Automated Complexity Analysis Based on Ordered Resolution

David Basin
Universität Freiburg, Germany

and
Harald Ganzinger
Max-Planck-Institut für Informatik, Saarbrücken, Germany

We define order locality to be a property of clauses relative to a term ordering. This property generalizes the subformula property for proofs where the terms appearing in proofs can be 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.

Categories and Subject Descriptors: F41 [**Mathematical Logic and Formal Languages**]: Mathematical Logic; I23 [**Artificial Intelligence**]: Deduction and Theorem Proving; F13 [**Computation by Abtract Devices**]: Complexity Measures and Classes

General Terms: Theory, Verification

Additional Key Words and Phrases: First-order theories, automated theorem proving, complexity analysis, ordered resolution

---

Preliminary versions of the results in this paper have been presented at the *11th IEEE Symposium on Logic in Computer Science* (New Brunswick, New Jersey, U.S.A., July 1996).
This research was supported in part by the ESPRIT Basic Research Working Groups 6028 (Construction of Computational Logics) and 6112 (COMPASS).
Name: David Basin
Affiliation: Institut für Informatik, Universität Freiburg
Address: D-79110 Freiburg, Germany
Name: Harald Ganzinger
Affiliation: Max-Planck-Institut für Informatik
Address: Stuhlsatzenhausweg 85, D-66123 Saarbrücken, Germany

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works, requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept, ACM Inc., 1515 Broadway, New York, NY 10036 USA, fax +1 (212) 869-0481, or `permissions@acm.org`.

1. INTRODUCTION

Many algorithms and decision problems can be reduced to ground entailment problems: determining if a ground formula is entailed by a given theory. For example, congruence closure is the problem of whether $s = t$ follows from ground equations $s_1 = t_1, \ldots, s_n = t_n$, in a theory that axiomatizes equality to be a congruence, i.e., reflexive, symmetric, transitive, and compatible with contexts. These axioms (see section 7) are Horn clauses and hence congruence closure is the decision problem of whether a ground Horn clause, $s_1 = t_1, \ldots, s_n = t_n \rightarrow s = t$, is entailed by a Horn clause theory.

Our research is motivated by McAllester [Givan and McAllester 1992; McAllester 1993] who gives procedures for recognizing sets of Horn clauses for which ground entailment is polynomial time decidable. He tests if Horn clauses have a property, which he calls *locality*, which corresponds to a kind of subformula property for Horn clause proofs; when a clause set has this property, then there is a bounded number of terms (polynomial in the size of the query) that need to appear in proofs and this is used to solve ground queries in polynomial time.

Our contributions are fourfold. First, we generalize locality by introducing *order locality*, which is a property of clauses, parameterized by term orderings. This is a strict generalization 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 how the complexity of the ground entailment problem for an order local clause set is determined by the ordering and the structure of the clauses. For example, congruence closure is local under a *polynomially bounded ordering*, which is one where every ground term has polynomially many smaller terms, and this implies polynomial time decidability. We show that tighter bounds are possible too and how other classes of orderings capture other complexity classes, such as co-NP and exponential time.

Second, we prove that order locality is closely related, and in many cases equivalent, to a clause set being *saturated up to redundancy under ordered resolution* (defined in section 3). This provides both an alternative characterization of when a clause set is local and yields a procedure whereby standard ordered-resolution based theorem provers can test locality. We have used the Saturate system [Ganzinger et al. 1994] to give machine checked proofs of the complexity of a number of nontrivial entailment problems. This includes automatically verifying 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 exponential time, and that the first-order theory of total orderings is in co-NP.

We show that saturation can not only detect locality in many cases (order locality and saturation are both undecidable), but for many non-local clause sets, saturation still terminates and yields a local presentation of the same theory; this is analogous to the rôle played by Knuth/Bendix completion in the equational setting. Although we do not explore them here, there are additional applications of our results to program transformation and optimization. Namely, Horn clauses constitute logic programs and saturation can produce a program that can be executed on ground queries more efficiently than the original.

Third, we show that order locality is not only sufficient for establishing the com-

plexity of entailment problems, but in many cases it is *necessary* too. We prove that when a decision problem belongs to one of the standard deterministic time complexity classes (polynomial time, exponential time, etc.) then it can be encoded by a clause set that is local under a corresponding ordering (polynomial, exponential, etc.). Hence the scope of order locality is quite wide; it can even be used to give an alternative formalization of these complexity classes.

Fourth, we show that order locality can be used for a proof-theoretic analysis of the structural properties of logical calculi. We give an extended example, for the propositional sequent calculus LK, where we use ordered resolution not only to bound the length of sequent calculus proofs, but also to show that the cut rule may be eliminated without sacrificing completeness. This illustrates the possibility of using ordered resolution techniques to automatically carry out proof-theoretic arguments, which traditionally require error-prone hand analysis of many cases.

Overall, the results of this paper are a first step towards computer-assisted analysis of the complexity and structural properties of logical theories and logic programs. The rôle of the user is to provide a theory presentation together with a well-founded ordering, including an analysis of how many smaller terms each ground term has. The automatic part is saturation up to redundancy of the presentation under ordered resolution. Redundancy analysis is essentially a process that aims at finding transformations for proofs that are too complex. The more intelligence one builds into this analysis, the less explicit the presentation and ordering need to be.

Organization

In section 2 we define 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 prove that saturation suffices to establish the locality of a clause set. Unfortunately, while saturation implies locality the converse is false. However, we show that the converse does hold for Horn clauses and a weaker notion of saturation. In section 5 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 ground entailment problem for N. In section 6 we prove a form of the converse: when the entailment problem for N belongs to certain complexity classes, then there is a Horn clause set encoding the problem, with an associated ordering, that is local. In section 7 we indicate what is involved in mechanizing our approach and present examples carried out with the Saturate system. In section 8 we give an extended example, of a theoretical nature, and show locality for a Horn theory of the propositional sequent calculus. This establishes both a complexity bound for propositional tautology checking and provides a basis for showing cut elimination. In section 9 we compare saturation under ordered resolution to the algorithm proposed by McAllester and Givan [Givan and McAllester 1992] and finally, in section 10, we draw conclusions.

2. PRELIMINARIES

2.1 Entailment and Proofs

We shall use standard terminology for clauses and clause-based proofs. *Terms* and *atoms* are defined relative to a (often implicit) *signature* Σ, which specifies

function and predicate symbols and their arity: terms are built from functions and variables, and atoms are built by applying predicates to n-tuples of terms. A *(full) clause* is a formula of the form $\Gamma \to \Delta$, where Γ and Δ are multisets of atoms; Γ represents the negative literals (the *antecedent* of the clause) and Δ the positive literals (the *succedent*). 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 $\{A_1, \ldots, A_m\} \to \{B_1, \ldots B_n\}$. We also make simplifications when one of the multisets is empty, e.g., writing A for the positive unit clause $\{\} \to A$, and $\neg A$ for $A \to \{\}$.

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 A is of the form $p(\ldots, t, \ldots)$ and t *occurs in a clause* if it occurs in an atom of the clause. An expression (i.e., a term, atom, or clause) is *ground* if it does not contain any variables. A ground instance of an expression is obtained by substituting ground terms for variables.

Given a set of clauses N, the *(ground) entailment problem* for N is to decide if $N \models C$, for ground clauses C, called *queries*. A *refutational proof* of $N \models C$ is a tree whose root is labeled by the empty clause, and whose internal nodes are labeled by results of resolution or factoring inferences from their direct descendents. Leaves are ground instances of clauses in N or unit clauses $\neg L$, with L a literal in C. When N is a set of Horn clauses, a *direct proof* of a ground atom A is an unordered tree, labeled by atoms, whose root is A and whose subtrees are direct proofs of atoms that constitute the antecedent of a ground instance of a clause in N with succedent A. Leaves are ground instances of positive unit clauses. The instance of the clause in N that is applied at the root of a direct proof is called the *end clause* of the proof. A direct proof of a ground clause $A_1, \ldots, A_n \to A$ from N is a direct proof of A from $N \cup \{A_1, \ldots, A_n\}$. A term t *occurs* in either kind of proof if t occurs in a clause or in an atom labeling the proof tree. A direct proof Π with leaves labeled $A_1, \ldots, A_n$ will also be written as $\Pi(A_1, \ldots, A_n)$. In this case, if Π_i, for $1 \leq i \leq m \leq n$, are direct proofs of A_i, then we denote by $\Pi(\Pi_1, \ldots, \Pi_m, A_{m+1}, \ldots, A_n)$ the proof that results from replacing each leaf labeled A_i in Π by the proof Π_i. For example, if N contains the two clauses $p(x) \to p(f(x))$ and $p(x), q(y) \to p(g(x, y))$, then

$$\Pi : \dfrac{\dfrac{p(a)}{p(f(a))}}{p(f^2(a))} \qquad \Pi' : \dfrac{\dfrac{p(f^2(a))}{p(f^3(a))} \qquad q(a)}{p(g(f^3(a), a))} \qquad \Pi'' : \dfrac{\dfrac{\dfrac{p(a)}{p(f(a))}}{p(f^2(a))}}{p(f^3(a))} \qquad \dfrac{q(a)}{p(g(f^3(a), a))}$$

are direct proofs $\Pi(p(a))$, $\Pi'(p(f^2(a)), q(a))$, and $\Pi''(p(a), q(a))$ of, respectively, $p(a) \to p(f^2(a))$, $p(f^2(a)), q(a) \to p(g(f^3(a), a))$, and $p(a), q(a) \to p(g(f^3(a), a))$ from N. We see that $\Pi'' = \Pi'(\Pi(p(a)), q(a))$, that is, it can be obtained by pasting Π into the $p(f^2(a))$ leaf of Π'.

We will introduce different *bounded entailment relations*, which are entailment relations based on restricted clause instances. Let Υ be a set of ground terms and C a clause $\Gamma \to \Delta$. We say that N *entails* C *with respect to* Υ, and write $N \models_\Upsilon C$,

if C is already entailed by the set of those ground instances of clauses in N in which all terms are subterms of terms in Υ. When Υ contains precisely all ground terms in N and C, we write $N \models\!\!\!\models C$ for $N \models_\Upsilon C$. Using $\dot{>}$ to denote the *strict subterm ordering*, that is, $s \dot{>} t$ if and only if t is a proper subterm of s, the property $N \models_\Upsilon C$ 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 $t \geq s$.

We shall make use of various ordering relations on expressions. A *(strict, partial) ordering* on a set X is a transitive and irreflexive binary relation on X. An ordering $\succ$ is said to be *well-founded* if there is no infinite descending chain $x_1 \succ x_2 \succ \cdots$ of elements. Any ordering $\succ$ on a set X can be extended to an ordering $\succ_{mul}$ on finite multisets over X as follows: $\Delta_1 \succ_{mul} \Delta_2$ if (i) $\Delta_1 \neq \Delta_2$ and (ii) whenever $\Delta_2(x) > \Delta_1(x)$ then $\Delta_1(y) > \Delta_2(y)$, for some y such that $y \succ x$. (Here $\Delta_i(x)$ denotes the number of occurrences of x in the multiset Δ_i, and $>$ denotes the standard "greater-than" relation on the natural numbers.) Given a multiset, any smaller multiset is obtained by (repeatedly) replacing an element by zero or more occurrences of smaller elements. If an ordering $\succ$ is total (respectively, well-founded), so is its multiset extension.

2.2 Locality

McAllester's work is based on Horn clauses.[1] 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 $N \models\!\!\!\models C$. Locality is a desirable property of clause sets because when N is local, then $N \models C$ if and only if $N' \models C$, where N' comprises those ground instances of clauses in N where all terms in N' are subterms in N or C.

Since all the clauses in N' are ground, its entailment problem can be solved efficiently, provided N, and hence N', is finite.

LEMMA 2.1 ([DOWLING AND GALLIER 1984]) For ground Horn clauses, $N \models C$ is decidable in time linear in the sum of the sizes of N and C.

Since the number of clauses in N' is polynomial in the size of C, it follows that when N is local, its ground entailment problem is tractable.

LEMMA 2.2 ([MCALLESTER 1993]) If N is a finite, local set of Horn clauses, then for all ground Horn queries C, $N \models C$ is polynomial time decidable in the size of C.

Note that this lemma provides a basis for solving queries in the universal fragment of a first-order theory in polynomial time if the matrix of the quantification is in conjunctive normal form and if the signature contains infinitely many constant symbols. The quantifiers can be eliminated by replacing quantified variables with fresh (Skolem) constants, not occurring in N or C. The resulting formula is entailed if each conjunct is entailed, and a clause $\Gamma \rightarrow A_1, \ldots, A_k$ is entailed by a Horn theory

[1] In [Givan and McAllester 1992] rules are Horn clauses and in [McAllester 1993] 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.

if and only if one of the Horn clauses $\Gamma \to A_i$ is entailed, which can be checked in polynomial time.

We generalize locality by allowing arbitrary term orderings and full clauses for our clause sets N and our queries C. A *term ordering* is a well-founded, (strict) partial ordering on ground terms. A term ordering $\succ'$ is called a *[total] extension* of a term ordering $\succ$ if $\succ' \supseteq \succ$ [and if $\succ'$ is a total ordering]. Given a term ordering $\succ$, we say that a set N of clauses *entails a clause C bounded by* $\succ$, and write $N \models_{\preceq} C$, if and only if there is a proof of $N \models C$ from those ground instances of clauses in N in which each term is smaller than or equal (under $\succeq$) to some term in C. A set of clauses N is *local with respect to* $\succ$ if whenever $N \models C$ for a ground clause C, then $N \models_{\preceq} C$. To distinguish this generalized notion of locality from McAllester's, we also refer to it as *order locality* in contrast to McAllester's definition, which we 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 $>$. 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 $\Gamma \to A$. However, there are several ways of reducing McAllester's definition to ours. For instance, for any N we can define a transformation that maps a query $\Gamma \to A$ to another query $\Gamma' \to A$ where $N \models\!\!\!\models \Gamma \to A$ if and only if $N' \models_{\leq} \Gamma' \to A$, for a correspondingly modified theory N'. Namely, when N contains the k ground terms $t_1, \ldots, t_k$, then we conservatively extend our theory with new uninterpreted, unary predicate symbols p_{t_i} and replace occurrences of t_i in a clause in N by a new variable x_{t_i}, adding the "guard" $p_{t_i}(x_{t_i})$ to the antecedent of the respective clause. We set $\Gamma' = p_{t_1}(t_1), \ldots, p_{t_k}(t_k), \Gamma$. 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 that is subterm local. With the methods presented in section 4.2, one can even show that the transformed theory N' is local with respect queries containing arbitrary atoms in the new symbols p_{t_i}.

An Example

As a simple example (for less trivial examples, see section 7), consider the following set N_1 of Horn clauses, which defines a theory about when the sum of two natural numbers (generated from 0 and the successor function s) is even.

$$
\begin{array}{l}
even(0) \\
odd(X) \to even(sX) \\
even(X) \to odd(sX) \\
sum(0, X, X) \\
sum(X, Y, Z) \to sum(sX, Y, sZ) \\
sum(X, Y, Z), even(Z) \to even_sum(X, Y)
\end{array}
$$

The set N_1 is not subterm local (nor order local under the subterm relation).

Consider, for example, a proof that the sum of one and one is even.

$$\dfrac{\dfrac{sum(0,s0,s0)}{sum(s0,s0,\underline{ss0})} \quad \dfrac{\dfrac{even(0)}{odd(s0)}}{even(\underline{ss0})}}{even_sum(s0,s0)}$$

Here we have underlined the problem: the proof requires the term $ss0$, which is not a subterm of either the terms appearing in the goal or in N_1. Moreover, there is no other way to prove $even_sum(s0,s0)$ and thus there does not exist any proof of this atom that only uses subterms.

Assuming that 0 and s are the only function symbols in the signature, it is, however, possible to come up with another formalization, N_2, with the same set of consequences, that is local. Hence locality is a syntactic property about sets of clauses as opposed to a semantic property about their models and entailment relations. In particular, N_2 is N_1 with the clauses defining $even_sum$ replaced by

$$even(X),\ even(Y) \rightarrow even_sum(X,Y)$$
$$odd(X),\ odd(Y) \rightarrow even_sum(X,Y)\,.$$

We may now prove the query $even_sum(s0,s0)$ using only the terms 0 and $s0$, both which appear as subterms in the query itself.

$$\dfrac{\dfrac{even(0)}{odd(s0)} \quad \dfrac{even(0)}{odd(s0)}}{even_sum(s0,s0)}$$

Moreover, it is possible to show that any ground atom (or, more generally, Horn clause), which is entailed by N_2, is provable only using subterms of terms appearing in it. Hence N_2 is subterm local and order local under the subterm ordering.

3. ORDERED RESOLUTION

We define the concept of saturation up to redundancy of a set of clauses by ordered resolution. The definitions in this section are with respect to (total) orderings on atoms rather than (partial) orderings on terms. By an *atom ordering* we mean any well-founded, (strict) *total* ordering on the ground atoms over the given signature. We call an atom ordering $\succ$ *compatible* with a term ordering $\succ'$ if $p(t_1,\ldots,t_k) \succ q(s_1,\ldots,s_n)$, whenever for any j with $1 \leq j \leq n$ there exists an i with $1 \leq i \leq k$ such that $t_i \succ' s_j$. Often we shall use the same symbol both for a term ordering and a compatible atom ordering. We will later discuss methods that can be used to generate compatible atom orderings. Note that if our signature of predicate symbols just consists of one element which, in addition, is unary, then total term orderings can be identified with compatible atom orderings. For finite signatures of predicate symbols, any total ordering on atoms that is compatible with a total (well-founded) term ordering is well-founded.

3.1 Clause Ordering

Given a term or atom ordering $\succ$, we may also use it to compare non-ground expressions by stipulating: $E_1 \succ E_2$ if $E_1\sigma \succ E_2\sigma$ for every ground substitution

σ. We will call an (either ground or non-ground) atom A [strictly] maximal with respect to a multiset of atoms Γ, if for any B in Γ we have $B \not\succ A$ [$B \not\succeq A$].

To extend an atom ordering $\succ$ to ground clauses 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 ground clauses C and C', we define $C \succ C'$ if and only if C $(\succ_{mul})_{mul}$ C' in the twofold multiset extension of $\succ$. This definition basically says that a clause is greater than another clause if the maximal atom of the former is greater than the maximal atom of the latter. In case they are identical, negative occurrences are considered greater than positive ones. If the maximal atom has identical occurrences in both clauses, non-maximal atoms are compared with respect to the atom ordering and polarity.

Clearly, if $\succ$ is a total and well-founded ordering on ground atoms, so is its extension to ground clauses. Given a ground clause C and a set of clauses N, by N_C we denote the set of ground instances D of clauses in 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 the compatible extension $\succ'$ of an atom ordering $\succ$ 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 based on [Bachmair and Ganzinger 1994a].

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 most general unifier of A and B such that $A\sigma$ is strictly maximal with respect to $\Gamma_1\sigma$,$\Delta_1\sigma$, and $A\sigma$ is maximal with respect to $\Gamma_2\sigma$,$\Delta_2\sigma$.

An inference by *ordered factoring* takes the form

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

where σ is the most general unifier of A and B such that $A\sigma$ is strictly maximal with respect to $\Gamma\sigma$, and $A\sigma$ is maximal with respect to $\Delta\sigma$.

We define a notion of redundancy that identifies clauses and inferences that are not needed for establishing refutational completeness of ordered resolution. It is based on the observation that ordered inferences are monotone (i.e., the conclusion is smaller than the maximal premise) and, hence, need only be applied to certain minimal clauses. A ground clause C (not necessarily an instance of a clause in N) is called *redundant in* N (with respect to $\succ$) if it is entailed by the instances of clauses in N that are smaller than C, i.e., if $N_C \models C$.[2]

[2]Note the difference between redundancy, i.e., $N_C \models C$, and bounded entailment, i.e., $N \models_{\preceq} C$. When using a compatible atom ordering, there are clauses, whose maximum term is the maximum term in C, which are disallowed in proofs of $N_C \models C$ but allowed in proving $N \models_{\preceq} C$; see the example in section 4.2. A less significant difference is that the former is based on the clause ordering and the latter on term orderings.

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 clauses in 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 of clauses in N that 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 under ordered resolution with respect to* $\succ$*)*, if any inference by ordered resolution or factoring from premises in N is redundant in N. As an inference is redundant whenever the conclusion is an instance of a clause in N, we can saturate N by systematically adding all conclusions of non-redundant inferences.

The previous definitions are based on orderings on *atoms*. Given a total *term* ordering $\succ$, we call N *saturated (up to redundancy under ordered resolution with respect to* $\succ$*)* if N is saturated with respect to some atom ordering that is compatible with $\succ$. If the term ordering $\succ$ is only partial, we call N *saturated (up to redundancy under ordered resolution with respect to* $\succ$*)* if N is saturated with respect to every total extension of $\succ$.

If N is saturated with respect to $\succ$, then N is inconsistent if and only if N contains the empty clause [Bachmair and Ganzinger 1990]. Moreover, in saturated sets N, any inference from premises both in N is redundant and redundancy of clauses and inferences is preserved by adding clauses and deleting redundant clauses that arise during theorem proving. This accounts for the following theorem, which states that entailment proofs with respect to a saturated N possess an N*-linear* form.

THEOREM 3.1 ([BACHMAIR AND GANZINGER 1994A]) Let N be a saturated set of clauses and C a clause. C follows from N iff there is a proof, by ordered resolution, of the empty clause from a (finite) set of ground instances in clauses in $N \cup \neg C$ such that for each inference at least one premise is not an instance of a clause in N.

4. SATURATION AND LOCALITY

4.1 Saturation implies Locality

We start by showing that saturation implies locality if sufficiently many terms are comparable under the ordering and if the term and atom ordering are compatible on theory clauses. Let $\succ$ and $\succ'$ be a term and an atom ordering, respectively. A set of clauses N is called *reductive* (with respect to $\succ$ and $\succ'$) if for each ground instance C of a clause in N, if A is the maximal atom in C under $\succ'$ then for each atom B in C and for each term t in B there exists a term s in A such that $s \succeq t$ in the term ordering. Reductivity of N is a weaker form of compatibility of $\succ'$ with $\succ$ and at the same time requires that sufficiently many ground terms in instances of clauses in N are comparable.

THEOREM 4.1 Let $\succ'$ be an atom ordering compatible with a term ordering $\succ$. Moreover, let N be saturated with respect to $\succ'$ and reductive with respect to $\succ$ and $\succ'$, 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. Conversely, assume $N \models C$ and that N is saturated and reductive with respect to the term and atom ordering $\succ$ and $\succ'$. By theorem 3.1 there exists an N-linear proof of the empty clause from a finite set of ground instances of clauses in $N \cup \neg C$. By induction on the depth of clauses in the proof we show that all terms are bounded (i.e., smaller than or equal under $\succeq$) by terms in C.

For the base case (depth one), consider any instance $\Gamma \to A, \Delta$ of a clause in N that is a leaf in the proof and is the first premise in an inference by ordered resolution. By N-linearity, the second premise must be a unit clause $A \to$ from $\neg C$. As the inference is ordered, A is the maximal atom with respect to Γ or Δ. Since N is reductive, any term in Γ, Δ is, therefore, bounded under $\succ$ by a term in A. Alternatively $A, \Gamma \to \Delta$ is an instance of a clause in N that is the second premise in an inference by ordered resolution of depth one, then the first premise must be a unit clause $\to A$ from $\neg C$. Again, any term in Γ, Δ is bounded under $\succ$ by a term in A. Note that an inference by ordered factoring at depth one is impossible since all clauses in $\neg C$ are unit. Hence the premises and the conclusions of all inferences at depth one satisfy the required property.

Now consider an ordered resolution inference of depth greater than one 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 inference does not introduce any new terms, the terms in the conclusion are again bounded by the terms in C. If only one, D say, of the two premises is an inner node, and the other, D' is an instance of a clause in 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 terms, and by the induction hypothesis all the old terms are bounded by terms in C. □

Observe that if the term ordering is total, compatibility of the atom ordering with the term ordering implies that any set of clauses is reductive. Therefore we obtain:

COROLLARY 4.2 Let $\succ$ be a total term ordering. Moreover, let N be saturated with respect to $\succ$ and let C be a ground clause. Then $N \models C$ if and only if $N \models_{\preceq} C$.

We show in section 5 that the complexity of entailment problems depends on how many smaller terms each ground term has under $\succ$. For most total orderings, a ground term has at least exponentially many smaller terms and therefore the direct use of these orderings would lead to poor complexity results. We get better complexity bounds by working instead with partial orders. If N is not reductive, and hence Theorem 4.1 is not applicable, we may still infer locality if the theory is saturated with respect to all total extensions.

THEOREM 4.3 Let N be a set of clauses and let $\succ$ be a (possibly partial) term ordering. If N is saturated with respect to $\succ$, then N is local with respect to $\succ$.

PROOF. Let N be saturated with respect to $\succ$. Then, by definition, N is saturated with respect to each total extension $\succ'$ of $\succ$. 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 a total extension of $\succ$ satisfying $s \succ' t$, 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 than or equal to some term in C with respect to $\succ$ become strictly larger with respect to $\succ'$ than any term in C. As N is saturated with respect to $\succ'$, we may apply corollary 4.2 and infer that $N \models_{\preceq'} C$ and by the definition of $\succ'$ we have $N \models_{\preceq} C$. □

4.2 Hyper-Resolution and Locality for Horn Clauses

The converse of the previous theorem is not true in general; however it does hold in restricted cases. Let us briefly give an example of a local theory that is not saturated. Suppose N is the following set of clauses.

$$\begin{aligned} p(f(x)) &\rightarrow q(x) \\ p'(f(x)) &\rightarrow q'(x) \\ p'(x) &\rightarrow p(x) \\ p(x) &\rightarrow p'(x) \end{aligned}$$

N is local with respect to the subterm ordering, but whatever ordering we choose on atoms, N will not be saturated. Suppose that a is a constant and that, for instance, $p'(f(a)) \succ p(f(a))$. Consider the inference

$$\frac{p(f(a)) \rightarrow p'(f(a)) \qquad p'(f(a)) \rightarrow q'(a)}{p(f(a)) \rightarrow q'(a)}$$

from ground instances of the second and fourth clause. The conclusion of this inference cannot be obtained by any other proof that does not involve the maximal atom $p'(f(a))$ of the inference. Therefore the inference is not redundant. This illustrates one of the main differences between locality and saturation: a local clause set might not be saturated when there are inferences in which the maximal term of the conclusion is not strictly smaller than the maximal term of the premises.

In order to analyze this difference in more detail, and to establish conditions under which the converse does hold, we change our setting slightly: we extend resolution from the binary case to hyper-inferences with selection and we restrict our attention to Horn clauses. Moreover, to simplify the technical treatment we will only consider the ground case (but allow clause sets to be infinite) as the lifting to the non-ground case of resolution inference systems poses no theoretical problems. *So, throughout this section we assume that all clauses are Horn and ground.*

Hyper-inferences are, in addition to a total atom ordering $\succ$, also based on a selection function for negative literals. 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.

Hyper-resolution based on $\succ$ and S is the inference rule:

$$\frac{\Gamma_1 \to A_1 \qquad \ldots \qquad \Gamma_n \to A_n \qquad A_1, \ldots, A_n, \Gamma \to \Delta}{\Gamma_1, \ldots, \Gamma_n, \Gamma \to \Delta}$$

where (i) for each i, A_i is strictly maximal with respect to Γ_i, (ii) no literal is selected in any of the Γ_i, and (iii) $n \geq 1$ and the selected atoms in the last premise are exactly the A_i, or else $n = 1$, nothing is selected in the last premise, and A_1 is maximal with respect to Γ, Δ. Observe that the restriction (ii) disallows clauses with selected literals to occur as *positive premises*, $\Gamma_i \to A_i$, of the inference.

The notion of redundancy for hyper-inferences is the same as in the binary case: an inference of the above form is redundant in a set of clauses N either if one of the premises is redundant in N, or else if the conclusion is entailed by clauses in N that are smaller with respect to $\succ$ than the *negative premise*, $A_1, \ldots A_n, \Gamma \to \Delta$, of the inference. With this, the notions of saturation are defined as before for the case of binary ordered resolution. And as before we say that N is saturated with respect to a term ordering $\succ$ if for each total extension $\succ'$ of $\succ$ there exists an atom ordering $\succ'$ that is compatible with $\succ'$ and a selection function S such that N is saturated with respect to $\succ'$ and S.

Hyper-resolution is refutationally complete for any atom ordering $\succ$ and selection function S. That is, a set of clauses N that is saturated up to redundancy is inconsistent if and only if it contains the empty clause [Bachmair and Ganzinger 1997].

To compare saturation by hyper-resolution with order locality, we shall use particular kinds of selection functions. Let $\succ$ be a term ordering. S is said to be *based on* $\succ$ if, for any clause C, $S(C)$ is the set of all negative atoms in C that contain a term that is maximal in C with respect to $\succ$. If $\succ$ is an atom ordering compatible with the term ordering $\succ$, we speak of *hyper-resolution based on* $\succ$ when we assume that the selection function is also based on $\succ$.

When selecting with respect to an ordering, one selects all negative literals that contain a maximal term. In that case, the atom ordering, when compatible with the term ordering, has no impact on the ordering constraints for the inference. In fact, either the antecedent of a clause does not contain a maximal term, in that case the succedent is strictly maximal with respect to any compatible atom ordering, or else some negative atom is selected, in which case there are no further ordering restrictions placed on that clause in any inference.[3]

Suppose now that $\succ$ is an atom ordering that is compatible with a term ordering, also denoted by $\succ$. One may distinguish two types of hyper-inferences. The first type, called *plateau inferences*, are those where in the succedent of the negative premise there exists an occurrence of a maximal term. For example, taking $\succ$ to be an ordering in which $f(a) \succ a \succ b \succ c$, the inference

$$\frac{p(a) \to p(f(a)) \qquad p(b) \to q(f(a), b) \qquad o(a), p(f(a)), q(f(a), b) \to r(f(a), f(a))}{o(a), p(a), p(b) \to r(f(a), f(a))}$$

is a plateau inference. The (unique) maximal term in the negative premise is $f(a)$ and occurs both positively and negatively in the negative premise.

[3] Atom orderings are still useful for obtaining a more fine-grained notion of redundancy for clauses and inferences.

N is called *plateau-saturated* if any plateau inference from premises in N is redundant. A clause in which the antecedent does not contain a maximal term of the clause and that is either in N, or else can be derived from N by a sequence of plateau inferences, is called a *slope* (in N). If N is a set of clauses in which positive unit clauses do not contain any terms, then any direct proof of a slope in N is a tree in which the non-leaf nodes are precisely the ones that are labeled by atoms that contain a maximal term among the terms occurring in the proof.

Inferences that are not of plateau type are of the second type, called *peak inferences*. For example

$$\frac{p(a) \to p(f(a)) \qquad p(a) \to q(f(a), a) \qquad o(a), p(f(a)), q(f(a), a) \to r(a, b)}{o(a), p(a), p(a) \to r(a, b)}$$

is a peak inference with respect to the ordering given before. Peak inferences generally enjoy the property that if t is a term in the conclusion of the inference, then there is a term in the negative premise that is strictly larger in $\succ$ than t. Note that no positive premises, in either a plateau or a peak inference, can have a negative occurrence of a maximal term. Otherwise, the respective literal would be selected and the clause would not be admitted as a positive premise in the inference.

We say that N is *peak-saturated* (with respect to $\succ$) if any peak inference with a negative premise in N and positive premises that are slopes in N is redundant. Remember that in general there are more slopes than just those clauses in N in which the antecedent does not contain a maximal term. Slope clauses also occur as conclusions of plateau inferences over N. In the non-ground case, the process of enumerating all consequences that might have slopes among their ground instances will usually be infinite, even for finite sets of clauses.

The various notions of saturation will again also be used when just given a term ordering $\succ$. As before, it is then assumed that the respective saturation property holds with respect to each member of some family of atom orderings containing, for each total extension $\succ'$ of $\succ$, at least one ordering compatible with $\succ'$. The set of clauses given at the beginning of this section is trivially peak-saturated with respect to $\succ$ as there are no peak inferences (between any of the ground instances of the clauses) with respect to any total extension of $\succ$.

Note that the definitions of peak and plateau inferences given above are well-defined also for partial term orderings. But saturation is again defined with respect to total extensions of partial orderings so that peaks and/or plateaus in which there is more than one maximal term are not relevant for our present investigations.

THEOREM 4.4 Let N be a set of clauses and let $\succ$ be a term ordering. If N is peak-saturated with respect to $\succ$, then N is local with respect to $\succ$.

PROOF. We have to show that if C is a clause such that $N \models C$ then $N \models_{\preceq} C$. We may assume that N does not contain any positive unit clauses containing terms. (Otherwise add a propositional constant p, a positive unit clause p, and replace every positive unit clause $q(\ldots, t, \ldots)$ by $p \to q(\ldots, t, \ldots)$. This transformation does not affect any saturation or locality properties.) Similarly, we may assume that C does not have an empty succedent, that is, C is of the form $A_1, \ldots, A_k \to A_0$. As in the proof of the theorem 4.3, we first choose a total term ordering $\succ'$ extending

$\succ$ that satisfies $t \succ' s$, for any ground term s in C and any ground term t for which there is no term u in C such that $u \succeq t$. According to the assumptions of the theorem, there exists an atom ordering (also denoted $\succ'$) compatible with $\succ'$ with respect to which all peak inferences are redundant.

Let Π be a minimal (with respect to the multiset of clauses which appear in Π) direct proof of C from N. We show that the maximal term t occurring in Π is smaller than or equal to (in $\succeq'$) the maximal term s of C. Suppose, to the contrary, that $t \succ' s$. Then Π is of the form $\Pi''(\Pi'(\Pi_1, \ldots, \Pi_n), A_1, \ldots, A_k)$, where Π' is a proof in N of a clause $\Gamma \to B$ such that (i) Π' contains at least one occurrence of t, (ii) every inner atom in Π' contains t, and (iii) the root B of Π', as well as any leaf in Π', does not contain t. (Note that we have assumed that positive unit clauses in N do not contain terms at all. Also, there may be additional occurrences of t in Π outside Π', but these are not relevant for the subsequent reasoning.) We now show that we can replace Π' by a smaller proof of $\Gamma \to B$, which contradicts the minimality of Π.

By the assumptions about Π', the end clause D of Π' is of the form

$$B_1, \ldots, B_m, B_{m+1}, \ldots, B_M \to B,$$

where $m \geq 1$, the B_i, $1 \leq i \leq m$, contain t, and the remaining $M - m \geq 0$ atoms B_j are in Γ and do not contain t.[4] B does not contain t either. Moreover, the direct subproofs Π'_i of Π' for the B_i, with $1 \leq i \leq m$, consist entirely of plateau inferences, hence are proofs of slopes $C_i = B_{i1}, \ldots, B_{in_i} \to B_i$ in N. (In fact, since the B_{ij} are leaves in Π', they do not contain the maximal term t.) D cannot be redundant as this would contradict the minimality of Π; replacing D by a proof from the smaller clauses that imply D would yield a smaller proof. Let us now consider the special case where none of the C_i is redundant in N. Then there exists a peak inference

$$\frac{B_{11}, \ldots, B_{1n_1} \to B_1 \quad \ldots \quad B_{m1}, \ldots, B_{mn_m} \to B_m \quad D}{B_{11}, \ldots, B_{1n_1}, \ldots, B_{m1}, \ldots, B_{mn_m}, B_{m+1}, \ldots, B_M \to B}$$

over N which, according to the assumption of the theorem, is redundant in N with respect to $\succ'$. The conclusion D' of the inference coincides with $\Gamma \to B$, the clause that is proved by the subproof Π'. As the premises are not redundant, D' follows from clauses in N that are smaller than D. Replacing Π', which contains D, by the proof of D' with all clauses smaller than D yields a smaller proof than Π of C, which is a contradiction.

More generally, some of the C_i might be redundant. In such a case there exists a direct proof Π^i of C_i from nonredundant clauses all smaller than C_i in $\succ'$. If C_i is not redundant we may define Π^i as the proof just consisting of C_i itself. In any case, the end clauses C'_i of the proofs Π^i are non-redundant and of the form $\Delta_i \to B_i$, with Δ_i not containing t. Any other clause appearing in Π^i is smaller than C_i and, hence, smaller than D. As the C'_i are slopes and non-redundant we may repeat the above reasoning, this time with regard to the redundancy of the

[4] Recall that clauses are multisets of literals, so that the order among the B_i, with $i \geq 1$, does not matter.

peak inference

$$\frac{\Delta_1 \to B_1 \qquad \ldots \qquad \Delta_m \to B_m \qquad D}{\Delta_1, \ldots, \Delta_m, B_{m+1}, \ldots, B_M \to B}\ .$$

That is, there exists a proof of $\Delta_1, \ldots, \Delta_m, B_{m+1}, \ldots, B_M \to B$ from clauses smaller than D. From this proof and from the (possibly empy) direct subproofs of the Π^i we are able to construct an alternative proof of D' from clauses all smaller that D. This proof is, therefore, smaller than Π', again contradicting the minimality of Π. □

As the example given at the beginning of this section is peak-saturated with respect to $\dot{>}$ it is, as claimed above, subterm local.

THEOREM 4.5 Let N be a set of clauses and let $\succ$ be a term ordering. If N is local with respect to $\succ$, then for every total extension of $\succ$ and every atom ordering $\succ'$ compatible with the extension, N is peak-saturated with respect to $\succ'$.

PROOF. Let $\succ'$ be an atom ordering, compatible with a total extension $\succ'$ of $\succ$, and consider any peak inference with negative premise D, conclusion C and positive premises $C_i = \Gamma_i \to A_i$, $1 \leq i \leq n$, which are slopes in N. Then the A_i are precisely those atoms in D that contain the maximal term (with respect to $\succ'$) t of the clause. Moreover, by the definition of peaks, t does not occur in the conclusion C of the inference, so that C only contains terms smaller in $\succ'$ than t. As N is local and C a consequence of N, C is also entailed by clauses in N that contain terms bounded by C with respect to $\succ$. All these clauses are smaller than D with respect to $\succ'$. Hence the inference is redundant in N. □

These theorems show that order locality and peak saturation are equivalent concepts for Horn clauses. We also observe that the way in which term orderings are extended to atoms does not matter: If there are two families of atom orderings, each containing compatible orderings for every total extension of the given term ordering, then if N is peak-saturated with respect to one of the families it is also peak-saturated with respect to the other.

COROLLARY 4.6 Let $\succ$ be a term ordering and suppose that N is plateau-saturated with respect to $\succ$. Then N is local with respect to $\succ$ if and only if N is saturated with respect to $\succ$.

Where term orderings and atom orderings coincide, locality and saturation coincide too, as exemplified by the following theorem.

THEOREM 4.7 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 term ordering. If N is local with respect to $\succ$, then N is saturated with respect to $\succ$.

PROOF. Let p be the unary predicate symbol, and let $\succ'$ be a total extension of $\succ$. If C is the negative premise of a plateau inference, then both the antecedent and the succedent contain the maximal term of C. Therefore C is of the form $\Gamma, p(t) \to p(t)$, that is, is a tautology. Hence, the inference is redundant. With this, the theorem follows from corollary 4.6 □

The preceding theorem also holds for binary resolution without selection. Factoring inferences may not be redundant if N also contains non-Horn clauses. One may generalize this 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 4.7 concerns the signature.

After having seen that more sophisticated inference systems and notions of saturation precisely characterize the concept order locality, for the remainder of this paper, in order to avoid uninteresting technical complications, we shall return to binary ordered resolution (without selection) as our main method for showing, or achieving, order locality. That is, the term "saturated" will, unless stated otherwise, always mean "saturated by ordered (binary) resolution (without selection)". We will also assume that the clause sets N under consideration are finite, and clauses may, again, contain variables.

5. COMPLEXITY BOUNDS FOR ENTAILMENT PROBLEMS

Using order locality we obtain new methods for establishing complexity bounds for ground entailment problems. 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.

5.1 Order-dependent Complexity Bounds

If a clause set is order local with respect to an ordering where ground terms have only finitely many smaller terms 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. Let the size of a clause be the sum of the sizes (the number of nodes in their tree representation) of the terms in the clause. We say that a term ordering $\succ$ is of *complexity* f, g, whenever for each clause of size n there exist $O(f(n))$ terms that are smaller or equal (under $\succeq$) to a term in the clause, and which may be enumerated in time $g(n)$. In the Horn case we have:

THEOREM 5.1 Let N be a set of Horn clauses that is local with respect to a term ordering $\succ$ of complexity f, g. 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. Furthermore, the constant k depends only on N.

PROOF. As N is local with respect to $\succ$, $N \models C$ iff $N \models_{\preceq} C$. Furthermore, this is equivalent to $N^C \models C$, where N^C is the set of ground clauses containing the ground instances of clauses in N whose terms are smaller than or equal to some term in C. We can construct N^C first by enumerating all terms bounded by terms in C and after using these to instantiate clauses in N. 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

[McAllester 1993].[5] By lemma 2.1, 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 5.2 Let N be a set of full clauses that is local with respect to a term ordering $\succ$ of complexity f, g. For C a ground clause, the disentailment problem $N \not\models C$ is nondeterministically decidable in time $O(f(m)^k + g(m))$, where, as before, m is size of C, and k is a constant depending on N.

Therefore, our saturation-based locality criteria given in the section 4, together with a complexity analysis of the underlying ordering, provide upper bounds for the complexity of entailment problems. We will later provide examples of these results. Note that McAllester's tractability result is a special case of ours for the Horn clause case and the subterm ordering: any term (and hence any clause) has only linearly many subterms which can be directly enumerated and hence locality implies that the ground Horn entailment problem is polynomial-time decidable. For full clauses, locality with respect to a term ordering of polynomial complexity (for f and g) means that ground entailment is in co-NP.

5.2 General Decidability Results

We consider here how saturation can be used to obtain general decidability results even when the underlying ordering does not have any finite complexity bound, that is, when there exist terms for which there are infinitely many smaller terms.

For Horn clause sets, the satisfiability and entailment problems are undecidable. This remains the case even when sets are saturated, but the underlying ordering $\succ$ does not have a finite complexity. To see this[6], suppose N is an arbitrary set of Horn clauses and let N' consist of the set of clauses $q(a), \Gamma \to \Delta$ such that $\Gamma \to \Delta$ is in N, with q a new predicate symbol and a a new constant not occurring in N. Choose an ordering such that a is the maximal term. Then N' is trivially saturated as no ordered inferences are possible from N' in any compatible atom ordering. It is undecidable if $N' \models \neg q(a)$, 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 5.3 Let N be saturated with respect to an atom ordering $\succ$. 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 $N \cup \neg C$ produces the empty clause. We show that with the required variable restriction, $N \cup \neg C$ can be finitely

[5] In particular, a clause C is said to be of *order* j if there are j subterms $e_1, \ldots, e_j$ of C, containing all of C's variables. The upper bound k is the maximum order of any clause in N.

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

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 clauses by factoring, i.e., by deleting multiple occurrences of atoms in the antecedent or succedent. 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 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 that is not an instance of a clause in 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 contradicts the well-foundedness of $\succ$. □

5.3 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, which can be enumerated in polynomial time. A simple example of such a *polynomial ordering* is the subterm ordering $\dot{>}$. A refinement, based on a precedence π on function symbols, is the following: $s \dot{>}_\pi t$ iff (i) $s \dot{>} t$, or else (ii) $s = f(\ldots)$, $t = g(t_1, \ldots, t_n)$, $s \dot{>} t_i$, 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. (The signature is assumed to be finite, and its size a constant.) 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 $s \dot{>} t$ 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) [Knuth and Bendix 1970]. 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 [Hofbauer and Lautemann 1989] 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 smaller 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 exponentially bounded orderings can be obtained by polynomial interpretations. In [Hofbauer and Lautemann 1989] 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.

5.4 Generalization: Saturation Modulo an Equivalence

Both the theory of ordered resolution and our complexity analysis can be straightforwardly extended to resolution modulo an equivalence $\sim$ on terms and atoms. The extension requires that the respective orderings are 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. To obtain complexity results in the spirit of section 5.1, the equivalence between ground terms should be decidable in polynomial time. Note that the complexity results are then independent of the cardinality of equivalence classes. The proof of theorems 5.1 and 5.2 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.

6. THE SCOPE OF ORDER LOCALITY

We have seen that the ground entailment problem for an order local clause set belongs to a complexity class characterized by the term ordering used. Here we prove a form of the converse: when a ground entailment problem — indeed, any decision problem — is decidable in polynomial (exponential, etc.) time, then there is a Horn clause set encoding the problem that is local under a polynomially (exponentially, etc.) bounded ordering. Hence the scope of order locality is quite wide and it is adequate to give an alternative formalization of these complexity classes.

Our proof is based on an encoding of Turing machines. One might hope to avoid a low level encoding and establish a stronger result such as if N is a clause set whose ground entailment problem is decidable in (say) polynomial time, then there is a polynomially bounded ordering under which N is saturated. Unfortunately, this is false. As our *even_sum* example demonstrates, order locality is a syntactic property

about sets of clauses. That is, it is a function of the presentation, and a given presentation may require proofs involving exponentially many different terms (in the size of the query), even when ground entailment is polynomial-time decidable by other means. Hence, to show that problems in a complexity class can be formalized by order local clauses sets with appropriate orderings, we encode Turing machines using Horn theories that are saturated with respect to orderings that reflect the successor relation between machine configurations.

Our construction is direct and conceptually simpler than similar results for polynomial time queries proven in [Givan and McAllester 1992] that are based on transforming the problem to one involving Datalog programs. Moreover, the encoding we use is not possible in McAllester's setting as it requires problem-specific orderings.

6.1 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 $\natural$, $\delta : Q \times \Delta \to \mathcal{P}(Q \times \Delta \times \{L, R\})$ 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* 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 Horn clause 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

$$c(w_m w_{m-1} \ldots w_1 \epsilon, q, a, v_1 v_2 \ldots v_n \epsilon)\,.$$

We previously listed three kinds of Turing machine transitions. Associated with each kind is a set of rules corresponding 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* $L(M)$.

6.2 Order Locality

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 $\rightsquigarrow^+$, that is, define $c(v, q, a, w) \succ_M c(v', q', a', w')$ iff $v(q, a)w \rightsquigarrow^+ v'(q', a')w'$.

LEMMA 6.1 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 succedent of each rule instance is always greater than the antecedent. As a result, there can be no ordered resolution steps, because we cannot satisfy the ordering requirements. Hence R_M is saturated with respect to every total extension $\succ'$ of $\succ_M$. So R_M is local with respect to $\succ_M$. □

When M is deterministic, the length of the unique computation history starting from the initial configuration of a string x is the number of terms smaller than x under $\succ_M$. Hence, from the previous lemma we have that:

THEOREM 6.2 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 that recognizes $L(M)$.

Hence any polynomial (exponential, etc.) time decidable problem can be formalized as a theory local under a polynomially (exponentially, etc.) bounded ordering.

7. EXPERIENCE WITH THE SATURATE SYSTEM

7.1 Background

Theorem provers based on ordered resolution with redundancy elimination, such as our Saturate system [Ganzinger et al. 1994], can be employed as procedures for detecting if a clause set is saturated and for saturating clause sets. We were able to take the Saturate system and, with very few additions, saturate a number of non-trivial examples. This requires that the prover supports appropriate classes of orderings and that it provides strong enough techniques to detect redundancy (ours are based on [Nivela and Nieuwenhuis 1993]). Among the two alternatives for saturation that we have discussed in the paper, Saturate implements saturation by ordered resolution with selection where hyper-inferences are simulated by sequences of binary inferences. However, the Saturate system does not distinguish plateaux from peaks as this is inessential for general first-order theorem proving.

From the results in section 4.2 one might, alternatively, consider implementing a peak saturation procedure (for Horn clauses) in which plateau inferences leading to slopes are handled specially. Non-redundant slopes have to be enumerated in order to check as to whether peak inferences are redundant. However, in the final set of clauses, only those clauses added by peak inferences are needed for order local proofs, whereas any of the consequences added from non-redundant plateau inferences can be ignored. Since the number and form of clauses in an order local theory affects the complexity of the entailment problem for the theory (recall the proof of theorem 5.1), being able to ignore plateau consequences could make a considerable difference in practice.

We now describe some of the techniques we use in the Saturate system. There are two basic problems to be dealt with: (i) how to check whether a set of clauses is saturated, and, in case the set cannot shown to be saturated, (ii) how to add non-redundant consequences so that the saturation process often terminates. The first problem is a subproblem of the second. We begin with the second problem and introduce additional notation in order to prove a decidability result about checking saturation.

Adding Non-Redundant Consequences. If a clause set is not saturated, consequences of non-redundant inferences can be systematically added and the extended clause set can again be checked for saturation. Hence, a key issue is how to appropriately represent the consequences of non-redundant inferences.

Inferences by ordered resolution must obey ordering constraints. Given an ordering constraint γ, it may be the case that γ is satisfied for some substitutions but violated for others. Therefore, if C is the conclusion of an ordered inference with constraint γ, only the instances of C that satisfy γ have to be checked for redundancy and, if they are not redundant, added to the clause set. This observation motivates working with *constrained clauses* $C \parallel \gamma$, which consist of a clause C and an *ordering constraint* γ. The ordering constraint is a propositional formula over inequations $A > B$ and $A \geq B$, where A and B are (possibly nonground) atoms. Equality constraints $A = B$ can also be admitted as abbreviations for conjunctions $(A \geq B) \wedge (B \geq A)$. Inferences are now computed over constrained clauses in a way such that the ordering constraint of the inference is formalized as part of the clause and inherited through subsequent inferences. The constraint variant of

binary ordered resolution is this inference on constrained clauses:

$$\frac{\Gamma_1 \to A, \Delta_1 \parallel \gamma_1 \qquad B, \Gamma_2 \to \Delta_2 \parallel \gamma_2}{\Gamma_1, \Gamma_2 \to \Delta_1, \Delta_2 \parallel \gamma_1 \wedge \gamma_2 \wedge A = B \wedge \gamma}$$

where γ is the *ordering constraint of the inference*

$$\bigwedge_{A' \in \Gamma_1, \Delta_1} A > A' \wedge \bigwedge_{A' \in \Gamma_2} B \geq A' \wedge \bigwedge_{A' \in \Delta_2} B > A' .$$

$A = B$ is called the *equality constraint* of the inference. The constraints are interpreted over the Herbrand domain of ground atoms, with $>$ the chosen atom ordering. A *solution* of the constraint is a substitution of ground terms for the variables that makes the constraint valid. Hence, a constrained clause *represents* the set of ground clauses that are instances of the clause part by solutions of the constraint part. In particular, a constrained clause in which the constraint part is unsolvable represents the empty set of clauses and, hence, can be deleted. Since a (non-ground) clause or inference is redundant when all its ground instances are, the notion is also well-defined for clauses with constraints, assuming the concept of ground instances as just defined.

An equality constraint such as $A = B$ is either unsolvable or has a most general solution σ that is the most general unifier of A and B. In the context of resolution it makes no essential difference as to whether one carries equality constraints along in constraint parts or eagerly computes their most general solution and applies it to both the clause and the constraint.

The constraint variant of hyper-resolution can be formalized similarly. There, the ordering constraints are used to identify maximal terms in clauses and are boolean combinations of inequalities between *terms*, to be interpreted over the given term ordering. As in our variant of hyper-resolution, as selection is based on the term ordering, the side-conditions of the inference, including those involving selection, can completely be expressed by ordering constraints between terms. As an example, consider the inference

$$\frac{q(x') \to p(x', b) \qquad p(y', a) \qquad p(x, y), p(y, z), r(z) \to p(x, z)}{q(x), r(a) \to p(x, a)} .$$

For the inference to satisfy the restrictions about selection, the maximal term must occur in both $p(x, y)$ and $p(y, z)$ but not in $r(z)$. Also, for the first positive premise to be eligible, its maximal term must not occur negatively, hence $b > x'$. Altogether we obtain the ordering constraint

$$(b > x') \wedge (y \geq x) \wedge (y > z)$$

which, when applying the most general solution of the equality constraint

$$(x' = x) \wedge (b = y) \wedge (y' = y) \wedge (a = z)$$

of the inference, simplifies to $(b > x) \wedge (b > a)$.

Checking Saturation. Given a term ordering $\succ$, a clause set is saturated if for any total extension $\succ'$ of the term ordering there exists a compatible atom ordering such that all inferences are redundant. Hence, the atom ordering may depend on the various possible extensions $\succ'$ of $\succ$. In this general setting, the problem of whether

a clause set is saturated is undecidable. However one may relax the problem by looking at specific methods of constructing compatible atom orderings. A method that turns out to be useful in practice is one we call *lexicographic atom orderings*: Let $\succ$ be a total term ordering, 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 $p(t_1, \ldots, t_m) \succ_\pi q(s_1, \ldots, s_n)$ 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. The relation $\succ_\pi$ 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.

With lexicographic atom orderings, saturation becomes a decidable property for many term orderings, including the subterm ordering.[7]

THEOREM 7.1 Let Σ be a signature with infinitely many constant symbols, and let π be a precedence on the predicate symbols in Σ. Given a finite set of clauses over Σ, it is decidable whether the set is saturated under ordered (binary) resolution or under hyper-resolution with respect to all $\succ'_\pi$, with $\succ'$ a total extension of $\cdot>$.

PROOF. We have to analyze the respective (hyper-)resolution inferences. Suppose that the inference is of the form

$$\frac{D_1 \quad \ldots \quad D_k \quad D}{C}$$

and let P denote the set of premises. Moreover, suppose that η and ω, respectively, are the equality and the ordering constraint of the inference. Without loss of generality we may assume that ω is a conjunction of strict inequalities of the form $s > t$. (If it is not, we may replace non-strict inequalities $s \geq t$ in ω by a disjunction $(s > t) \vee (s = t)$ and then transform the resulting constraint into disjunctive normal form. Each of the disjuncts will be of the form $\eta_j \wedge \omega_j$, with η_j an equality constraint and ω_j a conjunction of strict inequalities. Consider each of the cases seperately with, respectively, $\eta \wedge \eta_j$ as the equality and ω_j as the ordering constraint.) Suppose that ρ is the most general solution of the equality constraint η. (In other words, ρ is the most general unifier associated with the inference, and we may assume that C is of the form $C'\rho$, with all its variables also appearing in $P\rho$.) Suppose that the clauses in $P\rho$ contain n variables $x_1, \ldots, x_n$. Then, any ground instance of the

[7] This fact was observed, and communicated to us, by David McAllester.

inference (by σ) is of the form

$$\frac{D_1\rho\sigma \quad \ldots \quad D_k\rho\sigma \quad D\rho\sigma}{C\sigma}$$

where σ is a ground substitution for the x_i. Given σ and $\succ$, a total extension of $\dot{>}$, we can effectively check whether the ground instance of the inference by σ satisfies the constraint $\omega\rho$, that is, whether the correspondingly instantiated constraint $\omega\rho\sigma$ is valid under the term ordering $\succ$. We call the pair σ and $\succ$ essential if the constraint is satisfied. For essential pairs, the instance of the inference is admitted by the inference system and has to be checked for redundancy.

Let $\mathcal{C}$ be some fixed set of n constants from Σ that do not occur in N or P. (Note that we have assumed that the signature contains sufficiently many constants.) By a skolemizing substitution τ we mean any substitution that maps the x_i to constants in $\mathcal{C}$. Finally, if M is a set of ground clauses, let N^M denote those instances of clauses in N whose terms are smaller than or equal to terms in clauses in M under the subterm ordering $\dot{>}$.

We have to show that it is decidable whether for every essential σ and $\succ$ either one of the positive premises is redundant, that is $N^{\succ\pi}_{D_i\rho\sigma} \models D_i\rho\sigma$, or else the conclusion follows from instances smaller than the negative premise, that is, $N^{\succ\pi}_{D\rho\sigma} \models C\sigma$. (Note that the latter case also subsumes the case in which the negative premise is redundant. In fact, the positive premises in a (hyper-) inference are smaller than the negative premise, so if the latter is redundant, the conclusion of the inference is implied by clauses smaller than the negative premise.) To simplify matters technically, we disregard the possibility that a positive premise may be redundant and, as the extension is straightforward, merely demonstrate how to decide $N^{\succ\pi}_{D\rho\sigma} \models C\sigma$.

The crucial observation is that this property is satisfied if and only if $N^{\succ\pi}_{D\rho\tau} \cap N^{P\rho\tau} \models C\tau$, for every essential τ and $\succ$, with τ a skolemizing substitution.

If, for some essential τ and $\succ$, the clause $C\tau$ can only be proved from instances of clauses in N in which there are terms that are not bounded by some term in $P\rho\tau$ in the subterm ordering, we may choose an extension $\succ'$ of $\dot{>}$, coinciding with $\succ$ on the terms in $P\rho\tau$, such that any term not in $P\rho\tau$ is larger than any term in $D\rho\tau$. Then, the ordering constraint $\omega\rho\tau$ is also valid under $\succ'$ (hence, τ and $\succ'$ are essential) but the instance $C\tau$ of C is not entailed by $N^{\succ'\pi}_{D\rho\tau}$.

Conversely, suppose that $N^{\succ\pi}_{D\rho\tau} \cap N^{P\rho\tau} \models C\tau$, for every essential $\succ$ and skolemizing substitution τ. If σ is an arbitrary ground substitution of the x_i and $\succ$ an extension of $\dot{>}$ such that σ and $\succ$ are essential, we may choose a particular skolemizing substitution τ' such that $x_i\tau' = x_j\tau'$ if and only if $x_i\sigma = x_j\sigma$, for any two indices i and j, where $1 \leq i, j \leq n$. Moreover, we may choose an ordering $\succ'$ extending $\dot{>}$ such that for every two terms $s\tau'$ and $t\tau'$ in $N^{P\rho\tau'}$, such that $s\sigma \succ t\sigma$, we have $s\tau' \succ' t\tau'$. (If $s\tau' = s'\tau'$ then s and s' only differ in their variable names. There may be a variable x in s such that at the corresponding position in s' there is the variable y such that $x\tau' = y\tau'$. But then $x\sigma = y\sigma$. Therefore, whenever $s\tau' \geq t\tau'$ we also have $s\sigma \geq t\sigma$. In other words, a total term ordering extending $\geq$ with the required properties exists.) Since ω is a conjunction of strict inequalities, and $\succ$ satisfies $\omega\rho\sigma$, the ordering $\succ'$ satisfies $\omega\rho\tau'$, that is, τ' and $\succ'$ are essential. We may now apply the assumption and infer that $C_1\tau', \ldots, C_k\tau' \models C\tau'$, for certain

$C_i\tau'$ in $N^{\succ'_\pi}_{D\rho\tau'} \cap N^{P\rho\tau'}$. As the constants introduced by τ' do not occur in $N^{P\rho}$ or C, we may consider them as variables and instantiate them by σ. In other words, $C_1\sigma, \ldots, C_k\sigma \models C\sigma$ and, by the definition of $\succ'$, the $C_i\sigma$ are in $N^{\succ^\pi}_{D\rho\sigma} \cap N^{P\rho\sigma}$.

The decidability result now follows from the fact that there are only finitely many skolemizing substitutions τ, and for each of them $N^{P\rho\tau}$ is a finite set of ground clauses. Hence there are only finitely many orderings $\succ$ on the terms in $N^{D\tau}$ for which one has to check that $N^{\succ^\pi}_{D\tau} \cap N^{D\tau} \models C\tau$. □

The theorem can also be proved for most other computable term orderings in which for each term there exist at most finitely many smaller terms, and for any computable function that maps total term orderings into compatible atom orderings. However, since the procedure sketched here is quite inefficient, the Saturate system uses an incomplete approximation. The theorem would be even more significant for peak saturation procedures in the spirit of section 4.2. There we have seen that the choice of atom orderings does not have an impact on whether a peak is redundant. This does not contradict the fact that (subterm) locality is undecidable, as one in general needs to derive — by lifted forms of plateau inferences — infinitely many non-ground clauses which have slopes among their ground instances that may occur as positive premises in (lifted forms of) peak inferences.

The theorem 7.1 also holds for clauses with ordering constraints. In the proof of the theorem, the structure of the constraint ω plays no rôle since one only needs to check whether the constraint is satisfied for certain ground substitutions. In practice, however, one wants to avoid the enumeration of skolemizing substitutions as much as possible and therefore needs to check the satisfiability of ordering constraints with variables. 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 algorithms in [Nieuwenhuis 1993].

7.2 Examples Checked by Saturate

The following examples report on 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.

Congruence Closure. Consider the congruence closure example from the introduction. One presentation of this theory, over a binary function symbol f, is the following Horn clause set.

$$\rightarrow\ x = x \tag{5}$$

$$x = y\ \rightarrow\ y = x \tag{6}$$

$$x = y,\ y = z\ \rightarrow\ x = z \tag{7}$$

$$x = y,\ u = v\ \rightarrow\ f(x,u) = f(y,v) \tag{8}$$

This presentation is subterm local, and McAllester's systems can verify this. When given to Saturate, saturation terminates with the above clauses and the additional clause

$$x = y,\ u = v,\ f(x,u) = z \ \rightarrow\ f(y,v) = z \tag{9}$$

as the final saturated set. This behavior was observed in several examples of subterm local Horn theories. While Saturate was able to saturate many of them in finite time, the final system usually contained additional clauses that are either not redundant or whose redundancy could not be shown using Saturate's approximative techniques. Therefore, the fact that saturation is weaker than locality (cf. Section 4.2) does show up in practice.

The following variation of the congruence closure example indicates one of the advantages of the saturation-based approach to locality: saturation may transform a non-local theory presentation into a local one. Consider

$$\rightarrow\ x = x \tag{10}$$
$$x = y\ \rightarrow\ y = x \tag{11}$$
$$x = y,\ y = z\ \rightarrow\ x = z \tag{12}$$
$$x = y\ \rightarrow\ f(x,z) = f(y,z) \tag{13}$$
$$x = y\ \rightarrow\ f(z,x) = f(z,y)\ . \tag{14}$$

This presentation 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.

McAllester's locality procedures would stop here and identify a counter-example to the locality of the clause set. In our case, Saturate successfully transforms this non-local theory to a local one by adding the following new clauses

$$x = y,\ f(y,z) = u\ \rightarrow\ f(x,z) = u \tag{15}$$
$$x = y,\ f(z,y) = u\ \rightarrow\ f(z,x) = u \tag{16}$$
$$x = y,\ u = v\ \rightarrow\ f(x,u) = f(y,v) \tag{17}$$
$$x = y,\ u = v,\ f(y,v) = z\ \rightarrow\ f(x,u) = z \tag{18}$$

and deleting the clauses (13) and (14). Using theorem 5.1 (applying the refined analysis mentioned in the footnote to that theorem) we can conclude that congruence closure is decidable in time $O(n^3)$, which is a good bound, although not optimal.

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.

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

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

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. This is in contrast to an $O(n^2)$ bound possible using more refined techniques based on dynamic programming.

Propositional Calculus. We first formalize a tractable (incomplete) fragment of propositional logic and then a full, complete, calculus. Our tractable fragment is a slight modification of [McAllester 1993].

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

This theory presentation 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 using the modified subterm ordering with the precedence $or >_\pi not >_\pi \mathsf{ff}$; hence ground entailment is decidable in polynomial time. In this ordering $or(not(p), q)$ is greater than $not(not(p))$.

Now consider the following presentation of full propositional logic (defined over the connectives *or* and *not*).

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

The presentation is trivially saturated with respect to the subterm ordering; thus we obtain co-NP (in fact, co-nondeterministic linear time) as an upper bound for the entailment (tautology) problem for propositional logic.

Orderings . The following defines the Horn theory of nonstrict partial orderings.

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

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

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

from which we obtain a cubic upper bound for the entailment problem. This is an automatic proof of theorem 3.4 in [Nebel and Bürckert 1995]. (The latter paper also shows that this theory is in some sense the largest fragment of Allen's interval logic that is computationally tractable.)

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

8. SATURATION OF PROPOSITIONAL SEQUENT CALCULUS

We now give an example of a more theoretical nature than those previously considered: we use saturation not only to bound the length of propositional sequent calculus proofs, but also to show that, for provability (as opposed to derivability), the cut-rule is eliminable. This example was not run on the Saturate system for reasons that will become clear below. Still, it illustrates the possibility of using techniques based on ordered resolution to carry out proof-theoretic arguments.

8.1 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. A sequent is built from the binary sequent symbol $\vdash$ and its antecedent and succedent are possibly empty multisets of formulas built from propositional variables, conjunction and negation. Then LK consists of the inference rules given in figure 1.

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

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

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

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

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

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

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

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

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

Fig. 1. The Horn Theory LK

8.2 A Horn Encoding of LK

We will formulate LK as a set of Horn clauses $\mathcal{L}K$ where atomic formulae correspond to sequents and Horn clauses correspond to rules. Hence an atom entailed by $\mathcal{L}K$ corresponds to a provable sequent of LK and an entailed clause corresponds to a derivable rule (i.e., the succedent is provable under the assumption that the atoms in the antecedent are non-logical axioms). Despite the fact that cut-elimination holds

for sequents (cf. theorem 8.3), the cut rule is not redundant when one considers rules. For example, derived rules corresponding to natural deduction elimination rules, e.g.,

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

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.

In $\mathcal{L}K$ the ground atoms are built from a family $\rhd_{k,n}$ of $(k+n)$-ary predicate symbols taking (representatives 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 $A_1, \ldots, A_k \vdash B_1, \ldots, B_n$. We admit only 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 encode the multisets by variadic predicates 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 multiset extensions of certain orderings on formulas. If multisets of formulas were represented as terms 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, and C for formulas, and metavariables Δ, Γ, Λ and Π for sets of formulas. The expression $\rhd(H, G)$ 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 $\mathcal{L}K$ ground clauses. It is used for a finite "non-ground" representation of certain infinite sets of $\mathcal{L}K$ ground atoms and clauses. The ground terms of $\mathcal{L}K$ are simply the propositional formulas. Since we are working modulo idempotence, there is no need anymore for any explicit representation of the contraction rules.

The following list then is a finite presentation of $\mathcal{L}K$ using the concept of non-ground atoms just described.

$$\rightarrow \rhd(A, A) \tag{19}$$

$$\rhd(\Gamma, \Delta) \rightarrow \rhd(\Gamma + \Lambda, \Delta + \Pi) \tag{20}$$

$$\rhd(\Gamma + A + B, \Delta) \leftrightarrow \rhd(\Gamma + (A \wedge B), \Delta) \tag{21}$$

$$\rhd(\Gamma, \Delta + A), \rhd(\Gamma, \Delta + B) \rightarrow \rhd(\Gamma, \Delta + (A \wedge B)) \tag{22}$$

$$\rhd(\Gamma, \Delta + (A \wedge B)) \rightarrow \rhd(\Gamma, \Delta + A) \tag{23}$$

$$\rhd(\Gamma, \Delta + (A \wedge B)) \;\rightarrow\; \rhd(\Gamma, \Delta + B) \tag{24}$$

$$\rhd(\Gamma + \neg A, \Delta) \;\leftrightarrow\; \rhd(\Gamma, \Delta + A) \tag{25}$$

$$\rhd(\Gamma, \Delta + \neg A) \;\leftrightarrow\; \rhd(\Gamma + A, \Delta) \tag{26}$$

$$\rhd(\Gamma, \Delta + A), \rhd(\Lambda + A, \Pi) \;\rightarrow\; \rhd(\Gamma + \Lambda, \Delta + \Pi) \tag{27}$$

THEOREM 8.1 $\mathcal{L}K$ is saturated with respect to the subterm ordering on propositional formulas.

PROOF. Let $\succ$ be a total extension of the subterm ordering on propositional formulas. We have to provide a compatible extension of $\succ$ to $\mathcal{L}K$ ground atoms and to show the redundancy of all ordered resolution inferences between $\mathcal{L}K$-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 (27) 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 (20) 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 that is greater than A with respect to $\succ$. Otherwise $\rhd(\Gamma + \Lambda, \Delta + \Pi)$ would be a larger 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 (20)) and

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

(an instance of (27)). This is another proof of the same clause:

$$\dfrac{\dfrac{\rhd(\Gamma, A + \Delta) \quad \rhd(\Lambda' + A, \Pi')}{\rhd(\Gamma + \Lambda', \Delta + \Pi')}\,(27)}{\rhd(\Gamma + \Lambda + \Lambda', \Delta + \Pi + \Pi')}\,(20)$$

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 larger than any formula that is not in Γ, Δ or Π.

As another example consider the clause

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

obtained from

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

and

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

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

$$\dfrac{\dfrac{\rhd(\Lambda + A + B, \Pi) \qquad \dfrac{\rhd(\Gamma, (A \wedge B) + \Delta)}{\rhd(\Gamma, A + \Delta)}\,(23)}{\rhd(\Lambda + \Gamma + B, \Pi + \Delta)}\,(27) \qquad \dfrac{\rhd(\Gamma, (A \wedge B) + \Delta)}{\rhd(\Gamma, B + \Delta)}\,(24)}{\rhd(\Gamma + \Lambda, \Delta + \Pi)}\,(27)$$

We again utilize that $A \wedge B$ is larger 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. □

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

8.3 Consequences of the Saturation of $\mathcal{L}K$

THEOREM 8.2 The Horn theory of $\mathcal{L}K$, and hence the theory of derivable inference rules for LK, is decidable in exponential time.

PROOF. Since $\mathcal{L}K$ is saturated with respect to all extensions of the subterm ordering on formulas, it follows that any ground consequence Q of $\mathcal{L}K$ follows from $\mathcal{L}K$ 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 $\mathcal{L}K$ ground atoms with that property, hence exponentially many ground clauses of $\mathcal{L}K$ are needed for proving Q, and any of these clauses can be generated in linear time. □

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 with schematic lists of arguments to predicates is not the usual one.

As observed previously, the cut rule is not redundant for the Horn clause theory. However, it is not needed to prove sequents in LK or, equivalently, atomic formulas in $\mathcal{L}K$.

THEOREM 8.3 (CUT ELIMINATION) If a sequent S is provable in LK then there exists a proof of S without application of cut.

PROOF. Let S be the sequent $A_1, \ldots, A_k \vdash B_1, \ldots, B_n$ over the formulas A_i and B_j. We prove the theorem by showing that if S is provable in LK then $\hat{S} = \rhd_{k,n}(A_1, \ldots, A_k, B_1, \ldots, B_n)$ follows from $\mathcal{L}K \setminus \{(27)\}$. Suppose S has a proof in LK. Then $\hat{S}$ is a consequence from $\mathcal{L}K$. 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 $\mathcal{L}K$ 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 provable if and only if each of the sequents in the set is provable.

Let $\succ$ be a total ordering of the propositional variables. By theorem 8.2 there exists a linear proof of $\hat{S}$ in $\mathcal{L}K$ by ordered resolution with respect to $\succ$. Such a proof may be viewed as a tree with root $\hat{S}$ and each interior node is an instance of a clause in $\mathcal{L}K$. By locality, each node in the proof tree is an $\mathcal{L}K$-atom in which all terms are propositional variables in S. Moreover, any node is smaller in the atom ordering than its ancestor. The only $\mathcal{L}K$-clauses that can participate in a resolution step are axioms (19), weakening (20), and (27). We show, by induction over $\succ$, that inferences with (27) are redundant. Let

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

be an instance of (27) 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. □

THEOREM 8.4 (ANALYTIC CUTS) If a rule is derivable in LK then there exists a derivation of the rule in which all cuts are analytic, i.e., where the cut formula is a subformula of a formula occurring in the rule.

PROOF. Since $\mathcal{L}K$ is saturated with respect to the subterm ordering, $\mathcal{L}K$ is subterm local. The Horn theory of $\mathcal{L}K$ coincides with the set of derivable rules in LK. Therefore the assertion follows. □

9. COMPARISON WITH SUBTERM LOCALITY

In section 2 we compared the definition of subterm locality with order locality. We now compare algorithms for establishing locality. A full description of the algorithm given by [Givan and McAllester 1992] for testing subterm locality is beyond the scope of this paper. However, we do 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.,

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

9.1 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 $N \models\!\!\models C$; 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 that 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.

In [Givan and McAllester 1992] 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 their impossibility. 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

$$\Sigma, \Gamma \rightarrow A$$

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 Horn clause 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 containing at least x and z. Now, according to [Givan and McAllester 1992] 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

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

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 [Givan and McAllester 1992]); 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 proof normalization procedure: given a direct proof, of say $p(a, b)$ from N, the proof that the set of Horn clauses 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.

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

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

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

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 [Givan and McAllester 1992]) 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 set of Horn clauses is subterm local.[8]

9.2 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.

Let Σ be any signature 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

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

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

By theorem 5.1 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 3.1) and this form is used as the basis of showing the invariant that term size is bounded by the initial clause 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:

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

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 rôle analogous to backchaining in McAllester's: he 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. Second, and related to the first point, is that when we generate new clauses, they play a different rôle than 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

[8] 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.

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 set of Horn clauses N' deductively equivalent to a non-local one N.

We have mentioned before, in the discussion of the results of the section 4.2, that only conclusions of non-redundant peak inferences need to be kept in the final representation. Conclusions of non-redundant plateau inferences are only intermediate results for representing slopes and they can be deleted in the end. With this modification of the saturation procedure it should be possible to have the best of both worlds, namely turn a non-local theory into a local one, without adding clauses that are not needed for passing McAllester's locality test.

10. CONCLUSION AND FUTURE WORK

Our work generalizes subterm locality both to incorporate term-orderings and to handle full clauses. Even for polynomial time decidable theories, the flexibility offered by admitting orderings other than the subterm ordering is substantial. Our ordering-based approach also allows us to characterize entailment problems with respect to two complexity hierarchies: deterministic and co-nondeterministic time.

We have made a direct connection between order locality and standard techniques from resolution theorem proving, which enables us to use already implemented proof-procedures. Aside from providing a practical means for computer-assisted complexity analysis, this allows us to bring standard improvements, e.g., AC unification and the like, into our framework. Such generalizations, and implementing a procedure for checking locality using peak saturation, as described in section 4.2, are directions for future work.

Lemma 2.1 is the key to implementing a fast decision procedure for Horn entailment problems: appropriate ground instances are generated and then a linear time inference algorithm is applied to the resulting ground clause set. Another direction for future work is to avoid generating ground instances of some clauses and instead build the clauses into the inference algorithm. For example, the present approach always gives a lower bound of $O(n^3)$ for any clause set which includes transitivity; hence we get a suboptimal bound for problems like congruence closure. Transitivity and more general composition laws for binary relations can be built into ground inference systems. Ordered chaining calculi have been proposed and proved complete in the presence of a similar notion of redundancy in [Bachmair and Ganzinger 1994b]. We believe that the results of the present paper extend to these calculi.

ACKNOWLEDGMENTS

We thank David McAllester for helpful comments.

REFERENCES

Bachmair, L. and Ganzinger, H. 1990. On restrictions of ordered paramodulation with simplification. In M. Stickel Ed., *Proc. 10th Int. Conf. on Automated Deduction, Kaiserslautern*, Volume 449 of *Lecture Notes in Computer Science* (Berlin, 1990), pp. 427–441. Springer-Verlag.

Bachmair, L. and Ganzinger, H. 1994a. Rewrite-based equational theorem proving with selection and simplification. *J. Logic and Computation 4*, 3, 217–247. Revised version of Technical Report MPI-I-91-208, 1991.

Bachmair, L. and Ganzinger, H. 1994b. Rewrite techniques for transitive relations. In *Proc. 9th IEEE Symposium on Logic in Computer Science* (1994), pp. 384–393. IEEE Computer Society Press.

Bachmair, L. and Ganzinger, H. 1997. A theory of resolution. Research Report MPI-I-97-2-005, Max-Planck-Institut für Informatik, Saarbrücken, Saarbrücken. To appear in the Handbook of Automated Reasoning.

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

Ganzinger, H., Nieuwenhuis, R., and Nivela, P. 1994. The Saturate system. User manual maintained on the Web at http://www.mpi-sb.mpg.de/SATURATE/.

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

Hofbauer, D. and Lautemann, C. 1989. Termination proofs and the length of derivations. In N. Dershowitz Ed., *Proceedings of the 3rd International Conference on Rewriting Techniques and Applications*, Volume 355 of *Lecture Notes in Computer Science* (1989), pp. 167–177. Springer-Verlag.

Knuth, D. E. and Bendix, P. B. 1970. Simple word problems in universal algebras. In J. Leech Ed., *Computational Problems in Abstract Algebra*, pp. 263–297.

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

Nebel, B. and Bürckert, H.-J. 1995. Reasoning about temporal relations: A maximal tractable subclass of allen's interval algebra. *J. Association for Computing Machinery 42*, 1, 43–66.

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

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