MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1344 December 1991

Tractable Inference Relations

Robert Givan and David Allen McAllester

Abstract

We consider the concept of *local* sets of inference rules. Locality is a syntactic condition on rule sets which guarantees that the inference relation defined by those rules is polynomial time decidable. Unfortunately, determining whether a given rule set is local can be difficult. In this paper we define inductive locality, a strengthening of locality. We also give a procedure which can automatically recognize the locality of any inductively local rule set. Inductive locality seems to be more useful that the earlier concept of strong locality. There are many natural examples of inductively local rule sets that are not strongly local. We have not found any natural examples of local rule sets that fail to be inductively local. However, we show here that locality, as a property of rule sets, is undecidable in general.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the work described in this paper was provided in part by Misubishi Electric Research Laboratories, Inc. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124.

1 Introduction

Under what conditions does a given set of inference rules define a computationally tractable inference relation? This is a syntactic question about syntactic inference rules. There are a variety of motivations for identifying tractable inference relations. Tractable inference relations sometimes provide decision procedures for semantic theories. For example, the equational inference rules of reflexivity, symmetry, transitivity, and substitutivity define a tractable inference relation that yields a decision procedure for the entailment relation between sets of ground equations [Kozen, 1977], [Shostack, 1978]. Tractable inference relations can also provide useful incomplete inference in cases where complete inference is intractable. Many practical search algorithms use some form of incomplete inference to prune nodes in the search tree [Knuth, 1975], [Mackworth, 1977], [Pearl and Korf, 1987]. Incomplete inference also plays an important role in pruning search in constraint logic programming [Jaffar and Lassez, 1987], [van Hentenryck, 1989], [McAllester and Siskind, 1992]. Tractable inference relations can also be used to define a notion of "obvious inference" which can then be used in "Socratic" proof verification systems which require proofs to be reduced to obvious steps [McAllester, 1989], [Givan *et al.*, 1991].

Inference rules are syntactically similar to first order Horn clauses. In fact, most inference rules can be syntactically represented by a Horn clause in sorted first order logic. If R is a set of Horn clauses, Σ is set of ground atomic formulas, and Φ is a ground atomic formula, then we write $\Sigma \vdash_R \Phi$ if $\Sigma \cup R \vdash \Phi$ in first order logic. We write $\vdash_R$ rather than $\models_R$ because we think of R as a set of syntactic inference rules and $\vdash_R$ as the inference relation generated by those rules. Throughout this paper we use the term "rule set" and "Horn set" interchangeably. Technically this phrase to refers to a finite set of Horn clauses. We give nontrivial conditions on R which ensure that the inference relation $\vdash_R$ is polynomial time decidable.

One rather simple condition ensuring the tractability of $\vdash_R$ is that R be "superficial". We call a Horn clause $\Psi_1 \wedge \Psi_2 \wedge \cdots \wedge \Psi_n \Rightarrow \Phi$ superficial if every term appearing in the conclusion Φ also appears in some antecedent Ψ_i. Several authors have observed that if R is a finite set of superficial Horn clauses then $\vdash_R$ is polynomial time decidable [Aho and Ullman, 1979], [Papadimitriou, 1985]. Building on independent results of Vardi and Immerman, Papadimitriou showed that superficial Horn sets provide a characterization of the complexity class P [Vardi, 1982], [Immerman, 1986], [Papadimitriou, 1985]. Let $\mathcal{P}$ be any polynomial time predicate on k first order terms. Papadimitriou showed, in essence, that for any such $\mathcal{P}$ there exists a set R of superficial Horn clauses such that for any first order terms t_1, t_2, $\ldots$ t_k we have that $\mathcal{P}(t_1, t_2, \ldots t_n)$ if and only if $\texttt{Input}(t_1, t_2, \ldots t_k) \vdash_R \texttt{Accept}$

where Input is a predicate symbol and Accept is a distinguished proposition symbol. We can think of a set of Horn clauses as a Prolog program. Papadimitriou's result can be interpreted as saying that the class of superficial Prolog programs defines all and only polynomial time relations.

Superficial Horn sets are a special case of the more general class of *local* Horn sets [McAllester, 1990]. A set R of Horn clauses is local if whenever $\Sigma \vdash_R \Phi$ there exists a proof of Φ from Σ such that every term in the proof is mentioned in Σ or Φ. If R is local then $\vdash_R$ is polynomial time decidable. All superficial Horn sets are local but many local Horn sets are not superficial. The set of the four inference rules for equality is local but not superficial. The local inference relations provide a different characterization of the complexity class P. In section 5 we prove that for any predicate $\mathcal{P}$ of k first order terms, $\mathcal{P}$ can be computed in polynomial time if and only if there exists a local Horn set R such that for any terms $t_1, t_2, \ldots t_k$ we have that $\mathcal{P}(t_1, t_2, \ldots t_k)$ if and only if $\vdash_R P(t_1, t_2, \ldots t_k)$ where P is a predicate symbol representing $\mathcal{P}$.

Unlike superficiality, locality can be difficult to recognize. The set of four inference rules for equality is local but the proof of this fact is nontrivial. In section 5 we give a proof that locality, as a property of sets of Horn clauses, is undecidable. However, there are subclasses of local Horn sets which can be mechanically recognized. A notion of a strongly local rule set is defined in [McAllester, 1990] and a procedure is given which will automatically recognize the locality of any strongly local rule set. The set of the four basic rules for equality is strongly local. As another example of a strongly local rule set we give the following rules for reasoning about a monotone operator from sets to sets.

$$x \subseteq x, \quad x \subseteq y \wedge y \subseteq z \Rightarrow x \subseteq z$$

$$x \subseteq y \Rightarrow f(x) \subseteq f(y)$$

A variety of other strongly local rule sets are given [McAllester, 1990]. As an example of a rule set that is local but not strongly local we give the following rules for reasoning about union and intersection operations on sets.[1]

$$x \subseteq x, \quad x \subseteq y \wedge y \subseteq z \Rightarrow x \subseteq z$$

$$x \subseteq x \cup y, \quad y \subseteq x \cup y, \quad x \subseteq z \wedge y \subseteq z \Rightarrow x \cup y \subseteq z$$

$$x \cap y \subseteq x, \quad x \cap y \subseteq y, \quad z \subseteq x \wedge z \subseteq y \Rightarrow z \subseteq x \cap y$$

These rules remain local when the above monotonicity rule is added. With or without the monotonicity rule, the rule set is not strongly local.

[1]These rules are complete for reasoning about lattice operations. We phrase them as rules about sets because we believe this is how they are most commonly used.

In this paper we construct another machine-recognizable subclass of the local Horn sets which we call *inductively local* Horn sets. All of the strongly local rule sets given in [McAllester, 1990] are also inductively local. The procedure described in section 4 for recognizing inductively local rule sets has been implemented and has been used to determine that the above rule set is inductively local. Hence the inference relation defined by the rules is polynomial time decidable. We have been able to show that there are strongly local rule sets which are not inductively local, although our examples are somewhat artificial. We have not found any natural examples of local rule sets that fail to be inductively local. Polynomial time inference relations have a variety of applications. Inductively local rule sets provide a variety of mechanically recognizable polynomial time inference relations.

2 Basic Terminology

In this section we give more precise definitions of the concepts discussed in the introduction.

> **Definition:** A Horn clause is a first order formula of the form $\Psi_1 \wedge \Psi_2 \wedge \ldots \wedge \Psi_k \Rightarrow \Phi$ where Φ each Ψ_i is an atomic formula. For any set of Horn clauses R, any finite set Σ of ground terms, and any ground atomic formula Φ, we write $\Sigma \ \vdash_R \ \Phi$ if $\Sigma \cup U(R) \vdash \Phi$ in first order logic where $U(R)$ is the set of universal closures of Horn clauses in R.

There are a variety of inference relations defined in this paper. For any inference relation $\vdash$ and any sets of ground formulas Σ and Γ we write $\Sigma \vdash \Gamma$ if $\Sigma \vdash \Psi$ for each Ψ in Γ.

The inference relation $\vdash_R$ can be given a more direct syntactic characterization. This syntactic characterization is more useful in determining locality.

> **Definition:** A *derivation* of Φ from Σ using Horn set R is a sequence of ground atomic formulas $\Psi_1, \ \Psi_2, \ldots \Psi_n$ such that Ψ_n is Φ and for each Ψ_i there exists a Horn clause $\Theta_1 \wedge \Theta_1 \wedge \ldots \wedge \Theta_k \Rightarrow \Psi'$ in R and a ground substitution σ such that $\sigma[\Psi']$ is Ψ_i and each formula of the form $\sigma[\Theta_i]$ is either a member of Σ or a formula appearing in earlier in the derivation.

> **Lemma:** $\Sigma \ \vdash_R \ \Phi$ if and only if there exists a derivation of Φ from Σ using the Horn set R.

The following restricted inference relation plays an important role in the analysis of locality.

Definition: We write $\Sigma \Vdash_R \Phi$ if there exists a derivation of Φ form Σ such that every term appearing in the derivation apears as a subexpression of Φ or as a subexpression of some formula in Σ.

Lemma: For any finite Horn set R the inference relation $\Vdash_R$ is polynomial time decidable.

Proof: Let n be the number of terms that appear as subexpressions of Φ or a formula in Σ. If P is a predicate of k arguments that appears in the inference rules R then there are at most n^k formulas of the form $P(s_1, \ldots s_k)$ such that $\Sigma \Vdash_R P(s_1, \ldots s_k)$. Since R is finite there is some maximum arity over all the predicate symbols that appear in R. The total number of formulas that can be derived under the restrictions in the definition of $\Vdash_R$ is order n^k where k is the maximum arity of the predicates in R. ∎

Clearly, if $\Sigma \Vdash_R \Phi$ then $\Sigma \vdash_R \Phi$. But the converse does not hold in general. By definition, if the converse holds then R is local.

Definition: The Horn set R is *local* if the restricted inference relation $\Vdash_R$ is the same as the unrestricted relation $\vdash_R$.

Clearly, if R is local then $\vdash_R$ is polynomial time decidable.

3 Another Characterization of Locality

In this section we give an alternate characterization of locality. This characterization of locality plays an important role in both the definition of strongly local rule sets given in [McAllester, 1990] and in the notion of inductively local rule sets given here.

Definition: A *bounding set* is a set Υ of ground terms such that every subterm of a member of Υ is also a member of Υ.

Definition: A ground atomic formula Ψ is called a *label formula* of a bounding set Υ if every term in Ψ is a member of Υ.

Definition: For any bounding set Υ, we define the inference relation $\vdash_{R,\Upsilon}$ to be such that $\Sigma \vdash_{R,\Upsilon} \Phi$ if and only if there exists a derivation of Φ from Σ such that every formula in the derivation is a label formula of the term set Υ.

We have that $\Sigma \vdash_R \Phi$ if and only if $\Sigma \vdash_{R,\Upsilon} \Phi$ where Υ is the set of all terms appearing as subexpressions of Φ or formulas in Σ. The inference relation $\vdash_{R,\Upsilon}$ can be used to give another characterization of locality. Suppose that R is not local. In this case there must exist some Σ and Φ such that $\Sigma \not\vdash_R \Phi$ but $\Sigma \vdash_R \Phi$. Let Υ be the set of terms that appear in Σ and Φ. We must have $\Sigma \not\vdash_{R,\Upsilon} \Phi$. However, since $\Sigma \vdash_R \Phi$ we must have $\Sigma \vdash_{R,\Upsilon'} \Phi$ for some finite superset Υ' of Υ. Consider "growing" the bounding set one term at a time, starting with the terms that appear in Σ and Φ.

Definition: A *one step extension* of a bounding set Υ is a ground term α that is not in Υ but such that every proper subterm of α is a member of Υ.

Definition: An feedback event for R consists of a finite set Σ of ground formulas, a ground formula Φ, a bounding set Υ containing all terms that appear in Σ and Φ, and a one step extension α of Υ such that $\Sigma \vdash_{R,\Upsilon\cup\{\alpha\}} \Phi$ but $\Sigma \not\vdash_{R,\Upsilon} \Phi$

By abuse of notation, a feedback event will be written as $\Sigma \vdash_{R,\Upsilon\cup\{\alpha\}} \Phi$.

Lemma: R is local if and only if there are no feedback events for R.

Proof: First note that if R has a feedback event then R is not local — if $\Sigma \vdash_{R,\Upsilon\cup\{\alpha\}} \Phi$ then $\Sigma \vdash_R \Phi$ but if $\Sigma \not\vdash_{R,\Upsilon} \Phi$ then $\Sigma \not\vdash_R \Phi$. Conversely suppose that R is not local. In there case there is some Σ and Φ such that $\Sigma \not\vdash_R \Phi$ but $\Sigma \vdash_{R,\Upsilon} \Phi$ for some finite Υ. By considering a least such Υ one can show that a feedback event exists for R. ∎

The concepts of strong locality and inductive locality both involve the concept of a feedback event. We can define strong locality by first defining $C_R(\Sigma, \Upsilon)$ to be the set of of formulas Ψ such that $\Sigma \vdash_{R,\Upsilon} \Psi$. R is strongly local if it is local and there exists a natural number k such that whenever $\Sigma \vdash_{R,\Upsilon\cup\{\alpha\}} \Psi$ there exists a derivation of Ψ from $C_R(\Sigma, \Upsilon)$ such that every term in the derivation is a member of $\Upsilon \cup \{\alpha\}$ and such that the derivation is no longer than k. As mentioned above, the set of the four basic inference rules for equality is strongly local and there exists a procedure which can recognize the locality of any strongly local rule set. The definition of inductive locality is somewhat more involved and is given in the next section.

4 Inductive Locality

To define inductive locality we first define the notion of a feedback template. A feedback template represents a set of potential feedback events. We also define a backward chaining process which generates feedback templates from a Horn set R. We show that if there exists a feedback event for R then such an event will be found by this backchaining process. Furthermore, we define an "inductive" termination condition on the backchaining process and show that if the backchaining process achieves inductive termination then R is local.

Throughout this section we let R be a fixed but arbitrary set of Horn clauses. The inference relation $\vdash_{R,\Upsilon}$ will be written as $\vdash_{\Upsilon}$ with the understanding that R is an implicit parameter of the relation.

We define feedback templates as ground objects — they contain only ground first order terms and formulas. The process for generating feedback templates is defined as a ground process — it only deals with ground instances of clauses in R. The ground process can be "lifted" using a lifting transformation. Since lifting is largely mechanical for arbitrary ground procedures [McAllester and Siskind, 1992], the lifting operation is only discussed briefly here.

> **Definition:** A *feedback template* consists of a set of ground atomic formulas Σ, a multiset of ground atomic formulas Γ, a ground atomic formula Φ, a bounding set Υ, and a one step extension α of Υ such that Φ and every formula in Σ is a label formula of Υ, every formula in Γ is a label formula of $\Upsilon \cup \{\alpha\}$ that contains α, and such that $\Sigma \cup \Gamma \vdash_{\Upsilon\cup\{\alpha\}} \Phi$.

By abuse of notional a feedback template will be written as $\Sigma, \Gamma \vdash_{\Upsilon\cup\{\alpha\}} \Phi$. Γ is a multisetset of ground atomic formulas, each of which is a label formula of $\Upsilon \cup \{\alpha\}$ containing α, and such that the union of Σ and Γ allow the derivation of Φ relative to the bounding set $\Upsilon \cup \{\alpha\}$. A feedback template is a potential feedback event in the sense that an extension of Σ that allows a derivation of the formulas in Γ may result in a feedback event. The need to make Γ a multiset rather than a set is discussed below. Feedback templates for R can be constructed by backward chaining.

Procedure for Generating a Template for R:

1. Let $\Psi_1 \wedge \Psi_2 \wedge \cdots \wedge \Psi_n \Rightarrow \Phi$ be a ground instance of a clause in R.
2. Let α be a term that appears in the clause but does not appear in the conclusion Φ and does not appear as a proper subterm of any other term in the clause.

3. Let Υ be a bounding set that does not contain α but does contain every term in the clause other than α.
4. Let Σ be the set of antecedents Ψ_i which do not conain α.
5. Let Γ be the set of antecedents Ψ_i which do conaint α.
6. Return the feedback template $\Sigma, \Gamma \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$.

We let $\mathcal{T}_0[R]$ to be the set of all feedback templates that can be derived from R by an application of the above procedure. We leave it to the reader to verify that $\mathcal{T}_0[R]$ is a set of feedback templates. Now consider a feedback template $\Sigma, \Gamma \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$. We can construct a new template by backward chaining from $\Sigma, \Gamma \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ using the following procedure.

Procedure for Backchaining from $\Sigma, \Gamma \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$

1. Let Θ be a member of Γ
2. Let $\Psi_1 \wedge \Psi_2 \wedge \cdots \wedge \Psi_n \Rightarrow \Theta$ be a ground instances of a clause in R that has Θ as its conclusion and such that each Ψ_i is a label formula of $\Upsilon \cup \{\alpha\}$.
3. Let Σ' be Σ plus all antecedents Ψ_i which do not contain α.
4. Let Γ' be Γ minus Θ plus all antecedents Ψ_i which do contain α.
5. Return the template $\Sigma', \Gamma' \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$.

In step 4 of the above procedure, Γ' is constructed using multiset opertions. For example, if the multiset Γ contains two occurances of Θ, then "Γ minus Θ" contains one occurance of Θ. The use of multisets ensures that backchaining steps "commute". To see this let $\Sigma, \Gamma \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ be a template and let Θ_1 and Θ_2 be two elements of Γ. Select two Horn clauses for backchaining on Θ_1 and Θ_2 respectively and let $\Sigma'', \Gamma'' \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ be the result of performing these two backchaining steps. The use of multisets ensures that the template $\Sigma'', \Gamma'' \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ is independent of the order in which the two backchaining steps are done. Suppose we use sets rather than multisets and suppose that backchaining on Θ_1 generates Θ_2 as a subgoal. In this case backchaining on Θ_1 and then backchaining on Θ_2 eliminates all occurances of Θ_2, while backchining on Θ_2 and then backchaining on Θ_1 leaves an occurance of Θ_2. The use of multisets rather than sets guarantees that in this case the subgoal Θ_2 remains under either order of backchaining. The commutativity of backchaining steps simplifies the inductive termination test considered below.

For any set $\mathcal{T}$ of feedback templates we define $\mathcal{B}[\mathcal{T}]$ to be $\mathcal{T}$ plus all templates that can be derived from an element of $\mathcal{T}$ by an application of the above backchaining procedure. It is important to keep in mind that by definition $\mathcal{B}[\mathcal{T}]$ contains $\mathcal{T}$. We let $\mathcal{B}^n[\mathcal{T}]$ be $\mathcal{B}[\mathcal{B}[\cdots \mathcal{B}[\mathcal{T}]]]$ with n applications of $\mathcal{B}$.

Definition: A feedback template $\Sigma, \Gamma \vdash_{\Upsilon \cup \{\alpha\}} \Phi$ is called *critical* if Γ is empty.

If $\Sigma, \emptyset, \vdash_{\Upsilon \cup \{\alpha\}} \Phi$ is a critical template then $\Sigma \vdash_{\Upsilon \cup \{\alpha\}} \Phi$. If $\Sigma \not\vdash_{\Upsilon} \Phi$ then $\Sigma \vdash_{\Upsilon \cup \{\alpha\}} \Phi$ is a feedback event. By abuse of terminology, a critical template $\Sigma, \emptyset \vdash_{\Upsilon \cup \{\alpha\}} \Phi$ such that $\Sigma \not\vdash_{\Upsilon} \Phi$ will be called a feedback event. The following lemma provides the motivation for the definition of a feedback template and the backchaining process.

Lemma: There exists a feedback event for R if and only if there exists a j such that $\mathcal{B}^j[\mathcal{T}_0[R]]$ contains a feedback event.

Proof: To prove the above lemma suppose that there exists a feedback event for R. Let $\Sigma \vdash_{\Upsilon \cup \{\alpha\}} \Phi$ be a minimal feedback event for R, i.e., a feedback event for R which minimizes the length of the derivation of Φ from Σ under the bounding set $\Upsilon \cup \{\alpha\}$. The fact that this feedback event is minimal implies that every formula in the derivation other than Φ contains α. To see this suppose that Θ is a formula in the derivation other than Φ that does not involve α. We then have $\Sigma \vdash_{\Upsilon \cup \{\alpha\}} \Theta$ and $\Sigma \cup \{\Theta\} \vdash_{\Upsilon \cup \{\alpha\}} \Phi$. One of these two must be a feedback event — otherwise we would have $\Sigma \vdash_{\Upsilon} \Phi$. But if one of these is a feedback event then it involves a smaller derivation than $\Sigma \vdash_{\Upsilon \cup \{\alpha\}} \Phi$ and this contradicts the assumption that $\Sigma \vdash_{\Upsilon \cup \{\alpha\}} \Phi$ is minimal. Since every formula other than Φ in the derivation underlying $\Sigma \vdash_{\Upsilon \cup \{\alpha\}} \Phi$ contains α, the template $\Sigma, \emptyset \vdash_{\Upsilon \cup \{\alpha\}} \Phi$ can be derived by backchaining. ∎

The above lemma implies that if the Horn set is not local then backchaining will uncover a feedback event. However, we are primarilly interested in those cases where the Horn set is local. If the backchaining process is to establish locality then we must find a termination condition which guarantees locality. Let $\mathcal{T}$ be a set of feedback templates. In practice $\mathcal{T}$ can be taken to be $\mathcal{B}^j[\mathcal{T}_0[R]]$ for some finite j. We define a "self-justification" property for sets of feedback templates and prove that if $\mathcal{T}$ is self-justifying then there is no n such that $\mathcal{B}^n[T]$ contain a feedback event. In defining the self-justification property we treat each template in $\mathcal{T}$ as an independent induction hypothesis. If each template can be "justified" using the set of templates as induction hypotheses, then the set $\mathcal{T}$ is self-justifying.

Definition: We write $\Sigma, \Gamma \vdash_{\mathcal{T},\Upsilon} \Phi$ if $\mathcal{T}$ contains templates

$$\Sigma_1, \Gamma_1 \vdash_{\Upsilon \cup \{\alpha\}} \Psi_1,\ \ \Sigma_2, \Gamma_2 \vdash_{\Upsilon \cup \{\alpha\}} \Psi_2,\ \ \ldots\ \ \Sigma_k, \Gamma_k \vdash_{\Upsilon \cup \{\alpha\}} \Psi_k$$

where each Σ_i is a subset of Σ, each Γ_i is a subset of Γ and $\Sigma \cup \{\Psi_1,\ \Psi_2,\ \ldots\ \Psi_k\} \vdash_{\Upsilon} \Phi$.

Definition: $\mathcal{T}$ is said to *justify* a template $\Sigma, \Gamma \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ if there exists a $\Theta \in \Gamma$ such that for each template $\Sigma', \Gamma' \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ generated by backchaining from $\Sigma, \Gamma \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ by selecting Θ at step 1 of the backchaining procedure we have $\Sigma', \Gamma' \Vdash_{\mathcal{T},\Upsilon} \Phi$.

Definition: The set $\mathcal{T}$ is called *self-justifying* if every member of $\mathcal{T}$ is either critical or justified by $\mathcal{T}$, and $\mathcal{T}$ does not contain any feedback events.

Induction Theorem: If $\mathcal{T}$ is self-justifying then no set of the form $\mathcal{B}^n[\mathcal{T}]$ contains a feedback event.

Proof: We must show that for every critical template $\Sigma, \emptyset \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ in $\mathcal{B}^n[\mathcal{T}]$ we have that $\Sigma \Vdash_{\Upsilon} \Phi$. The proof is by induction on n. Consider a critical template $\Sigma, \emptyset \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ in $\mathcal{B}^n[\mathcal{T}]$ and assume the theorem for all critical templates in $\mathcal{B}^j[\mathcal{T}]$ for j less than n. The critical template $\Sigma, \emptyset \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ must be derived by backchaining from some template $\Sigma', \Gamma' \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ in $\mathcal{T}$. Note that Σ' must be a subset of Σ. If Γ' is empty then Σ' equals Σ and $\Sigma \Vdash_{\Upsilon} \Phi$ because $\mathcal{T}$ is assumed not to contain any feedback events. If Γ' is not empty then, since $\mathcal{T}$ is self justifying, there must exist some Θ in Γ' such that for each template $\Sigma'', \Gamma'' \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ derived from $\Sigma', \Gamma' \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ by backchaining on Θ we have $\Sigma'', \Gamma'' \Vdash_{\mathcal{T},\Upsilon} \Phi$. We noted above that backchaining operations commute. By the commutativity of backchaining steps there exists a backchaining sequence from $\Sigma', \Gamma' \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ to $\Sigma, \emptyset \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ such that the first step in that sequence is a backchaining step on Θ. Let $\Sigma'', \Gamma'' \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$ be the template that results from this backchaining step from $\Sigma', \Gamma' \Vdash_{\Upsilon \cup \{\alpha\}} \Phi$. Note that Σ'' is a subset of Σ. We must now have $\Sigma'', \Gamma'' \Vdash_{\mathcal{T},\Upsilon} \Phi$. By definition, $\mathcal{T}$ must contain templates

$$\Sigma_1, \Gamma_1 \Vdash_{\Upsilon \cup \{\alpha\}} \Psi_1,\ \ \Sigma_2, \Gamma_2 \Vdash_{\Upsilon \cup \{\alpha\}} \Psi_2,\ \ \ldots\ \ \Sigma_k, \Gamma_k \Vdash_{\Upsilon \cup \{\alpha\}} \Psi_k$$

such that each Σ_i is a subset of Σ'', each Γ_i is a subset of Γ'', and $\Sigma'' \cup \{\Psi_1,\ \Psi_2,\ \ldots\ \Psi_k\} \Vdash_{\Upsilon} \Phi$. Note that each Σ_i is a subset of Σ. Since Γ_i is a subset of Γ'' there must be a sequence of *fewer than* n backchaining steps that leads from $\Sigma_i, \Gamma_i \Vdash_{\Upsilon \cup \{\alpha\}} \Psi_i$ to a critical template $\Sigma_i', \emptyset \Vdash_{\Upsilon \cup \{\alpha\}} \Psi_i$. Note that Σ_i' is a subset of Σ. This critical template is a member of $\mathcal{B}^j[\mathcal{T}]$ for j less than n so we have $\Sigma_i' \Vdash_{\Upsilon} \Psi_i$ and thus $\Sigma \Vdash_{\Upsilon} \Psi_i$. But if $\Sigma \Vdash_{\Upsilon} \Psi_i$ for each Ψ_i, and $\Sigma \cup \{\Psi_1,\ \Psi_2,\ \ldots\ \Psi_k\} \Vdash_{\Upsilon} \Phi$, then $\Sigma \Vdash_{\Upsilon} \Phi$. ∎

We now come the main definition and theorem of this section.

Definition: A Horn set R is called *inductively local* if there exists some n such that $\mathcal{B}^n[\mathcal{T}_0[R]]$ is self-justifying.

Theorem: There exists a procedure which, given any finite set R of Horn clauses, will terminate with a feedback event whenever R is not local, terminate with "success" whenever R is inductively local, and fail to terminate in cases where R is local but not inductively local.

The procedure is derived by lifting the above ground procedure for computing $\mathcal{B}^n[\mathcal{T}[R]]$. Lifting can be formalized as a mechanical operation on arbitrary nondeterministic ground procedures [McAllester and Siskind, 1992]. In the lifted version the infinite set $\mathcal{B}^j[\mathcal{T}_0[R]]$ is represented by a finite set of "template schemas" each of which consists of a template expression $\Sigma, \Gamma \vdash_{\Upsilon \cup \{\alpha\}} \Phi$ involving variables plus a set of constraints on those variables. Lifting always converts a ground computation to a lifted computation resulting in a data structure containing variables together with constraints on those variables. In this case all of the terms in $\Sigma, \Gamma \vdash_{\Upsilon \cup \{\alpha\}} \Phi$ may contain first order variables and Υ is a variable ranging over bounding sets. The variable Υ is subject to the constraint $\alpha \notin \Upsilon$ and a set of constraints of the form $t \in \Upsilon$ where t is a first order term. There exists a simple procedure for determining if these constraints are satisfiable, i.e., if there exists a ground substitution σ and a ground bounding set Υ satisfying these constraints. It is also possible to construct a procedure that determines if $\mathcal{B}^n[\mathcal{T}_0[R]]$ is self-justifying by examining the finite set of template schemas constructed by the lifted computation.

5 Locality is Undecidable

To prove that locality is undecidable we start with Papadimitriou's result that superficial Horn sets can represent any polynomial time predicate. Let $\mathcal{P}$ be any polynomial time predicate on first order terms. Papadimitriou proved that a polynomial time bounded Turing machine for computing $\mathcal{P}$ can be mechanically converted to a superficial Horn set R such that for any first order terms $t_1, t_2, \ldots t_k$ we have that $\mathcal{P}(t_1,\ t_2,\ \ldots\ t_n)$ if and only if $\texttt{Input}(t_1,\ t_2,\ \ldots\ t_k)\ \vdash_R \texttt{Accept}$ where `Input` is a predicate symbol and `Accept` is a distinguished proposition symbol.[2] The first step in proving that locality is undecidable is to convert R to a local Horn set R' such that for any $t_1, t_2, \ldots t_k$ we have that $\mathcal{P}(t_1,\ t_2,\ \ldots\ t_k)$ if and only if $\vdash_R P(t_1,\ t_2,\ \ldots\ t_k)$.

[2] Papadimitriou's result is actually somewhat different. He only uses Horn clauses that do not contain any function symbols other than constants but then requires more input premises. Papadimitriou's result can be used to prove the result stated here by constructing additional clauses that allow the additional premises needed by Papadimitriou's clauses to be proved by superficial inference rules operating on the terms $t_1, t_2, \ldots t_n$.

Let $\mathcal{P}$ be a polynomial time predicate on k first order terms and let R be a superficial Horn set satisfying Papadimitriou's condition for $\mathcal{P}$. For each predicate symbol Q of m arguments appearing in R let Q' be a new predicate symbol of $k+m$ arguments. We define the Horn set R' to be the Horn set containing the following clauses.

- $\mathtt{Input}'(x_1, \ldots x_k, x_1, \ldots x_k)$
- All clauses of the form

  $Q'_1(x_1, \ldots x_k, t_{1,1}, \ldots t_{1,m_1}) \wedge \cdots \wedge Q'_n(x_1, \ldots x_k, t_{n,1}, \ldots t_{n,m_n}) \Rightarrow W'(x_1, \ldots x_k, s_1, \ldots s_j)$

  where the clause $Q_1(t_{1,1}, \ldots t_{1,m_1}) \wedge \cdots \wedge Q_n(t_{n,1}, \ldots t_{n,m_n}) \Rightarrow W(s_1, \ldots s_j)$ is in R.
- The clause $\mathtt{Accept}'(x_1, \ldots x_k) \Rightarrow P(x_1, \ldots x_k)$.

Given the above definition we have that $\vdash_{R'} Q'(t_1, \ldots t_k, s_1, \ldots s_m)$ if and only if $\mathtt{Input}(t_1, \ldots t_k) \vdash_R Q(s_1, \ldots s_m)$. So $\mathtt{Input}(t_1, \ldots t_k) \vdash_R \mathtt{Accept}$ if and only if $\vdash_{R'} P(t_1, \ldots t_k)$. It remains only to show that R' is local. Suppose that $\Sigma \vdash_{R'} \Phi$. We must show that $\Sigma \Vdash_{R'} \Phi$. Let $t_1, \ldots t_k$ be the first k arguments in Φ. Every inference based on R' involves formulas which all have the same first k arguments. Given that $\Sigma \vdash_{R'} \Phi$ we must have that $\Sigma' \vdash_{R'} \Phi$ where Σ' is the set of formulas in Σ that have $t_1, \ldots t_k$ as their first k arguments. Let Σ'' be and Φ' be the result of replacing each formula $Q'(t_1, \ldots t_k, s_1, \ldots, s_m)$ by $Q(s_1, \ldots, s_m)$. Since $\Sigma' \vdash_{R'} \Phi$ we must have $\{\mathtt{Input}(t_1, \ldots t_k)\} \cup \Sigma'' \vdash_R \Phi'$. But since R is superficial this implies that every term in the derivation underlying $\{\mathtt{Input}(t_1, \ldots t_k)\} \cup \Sigma'' \vdash_R \Phi'$ either appears in some t_i or appears in Σ''. This implies that every term in the derivation appears in either Σ' or Φ. This implies $\Sigma \Vdash_{R'} \Phi$.

We can now reduce the halting problem to the problem of determining locality. Let T be a specification of a Turing machine. Let $\mathcal{P}_T$ be the predicate on first order terms which is true of a term c just in case c is a representation of a computation history of T which ends in a halt state. $\mathcal{P}_T$ is a polynomial time predicate on first order terms. Furthermore, a polynomial time Turing machine for $\mathcal{P}_T$ can be computed from the representation of T. By the above construction, the Turing machine for $\mathcal{P}_T$ can be converted to a local rule set R such that $\vdash_R P_T(c)$ if and only if $\mathcal{P}_T(c)$. Now let R' be R plus the single clause $P_T(x) \Rightarrow \mathtt{Halts}$ where $\mathtt{Halts}$ is a new proposition symbol. We claim that R' is local if and only if T does not halt. First note that if T halts then we have $\vdash_{R'} \mathtt{Halts}$ but $\not\Vdash_{R'} \mathtt{Halts}$. Conversely, suppose that T does not halt and suppose that $\Sigma \vdash_{R'} \Phi$. We must show that $\Sigma \Vdash_{R'} \Phi$. Suppose Φ is some formula other than $\mathtt{Halts}$. In this case $\Sigma \vdash_{R'} \Phi$ is equivalent to $\Sigma \vdash_R \Phi$. Since R is local we must have $\Sigma \Vdash_R \Phi$ and thus $\Sigma \vdash\!\!\dashv_{R'} \Phi$. Now suppose Φ is the formula `Halts`. Since $\Sigma \vdash_{R'}$ `Halts` we must have $\Sigma \vdash_{R'} P_T(c)$ for some term c. To show $\Sigma \vdash\!\!\dashv_{R'}$ *Halts* it now suffices to show that c is mentioned in Σ. By the preceding argument we then have $\Sigma \vdash\!\!\dashv_{R} P_T(c)$. Since the rule set R was generated by the construction given above, we have that every inference based on a clause in R is such that that every formula in the inference has the same first argument. This implies that $\Sigma' \vdash\!\!\dashv_{R} P_T(c)$ where Σ' is the subset of formulas in Σ that have c as a first argument. We have assumed that T does not halt, and thus $\not\vdash_R P_T(c)$. Hence Σ' must not be empty. So Σ must mention c and since $\Sigma \vdash\!\!\dashv_{R} P_T(c)$ we have $\Sigma \vdash\!\!\dashv_{R'}$ `Halts`.

The use of Papadimitriou's result can be eliminated from the proof that locality is undecidable. We could have started by giving a direct proof that one can not in general determine for an arbitrary superficial Horn set R whether there exists a c such that `Input`$(c) \vdash_R$ `Accept`. We could then translate an arbitrary superficial Horn set R to a Horn set R' such that R' is local if and only if there does not exist a c such that `Input`$(c) \vdash_R$ `Accept`. This can be done using the construction given above.

6 Discussion

Local rule sets provide a characterization of the set of relations decidable in polynomial time. An understanding of local rule sets can lead to streamlined proofs of polynomial time decidability. For example, consider the fact that any context free language can be recognized in cubic time. This fact is easily proven by giving a translation of grammars into local rule sets. Consider a grammar with productions of the form $D \rightarrow d$ and $A \rightarrow BC$ where A, B, C, and D are nonterminal symbols and d is a terminal symbol. One can translate these productions to Horn clauses of the form `Terminal`$(d,\ i,\ j) \Rightarrow D(i,\ j)$ and $B(i,\ j) \wedge C(j,\ k) \Rightarrow A(i,\ k)$. One can then add the clause `Terminal`$(x,$ `cons`$(x,\ y),\ y)$. One can readily verify that this set of clauses is local and that a string x (represented as a list) parses as nonterminal A if and only if $\vdash A(x,$ `nil`$)$. Hence context free parsing is polynomial time. General methods for analyzing the order of running time of local rule sets can be used to immediately give that these clauses can be run to completion in order n^3 time where n is the length of the input string. We have implemented a compiler for converting local rule sets to efficient inference procedures. This compiler can be used to automatically generate a polynomial time parser from the above inference rules.

In addition to providing a characterization of the complexity class P, and a convenient way of representing some polynomial time algorithms, local inference relations provide a variety of efficient inference techniques that can be applied in situations where complete inference is intractable. The inference rules for set inclusion given in the introduction are incomplete (as rules as about set inclusion) but are useful nonetheless. Efficient but incomplete inference relations can play an important role in pruning search. We hope that progress in understanding efficient inference will lead to better practical algorithms.

In closing we note some open problems. First, although there is a close relationship between the complexity class P and local rule sets, it is not known whether every polynomial time inference relation can be generated by a local rule set. More precisely, let $\vdash$ be any polynomial time inference relation such that $\Phi \vdash \Phi$ for any ground atomic formula Φ and if $\Sigma \vdash \Phi$ then $\Sigma \cup \{\Psi\} \vdash \Phi$. Does there always exist a local rule set which generates $\vdash$, or perhaps a conservative extension of $\vdash$? Our other problems are less precise. Can one find a "natural" rule set that is local but not inductively local? A related question is whether there are useful machine recognizable subclasses of the local rule sets other than the classes of strongly local and inductively local rule sets?

Acknowledgement

We would like to thank Franz Baader for his collaboration in a rediscovery of Papadimitriou's results on superficial Horn sets.

References

[Aho and Ullman, 1979] A. Aho and J. Ullman. Universality of data retrieval languages. In *Proc. of 6th ACM POPL*, pages 110–117, 1979.

[Givan *et al.*, 1991] Robert Givan, David McAllester, and Sameer Shalaby. Natural language based inference procedures applied to schubert's steamroller. In *AAAI-91*, pages 915–920. Morgan Kaufmann Publishers, July 1991.

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

[Jaffar and Lassez, 1987] J. Jaffar and J. L. Lassez. Constraint logic programming. In *Proceedings of POPL-87*, pages 111–119, 1987.

[Knuth, 1975] Donald E. Knuth. Estimating the efficiency of backtrack programs. *Mathematics of Computation*, 29(129):121–136, January 1975.

[Kozen, 1977] Dexter C. Kozen. Complexity of finitely presented algebras. In *Proceedings of the Ninth Annual ACM Symposium on the Theory of Compututation*, pages 164–177, 1977.

[Mackworth, 1977] A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–181, 1977.

[McAllester and Siskind, 1992] David Allen McAllester and Jeffrey Mark Siskind. Lifting transformations. Submitted to LICS92, 1992.

[McAllester, 1989] David A. McAllester. *Ontic: A Knowledge Representation System for Mathematics.* MIT Press, 1989.

[McAllester, 1990] D. McAllester. Automatic recognition of tractability in inference relations. Memo 1215, MIT Artificial Intelligence Laboratory, February 1990. To appear in JACM.

[Papadimitriou, 1985] Christos H. Papadimitriou. A note on the expressive power of prolog. *EATCS Bulletin*, 26:21–23, 1985.

[Pearl and Korf, 1987] Judea Pearl and Richard Korf. Search techniques. *Ann. Rev. Comput. Sci.*, 2:451–467, 1987.

[Shostack, 1978] R. Shostack. An algorithm for reasoning about equality. *Comm. ACM.*, 21(2):583–585, July 1978.

[van Hentenryck, 1989] Pascal van Hentenryck. *Constraint Satisfaction in Logic Programming.* MIT Press, 1989.

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