to each extension $\succ'$ of $\succ$ that is well-founded and total on ground terms. We show that N is local with respect to $\succ$. Let C be a ground clause such that $N \models C$. We have to prove that $N \models_{\preceq} C$. To this end, let $\succ'$ be an ordering that (i) extends $\succ$, (ii) is well-founded, (iii) is total on ground terms, and satisfies (iv) $s \succ' t$, for any ground term t in C and any ground term s for which there is no term u in C such that $u \succeq s$. In other words, we construct $\succ'$ in a way such that all ground terms that are not smaller or equal to some term in C with respect to $\succ$ become strictly bigger with respect to $\succ'$ than any term in C. As N is saturated with respect to $\succ'$, we may apply Lemma 3 and infer that $N \models_{\preceq'} C$ and by (iv) we have $N \models_{\preceq} C$. $\square$

4.2 Saturation is Weaker than Locality

The converse of the previous theorem is not true in general; however it does hold in restricted cases.

Theorem 4 *Suppose that we have just one predicate symbol which is, in addition, unary. Let N be a set of Horn clauses and let $\succ$ be a well-founded partial ordering on terms. If N is local with respect to $\succ$ then N is saturated up to redundancy with respect to $\succ$.*

Proof. Let p be the unary predicate symbol, and let $\succ'$ be an extension of $\succ$ that is well-founded and total on ground terms. The ordering on ground atoms, defined by $p(s) \succ' p(t)$ iff $s \succ' t$, is total, well-founded and compatible with the term ordering.

Consider an inference by ordered resolution with respect to $\succ'$ from ground instances of clauses in N:

$$\frac{\Gamma_1 \rightarrow A \qquad A, \Gamma_2 \rightarrow B}{\Gamma_1, \Gamma_2 \rightarrow B}$$

A is strictly greater (in $\succ'$) than any of the atoms in the clause $C = \Gamma_1, \Gamma_2' \rightarrow B$, where $\Gamma_2' = \Gamma_2 \setminus \{A\}$. Since C is a logical consequence of N and N is local with respect to $\succ$, we have $N \models_{\preceq} C$, that is, C is implied by those instances of N in which all atoms are smaller or equal (in $\succeq$) than some atom in C. These instances of N are all smaller than A, and, in particular, smaller than $A, \Gamma_2 \rightarrow B$ with respect to $\succ'$. What we have shown is that $N^{\succ'}_{A,\Gamma_2 \rightarrow B} \models C$. As C implies the conclusion of the inference we have shown that the inference is redundant. $\square$

Factoring inferences may not be redundant if N also contains non-Horn clauses. One may generalize the Theorem to full clauses if factorization is built into the resolution inferences, that is, if one considers clauses as sets, rather than multisets, of literals. However, the main restriction in Theorem 4 concerns the signature.

We have shown that saturation implies locality in a very general setting that includes selection in addition to ordering restrictions. Let us briefly demonstrate why this combination gives us increased power in verifying order locality. Suppose N is the following set of clauses.

$$\begin{array}{rcl}
p(f(x)) & \rightarrow & q(x) \\
p'(f(x)) & \rightarrow & q'(x) \\
p'(x) & \rightarrow & p(x) \\
p(x) & \rightarrow & p'(x) \\
r(f(x)) & \rightarrow & p(x) \\
r'(f(x)) & \rightarrow & p'(x)
\end{array}$$

N is local with respect to the subterm ordering, but whatever ordering we choose on atoms, N will not be saturated under ordered resolution if the selection function is empty. Suppose that a is a constant and that, for instance, $p'(f(a)) \succ p(f(a))$. Then $p(f(a)) \rightarrow q'(a)$ can be obtained by ordered resolution from instances of the second and fourth clause, but cannot be obtained by any other proof that does not involve the maximal atom $p'(f(a))$ of the inference. However, if we select in (the ground instances of) the third and fourth clause the negative literal, the set of clauses is trivially saturated as no inference is possible: A clause that has a selected negative literal need not be considered as the first premise in a resolution inference.

We observe that requiring atoms to be ordered may be too restrictive when they contain the same maximal term. Moreover, clauses that contain the maximal term both in the antecedent and in the succedent may require one to consider non-local proofs of potentially unbounded depth in which a path of the form $A, A_1, \ldots, A_n, B$ (a plateau) exists such that the A_i, but neither A nor B, contain the maximal term. Selection improves the situation in some, but not in all cases. For instance, if we add the clauses

$$\begin{array}{rcl}
s(x) & \rightarrow & p(g(x)) \\
s'(x) & \rightarrow & p'(g(x))
\end{array}$$

to the above clause set, locality is maintained, but saturation is not. In general, to eliminate "Plateaus" of unbounded width requires a stronger form of redundancy that we will discuss in the following section.

5 The Horn Clause Case

For full clauses we have shown that saturation implies locality; however, only for a quite restricted class of signatures and orderings we have been able to prove the converse too (cf. Theorem 4). The purpose of this section is to show that saturation up to redundancy and order locality are in fact equivalent concepts in the case of Horn clauses provided the notion of redundancy is appropriately extended. *Throughout this section we assume that all clauses are Horn.*

5.1 Monotonicity

To simplify the technical treatment, we will generalize order locality to *monotonicity* by restricting the class of queries considered. A set of Horn clauses N is *monotone* with respect to a well-founded partial ordering $\succ$ on terms if for any ground *atom* A we have $N \models A$ if and only if $N \models_{\preceq} A$. A monotone theory has the property (motivating the name 'monotone') that every provable ground atom has a direct proof in which any term that occurs in some atom in the proof is smaller or equal to some term in its parent atom.

We will show that monotonicity is equivalent to saturation up to redundancy. Although restricting the queries may seem to reduce the usefulness of monotonicity, we show that there is no significant difference as a Horn query may be transformed to an equivalent atomic query over an extended signature. We justify this with the following construction.

Let N be a Horn theory over a signature Σ and $\succ$ a partial, well-founded ordering on ground terms over Σ. We construct a presentation N' over an extended signature Σ' that will be "equivalent" to N: N' will be monotone with respect to a "closely related" ordering $\succ'$ if and only if N is local with respect to $\succ$. Since locality says something about entailment of Horn clauses $\Gamma \rightarrow A$ from N whereas monotonicity just refers to atomic consequences A, we pass Γ as an additional parameter to predicates in N'. The function symbols in Σ' are, therefore, the function symbols of Σ, the predicate symbols of Σ, together with an additional binary (right-associative) function symbol ",". For each n-ary predicate p, Σ' contains an $n+1$-ary predicate p'. The extra argument in p' serves to pass the antecedent Γ. In addition, Σ' contains a binary predicate member to denote membership in lists constructed from ",".

Let A $[T]$ denote the set of ground atoms [terms] over Σ. In Σ' the atoms in A are terms. To distinguish Σ-atoms from Σ-terms when considered as terms over Σ', in Σ' we assume a three-sorted type discipline where A and T are two of the sorts, and where S is an additional sort, denoting A-sequences. The predicates p' have typing $p' \subseteq S{*}T{*}\ldots{*}T$, and $\mathsf{member} \subseteq A{*}S$. "," is of type $A{*}S \rightarrow S$ and the predicates p and functions f in Σ, respectively, as functions symbols in Σ' have the type $p : T{*}\ldots{*}T \rightarrow A$ and $f : T * \ldots * T \rightarrow T$.

The clauses of N' are the following:

$$
\begin{array}{rcl}
p'_1(s,\vec{t_1}),\ldots,p'_n(s,\vec{t_n}) & \rightarrow & p'_0(s,\vec{t_0}), \\
 & & \text{for } p_1(\vec{t_1}),\ldots,p_n(\vec{t_n}) \rightarrow p_0(\vec{t_0}) \text{ in } N \\
\mathsf{member}(p(x_1,\ldots,x_n),s) & \rightarrow & p'(s,x_1,\ldots,x_n), \\
 & & \text{for any } n\text{-ary predicate symbol } p \text{ in } N \\
 & \rightarrow & \mathsf{member}(a,(a,s)) \\
\mathsf{member}(a,s) & \rightarrow & \mathsf{member}(a,(b,s))
\end{array}
$$

On Σ' we consider the smallest ordering $\succ'$ containing $\succ$ and satisfying $(a, t) \succ' a$, $(a, t) \succ' t$, and $p(t_1, \ldots, t_k) \succ' t_j$, for all $p : T * \ldots * T \to A$ in Σ'.

The following properties are immediate by construction:

Lemma 4 *(i) Let $C = p_1(\vec{t_1}), \ldots, p_n(\vec{t_n}) \to p_0(\vec{t_0})$ be a ground clause over Σ. Then $N \models C$ if and only if $N' \models p'_0((p_1(\vec{t_1}), \ldots, p_n(\vec{t_n})), \vec{t_0})$.*

(ii) N' is monotone with respect to $\succ'$ if and only if N is local with respect to $\succ$

(iii) If Σ-terms of size n have at most $f(n)$ smaller terms under $\succ$, then Σ'-terms of size n have at most $O(max(n, n * f(n)))$ smaller terms under $\succ'$.

The significance of the lemma lies in the fact that from now on it will suffice to characterize monotonicity in terms of saturation by ordered resolution. For Horn clauses, monotonicity and locality are essentially equivalent concepts.

5.2 Monotonicity implies Saturation

The definition of redundancy that we have given in section 3.2 was based on the notion of entailment. Entailment then referred to *all* models of a set of clauses. In the context of Horn clauses a stronger form of entailment, based on the minimal model semantics [9], is equally important and useful. Throughout this section, entailment $\models$ will now only refer to the *minimal model*. That is, a Horn clause C is entailed by a set of Horn clauses N either if N is inconsistent or else if C is true in the minimal model of N. Under this new definition of entailment, a ground clause C is redundant in N if and only if either N_C is inconsistent, or else C is true in the minimal model of N_C. By basing redundancy on a particular model, rather than on all models, more inferences and clauses become redundant. This will be the key for showing that monotonicity and saturation are equivalent.

Theorem 5 *Let N be a set of clauses and let $\succ$ be a well-founded partial ordering on terms. Then N is monotone with respect to $\succ$ if and only if N is saturated up to redundancy with respect to $\succ$.*

Proof. The "if" direction of the theorem is proved similar to the proof of Theorem 3. Let us now consider the reverse direction and assume that N is monotone with respect to $\succ$. Moreover, we may assume that N does not contain the empty clause since otherwise N would be trivially saturated. Let $\succ'$ be any well-founded, total extension of $\succ$ on ground terms. We have to choose an atom ordering compatible with $\succ'$ (again denoted by $\succ'$) and a selection function S for which we will be able to prove that N is saturated with respect to $\succ'$ and S. For S we define $S(C) = \emptyset$, for any clause C, i.e., we do not select any negative literal in any clause. The definition of $\succ'$ proceeds in parallel to the inductive definition of a Herbrand interpretation J, which we later shall see is the minimal model of N.

Formally we apply induction over $\succ'$ (on terms) to define Herbrand interpretations J_s and J^s for all ground terms s as follows: J_s is the set $\bigcup_{s \succ' t} J^t$. For the construction of J^s let N^s be the set of ground instances of N in which s occurs and is the maximal term with respect to $\succ'$. By P^s we denote the subset of N^s in which s occurs positively. Moreover let T denote the standard "T_P"-operator used to define the semantics of Horn programs [9]. Then we define:

$$J^s = \bigcup_{n \geq 0} T^n_{P^s}(J_s) \ , \qquad N_s = \bigcup_{s \succ' t} N^t \ .$$

In parallel we define $\succ'$ on ground atoms inductively such that it is compatible with the term ordering $\succ'$ and reflects the logical dependencies between true atoms as given by the T_P-operator. Formally, let A and B be in J^s with s the maximal term in both A and B. Then there exist (minimal) indices $n, m \geq 1$ such that A is in $T^n_{P^s}(J_s) \setminus T^{n-1}_{P^s}(J_s)$ and B is in $T^m_{P^s}(J_s) \setminus T^{m-1}_{P^s}(J_s)$. Now, $A \succ' B$ if and only if $n > m$. When $n = m$ or when the ground atoms A and B do not share a maximal term, then orient them arbitrarily under $\succ'$ as compatible with the term ordering $\succ'$.

Now that we have extended $\succ'$ to atoms, we can construct a second, essentially equivalent family of partial interpretations I_C, for all ground instances C of N, by induction over the extension of $\succ'$ to clauses. Formally we define, for each such C, sets of ground atoms I_C and E_C such that I_C is $\bigcup_{C \succ D} E_D$. Furthermore, if C is a clause $\Gamma \to A$, where A is strictly maximal with respect to Γ, such that C is false in I_C then $E_C = \{A\}$. In that case, we also say that C is *productive* and that it *produces* A. In all other cases, $E_C = \emptyset$. Finally, let I be $\bigcup_C E_C$.

In what follows we shall also use the notation I^C for $I_C \cup E_C$. Hence I_C is the partial interpretation defined by clauses smaller than C, whereas I^C additionally includes the effect of C in this construction. Moreover, for any ground term s, let I^s $[I_s]$ denote the union of the E_C for all clauses C with a maximal term t such that $s \succeq' t$ $[s \succ' t]$.

For all ground terms s and ground instances C of N with maximal term s we have:

(i) $I^s = J^s$

(ii) I_C is the minimal model of N_C.

(iii) if C is a clause in N^s of the form $\Gamma \to A$ such that $A \succ' \Gamma$ then C is true in I^C

(iv) if C is a clause in N^s of the form $A, \Gamma \to \Delta$ such that A is maximal with respect to Γ, Δ (in $\succ'$), then C is true in I_C.

We prove these facts by induction over $\succ'$ on terms s with an inner induction over $\succ'$ on clauses with maximal term s.

Ad (i): In the construction of I only clauses C in which the head A is strictly maximal may contribute to I. Then C is in P^s. C is productive if, in addition, any of its body atoms is true already in I_C. On the other hand, if a clause contributes to J via the T_P-operator then its head A appears later in the iteration than any of its body atoms B. But then $\succ'$ has been fixed such that $A \succ' B$.

Ad (ii): We may use the induction hypothesis for (iii) and (iv) to infer that any ground instance D of N with $D \prec' C$ is true in I_C. Hence I_C is a model of N_C. Its minimality is obvious from the construction.

Ad (iii): Suppose C is false in I^C. Then Γ is true, but A is false in I^C. But then C would have to be productive, which is a contradiction.

Ad (iv): First, suppose that s occurs in Δ. Then C, as a clause of P^s, is true in J^s. Suppose C is false in I_C. Then A, Γ is true in I_C, hence Δ is an atom B which is true in J^s. Since $J^s = I^s$, B must be contained in I^s, and hence in I_C which is a contradiction. (In fact, the extension of $\succ'$ to clauses is such that clauses $D \succeq' C$ cannot produce atoms $B \preceq' A$ into I.)

Otherwise, assume that s occurs negatively, but not positively in C, and that C is false in I_C, hence A, Γ is true in I_C. By induction hypothesis for (iii) I_C is the minimal model of N_C. Therefore $N \models \Delta$. If Δ is empty and N is local, N contains the empty clause, which is a contradiction. Otherwise, $\Delta = B$, and if t is the maximal term in B, then $t \prec' s$, hence $t \not\succeq s$. As N is local, B follows from ground instances of N in which the maximal terms (with respect to $\succ$) are smaller or equal (with respect to $\succ$) than t. These clauses are all in N_C. Hence B is true in I_C, which is a contradiction. We may infer that C is true in I_C.

Now it is easy to see that N is saturated with respect to $\succ'$. By (iv) above, all ground instances of N that may possibly occur as the second premise in an ordered resolution inference are redundant, hence so is the inference. □

The preceding result is with respect to the notion of redundancy based on the minimal model semantics of Horn clauses. It is easy to see that if a set of Horn clauses is saturated up to redundancy with respect to the loose (all models) semantics then it is in particular also saturated with respect to the minimal model semantics.

6 Complexity Bounds for Entailment Problems

Using order locality we obtain new methods for establishing complexity bounds for entailment of ground clauses. We first present our main results which provide complexity upper bounds for entailment dependent on the orderings, or class of orderings, used and whether the problem is for Horn or full clauses. Afterwards, we consider how ordered resolution itself can serve as a decision procedure.

For the remainder of the paper we will assume that clause sets N under consideration are finite.

6.1 Order-dependent Complexity Bounds

If a clause set is saturated under an ordering that has the property that ground atoms have only finitely many smaller atoms that may be enumerated in finite time, then decidability of the ground entailment problem is obtained by enumerating an appropriate finite set of ground instances of the clauses. More precisely, let the size of a clause be the sum of the sizes (the number of nodes in its tree representation) of the terms in the clause. We may now express the complexity of ground entailment as a function of the size of the query. We begin with the Horn case.

Theorem 6 *Let N be a set of Horn clauses that is saturated up to redundancy with respect to $\succ$. Then for ground Horn clauses C, the entailment problem for $N \models C$ is decidable in time $O(f(m)^k + g(m))$ where m is the size of C and each term of size n has $O(f(n))$ smaller terms (under $\succ$) which may be enumerated in time $g(n)$. Furthermore, the constant k depends only on N.*

Proof. As N is saturated with respect to any total extensions of $\succ$, then by Theorem 3, N is local with respect to $\succ$. Hence $N \models C$ iff $N \models_{\preceq} C$. The set of ground instances N^C of N in which terms are smaller than or equal to some term in C may be generated first by enumerating all terms bounded by terms in C and after using these to instantiate clauses in N to form the ground clauses in N^C. The enumeration step requires $O(g(m))$ time and instantiation step requires time at most $O(f(m)^k)$ where k is a constant that depends on N only; in particular, an upper bound for k is the maximum number of variables in any clause in N. A tighter bound can be found using analysis similar to McAllester's "refined tractability theorem" in [10]. By lemma 1, we can decide $N^C \models C$ in time linear in the sizes of N^C and C. □

For full clauses we obtain a similar result by reduction to the satisfiability problem for $O(f(m)^k)$ ground clauses of bounded size:

Theorem 7 *Let N be a set of (full) clauses that is saturated up to redundancy with respect to $\succ$. Then the disentailment problem $N \not\models C$ is non-deterministically decidable in time $O(f(m)^k + g(m))$, where, as before, m is size of C, f and g the number of smaller terms and their enumeration time, and k a constant depending on N only.*

We will later provide examples of this. Note that McAllester's tractability result is a special case of ours for the Horn clause case and the subterm ordering: any term has only linearly many subterms which can be directly enumerated and hence locality implies that the ground Horn entailment problem is polynomial-time decidable.

6.2 General Decidability Results

We consider here how saturation can be used to obtain general decidability results. Afterwards we specialize this to yield particular complexity bounds.

For Horn clause sets, the satisfiability and entailment problems are undecidable. This remains the case even when sets are saturated up to redundancy with respect to an arbitrary ordering $\succ$. To see this[2], suppose N is an arbitrary set of Horn clauses and let N' consist of the set of clauses $q, \Gamma \to \Delta$ such that $\Gamma \to \Delta$ is in N, with q a new propositional variable not occurring in N. Choose an ordering such that q is the maximal atom. Then N' is trivially saturated as no ordered inferences are possible from N'. It is undecidable if $N' \models \neg q$, since this is equivalent to the inconsistency of N.

To obtain decidability, one must either constrain the ordering or the syntactic format of the clauses. The subsequent theorem shows that restrictions on occurrences of variables can lead to decidability for full, and hence also Horn, clauses.

Theorem 8 *Let N be saturated up to redundancy with respect to $\succ$. Assume, for each clause in N, that whenever one of its literals is maximal, then it contains all the variables of the clause. Then the entailment problem for N is decidable.*

Proof. As before we make use of the fact that $N \models C$ if and only if $N \cup \neg C$ is inconsistent if and only if (N-linear) saturation of $N \cup \neg C$ up to redundancy produces the empy clause. We show that with the required variable restriction, $N \cup \neg C$ can be finitely saturated. This is based on three facts. First, N-linear inferences that satisfy the ordering constraints can only produce ground clauses; since the maximal atoms of clauses in N contain all variables, these are all instantiated by ground terms upon any ordered inference in which at most one premise is in N, and hence, the other premise is a ground clause. Second, we may simplify newly generated ground clause by factoring, i.e., by deleting multiple occurrences of atoms in the antecedent or consequent. Such a simplification renders the original clause redundant, hence may be applied eagerly, without affecting refutational completeness. Third, with each ground clause D which is either in $\neg C$ or is produced by an N-linear inference (possibly followed by a factoring) we can associate a chain of ground atoms $A_1 \succ \ldots \succ A_n = A'$ such that A_1 is an atom in the query C, A' is the maximal atom in D, and n is the depth of the proof of D. We show this fact by induction over n. For the base case, $n = 1$, D is a literal in $\neg C$, and the case is trivial. If $n > 0$ then D is produced from either a resolution or a factoring inference. Suppose that D is the conclusion of an ordered resolution inference from premises $\Gamma \to A, \Delta$ and $A, \Gamma' \to \Delta'$. Suppose that the second premise $A, \Gamma' \to \Delta'$ is a ground clause to which the induction hypothesis may be applied, and for which the

[2]The following simple proof is due to Uwe Waldmann.

proof depth is $n-1$. Therefore we have a chain $A_1 \succ \ldots \succ A_{n-1}$ of ground atoms leading from some query atom to the maximal atom $A_{n-1} = A$ of this premise of D. As the premises of the inference may be assumed as fully factored, $A_{n-1} \succ A_n$, if A_n is the maximal atom in D, and thus we have extended the chain to one of length n. If the first premise is a ground clause which is not an instance of N and which has proof depth $n-1$ then the reasoning is essentially the same. Factoring inferences are redundant since we factor computed clauses eagerly.

Now, if this strategy of N-linear saturation for $N \cup \neg C$ were to produce an unbounded number of (ground) clauses, the maximal proof depth of this set of clauses would be unbounded. But that would imply the existence of an unbounded decreasing chain of ground atoms starting out from some atom in the query C. This is a contradiction to the well-foundedness of $\succ$. □

6.3 Decidability/Complexity (Mixed Case)

It should be possible to refine the previous theorem to give an upper bound on the complexity of entailment problems as a function of the maximal length $l(n)$ of decreasing chains of atoms that start from any given atom of size n. Instead, we show how this result can be combined with the previous to provide upper bounds for entailment problems.

Suppose that N is saturated with respect to an ordering $\succ$. Moreover, assume that $\succ$ is an extension of a possibly smaller ordering $\succ'$ such that for each clause C in N there is an atom A in C that contains all variables of C and such that $A \succ' B$, for all other atoms B in the clause (in an atom ordering $\succ'$ that is compatible with the term ordering $\succ'$). Finally, let ground terms of size n have $O(f(n))$ smaller terms with respect to $\succ'$, which may be enumerated in $O(g(n))$ time.

Theorem 9 *Under the assumptions listed above, entailment is decidable in time $O(f(m)^k + g(m))$, if N is a Horn clause set. Disentailment is decidable non-deterministically in time $O(f(m)^k + g(m))$ if N is a full clause set. In both cases m refers to the size of the query.*

Proof. When N entails C then Theorem 8 provides the existence of a ground proof in which all atoms are smaller under $\succ$ then those in C. Under the additional assumptions, they are also smaller under $\succ'$ and so N is local with respect to $\succ'$. Hence, Theorems 6 and 7 apply. □

6.4 Special Cases: Polynomial and Exponential Orderings

We obtain polynomial bounds for entailment or disentailment if orderings are employed for which terms have only polynomially many smaller ones, and which can be enumerated in polynomial time. A simple example of such a *polynomial ordering* is the subterm ordering $\cdot\!>$. A refinement, based on a precedence π on function symbols, is the following: $s \cdot\!>_\pi t$ iff (i) $s \cdot\!> t$, or else (ii) $s = f(...)$, $t = g(t_1, \ldots, t_n)$, $s \cdot\!> t_i$, for $1 \leq i \leq n$, and $f >_\pi g$. In this ordering each term can have at most $O(|t|^n)$ smaller terms, where n is the maximal arity of function symbols. We will refer to this ordering as the *π-modified subterm ordering*.

These orderings remain polynomial if we extend them by arbitrarily many additional ground facts $s \cdot\!> t$ in a well-founded and query-independent way.

We obtain exponential bounds for entailment or disentailment if orderings are employed for which terms have only exponentially many smaller ones, and which can be enumerated in exponential time. An example of such an *exponential ordering* is the Knuth-Bendix ordering (KBO) [8]. An instance of the KBO is specified by a weight function and a precedence on the function symbols. Ground terms are compared by first considering their weights. If the weights are equal, the topmost function symbols are compared with respect to their precedence. If these are the same, their arguments are compared lexicographically. It has been shown in [6] that under certain restrictions decreasing chains $t_1 \succ t_2 \succ \ldots$ of ground terms can be of at most exponential length if $\succ$ is a KBO. The statement holds, for instance, if all function symbols have an arity less than or equal to 1. It is true for an arbitrary signature if unary function symbols (and constants) have a weight greater than 0. Function symbols of arity greater than 1 can have an arbitrary non-negative weight (including 0). Since KBOs are total orderings (for total precedences) this implies that there are at most exponentially many smaller terms for any given ground term.

Double exponential orderings can be obtained by polynomial interpretations. In [6] it was also shown that decreasing chains of ground terms with respect to polynomial interpretations have a double exponential length bound. Hence if one allows arbitrary polynomial interpretations in a KBO (rather than just linear, weight-defined term interpretations) one obtains orderings where terms have at most double exponentially many smaller terms.

7 Characterizing Complexity Classes

In this section we will restrict ourselves again to Horn clauses. We have previously seen that order locality can be used to identify the complexity of entailment problems for clause sets. Here we consider the converse problem: we show that when a ground entailment problem is decidable in polynomial (exponential, ...) time, then there is a Horn clause set defining that entailment relation which is local under a polynomial (exponential, ...) ordering. This result is in the spirit of results in descriptive complexity theory, for example that datalog programs precisely capture the set of queries on a finite ground database decidable in finite time [7, 14]. In our setting, in the problem $R \models \Gamma \rightarrow A$, the atoms in Γ play the role of the finite ground database and A and R determine the ground query.[3] Our proof is general and applies to complexity classes associated with other orderings (e.g., exponential orderings) as well.

We proceed by showing how Turing machines can be encoded as Horn theories that are trivially saturated under orderings which reflect the successor relation between configurations in computation histories. Note that our proof is quite direct and conceptually simpler then similar results for polynomial time queries proven in [5]; moreover, the encoding we use are not possible in McAllester's setting as it requires problem specific orderings.

[3] Note that in Datalog, A need not be ground, but since the data universe is assumed to be finite this is equivalent to testing polynomially many (n^k, where n is the size of the data-universe and k the arity of A) ground queries.

7.1 Encoding

A Turing machine is a four-tuple $M = \langle Q, \Delta, \delta, F\rangle$ where Q is a finite set of states, with starting state $q_0 \in Q$, Δ is a finite tape alphabet, which contains an empty tape square symbol $\natural$, $\delta : Q \times \Delta \to Q \times \Delta \times \{L, R\}$ is the transition function and, $F \subseteq Q$ is the final set of states.

We associate machine configurations with strings in $\Delta^*(Q \times \Delta)\Delta^*$: if $w(q, a)v$ is a configuration then wav is the tape content and the Turing machine head points to a and is in state q. Let $C = w(q, a)v$ and $C' = w'(q', a')v'$ be configurations, then C' is a successor (the unique successor, when M is deterministic) of C, which we write as $C \rightsquigarrow C'$ if

1. $(q', b, L) \in \delta(q, a)$, $w = w'a'$, and $v' = bv$, or
2. $(q', b, R) \in \delta(q, a)$, $w' = wb$, and $v = a'v'$, or
3. $(q', b, R) \in \delta(q, a)$, $w' = wb$, $v = v' = \epsilon$, and $a' = \natural$.

A configuration $w(q, a)v$ is a *halting* configuration if $\delta(q, a) = \emptyset$, *accepting* if $q \in F$, and *initial* with respect to the string $x = a_1 \ldots a_n$ if $q = q_0$, $w = \epsilon$, $a = a_1$, and $v = a_2 \ldots a_n$. A *computation history* (of length $k + 1$) on input x is a sequence $C_0, \ldots,$ C_k of configurations where $C_i \rightsquigarrow C_{i+1}$, and C_0 is initial with respect to x. The history is accepting (halting) if C_k is accepting (halting).

Given a machine M we construct a signature and a rule set R_M that represents M's computation. In the signature, each $q \in Q$ corresponds to a constant (given the same name) and each $a \in \Delta$ corresponds to a unary function, which we will apply by juxtaposition, e.g., at instead of $a(t)$. In addition we have a new constant ϵ, a function symbol c of arity 4 for representing configurations, and a unary predicate symbol tm. We represent a configuration

$$w_1 \ldots w_m(q, a)v_1 \ldots, v_n$$

by the ground term

$$c(w_m w_{m-1} \ldots w_1\epsilon, q, a, v_1v_2 \ldots v_n\epsilon)\,.$$

We previously listed three kinds of Turing machine transitions. Associated with each kind is a set of rules correspond to each possible transition. Rules for the first kind are of the form

$$tm(c(W', q', a', bV)) \to tm(c(a'W', q, a, V))\,, \tag{1}$$

which formalizes the relationship between the successor configuration (on the left) and the predecessor (on the right). Note that lower case letters range over states and constants determined by M's transition function δ and upper case variables are universally quantified in the rule. Rules for the second and third kind of transition are given by

$$tm(c(bW, q', a', V')) \to tm(c(W, q, a, a'V')) \tag{2}$$

and

$$tm(c(bW, q', \natural, \epsilon)) \to tm(c(W, q, a, \epsilon))\,. \tag{3}$$

Finally, there is an additional set of rules, for each $f \in F$, which reflects accepting computations.

$$\to tm(c(W, f, A, V)) \tag{4}$$

Let M be a Turing machine and let R_M be the set of all rules derived from M. Given a string x, let $\lceil x \rceil$ denote the representation of the initial configuration with respect to x. By construction, for every $x \in \Delta^*$,

$$x \in L(M) \quad \leftrightarrow \quad R_M \models tm(\lceil x \rceil)\,.$$

Let us say that any R_M with this property *recognizes the language* $L(M)$.

7.2 Saturation and Characterization of Complexity Classes

Let a Turing machine M and its corresponding theory be given as above. Call M *uniformly terminating* when all computations from any initial configuration terminate. For such an M, it is simple to find an ordering $\succ_M$ with respect to which R_M is order local: take the transitive closure of the transition relation $\leadsto^+$, that is, define $c(v,q,a,w) \succ_M c(v',q',a',w')$ iff $v(q,a)w \leadsto^+ v'(q',a')w'$.

Lemma 5 *Let M be uniformly terminating. Then R_M is local with respect to $\succ_M$.*

Proof. As M is uniformly terminating, $\succ_M$ defines a well-founded partial order. Let $\succ'$ denote any total well-founded extension of $\succ_M$. We extend $\succ'$ to atoms by $tm(c(v,q,a,w)) \succ' tm(c(v',q',a',w'))$ if and only if $c(v,q,a,w) \succ' c(v',q',a',w')$. Now, observe that the right-hand side of each rule (the positive atom) is always greater than the left-hand side (the negative atom). As a result, there can be no ordered resolution steps, because we cannot satisfy the ordering requirement for the second premise when the resolved atom occurs negatively. Hence R_M is saturated up to redundancy with respect to every total well-founded extension $\succ'$ of $\succ_M$. So R_M is local with respect to $\succ_M$. □

Each rule in R directly simulates a Turing machine transition except for the final rule which simulates acceptance. Hence when M is deterministic, the length of the unique computation history starting from the initial configuration of a string x is precisely the number of terms smaller than x under $>$. Combining this with the previous lemma gives us the following:

Lemma 6 *Let M be a deterministic Turing machine whose running time for all input strings x is bounded by $f(|x|)$. Then there is an ordering $\succ$, in which each ground term t has at most $f(|t|)$ $\succ$-smaller terms and a $\succ$-local theory R which recognizes $L(M)$.*

This is not too surprising. Our Horn proofs trees are simply unbranching linear sequences of atoms that directly reflect a computation histories for M. Note that if M were nondeterministic, then terms would have $O(k^{f(|x|)})$ $\succ$-smaller terms where k is determined by δ and is the maximum number of transitions for any pair of $q \in Q$ and $a \in \Delta$.

Finally, we relate this result to polynomial time rule sets R.

Theorem 10 *Let R be a Horn theory over a finite signature for which ground entailment is polynomial time decidable. Then there is a Horn theory S over an extended signature that (1) is order local under a polynomial ordering and (2) for all Horn clauses C over the signature of R, $R \models C \quad \leftrightarrow \quad S \models C$.*

Proof. If the entailment problem for R is polynomial time decidable then there is some deterministic Turing machine M that operates on strings (polynomially) encoding ground Horn clauses and accepts exactly those entailed by R. Further, the time required for any computation is bounded by a polynomial. Using Lemma 6 it follows that that there is a Horn clause theory S that is local under a polynomial ordering and recognizes $L(M)$.

To finish the proof requires extending the signature and some metaprogramming (a bit in the spirit of the construction in section 5). We augment the signature of S with declarations for R; we assume that the names of representatives in S are distinct from the functions and predicates in R which they represent. The resulting theory, however, doesn't prove the same Horn clauses C as R but instead C encoded, using the tm predicate, as the appropriate initial Turing machine configuration. If S is to accept C itself we must add rules for a metaprogram such that for each C, $S \models tm(...) \rightarrow C$ where $tm(...)$ is the corresponding initial configuration for C in the theory S. This requires adding rules for a Horn metaprogram which translate clauses C to configurations $tm(...)$. We omit details here but this can be done in linear time with a simple metaprogram that can be added to S preserving locality under a polynomial ordering. □

For concreteness we have proven the theorem for polynomial time. But we can uniformly replace the word "polynomial" by exponential or any other large complexity class and the proof remains valid. Hence polynomial (exponential, etc.) time entailment corresponds precisely to locality under the corresponding orderings.

8 Order Locality in Practice

8.1 Finite Saturation of Clause Sets

When a set of clauses is not saturated we may try to saturate it by adding all consequences that can be obtained from non-redundant resolution inferences. Saturation-based theorem provers such as our Saturate system [4] may be employed for that purpose provided that they support appropriate classes of orderings and that they provide strong enough techniques for simplification and for the detection of redundancy. The methods for proving redundancy that we have implemented in the Saturate system have proved sufficiently powerful for saturating a number of non-trivial examples, including congruence closure, tree embeddings, various fragments of propositional logic, and total orderings. Most of the methods that we currently apply for detecting redundancy are described in [13].

In order to apply the Theorem 3 we need to simultaneously saturate with respect to all total extensions $\succ'$ of a given term ordering $\succ$. This in particular includes the choice of a method for extending the $\succ'$ to atoms. The following method has turned out to be useful in this context. Let $\succ'$ be a well-founded, total ordering on ground terms, and let $>_\pi$ denote a well-founded, total ordering (a precedence) on predicates. By the *lexicographic extension* $\succ'_\pi$ of $\succ'$ to ground atoms (with respect to $>_\pi$) we mean the ordering defined as follows: $p(t_1, \ldots, t_m) \succ'_\pi q(s_1, \ldots, s_n)$ if and only if:

(i) $p >_\pi q$, and for all j there exists an i such that $t_i \succeq' s_j$, or

(ii) $p = q$ and there exists a k such that $s_i = t_i$, for $1 \leq i < k$, and $t_k \succ' s_k$, and for all $k < j \leq n$ there exists an i such that $t_i \succeq' s_j$, or

(iii) there exists an i such that $t_i \succ' s_j$, for all j.

Note that the disjunction (ii) or (iii) is not exclusive and that (iii) implies (i) if $p >_\pi q$. (iii) implies compatibility with the term ordering.

The relation $\succ'_\pi$ coincides with the lexicographic path ordering if ground terms are considered as constants (their structure is forgotten), and where the precedence is given by the term ordering $\succ'$, by the precedence $>_\pi$ on the predicates, and by assuming that each predicate is smaller than any term. $\succ'_\pi$ is a total and well-founded ordering on ground atoms. An alternative method might be based on the multiset extension of $\succ'$ applied to the sets or multisets, respectively, of arguments of the atoms.

Given $>_\pi$, to saturate a set of clauses with respect to all the $\succ'_\pi$ requires solving certain ordering constraints. An ordering constraint is a propositional formula γ over inequations $A > B$ and $A \geq B$, where A and B are (possibly nonground) atoms. γ is called solvable if there is (i) an extension of the given signature of function symbols, (ii) a total and well-founded ordering $\succ'$ on ground terms over the extended signature extending $\succ$, and (iii) a ground substitution σ for the variables in γ, such that $\gamma\sigma$ is satisfied when interpreting $>$ and $\geq$ as the strict and the non-strict version $\succ'_\pi$ and $\succeq'_\pi$, respectively, of the lexicographic extension of $\succ'$ with respect to $>_\pi$. A procedure that checks for satisfiability of such constraints can be incomplete in that unsolvable constraints may be classified as "possibly solvable", but not vice-versa. Saturation may now employ this constraint solver to discard any inference for which the ordering constraint is shown to be unsolvable. Moreover, when proving the redundancy of an inference only clauses smaller than the maximal premise may be employed. Sufficient criteria for this problem may be formulated as constraints for which the classification "unsolvable" indicates that the clause in question is indeed sufficiently small. Theories that are treated according to this use of ordering constraints will, upon termination of the saturation process, be saturated with respect to all the $\succ'_\pi$. Polynomial complexity bounds for entailment follow if the ordering $\succ$ is polynomial.

We have mentioned that lexicographic atom orderings are special cases of lexicographic path orderings if one forgets the structure of terms and considers them as constants. Based on that observation we have implemented a constraint solver as a special instance of the constraint solver for the lexicographic path ordering that we had available in our system as an implementation of the algorithm in [12].

8.2 Saturation Modulo an Equivalence

The theory of ordered resolution and its applications as outlined above can be straightforwardly extended to resolution modulo an equivalence $\sim$ on terms and atoms. The extension requires that the respective orderings be compatible with $\sim$ in the sense that if $s \succ t$ and $s \sim s'$ and $t \sim t'$ then $s' \succ t'$. Moreover, unification modulo $\sim$ should be computable and finitary. Finally, for obtaining complexity results in the spirit of section 6.1, the equivalence between ground terms should be decidable in linear time or at most polynomial time. Note that the complexity results are then independent of the cardinality of equivalence classes. The proof of Theorems 6 and 7 can be refined by describing the details of a resolution procedure that employs $\sim$-matching on ground atoms rather than enumerating all equivalent $\sim$-variants of the required ground clauses.

As an example, suppose that $\sim$ is just the symmetry of a binary relation symbol R, that is, $R(s,t) \sim R(t,s)$, for any two terms s and t. Then the multiset extension of a given term ordering (applied to the arguments of R-atoms) yields a compatible atom ordering.

9 Examples Checked by Saturate

The following examples report about cases in which we were able to successfully saturate a theory with the Saturate system. When we speak of "the" saturation we refer to the particular strategy of saturation as it is implemented in the Saturate system. Different strategies might lead to different results. While Knuth-Bendix completion for unit equations produces a uniquely determined (possibly infinite) canonical system (depending on the ordering), no related result is known for clauses and saturation under ordered resolution.

9.1 Congruence Closure

$$\begin{array}{rcl} & \rightarrow & x = x \\ x = y & \rightarrow & y = x \\ x = y,\, y = z & \rightarrow & x = z \\ x = y & \rightarrow & f(x, z) = f(y, z) \\ x = y & \rightarrow & f(z, x) = f(z, y) \end{array}$$

This presentation of the axioms for congruences over a binary function symbol is not subterm local. For instance,

$$a = b,\ c = d \rightarrow f(a, c) = f(b, d)$$

cannot be derived without employing an intermediate term $f(a, d)$ or $f(b, c)$, neither of which is a subterm of the query. The given presentation can be saturated with respect to the subterm ordering, which proves that the congruence closure problem has a polynomial time upper bound. The saturated system we obtain with Saturate is the following:

$$\begin{array}{rcl} & \rightarrow & x = x \\ x = y & \rightarrow & y = x \\ x = y,\, y = z & \rightarrow & x = z \\ x = y & \rightarrow & f(x, z) = f(y, z) \\ x = y & \rightarrow & f(z, x) = f(z, y) \\ x = y,\, f(y, z) = u & \rightarrow & f(x, z) = u \\ x = y,\, f(z, y) = u & \rightarrow & f(z, x) = u \\ x = y,\, u = v & \rightarrow & f(x, u) = f(y, v) \\ x = y,\, u = v,\, f(y, v) = z & \rightarrow & f(x, u) = z \end{array}$$

If Saturate did a more sophisticated backward checking of redundancy, the clauses 4–7 could have been deleted after the last two clauses are generated. We obtain $O(n^3)$ as an upper bound which obviously is not a very tight one.

9.2 Embedding of Trees

The following defines the strict embedding relation $>$ on terms over one binary function symbol f as a transitive, irreflexive relation that includes the subterm relation and is compatible with contexts.

$$
\begin{aligned}
x > y,\ y > z &\ \rightarrow\ x > z \\
x > x &\ \rightarrow \\
&\ \rightarrow\ f(x,\ y) > x \\
&\ \rightarrow\ f(x,\ y) > y \\
x > y &\ \rightarrow\ f(x, z) > f(y, z) \\
x > y &\ \rightarrow\ f(z, x) > f(z, y)
\end{aligned}
$$

The given presentation is not subterm-local. Saturation with respect to the subterm ordering gives:

$$
\begin{aligned}
x > y,\ y > z &\ \rightarrow\ x > z \\
x > x &\ \rightarrow \\
&\ \rightarrow\ f(x,\ y) > x \\
&\ \rightarrow\ f(x,\ y) > y \\
x > y &\ \rightarrow\ f(x, z) > f(y, z) \\
x > y &\ \rightarrow\ f(z, x) > f(z, y) \\
x > y &\ \rightarrow\ f(x, z) > y \\
x > f(y, z) &\ \rightarrow\ x > y \\
x > y &\ \rightarrow\ f(z, x) > y \\
x > f(y, z) &\ \rightarrow\ x > z \\
x > y,\ f(y, z) > u &\ \rightarrow\ f(x, z) > u \\
x > y,\ z > f(x, u) &\ \rightarrow\ z > f(y, u) \\
x > y,\ f(z, y) > u &\ \rightarrow\ f(z, x) > u \\
x > y,\ u > f(z, x) &\ \rightarrow\ u > f(z, y) \\
x > y,\ z > u &\ \rightarrow\ f(z, x) > f(u, y) \\
x > y,\ u > z,\ f(z, y) > v &\ \rightarrow\ f(u, x) > v \\
x > y,\ z > u,\ v > f(z, x) &\ \rightarrow\ v > f(u, y)
\end{aligned}
$$

We obtain $O(n^3)$ as an upper bound for its Horn theory.

9.3 Propositional Calculus

The following example, formalizing a fragment of propositional logic as Horn clauses, is a slight modification of [10].

$$
\begin{aligned}
true(p) &\ \rightarrow\ true(or(p, q)) \\
true(p) &\ \rightarrow\ true(or(q, p)) \\
true(or(p, q)), true(not(p)) &\ \rightarrow\ true(q) \\
true(or(p, q)), true(not(q)) &\ \rightarrow\ true(p) \\
true(not(p)), true(not(q)) &\ \rightarrow\ true(not(or(p, q))) \\
true(not(or(p, q))) &\ \rightarrow\ true(not(p))
\end{aligned}
$$

$$
\begin{array}{rcl}
true(not(or(p,q))) & \rightarrow & true(not(q)) \\
true(not(p)), true(p) & \rightarrow & true(\mathsf{ff}) \\
true(\mathsf{ff}) & \rightarrow & true(p) \\
true(not(not(p))) & \leftrightarrow & true(p)
\end{array}
$$

This is a theory presentation that is not subterm local. In fact, to derive $true(or(not(p),q)), true(p) \rightarrow true(q)$ requires $true(not(not(p))$ as an intermediate atom. It is, however, saturated with to the modified subterm ordering over the precedence $or \succ not \succ \mathsf{ff}$, giving us $O(n^2)$ as an upper bound for its Horn theory. In this ordering $or(not(p),q)$ is greater than $not(not(p))$.

As another simple example, consider the following presentation of full propositional logic.

$$
\begin{array}{rcl}
true(A) & \rightarrow & true(or(A,B)) \\
true(A) & \rightarrow & true(or(B,A)) \\
false(A), false(B) & \rightarrow & false(or(A,B)) \\
 & \rightarrow & true(A), false(A) \\
false(not(A)) & \rightarrow & true(A) \\
true(A) & \rightarrow & false(not(A))
\end{array}
$$

The presentation is trivially saturated under the subterm ordering, hence we obtain CoNP as an upper bound for the entailment (tautology) problem.

9.4 Orderings

The following defines the Horn theory of nonstrict partial orderings.

$$
\begin{array}{rcl}
 & \rightarrow & x = x \\
x = y & \rightarrow & y = x \\
x = y,\ y = z & \rightarrow & x = z \\
x = y & \rightarrow & x \leq y \\
x \leq y,\ y \leq z & \rightarrow & x \leq z \\
x \leq y,\ y \leq x & \rightarrow & x = y
\end{array}
$$

Saturation with respect to the subterm ordering (and the precedence $\leq \succ =$) yields

$$
\begin{array}{rcl}
 & \rightarrow & x = x \\
x = y & \rightarrow & y = x \\
x = y,\ y = z & \rightarrow & x = z \\
x = y & \rightarrow & x \leq y \\
x \leq y,\ y \leq z & \rightarrow & x \leq z \\
x \leq y,\ y \leq x & \rightarrow & x = y \\
x = y,\ y \leq z & \rightarrow & x \leq z \\
x \leq y,\ y = z & \rightarrow & x \leq z
\end{array}
$$

from which we obtain a cubic upper bound for the entailment problem. This is an automatic proof of Theorem 3.4 in [11]. (The latter paper also shows that this theory is in fact the in some sense largest fragment of Allen's interval logic which is computationally tractable.)

In a similar way one obtains a saturated version of the first-order theory of a *total* ordering, giving CoNP an upper bound for its universal fragment.

10 Locality of Propositional Sequent Calculus

The previous examples were checked with the Saturate system. The following example is of a more theoretical nature and was not run on the Saturate system for reasons that will become clear below.

10.1 The Definition of LK

We shall assume that propositions are built from two logical symbols, conjunction and negation, from which other standard connectives can be defined. Then LK consists of the inference rules given in figure 1, in which $\triangleright$ is used to separate antecedent and succedent in sequents. Antecedent and succedent are possibly empty multisets of formulas built from propositional variables, conjunction and negation.

$$A \triangleright A \qquad \text{(axiom)}$$

$$\frac{\Gamma \triangleright \Delta}{\Gamma, \Lambda \triangleright \Delta, \Pi} \qquad \text{(weakening)}$$

$$\frac{\Gamma, A, A \triangleright \Delta}{\Gamma, A \triangleright \Delta} \qquad \text{(contraction left)}$$

$$\frac{\Gamma \triangleright \Delta, A, A}{\Gamma \triangleright \Delta, A} \qquad \text{(contraction right)}$$

$$\frac{\Gamma, A, B \triangleright \Delta}{\Gamma, A \wedge B \triangleright \Delta} \qquad \text{(conjunction left)}$$

$$\frac{\Gamma \triangleright \Delta, A \quad \Gamma \triangleright \Delta, B}{\Gamma \triangleright \Delta, A \wedge B} \qquad \text{(conjunction right)}$$

$$\frac{\Gamma \triangleright \Delta, A}{\Gamma, \neg A \triangleright \Delta} \qquad \text{(negation left)}$$

$$\frac{\Gamma, A \triangleright \Delta}{\Gamma \triangleright \Delta, \neg A} \qquad \text{(negation right)}$$

$$\frac{\Gamma \triangleright \Delta, A \quad \Lambda, A \triangleright \Pi}{\Gamma, \Lambda \triangleright \Delta, \Pi} \qquad \text{(cut)}$$

Figure 1: The Horn Theory LK

The Horn theory of LK can be seen as a theory of derivable inference rules. Despite the fact that cut-elimination holds for sequents, the cut rule is not redundant when one considers the Horn theory. For example, derived rules corresponding to natural deduction elimination, e.g.,

$$\frac{\Gamma \rhd \Delta, A \wedge B}{\Gamma \rhd \Delta, A}$$

are logical consequences of the logical rules and cut. In the presence of cut and weakening, the contraction rules and the logical rules of LK are in fact equivalences which may be used to eliminate the logical symbols from sequents and to remove multiple occurrences of propositional variables.

10.2 An Encoding of LK as a Horn Theory

In our encoding $\mathcal{LK}$ of LK the ground atoms are built from a family $\rhd_{k,n}$ of $(k+n)$-ary predicate symbols taking (representations of propositional) formulas as arguments. A ground atom $\rhd_{k,n}(A_1, \ldots, A_k, B_1, \ldots, B_n)$, with propositional formulas A_i and B_i, represents the sequent $A_1, \ldots, A_k \rhd B_1, \ldots, B_n$. We admit only such atoms $\rhd_{k,n}(A_1, \ldots, A_k, B_1, \ldots, B_n)$ in which the A_i and B_i subsequences are ordered with respect to the formula orderings that will be defined later, and in which none of the A_i and B_i, respectively, occurs more than once in their respective subsequences. Alternatively we might say that we identify any two ground atoms whenever their *sets* of positive and negative formulas are the same, that is, their antecedents and succedents, respectively, are equivalent modulo ACI1.

The reason why we introduce the multisets on the meta-level is that the only class of orderings for which we will be able to show that LK is saturated is the class of orderings that are multi-set extensions of certain orderings on formulas. If multisets were on the object level we would have to consider *any* extension of orderings on formulas to multisets of formulas.

On the non-ground level, atoms are written as expressions of the form $\rhd(H, G)$, where H and G are expressions denoting sets of formulas. These set expressions are built from "+", denoting union of formula sets, from propositional formulas, from metavariables A, B, C for formulas, and meta-variables Δ, Γ, Λ and Π for sets of formulas. Then $\rhd(H, G)$ represents all ground atoms of the form $\rhd_{k,n}(A_1, \ldots, A_k, B_1, \ldots, B_n)$ for which $\{A_1, \ldots, A_k\}$ and $\{B_1, \ldots, B_n\}$, respectively, are sets of propositional formulas denoted by the set expressions H and G. Note that "+" is *not* in the language of $\mathcal{LK}$ ground clauses. It is used for a finite "non-ground" representation of certain infinite sets of $\mathcal{LK}$ ground atoms and clauses. The ground terms of $\mathcal{LK}$ are simply the propositional formulas. Since we are working modulo idempotence, there is no need anymore for any explicit representation of the contraction rules.

It follows the list of the axioms of $\mathcal{LK}$, finitely presented by using the concept of non-ground atoms as described before.

$$\begin{aligned}
&\to \rhd(A, A) && (5)\\
\rhd(\Gamma, \Delta) &\to \rhd(\Gamma + \Lambda, \Delta + \Pi) && (6)\\
\rhd(\Gamma + A + B, \Delta) &\leftrightarrow \rhd(\Gamma + (A \wedge B), \Delta) && (7)\\
\rhd(\Gamma, \Delta + A), \rhd(\Gamma, \Delta + B) &\to \rhd(\Gamma, \Delta + (A \wedge B)) && (8)\\
\rhd(\Gamma, \Delta + (A \wedge B)) &\to \rhd(\Gamma, \Delta + A) && (9)\\
\rhd(\Gamma, \Delta + (A \wedge B)) &\to \rhd(\Gamma, \Delta + B) && (10)\\
\rhd(\Gamma + \neg A, \Delta) &\leftrightarrow \rhd(\Gamma, \Delta + A) && (11)\\
\rhd(\Gamma, \Delta + \neg A) &\leftrightarrow \rhd(\Gamma + A, \Delta) && (12)
\end{aligned}$$

$$\rhd(\Gamma, \Delta + A), \rhd(\Lambda + A, \Pi) \quad \rightarrow \quad \rhd(\Gamma + \Lambda, \Delta + \Pi) \tag{13}$$

Theorem 11 *$\mathcal{LK}$ is saturated up to redundancy by ordered resolution with respect to any total and well-founded extension of the subterm ordering on propositional formulas.*

Proof. Let $\succ$ be a total and well-founded extension of the subterm ordering on propositional formulas. We have to provide a compatible extension of $\succ$ to $\mathcal{LK}$ ground atoms and to show the redundancy of all ordered resolution inferences between $\mathcal{LK}$-clauses. Let $\succ$ be an ordering on atoms for which $\rhd_{k,n}(A_1, \ldots, A_k, B_1, \ldots, B_n) \succ \rhd_{k',n'}(A'_1, \ldots, A'_{k'}, B'_1, \ldots, B'_{n'})$ whenever the *set* $\{A_1, \ldots, A_k, B_1, \ldots, B_n\}$ is greater than the *set* $\{A'_1, \ldots, A'_{k'}, B'_1, \ldots, B'_{n'}\}$ in the multiset extension of $\succ$. With this ordering we can now analyze possible inferences.

The only nontrivial inferences are those involving weakening and/or cut. First we observe that instances

$$\rhd(\Gamma, \Delta + A), \rhd(\Lambda + A, \Pi) \rightarrow \rhd(\Gamma + \Lambda, \Delta + \Pi)$$

of the encoding (13) of the cut rule are redundant if the cut formula A is in any of the sets Γ, Δ, Λ or Π. These instances are subsumed by weakening (6) and hence need not be considered further. Moreover, for $\rhd(\Gamma, \Delta + A)$ to be the strictly maximal atom of an instance of cut it is not possible that $\Lambda \setminus \Gamma$ or $\Pi \setminus \Delta$ contains a propositional formula which is greater than A with respect to $\succ$. Otherwise $\rhd(\Gamma + \Lambda, \Delta + \Pi)$ would be a bigger atom. A dual property holds when one assumes the maximality of $\rhd(\Lambda + A, \Pi)$. If $\rhd(\Gamma + \Lambda, \Delta + \Pi)$ is assumed to be maximal, then A cannot be maximal in either $\Gamma + \Lambda$ or $\Delta + \Pi$. We now consider some representative cases of inferences involving cut.

Consider the clause

$$\rhd(\Gamma, A + \Delta), \rhd(\Lambda' + A, \Pi') \rightarrow \rhd(\Gamma + \Lambda + \Lambda', \Delta + \Pi + \Pi')$$

obtained by resolution from the premises

$$\rhd(\Gamma, A + \Delta) \rightarrow \rhd(\Gamma + \Lambda, A + \Delta + \Pi)$$

(an instance of (6)) and

$$\rhd(\Gamma + \Lambda, A + \Delta + \Pi), \rhd(\Lambda' + A, \Pi') \rightarrow \rhd(\Gamma + \Lambda + \Lambda', \Delta + \Pi + \Pi')$$

(an instance of (13)). This is another proof of the same clause:

$$\cfrac{\cfrac{\rhd(\Gamma, A + \Delta) \quad \rhd(\Lambda' + A, \Pi')}{\rhd(\Gamma + \Lambda', \Delta + \Pi')}\,(13)}{\rhd(\Gamma + \Lambda + \Lambda', \Delta + \Pi + \Pi')}\,(6)$$

All $\rhd$-atoms in that new proof are smaller than the (strictly) maximal atom $\rhd(\Gamma + \Lambda, A + \Delta + \Pi)$ of the resolution inference. In particular, $\rhd(\Gamma + \Lambda + \Lambda', \Delta + \Pi + \Pi')$ is smaller as A is bigger than any formula which is not in Γ, Δ or Π.

As another example consider the clause

$$\rhd(\Lambda + A + B, \Pi), \rhd(\Gamma, (A \wedge B) + \Delta) \rightarrow \rhd(\Gamma + \Lambda, \Delta + \Pi)$$

obtained from

$$\rhd(\Lambda + A + B, \Pi) \rightarrow \rhd(\Lambda + (A \wedge B), \Pi)$$

and

$$\rhd(\Gamma, (A \wedge B) + \Delta), \rhd(\Lambda + (A \wedge B), \Pi), \rightarrow \rhd(\Gamma + \Lambda, \Delta + \Pi),$$

where $\rhd(\Lambda + (A \wedge B), \Pi)$ is the strictly maximal atom in either clause. This is another proof of the same clause:

$$\dfrac{\dfrac{\rhd(\Lambda + A + B, \Pi) \qquad \dfrac{\rhd(\Gamma, (A \wedge B) + \Delta)}{\rhd(\Gamma, A + \Delta)}\,(9)}{\rhd(\Lambda + \Gamma + B, \Pi + \Delta)}\,(13) \qquad \dfrac{\rhd(\Gamma, (A \wedge B) + \Delta)}{\rhd(\Gamma, B + \Delta)}\,(10)}{\rhd(\Gamma + \Lambda, \Delta + \Pi)}\,(13)$$

We again utilize that $A \wedge B$ is bigger than any formula in $\Gamma \setminus \Lambda$ and $\Delta \setminus \Pi$. A and B are both smaller than $A \wedge B$. Hence all atoms of the proof are smaller than $\rhd(\Lambda + (A \wedge B), \Pi)$, and the inference is redundant.

The other cases of inferences between cut and logical rules follow the same pattern and are left to the reader. $\Box$

$\mathcal{LK}$ without cut can be shown saturated with SATURATE by ordered resolution modulo AC1 for "+" with a linear polynomial interpretation such that exponential-time decidability follows.

10.3 Consequences of the Locality of $\mathcal{LK}$

Theorem 12 *The Horn theory of $\mathcal{LK}$, and hence the theory of derivable inference rules for* LK*, is decidable in exponential time.*

Proof. Since $\mathcal{LK}$ is saturated under all extensions of the subterm ordering on formulas, it follows that any ground consequence Q of $\mathcal{LK}$ follows from $\mathcal{LK}$ ground clauses in which all terms (i.e., propositional formulas) are subformulas of formulas in Q. Depending on the size of Q there are exponentially many $\mathcal{LK}$ ground atoms with that property, hence exponentially many ground clauses of $\mathcal{LK}$ are needed for proving Q, and any of these clauses can be generated in linear time. $\Box$

Note that the term ordering (terms are the propositional formulas) in the proof of locality is the subterm ordering. We nevertheless do not get a polynomial bound as our notion of non-ground clauses is not the usual one.

Theorem 13 (Cut Elimination) *If a sequent S is derivable in LK then there exists a derivation of S without application of cut.*

Proof. Let S be the sequent $A_1, \ldots, A_k \;\rhd\; B_1, \ldots, B_n$ over the formulas A_i and B_j. We prove the theorem by showing that if S is derivable in LK then $\rhd_{k,n}(A_1, \ldots, A_k, B_1, \ldots, B_n)$ follows from $\mathcal{LK} \setminus \{(13)\}$. Suppose S is derivable in LK. Then $\hat{S} = \rhd_{k,n}(A_1, \ldots, A_k, B_1, \ldots, B_n)$ is a consequence from $\mathcal{LK}$. We may restrict ourselves to the case where the A_i and B_j are propositional variables. This follows since we may employ the logical equivalences in $\mathcal{LK}$ to eliminate any logical symbol that might have been present in S. The result of that elimination is a set of sequents without logical symbols such that the original sequent S is derivable if and only if each of the sequents in the set is derivable.

Let $\succ$ be a total ordering of the propositional variables. By Theorem 12 there exists a linear proof of $\hat{S}$ in $\mathcal{LK}$ by ordered resolution with respect to $\succ$. Such a proof may be viewed as a tree with $\hat{S}$ as root and each interior node an instance of a clause is $\mathcal{LK}$. By locality, each node in the proof tree is an $\mathcal{LK}$-atom in which all terms are propositional variables in S. Moreover, any node is smaller in the atom ordering than its ancestor. The only $\mathcal{LK}$-clauses that can participate in a resolution step are axioms (5), weakening (6) and (13). We show, by induction over $\succ$, that inferences with (13) are redundant. Let

$$\rhd(\Gamma, \Delta + A), \rhd(\Lambda + A, \Pi) \to \rhd(\Gamma + \Lambda, \Delta + \Pi)$$

be an instance of (13) in any given proof of $\hat{S}$. Then $S' = \rhd(\Gamma, \Delta + A)$ and $S'' = \rhd(\Lambda + A, \Pi)$ are both smaller than $\rhd(\Gamma + \Lambda, \Delta + \Pi)$, and A is a propositional variable in S. By the induction hypothesis, the proofs of S' and S'' only employ axioms and weakening. But then $\rhd(\Gamma + \Lambda, \Delta + \Pi)$ is also the weakening of an axiom. $\Box$

11 Comparison between Subterm Locality and Order Locality

A full description of the algorithm given by [5] for testing subterm locality is beyond the scope of this paper. However, here we compare the main ideas behind McAllester's locality procedure and our saturation based procedure, especially how both constructively correspond to proof-normalization arguments. For concreteness, we compare both methods on a simple example: the theory of transitive closure where N consist of the transitivity clause for some predicate p, i.e.,

$$N = \{p(x, y),\ p(y, z) \to p(x, z)\}.$$

11.1 Subterm Locality via Backward Chaining

Given a set of terms Υ, let $st(\Upsilon)$ be the set of subterms in Υ. As indicated in Section 2, a set of Horn clauses N is subterm local if for every ground Horn clause C, we have $N \models C$ if and only if $N \models\!\!\!| C$; this means that there is a set Υ consisting of the ground terms in N and C where every term occurring in the proof of C from N contains only terms in $st(\Upsilon)$. Subterm locality can be tested by looking for a *feedback-event with respect to* N, which is a set of terms Φ and term α, where (i) $st(\Upsilon) \subseteq st(\Phi)$, (ii) the proper subterms of α are contained in $st(\Phi)$, (iii) $N \not\models_\Phi C$, and (iv) $N \models_{\Phi\cup\{\alpha\}} C$. The intuition is that a feedback event identifies a set of terms Φ which is "just too small" to bound the terms needed to derive C: the additional *feedback witness* α is needed for some derivation. It is easy to show that N is local iff there is no feedback event with respect to N.

It [5] it is shown that subterm locality, in general, is undecidable; however, a procedure is given that identifies a non-trivial subclass of local clause sets. This procedure is based on backchaining, which is Horn clause resolution, augmented with book-keeping information. The additional information supports termination analysis, i.e., identifying feedback events or the impossibility of such events. In particular, for Σ, Γ sets of atoms, and A an atom, a *template*,

$$\Sigma, \Gamma \vdash_{\Upsilon\cup\{\alpha\}} A$$

represents the Horn clause

$$\Sigma, \Gamma \to A$$

where (i) Υ is subterm closed and contains all proper subterms of α, but not α itself, (ii) $st(\Sigma \cup \{A\}) \subseteq \Upsilon$, and (iii) $st(\Gamma) \subseteq \Upsilon \cup \{\alpha\}$ and every atom in Γ contains α. This additional book-keeping information is used for termination: when Γ is empty, the resulting template represents a potential feedback event, with feedback witness α. Moreover, in some cases it is possible to terminate with success and indicate that the rule set is local.

In our example, we start with

$$\{\},\ \{p(x,y), p(y,z)\} \vdash_{\Upsilon \cup \{y\}} p(x,z),$$

which is the only possible template (in general there may be many) corresponding to the single clause in N. The variable Υ represents all possible sets meeting the above three conditions contain at least x and z. Now, according to [5] we may backchain through this template by resolving some atom in Γ, say $p(x,y)$, with any clause in N (here, there is only one) and partition the resulting atoms into Γ, when they contain α and otherwise into Σ.

In our example, backchaining through $p(x,y)$ yields two new templates:

$$\{p(x,w)\},\ \{p(w,y), p(y,z)\} \vdash_{\Upsilon \cup \{y\}} p(x,z)$$

and

$$\{\},\ \{p(x,y), p(y,y), p(y,z)\} \vdash_{\Upsilon \cup \{y\}} p(x,z)\,.$$

These both are instances of the clause

$$p(x,w), p(w,y), p(y,z) \to p(x,z)$$

but two templates are needed to represent the cases where w is distinct from the feedback witness y, or identical.

Backchaining may terminate with success (indicating locality) when all new templates can be *justified* from previously generated templates. Justification is complex and we omit details here (the technical details are considerable, see [5]); however, the idea is roughly that new templates are justified when previously generated templates (considered as rules) entail atoms $A_1, \ldots A_k$ under $\models_{\Upsilon \cup \{\alpha\}}$ and these A_i entail (from the original set of rules) the desired goal using $\models_\Upsilon$. In our example, the second template generated is justified as, if we drop the second hypothesis $p(y,y)$, then it is an instance of the starting template. The first template is justified since an instance of the starting template is

$$\{\},\ \{p(w,y), p(y,z)\} \vdash_{\Upsilon \cup \{y\}} p(w,z)$$

and from $p(x,w)$ and $p(w,z)$ we can prove $p(x,z)$ without recourse to atoms mentioning the feedback witness y.

Justification, and its proof of correctness, can be interpreted constructively as a kind of proof normalization procedure: given a direct proof, of say $p(a,b)$ from N, the proof that the rule set N is justified, constructively explains how to transform any proof mentioning a feedback witness α to one without α. Reasoning about backchaining amounts to reasoning about transforming the composition of inference steps. In our example, justifying the two rules generated by backchaining, gives rise to two transformations for "pushing" α (in our case y) upwards towards the leaves of a proof. For example, the argument that justifies the first derived template can be interpreted constructively as explaining how to transforming a proof tree like that on the left to that on the right.

$$\dfrac{\dfrac{\begin{matrix}P_1\\ \vdots\\ R(a,w)\end{matrix} \quad \begin{matrix}P_2\\ \vdots\\ R(w,y)\end{matrix}}{R(a,y)} \qquad \begin{matrix}P_3\\ \vdots\\ R(y,b)\end{matrix}}{R(a,b)} \quad \Longrightarrow \quad \dfrac{\begin{matrix}P_1\\ \vdots\\ R(a,w)\end{matrix} \qquad \dfrac{\begin{matrix}P_2\\ \vdots\\ R(w,y)\end{matrix} \quad \begin{matrix}P_3\\ \vdots\\ R(y,b)\end{matrix}}{R(w,b)}}{R(a,b)}$$

Similarly the justification for the second template corresponds to the following proof transformation:

$$\dfrac{\dfrac{\begin{matrix}P_1\\ \vdots\\ R(a,y)\end{matrix} \quad \begin{matrix}P_2\\ \vdots\\ R(y,y)\end{matrix}}{R(a,y)} \qquad \begin{matrix}P_3\\ \vdots\\ R(y,b)\end{matrix}}{R(a,b)} \quad \Longrightarrow \quad \dfrac{\begin{matrix}P_1\\ \vdots\\ R(a,y)\end{matrix} \quad \begin{matrix}P_3\\ \vdots\\ R(y,b)\end{matrix}}{R(a,b)}$$

The first transformation, which is under the assumption that $y \neq w$, moves the potential feedback witness y closer to the leaves. The second transformation, when $y = w$ is even simpler. Application of such transformations must terminate (and the termination ordering corresponds to the well-founded ordering implicit in the induction used in [5]) and each application makes progress moving y upwards. Moreover, a feedback witness may not occur in any leaves, since these correspond to assumptions and, by definition, feedback witnesses must be strict superterms of any term appearing in the Horn clause to be proven. Hence, for every proof with feedback, there is a corresponding proof without feedback, so the rule set is subterm local.[4]

11.2 Order Locality via Saturation

Saturation may be directly applied to show that the transitivity example is order local. Indeed, this example is so simple, that one can easily show by hand that it is saturated with respect to any term ordering.

We now consider the same example, using order locality. Let Σ be any vocabulary of function symbols and let $\succ$ be any total ordering on the ground terms over Σ. N is saturated, and hence local, with respect to the lexicographic extension of $\succ$ to atoms. In fact, consider an inference by ordered resolution of the form

$$\frac{p(x,u),\ p(u,y) \to p(x,y) \quad p(x,y),\ p(y,z) \to p(x,z)}{p(x,u),\ p(u,y),\ p(y,z) \to p(x,z)}$$

For the inference to be ordered we must have $x \succ y \succ z$ and $x \succ y \succ u$. But then

$$\frac{p(u,y),\ p(y,z) \to p(u,z) \quad p(x,u),\ p(u,z) \to p(x,z)}{p(x,u),\ p(u,y),\ p(y,z) \to p(x,z)}$$

[4] In the above we only considered resolution with the left atom in the initial rule, which corresponded to a transformation on the left-hand branch. There are similar cases for resolution with the right atom giving rise to similar transformations on the right side. Interestingly, McAllester's proof of correctness for justifications allows one to consider only backchaining (resolution) at one position and his explanation for this corresponds, constructively, to a kind of commutation condition on resolution proofs.

gives us a proof of $p(x, u),\ p(u, y),\ p(y, z) \rightarrow p(x, z)$ in which the two premises are smaller than $p(x, y),\ p(y, z) \rightarrow p(x, z)$.

By Theorem 6 we obtain the fact that the entailment problem for transitivity is polynomial. This result is also based on proof transformations. Any proof in a saturated theory N has an N-linear form (Theorem 2) and this form is used as the basis of showing the invariant that term size is bounded by the initial clause C at each step. Proof transformation lies behind the proof of the normal form theorem. Namely, to show that this linear-form exists, it suffices to consider a non-linear resolution step (the argument for factoring is similar) between two clauses in N:

$$\frac{\Gamma_1 \rightarrow A, \Delta_1 \qquad B, \Gamma_2 \rightarrow \Delta_2}{\Gamma_1\sigma, \Gamma_2\sigma \rightarrow \Delta_1\sigma, \Delta_2\sigma}$$

Since N is saturated, the step is redundant, and can be replaced by one involving smaller instances of clauses in N.

There are several other relationships between our approach and McAllester's worth noting. First, redundancy testing in our setting plays a role analogous to backchaining in McAllester's: McAllester generates new clauses to test subterm locality whereas when a clause set is order local, saturation will not generate any new clauses. Instead, testing order locality consists of checking if each possible ordered resolution step is redundant. In general, this is undecidable, but then so is locality as a property of (Horn or full) clause sets. Second, and related to the first point, is that when we generate new clauses, they play a different role then clauses generated by McAllester's procedure. Namely, when he begins with a clause set N and terminates successfully with a set N', he can conclude that N itself was local; the new clauses were generated just to test that fact. On the other hand, when we terminate with a larger set N' we conclude that the larger set is local, but not necessarily the smaller set. This has advantages in the sense that it allows us, not merely to test locality, but (analogous to completion procedures) find a local rule set N' deductively equivalent to a non-local one N.

12 Conclusion

In this paper we have generalized subterm locality to locality with respect to arbitrary well-founded term orderings. This provides a characterization of the complexity of entailment problems with respect to any deterministic time complexity class above and including P. Order locality is conceptually more general than subterm locality, also in that it applies to full clauses, and, hence also captures certain non-deterministic time complexity classes.

As an operational tool for testing order locality we have proposed the concept of ordered resolution (with selection) up to redundancy. In fact, we have been able to show that order locality is in most cases equivalent to being saturated up to redundancy by ordered resolution. This links automated complexity analysis to standard techniques from resolution theorem proving. It also allows one to bring in standard improvements, e.g., AC unification and the like, into the framework. Saturation is a compiler which, when it terminates, has transformed a theory presentation that was not local into one that is. We have demonstrated the usefulness of our Saturate theorem prover in this regard.

References

[1] L. Bachmair and H. Ganzinger. On restrictions of ordered paramodulation with simplification. In M. Stickel, editor, *Proc. 10th Int. Conf. on Automated Deduction, Kaiserslautern*, volume 449 of *Lecture Notes in Computer Science*, pages 427–441, Berlin, 1990. Springer-Verlag.

[2] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994. Revised version of Technical Report MPI-I-91-208, 1991.

[3] W. Dowling and J. Gallier. Linear-time algorithms for testing the satisfiability of propositional horn formulae. *Journal of Logic Programming*, 3:267–284, 1984.

[4] H. Ganzinger. The Saturate system. Available on the world-wide web under URL http://www.mpi-sb.mpg.de/SATURATE/Saturate.html, 1994.

[5] Robert Givan and David McAllester. New results on local inference relations. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR-92)*, pages 403–412. Morgan Kaufmann Press, October 1992.

[6] D. Hofbauer and C. Lautemann. Termination proofs and the length of derivations. In N. Dershowitz, editor, *Proceedings of the 3rd International Conference on Rewriting Techniques and Applications*, volume 355 of *Lecture Notes in Computer Science*, pages 167–177. Springer-Verlag, 1989.

[7] Neil Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[8] Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[9] Robert A. Kowalski and M. H. van Emden. The semantics of predicate logic as a programming language. *Journal of the Association for Computing Machinery*, 23:733–742, October 1976.

[10] David A. McAllester. Automatic recognition of tractability in inference relations. *Journal of the Association for Computing Machinery*, 40(2):284–303, April 1993.

[11] Bernhard Nebel and Hans-Jürgen Bürckert. Reasoning about temporal relations: A maximal tractable subclass of allen's interval algebra. *Journal of the Association for Computing Machinery*, 42(1):43–66, 1995.

[12] R. Nieuwenhuis. Simple LPO constraint solving methods. *Information Processing Letters*, 47(2):65–69, August 1993.

[13] P. Nivela and R. Nieuwenhuis. Saturation of first-order (constrained) clauses with the *Saturate* system. In Claude Kirchner, editor, *Rewriting Techniques and Applications, 5th International Conference, RTA-93*, volume 690 of *Lecture Notes in Computer Science*, pages 436–440, Montreal, Canada, June 16–18, 1993. Springer-Verlag.

[14] M. Vardi. Complexity of relational query languages. In *14th Symposium on Theory of Computation*, pages 137–146, 1982.