Fragments of Language

IAN PRATT-HARTMANN

Department of Computer Science,
Manchester University,
Manchester M13 9PL,
UK

ABSTRACT: By a fragment *of a natural language we mean a subset of that language equipped with semantics which translate its sentences into some formal system such as first-order logic. The familiar concepts of* satisfiability *and* entailment *can be defined for any such fragment in a natural way. The question therefore arises, for any given fragment of a natural language, as to the computational complexity of determining satisfiability and entailment within that fragment. We present a series of fragments of English for which the satisfiability problem is polynomial, NP-complete, EXPTIME-complete, NEXPTIME-complete and undecidable. Thus, this paper represents a case study in how to approach the problem of determining the logical complexity of various natural language constructions. In addition, we draw some general conclusions about the relationship between natural language and formal logic.*

KEYWORDS: *Logic, semantics, decidable fragments, computational complexity*

1 Introduction

Let us begin, as logic itself began, with the syllogism. Consider the set of all English sentences subsumed by the following six schemata, where A and B are any common (count) nouns and S is any proper noun:

Every A is a B	No A is a B
Some A is a B	Some A is not a B
S is a B	S is not a B.

For any sentence in this set, we call the constituents A, B and S the *content* of that sentence, and we call the schema it falls under its *form*. Given such a distinction between content and form, a set E of sentences is said to be *satisfiable* if there is some way of varying the semantic contributions made by the content of the sentences in E so as to render every sentence in E simultaneously true. Similarly, a set of sentences E is said to *entail* a sentence e if every way of varying the semantic contributions made by the content of the sentences in $E \cup \{e\}$ renders either e true or some sentence in E false. For a given fragment of English, the problem of determining whether a finite set of sentences in that fragment is satisfiable is referred to as the *satisfiability problem* for that fragment; the problem of determining whether a finite set of sentences in that fragment entails another such sentence is referred to as the *entailment problem*. For fragments closed under negation, the two problems are visibly equivalent.

These are, of course, old problems. Aristotle's *Prior Analytics* proposes—in effect—a solution to the entailment problem for the fragment of English defined above. The solution takes the form of a list of *syllogisms* such as, e.g.

$$\frac{\begin{array}{l}\textsf{Every } A \textsf{ is a } B\\ \textsf{Every } B \textsf{ is a } C\end{array}}{\textsf{Every } A \textsf{ is a } C} \qquad \frac{\begin{array}{l}\textsf{Some } A \textsf{ is a } B\\ \textsf{No } B \textsf{ is a } C\end{array}}{\textsf{Some } A \textsf{ is not a } C}$$

which may be used to construct chains of argument via intermediate conclusions in the familiar way. It is tempting, though probably historically inaccurate, to interpret Chapter A 25 of the *Prior Analytics* as a (not desperately convincing) argument for the completeness of this procedure: if E really does entail e, then it should be possible to derive e from E by a sequence of two-premise inference steps.

Despite the antiquity of the issue, modern developments in mathematical logic, natural language semantics and computational complexity theory afford us a perspective unavailable to the ancients. Three features characterize this new perspective. The first is generality: we view the language of the syllogism as just one of many fragments for which the satisfiability and entailment problems can be posed. The second is the requirement of validation: we demand that any method of determining satisfiability in some fragment of a natural language be accompanied by a formal assurance of its correctness. The third is an interest in computation: one of the key issues regarding any fragment of natural language is the complexity class of the satisfiability and entailment problems for that fragment. The purpose of this paper is to describe some of the technical results that these modern developments have made available, setting them in the broader context of the relationship between natural and formal languages.

2 The syllogistic fragment

Before proceeding, we provide an alternative specification of the language of the syllogism, on lines more amenable to the generalizations we envisage. Consider the following grammar rules.

Syntax	Formal lexicon	Content lexicon
IP $\rightarrow$ NP, I′	Det $\rightarrow$ some	N $\rightarrow$ man
I′ $\rightarrow$ is a N′	Det $\rightarrow$ every	N $\rightarrow$ mortal
I′ $\rightarrow$ is not a N′	Det $\rightarrow$ no	...
NP $\rightarrow$ PropN		PropN $\rightarrow$ Socrates
NP $\rightarrow$ Det, N′		...
N′ $\rightarrow$ N.		

The node labels IP, NP, etc. gesture in the direction of familiar phrasal categories, though of course linguistic orthodoxy has to some extent been sacrificed for simplicity of exposition, particularly in the handling of negation. These rules generate a set of English sentences (phrases of category IP), complete with phrase-structures, via successive expansion of nodes in the usual way. The sentences thus generated are simply those having the forms described by the schemata of Section 1, together with the two additional schemata

$$\text{Every } A \text{ is not a } B \qquad\qquad \text{No } A \text{ is not a } B.$$

Since these are equivalent to No A is a B and Every A is a B, respectively, they do not affect the expressive power of the fragment.

The above grammar rules have been divided into three groups: a *syntax*, consisting of the rules for non-terminal categories, a *formal lexicon*, consisting of the rules for the syncategoremata all, some, no, and a *content lexicon*, consisting of an indefinite number of rules for common and proper nouns. Thus, the syntax and formal lexicon contribute exclusively to *form*, while the content lexicon contributes exclusively to *content*, as defined for this fragment in Section 1. It helps to think of the syntax and formal lexicon as together defining a *family* of fragments of English, each member of which is determined by its content lexicon. We denote this family of English fragments by $\mathcal{E}_0$. To avoid cumbersome formulations and notation, however, we occasionally speak of “the fragment $\mathcal{E}_0$” to refer to the *union* of all these fragments. In practice, no confusion need arise from this abuse of terminology.

The meanings of sentences of $\mathcal{E}_0$ can be provided using the techniques of Montague semantics. The idea is to augment the grammar rules with information specifying the semantic value of each node in the phrase-structure of a sentence. The semantic values of terminal nodes are given directly by the formal lexicon and the content lexicon; the semantic values of nonterminal nodes are computed from the semantic values of their daughters as specified by the syntax. The augmented grammar rules for the syntax and formal lexicon are as follows.

Syntax	Formal lexicon
IP/$\phi(\psi) \to$ NP/ϕ, I$'$/ψ	Det/$\lambda p \lambda q[\exists x(p(x) \wedge q(x))] \to$ some
I$'$/$\phi \to$ is a N$'$/ϕ	Det/$\lambda p \lambda q[\forall x(p(x) \to q(x))] \to$ every
I$'$/$\lambda p \lambda x[\neg p(x)](\phi) \to$ is not a N$'$/ϕ	Det/$\lambda p \lambda q[\forall x(p(x) \to \neg q(x))] \to$ no
NP/$\phi \to$ PropN/ϕ	
NP/$\phi(\psi) \to$ Det/ϕ, N$'$/ψ	
N$'$/$\phi \to$ N/ϕ.	

Here, $\phi(\psi)$ denotes the result of applying the function ϕ to the argument ψ. Thus, the augmented grammar rule for IP states that the meaning of an IP consisting of an NP and an I$'$ is computed by applying the meaning of the NP (as a function) to the meaning of the I$'$. The augmented grammar rules for the content lexicon assign meanings to common and proper nouns according to the following pattern.

Content lexicon	
N/$\lambda x[\mathrm{man}(x)] \to$ man	PropN/$\lambda p[p(\mathrm{socrates})] \to$ Socrates
N/$\lambda x[\mathrm{mortal}(x)] \to$ mortal	...
...	

Every such content lexicon defines a first-order signature, where each common noun corresponds to a unary predicate and each proper noun to an individual constant.

It is straightforward to verify that the semantically augmented grammar rules for $\mathcal{E}_0$ map the six schemata of the previous section into formula schemata of first-order logic as follows.

Every A is a B	$\forall x(a(x) \to b(x))$	No A is a B	$\forall x(a(x) \to \neg b(x))$
Some A is a B	$\exists x(a(x) \wedge b(x))$	Some A is not a B	$\exists x(a(x) \wedge \neg b(x))$
S is a B	$b(s)$	S is not a B	$\neg b(s)$,

where a, b and s are the elements of the signature corresponding to the content lexicon entries A, B and S. In contrast to the methods of traditional logic, we have adopted the now standard non-presuppositional account of universal quantifiers, according to which a sentence Every A is a B is true if no A exist. (Traditional accounts of the syllogism assign the reverse truth-value in this case.) Modulo the issue of presuppositionality, we take it that the above translations faithfully capture the meanings of sentences in $\mathcal{E}_0$ as understood by English speakers. It is not possible formally to validate this assumption, since it relies on a pre-theoretic notion of meaning for $\mathcal{E}_0$-sentences. In practice, however, it does not appear open to serious doubt.

The above semantics reduce the satisfiability and entailment problems for $\mathcal{E}_0$, as defined in Section 1, to their counterparts in first-order logic. In particular, if $E \subseteq \mathcal{E}_0$, and Φ is the corresponding set of first-order formulas, then E is satisfiable in the sense of Section 1 if and only if Φ is satisfiable in the usual sense of first-order logic. This reduction rests on the correspondence between content lexicon and first-order signature. More generally, the notions of satisfiability and entailment in fragments of English defined in Section 1 rely crucially on a given—and indeed perhaps arbitrary—separation of content and form. For $\mathcal{E}_0$, this separation was provided by the notion of a distinguished content lexicon. We will maintain this separation in all the fragments of English considered below.

In the sequel, if X is any expression or set of expressions (either in a natural or a formal language), we take the *size* of X, denoted $|X|$, to be the number of atomic symbol (tokens) occurring in X. For natural languages, the atomic symbols are the lexemes (and perhaps morphemes); for formal languages, the atomic symbols are the logical connectives, the variables and the members of the signature. Using this notion of size, we can formulate questions about computational complexity in the usual way. Thus, the complexity of the satisfiability problem for some English fragment $\mathcal{E}$ is the number of steps of computation required to determine algorithmically whether a given finite set of sentences $E \subset \mathcal{E}$ is satisfiable, expressed as a function of $|E|$.

Our first result states that satisfiability in the fragment $\mathcal{E}_0$ is computationally tractable.

THEOREM 2.1
The problem of determining the satisfiability of a set of sentences in $\mathcal{E}_0$ is in PTIME.

PROOF. Let E be a finite set of sentences of $\mathcal{E}_0$, and let Φ be their translations into first-order logic. It is obvious that Φ can be computed in linear time. Since we have argued that Φ is satisfiable if and only if E is satisfiable, it suffices to show that the satisfiability of Φ can be determined in polynomial time.

Replace all formulas of the form $\exists x \phi(x)$ in Φ by the corresponding formulas $\phi(s)$, where the s are new individual constants. The resulting set of formulas Φ' is then satisfiable if and only if Φ is satisfiable, and will involve a signature whose size is bounded by $|E|$. Now re-write Φ' as a set of clauses (disjunctions of literals). After factoring and eliminating tautologies, all these clauses will be of the forms $\neg a(x) \vee b(x)$, $\neg a(x) \vee \neg b(x)$, $\neg a(x)$, $a(s)$ or $\neg a(s)$. It is easy to see that resolution applied to such clauses will produce only more clauses of this form. The number of such clauses is quadratic in the size of the signature of Φ'. Hence the resolution procedure will either produce a contradiction or will reach saturation after $O(|E|^2)$ steps. ∎

This simple result suggests a programme of work: take a fragment of English delineated in terms which respect the syntax of the language; then determine the computational complexity of deciding satisfiability in that fragment, if, indeed, the fragment is decidable. From this standpoint, the syllogistic can be regarded as just one such fragment, with very restricted syntax, and a correspondingly efficient decision procedure. In the sequel, we shall investigate what happens as we expand our syntactic horizons.

3 Relative Clauses

One way to generalize the fragment $\mathcal{E}_0$ is to add relative clauses, thus accommodating arguments such as:

$$\begin{array}{l} \textsf{Every philosopher who is not a stoic is a cynic} \\ \textsf{Every stoic is a man} \\ \underline{\textsf{Every cynic is a man}} \\ \textsf{Every philosopher is a man.} \end{array}$$

In this section we investigate the computational consequences of this generalization.

Our approach to the semantics of relative clauses loosely follows that of Heim and Kratzer [9]. Let $\mathcal{E}_1$ be the fragment defined by the grammar rules for $\mathcal{E}_0$ together with the following additional grammar rules and formal lexicon rules.

Syntax	Formal lexicon
$\mathrm{N}'/\phi(\psi) \to \mathrm{N}/\psi, \mathrm{CP}/\phi$	$\mathrm{RelPro}/\lambda q \lambda p \lambda x[p(x) \wedge q(x)] \to \textsf{who}, \textsf{which}$
$\mathrm{CP}/\phi(\psi) \to \mathrm{CSpec}_t/\phi, \mathrm{C}'_t/\psi$	$\mathrm{C} \to$
$\mathrm{C}'_t/\lambda t[\phi] \to \mathrm{C}, \mathrm{IP}/\phi$	
$\mathrm{NP}/\phi \to \mathrm{RelPro}/\phi$	
$\mathrm{CSpec}_t \to$	

In addition, we assume that, following generation of an IP by these rules, relative pronouns are subject to wh-movement to produce the observed word-order. For our purposes, we may take the wh-movement rule to require: (i) the empty position CSpec_t must be filled by movement of a RelPro from within the IP which forms its right-sister (i.e. which it C-commands); (ii) every RelPro must move to some such CSpec_t position; (iii) every RelPro moving to CSpec_t leaves behind a trace t, which contributes the semantic value $\lambda p[p(t)]$. The semantic information with which these rules are augmented can then be understood as for $\mathcal{E}_0$, with meanings computed after wh-movement. Figure 1 illustrates the structure of the first sentence in the above argument, with the arrow representing wh-movement in the obvious way.

For the sake of clarity, we have ignored the issue of agreement of relative pronouns with their antecedents—animate or inanimate. This detail aside, we claim that the above rules result in intuitively correct meanings for $\mathcal{E}_1$-sentences. This claim may be verified by hand-checking or (better) by direct implementation as a Prolog program. Thus, for example, the above argument translates to:

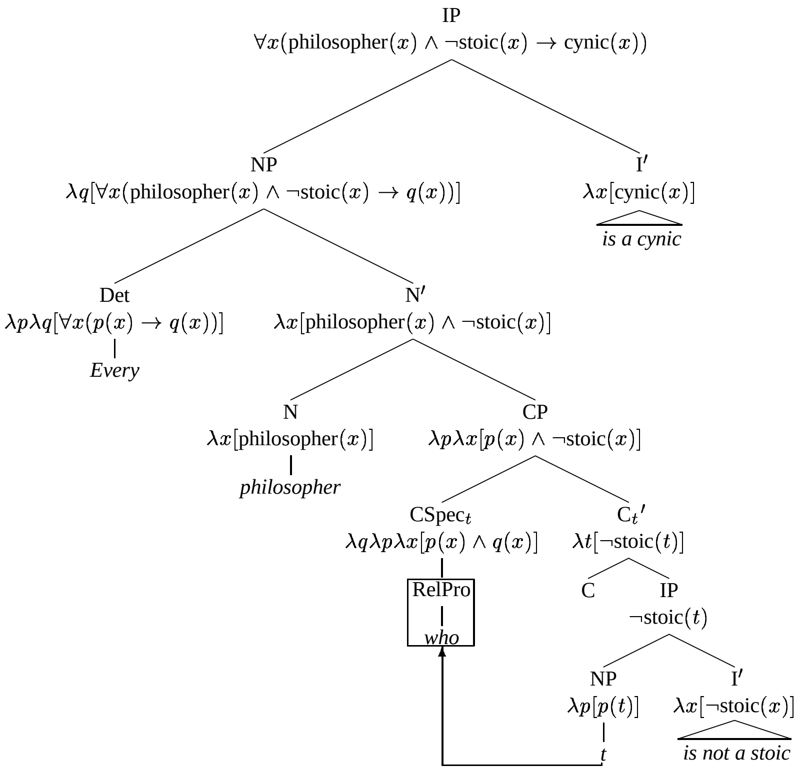

FIG. 1. Typical phrase-structure in the fragment $\mathcal{E}_1$

$$\begin{array}{l}
\forall x(\text{philosopher}(x) \wedge \neg\text{stoic}(x) \rightarrow \text{cynic}(x)) \\
\forall x(\text{stoic}(x) \rightarrow \text{man}(x)) \\
\forall x(\text{cynic}(x) \rightarrow \text{man}(x)) \\
\hline
\forall x(\text{philosopher}(x) \rightarrow \text{man}(x)).
\end{array}$$

As with $\mathcal{E}_0$, so too with $\mathcal{E}_1$, it is not possible formally to validate the proposed semantics; again, however, the fragment in question is so simple that they are not open to reasonable doubt. It follows that a set of $\mathcal{E}_1$-sentences is satisfiable in the sense of Section 1 if and only if its translation is satisfiable in the usual sense of first-order logic.

We have the following result:

THEOREM 3.1
The problem of determining the satisfiability of a set of sentences in $\mathcal{E}_1$ is NP-complete.

PROOF. To show membership in NP, let E be a finite set of sentences of $\mathcal{E}_1$, let Φ

be their translations into first-order logic, and let Φ' be the result of replacing all existential quantifiers in Φ with new constants. Clearly, Φ' can be computed in linear time, and is satisfiable if and only if E is satisfiable. Thus, it suffices to show that, if Φ' is satisfiable, then it has a model of size bounded by $|\Phi'|$. But this is obvious since Φ is only universally quantified, and hence its models are closed under taking substructures.

To show NP-hardness, we reduce 3SAT to the problem of determining satisfiability in $\mathcal{E}_1$. Let C be a set of propositional clauses, each of which has at most three literals. Without loss of generality, we may assume all the clauses in C to be of the forms $p \vee q$, $\neg p \vee \neg q$ or $\neg p \vee \neg q \vee r$. We then map each clause in C to a sentence of $\mathcal{E}_1$ as follows:

Clause	$\mathcal{E}_1$-sentence
$p \vee q$	Every element which is not a q is a p
$\neg p \vee \neg q$	No p is a q
$\neg p \vee \neg q \vee r$	Every p which is a q is an r,

and finally add the $\mathcal{E}_1$-sentence Some element is an element. Let the resulting set of $\mathcal{E}_1$-sentences be E. These sentences translate to first-order logic according to the following table:

$\mathcal{E}_1$-sentence	Formula
Every element which is not a q is a p	$\forall x(\text{element}(x) \wedge \neg q(x) \rightarrow p(x))$
No p is a q	$\forall x(p(x) \rightarrow \neg q(x))$
Every p which is a q is an r	$\forall x(p(x) \wedge q(x) \rightarrow r(x))$
Some element is an element	$\exists x(\text{element}(x) \wedge \text{element}(x))$.

Let Φ be the set of formulas thus obtained. Thus E is satisfiable if and only if Φ is satisfiable. But it is routine to transform any satisfying assignment for C into a model of Φ and vice versa. This completes the reduction of 3-SAT to $\mathcal{E}_1$-satisfiability. ∎

One question that sometimes arises in discussions of traditional logic, and indeed, in exegesis of logical works from earlier epochs, is whether certain arguments can, by careful massaging, be accommodated within the syllogistic framework. For example, one might wonder whether arguments expressed in $\mathcal{E}_1$ have this property. Theorems 2.1 and 3.1 provide strong evidence that they do not. Unless $\text{P} = \text{NP}$, satisfiability in the fragment $\mathcal{E}_1$ cannot tractably be reduced to satisfiability in the fragment $\mathcal{E}_0$.

4 Non-copula verbs

Despite its greater computational complexity, $\mathcal{E}_1$ is still representationally impoverished: it is in no better position than $\mathcal{E}_0$ to satisfy Augustus de Morgan's famous demand to account for the evident validity of the argument

$$\frac{\text{Every horse is an animal}}{\text{Every horse's head is an animal's head.}}$$

Let us see what happens when we add relations to our fragment.

Let $\mathcal{E}_2$ be the fragment defined by the grammar rules for $\mathcal{E}_1$ together with the following additional grammar rules.

Syntax	Content Lexicon
$\mathrm{I}'/\phi \to \mathrm{VP}/\phi$	$\mathrm{V}/\lambda s \lambda x[s(\lambda y[\mathrm{admire}(x,y)])] \to$ admires
$\mathrm{VP}/\phi(\psi) \to \mathrm{V}/\phi,\ \mathrm{NP}/\psi$	...

The wh-movement rule is carried over from $\mathcal{E}_1$. Note that this rule allows any NP in a relative clause—in either subject or object position—to fill the appropriate CSpec.

The fragment $\mathcal{E}_2$ includes the sentences in the following (lightly transcribed) version of the de Morgan argument.

$$\frac{\text{Every horse is an animal}}{\text{Every head which some horse has is a head which some animal has.}}$$

The reader may verify that the above rules translate this argument as follows.

$$\frac{\forall x(\mathrm{horse}(x) \to \mathrm{animal}(x))}{\forall x(\mathrm{head}(x) \wedge \exists y(\mathrm{horse}(y) \wedge \mathrm{has}(y,x)) \to \mathrm{head}(x) \wedge \exists y(\mathrm{animal}(y) \wedge \mathrm{has}(y,x))).}$$

We claim that our semantics correctly capture the meanings of $\mathcal{E}_2$-sentences. Again, therefore, a set of $\mathcal{E}_2$-sentences is satisfiable in the sense of Section 1 if and only if its translation into first-order logic is satisfiable in the usual sense of first-order logic.

The reader may be wondering why our fragment $\mathcal{E}_2$ does not allow VPs to be directly negated (though their NP complements may contain the negative determiner no). The reason for this restriction is simply to avoid complications of a purely linguistic nature concerning quantifier scoping and negative polarity determiners, for example in sentences such as

Every/no farmer does not own some/any/a/every horse.

In fact, adding full negation for non-copula verbs would not affect the computational complexity of the satisfiability problem for $\mathcal{E}_2$, which is established in the following theorem. To shorten the proof, we have also ignored proper nouns altogether; the reader may easily verify that this feature of $\mathcal{E}_2$ does not affect its computational complexity either.

THEOREM 4.1
The problem of determining the satisfiability of a set of sentences in $\mathcal{E}_2$ is EXPTIME-complete.

PROOF. To show membership in EXPTIME, let E be a finite set of sentences of $\mathcal{E}_2$ and let Φ be the set of their translations into first-order logic. Define an *$\mathcal{E}_2$-formula* recursively as follows.

1. If a is a unary predicate and x a variable, then $a(x)$ is an $\mathcal{E}_2$-formula.
2. If a is a unary predicate, R a binary predicate, x, y variables and $\pi(x)$ an $\mathcal{E}_2$-formula, then

$$\begin{array}{ll} a(x) \wedge \pi(x) & \neg\pi(x) \\ \exists y(\pi(y) \wedge R(x,y)) & \exists y(\pi(y) \wedge R(y,x)) \\ \forall y(\pi(y) \rightarrow R(x,y)) & \forall y(\pi(y) \rightarrow R(y,x)) \end{array}$$

   are $\mathcal{E}_2$-formulas.

A simple induction on the phrase-structures of $\mathcal{E}_2$-sentences shows that every N$'$ contributes a meaning of the form $\lambda x[\psi(x)]$, where ψ is (modulo trivial logical manipulations) an $\mathcal{E}_2$-formula. It follows that, by moving negations inwards and introducing new unary predicate letters for subformulas, we can transform Φ into an equisatisfiable set Φ' of formulas of the forms

$$\begin{array}{ll} \exists x(p(x) \wedge q(x)) & \exists x(p(x) \wedge \neg q(x)) \\ \forall x(p(x) \rightarrow q(x)) & \forall x(p(x) \rightarrow \neg q(x)) \\ \forall x(\neg p(x) \rightarrow q(x)) & \forall x(p(x) \rightarrow (q(x) \vee r(x))) \\ \forall x(p(x) \rightarrow \exists y(q(y) \wedge R(x,y))) & \forall x(p(x) \rightarrow \exists y(q(y) \wedge R(y,x))) \\ \forall x(p(x) \rightarrow \exists y(q(y) \wedge \neg R(x,y))) & \forall x(p(x) \rightarrow \exists y(q(y) \wedge \neg R(y,x))) \\ \forall x(p(x) \rightarrow \forall y(q(y) \rightarrow R(x,y))) & \forall x(p(x) \rightarrow \forall y(q(y) \rightarrow R(y,x))) \\ \forall x(p(x) \rightarrow \forall y(q(y) \rightarrow \neg R(x,y))) & \forall x(p(x) \rightarrow \forall y(q(y) \rightarrow \neg R(y,x))), \end{array}$$

where p, q and r are unary predicates and R is a binary predicate. Since Φ' can be computed in polynomial time, it suffices to show that the satisfiability of Φ' can be decided in EXPTIME.

Suppose Φ' is converted into a set of clauses Γ in the usual way. The key observation is that every clause in Γ contains at most one occurrence of a binary predicate. Consider the partial order on the set of atomic formulas defined by declaring every atom involving a binary predicate to be greater than every atom involving a unary predicate. Since this ordering is liftable, resolution under the restriction that only maximal literals in clauses may be resolved upon is complete.

Suppose we now saturate Γ under resolution on atoms involving binary predicates. Since each clause in Γ contains at most one binary predicate, this step can be computed in quadratic time. Clauses containing any binary predicates cannot now take any further part in ordered resolution, and so may be discarded. The result will be a set of clauses Γ', such that: (i) Γ has a model if and only if Γ' has; (ii) Γ' features only unary (not binary) predicates and only unary function-symbols; and (iii) $|\Gamma'|$ is bounded by a polynomial function of $|\Gamma|$.

By (i), Φ' is satisfiable if and only if Γ' has a model. By (ii) we can apply the splitting rule to every clause in Γ' to obtain clauses involving only one variable. (The splitting rule allows us to replace a clause $\pi \vee \psi$ nondeterministically by π or ψ provided that

π and ψ have no variables in common.) By (iii) the number of backtracking choices generated by the splitting rule is at most exponential in $|\Phi'|$. For each choice of how to split clauses, the subsequent resolution procedure is confined to clauses in one variable with bounded functional depth, and can easily be seen to reach saturation after at most exponentially many steps.

To show EXPTIME-hardness, recall that the logic K^U is the modal logic K together with an additional universal modality, whose semantics are given by

$$\models_w U\phi \text{ if and only if } \models_{w'} \phi \text{ for all worlds } w'.$$

The satisfiability problem for K^U is EXPTIME-hard. (The proof is an easy adaptation of the corresponding result for propositional dynamic logic: see, e.g. Harel *et al.* [8], pp. 216 ff.) It suffices, therefore, to reduce this problem to satisfiability in $\mathcal{E}_2$. Let ϕ be a formula of K^U. For convenience, we take V to be the dual modality to U. For every proper or improper subformula ψ of ϕ, let A_ψ be a noun. Let Es and Rs be verbs. Now define, for each such ψ a set of formulas $T_\psi \subset \mathcal{E}_2$ inductively as follows:

$$\begin{aligned}
T_p &= \emptyset \text{ if } p \text{ is atomic}\\
T_{\psi\wedge\pi} &= T_\psi \cup T_\pi \cup \{\text{Every } \mathsf{A}_\psi \text{ which is an } \mathsf{A}_\pi \text{ is an } \mathsf{A}_{\psi\wedge\pi},\\
&\qquad\qquad\qquad \text{Every } \mathsf{A}_{\psi\wedge\pi} \text{ is an } \mathsf{A}_\psi, \text{Every } \mathsf{A}_{\psi\wedge\pi} \text{ is an } \mathsf{A}_\pi\}\\
T_{\neg\psi} &= T_\psi \cup \{\text{Every element which is not an } \mathsf{A}_\psi \text{ is an } \mathsf{A}_{\neg\psi},\\
&\qquad\qquad \text{No } \mathsf{A}_\psi \text{ is an } \mathsf{A}_{\neg\psi}\}\\
T_{\diamond\psi} &= T_\psi \cup \{\text{Every element which Rs some } \mathsf{A}_\psi \text{ is an } \mathsf{A}_{\diamond\psi},\\
&\qquad\qquad \text{Every } \mathsf{A}_{\diamond\psi} \text{ Rs some } \mathsf{A}_\psi\}\\
T_{V\phi} &= T_\psi \cup \{\text{Every element which Es some } \mathsf{A}_\psi \text{ is an } \mathsf{A}_{V\psi},\\
&\qquad\qquad \text{Every } \mathsf{A}_{V\psi} \text{ Es some } \mathsf{A}_\psi\}.
\end{aligned}$$

Now let S_ϕ be the collection of $\mathcal{E}_2$-sentences

$$\begin{aligned}
&\{\text{Every } \mathsf{A}_\psi \text{ is an element} \mid \psi \text{ a subformula of } \phi\}\cup\\
&\{\text{Some } \mathsf{A}_\phi \text{ is an } \mathsf{A}_\phi, \text{ Every element Es every element}\}.
\end{aligned}$$

It is routine to show that ϕ is satisfiable if and only if $T_\phi \cup S_\phi$ is satisfiable. ■

5 Anaphora

There are still many simple arguments that cannot be captured by the fragment $\mathcal{E}_2$. Here is one:

Every artist despises some bureaucrat
Every bureaucrat admires every artist who despises him

Every artist despises some bureaucrat who admires him.

(We assume that the pronouns above are resolved intrasententially.) So the next question is what happens to the computational complexity of the satisfiability problem when pronouns (he, him) and reflexives (himself) are admitted.

Let the fragment $\mathcal{E}_3$ be defined by adding the following grammar rules to those defining $\mathcal{E}_2$

Syntax	Formal lexicon
NP $\rightarrow$ Reflexive	Reflexive $\rightarrow$ itself (him/herself)
NP $\rightarrow$ Pronoun	Pronoun $\rightarrow$ it (he/she/him/her)
I$'$ $\rightarrow$ NegP, VP	NegP $\rightarrow$ does not.

For technical reasons, we have added negation for non-copula verbs as well. (The issue of verb-inflections in such sentences has been ignored, however.) To avoid problems of a purely linguistic nature concerning quantifier scoping and negative polarity determiners, we insist that the VP in the rule I$'$ $\rightarrow$ NegP, VP contain no determiner at all after wh-movement. (Hence it contains a proper noun, pronoun, reflexive or wh-trace.) This limited form of negation is all that is required to obtain the complexity results below.

The content lexicon (for nouns and verbs) and the wh-movement rule are carried over from $\mathcal{E}_2$. Furthermore, we take pronouns and reflexives, which are assumed always to have antecedents in the sentences in which they occur, to be subject to the usual rules of binding theory, and in addition to obey a further restriction explained below. We shall not rehearse binding theory here, referring the reader instead to a standard text, such as Cowper [3], p. 171. For present purposes, we can use our linguistic intuitions to determine which NPs a given reflexive or pronoun can take as antecedent. We also forego a formal account of the semantics for $\mathcal{E}_3$, in order not to be detained by the technicalities of handling bound-variable anaphora within the framework of Montague semantics. A full semantic analysis of $\mathcal{E}_3$ (with some inessential variations) is given in Pratt-Hartmann [14]; for an approachable general account of bound-variable anaphora and Montague semantics, see Heim and Kratzer [9], Chh. 9, 11. Accordingly, we shall simply assume in the present paper some mechanism for producing faithful first-order translations of sentences of $\mathcal{E}_3$ along the lines outlined for the other fragments considered above. For example, we expect the above argument to be translated as follows:

$$\begin{array}{l}\forall x(\text{artist}(x) \rightarrow \exists y(\text{bureaucrat}(y) \wedge \text{despises}(x,y)))\\ \underline{\forall x(\text{bureaucrat}(x) \rightarrow \forall y(\text{artist}(y) \wedge \text{despises}(y,x) \rightarrow \text{admires}(x,y)))}\\ \forall x(\text{artist}(x) \rightarrow \exists y(\text{bureaucrat}(y) \wedge \text{admires}(y,x) \wedge \text{despises}(x,y))).\end{array}$$

We now come to the additional restriction on pronoun resolution mentioned above. By way of introduction, consider the sentence

Every artist who employs a caretaker despises every bureaucrat who admires himself.

The phrase-structure of this sentence is shown in Figure 2; its translation into first-order logic is

$$\forall x_1(\text{artist}(x_1) \wedge \exists x_2(\text{caretaker}(x_2) \wedge \text{employ}(x_1, x_2)) \rightarrow \\ \forall x_3(\text{bureaucrat}(x_3) \wedge \text{admire}(x_3, x_3) \rightarrow \text{despise}(x_1, x_3))).$$

That the arguments of the predicate admire in this formula are identical is due to the use of the reflexive himself, which, according to binding theory, must coindex with the NP headed by bureaucrat. As we see from Figure 2, this NP is the closest NP to the reflexive himself in the phrase-structure of the sentence.

By contrast, the sentence

Every artist who employs a caretaker despises every bureaucrat who admires him

exhibits an anaphoric ambiguity, according to whether the antecedent of the pronoun him is the NP headed by artist or the NP headed by caretaker. (The NP headed by bureaucrat is not available as a pronoun antecedent here.) The translations of these two readings into first-order logic, are, respectively,

$$\forall x_1(\text{artist}(x_1) \wedge \exists x_2(\text{caretaker}(x_2) \wedge \text{employ}(x_1, x_2)) \rightarrow \\ \forall x_3(\text{bureaucrat}(x_3) \wedge \text{admire}(x_3, x_1) \rightarrow \text{despise}(x_1, x_3)))$$

and

$$\forall x_1 \forall x_2(\text{artist}(x_1) \wedge \text{caretaker}(x_2) \wedge \text{employ}(x_1, x_2) \rightarrow \\ \forall x_3(\text{bureaucrat}(x_3) \wedge \text{admire}(x_3, x_2) \rightarrow \text{despise}(x_1, x_3))).$$

We see from Figure 2 that the NP headed by artist is closer to the pronoun, as measured along the edges of the phrase-structure, than is the NP headed by caretaker. Given that the NP headed by bureaucrat is disallowed as an antecedent to the pronoun in this sentence, the first reading is thus the one in which the pronoun takes its *closest allowed* antecedent.

We notice something else about this first reading. Consider again its translation into first-order logic. Although there are three variables in the formula, corresponding to the three nouns in the sentence, the variables x_2 and x_3 never occur free in the same subformula. Hence, we can replace x_3 by x_2, to get the equivalent formula:

$$\forall x_1(\text{artist}(x_1) \wedge \exists x_2(\text{caretaker}(x_2) \wedge \text{employ}(x_1, x_2)) \rightarrow \\ \forall x_2(\text{bureaucrat}(x_2) \wedge \text{admire}(x_2, x_1) \rightarrow \text{despise}(x_1, x_2))).$$

The reader is invited to verify that no such move is available for the second reading. This observation generalizes: it is shown in Pratt-Hartmann [14], Theorem 1, that every sentence in $\mathcal{E}_3$ translates to a formula which may be written with exactly two variables. (The details of the proof are somewhat tedious.)

Despite this limitation, it should come as no surprise that $\mathcal{E}_3$ is more expressive than $\mathcal{E}_2$. Table 1 lists some $\mathcal{E}_3$-sentences and the formulas they translate to. In giving these translations, we have supressed reference to the unary predicate corresponding

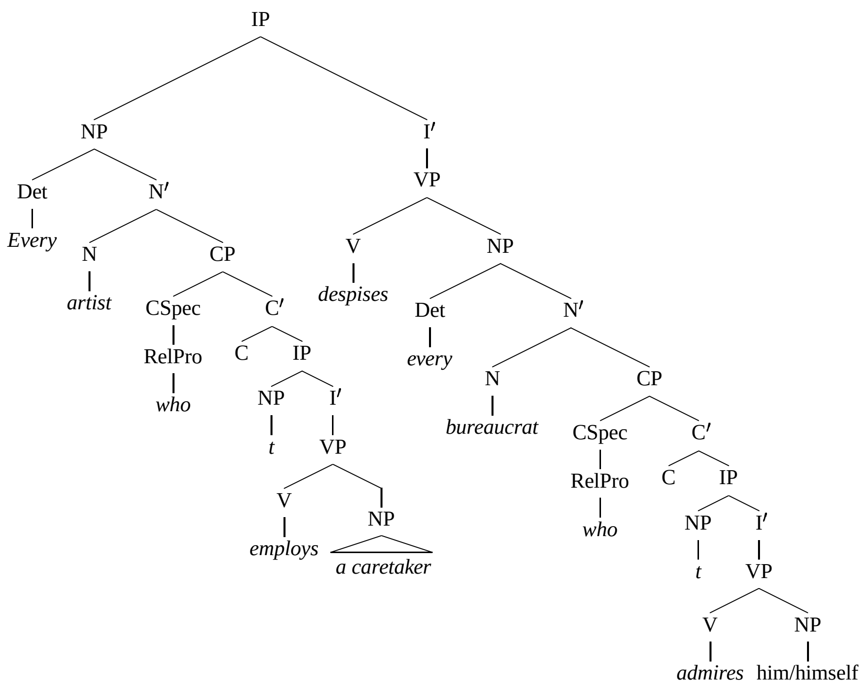

FIG. 2. Typical phrase-structure in the fragment $\mathcal{E}_3$

Every thing bees every thing which ays it	$\forall x \forall y(a(x,y) \rightarrow b(y,x))$
Every thing which ays some thing bees itself	$\forall x(\exists y a(x,y) \rightarrow b(x,x))$
Every thing which some thing ays bees itself	$\forall y(\exists x a(x,y) \rightarrow b(y,y))$
Every thing which bees itself ays every thing	$\forall x(b(x,x) \rightarrow \forall y a(x,y))$
Every thing ays every thing which bees itself	$\forall x(b(x,x) \rightarrow \forall y a(y,x))$
Nothing bees some thing which it ays	$\forall x \forall y(a(x,y) \rightarrow \neg b(x,y))$
Every thing bees every thing which it ays	$\forall x \forall y(a(x,y) \rightarrow b(x,y))$
Every thing ays every thing	$\forall x \forall y a(x,y)$
Every thing ays some thing	$\forall x \exists y a(x,y)$
Every thing ays every thing which it does not bee	$\forall x \forall y(\neg b(x,y) \rightarrow a(x,y))$
Every thing which bees some thing which it cees ays it	$\forall x \forall y((b(x,y) \wedge c(x,y)) \rightarrow a(x,y))$.

TABLE 1. Expressiveness of $\mathcal{E}_3$

to the noun thing, since all quantification is restricted to its extension anyway. Notice the use of 'donkey-anaphora' in the last row of this table. This type of anaphora is permitted in $\mathcal{E}_3$, subject of course to the restriction that pronouns always take their closest allowed antecedent. In fact, these exampes suffice to establish the following theorem:

THEOREM 5.1
The problem of determining the satisfiability of a set of sentences in $\mathcal{E}_3$ is NEXPTIME-complete.

PROOF. It is well-known that satisfiability in the two-variable fragment of first-order logic is NEXPTIME-complete (see, e.g. Börger, Gurevich and Grädel [2], Ch. 8.1). Given the result that every $\mathcal{E}_3$-sentence translates into this fragment, membership in NEXPTIME is immediate. For the hardness result, we may use the standard normal form re-writing techniques to replace any formula ϕ in the two-variable fragment of first-order logic by an equisatisfiable conjunction of formulas of the forms appearing in the right-hand column of Table 1. (We may assume without loss of generality that ϕ involves only binary predicates.) The corresponding set of $\mathcal{E}_3$-sentences will then be satisfiable if and only if ϕ is satisfiable. ■

Finally, we consider what happens when we relax the artificial restriction on pronoun interpretation in $\mathcal{L}_3$, allowing a pronoun to take any allowed antecedent in the sentence in which it occurs. As we have seen, this relaxation results in ambiguous sentences, for example

Every artist who employs a caretaker despises every bureaucrat who admires him.

In order to eliminate this ambiguity, we suppose such sentences to come complete with (allowable) indexation patterns indicating the antecedents of any pronouns. For instance, the above sentence would be replaced by the two sentences:

Every artist$_i$ who employs a caretaker$_j$ despises every bureaucrat$_k$ who admires him$_i$

Every artist$_i$ who employs a caretaker$_j$ despises every bureaucrat$_k$ who admires him$_j$.

Let $\mathcal{L}_4$ be the fragment of ‘English’ thus obtained.

We can now formulate the satisfiability question for $\mathcal{E}_4$ as before, and ask what its complexity is. Given that the three-variable fragment of first-order logic is undecidable, it is no surprise that the same fate befalls $\mathcal{E}_4$.

THEOREM 5.2
The problem of determining the satisfiability of a set of sentences in $\mathcal{E}_4$ is undecidable.

Actually, a slightly stronger result is shown in Pratt-Hartmann [13], Corollary 1: the entailment problem for the positive fragment of $\mathcal{E}_4$ (no negation or negative determiners) is also undecidable. Again, the details are tedious, and we omit them here.

6 Conclusions

The technical content of this paper is easily summarized. The following table lists the five English fragments we have introduced, briefly describes their distinguishing features, and gives the complexity class of the corresponding satisfiability problem.

Fragment	Distinguishing features	Computational complexity
$\mathcal{E}_0$	Syllogism	PTIME
$\mathcal{E}_1$	Relative clauses	NP-complete
$\mathcal{E}_2$	Non-copula verbs	EXPTIME-complete
$\mathcal{E}_3$	Restricted anaphora	NEXPTIME-complete
$\mathcal{E}_4$	Unrestricted anaphora	undecidable.

So one could go on. There are many more fragments of English we could have analysed, of all levels of complexity. The utility of such an analysis from the point of view of natural language engineering, at least in principle, should be obvious. Moreover, the techniques required to carry out analyses of other fragments will be, in essence, those used above: a specification of the fragment in terms of syntax rules, a formal semantics mapping the fragment into first-order logic (or some other formal logic), and the deployment of standard methods of computational complexity theory on the resulting fragment of formal logic. It should perhaps be pointed out that we cannot expect the determination of all linguistically salient fragments to be quite as straightforward as those considered here.

More generally, our analysis allows us to view the relationship between traditional logic and mathematical logic in a more conciliatory light than has sometimes been the case (Englebretsen [4], Sommers [16]). It would be wrong to think of the logic of the *Principia Mathematica* as being so pitilessly superior to that of the *Prior Analytics* that we can simply forget about the latter. Yes, first-order logic is more expressive than the language of the syllogism; but expressiveness is a double-edged sword, because it correlates, loosely at least, with computational complexity. Indeed, the very recent history of logic, especially within Computer Science, is dominated by the search for logics of limited expressive power whose satisfiability problems are decidable. The kinds of fragments which have drawn most attention, for example various prefix classes (see Börger, Grädel and Gurevich [2] for a survey), the guarded fragment (Andréka, van Benthem and Németi [1], Grädel [6]) and the two-variable fragment (Mortimer [12], Grädel and Otto [7]), owe their salience to purely logic-internal considerations. But there is every reason to consider also those logics arising from fragments of natural languages. The syllogistic is one such logic. And if that logic is too inexpressive to be of much practical use, perhaps its natural generalizations are not. We have presented a selection of such generalizations in this paper.

Finally, the foregoing analysis should help to lay to rest some appealing but ultimately confused ideas concerning the value of natural-language-friendly logic. According to its proponents, we obtain a better (i.e. more efficient) method of assessing the validity of arguments couched in natural language if we reason within a logical calculus whose syntax is closer to that of natural language than is—say—first-order logic. This idea is attractive because it suggests an ecological dictum: treat the syntax of natural language with the respect it is due, and your inference processes will run faster. Writers apparently expressing support for such views include Fitch [5], Hintikka [10], Suppes [17], Purdy [15] and (perhaps) McAllister and Givan [11]. The observations of this paper lend no support to such views, and indeed cast doubt on them. There is no reason, having identified a fragment of a natural language, why satisfiability within that fragment should not be decided by first translating into first-order logic and then using procedures appropriate to the fragment of first-order logic so obtained. Indeed, from a complexity-theoretic point of view, there is every reason to believe that, for all but the most impoverished fragments, reasoning using schemata based on the syntax of natural language will confer no advantage whatever.

References

[1] Hajnal Andréka, Johan van Benthem, and István Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.

[2] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1997.

[3] Elizabeth A. Cowper. *A Consice Introduction to Syntactic Theory*. University of Chicago Press, Chicago, 1992.

[4] George Englebretsen. *Three Logicians*. Van Gorcum, Assen, 1981.

[5] Frederic B. Fitch. Natural deduction rules for English. *Philosophical Studies*, 24:89–104, 1973.

[6] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64:1719–1742, 1999.

[7] Erich Grädel and Martin Otto. On logics with two variables. *Theoretical Computer Science*, 224(1–2):73–113, 1999.

[8] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic Logic*. MIT Press, Cambridge, MA., 2000.

[9] I. Heim and A. Kratzer. *Semantics in Generative Grammar*. Blackwell, Oxford, 1998.

[10] Jaakko Hintikka. Quantifiers vs quantification theory. *Inquiry*, 5:153–77, 1974.

[11] David A. McAllester and Robert Givan. Natural language syntax and first-order inference. *Artificial Intelligence*, 56:1–20, 1992.

[12] M. Mortimer. On languages with two variables. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 21:135–140, 1975.

[13] Ian Pratt-Hartmann. On the semantic complexity of some fragments of English. Technical Report UMCS-00-5-1, University of Manchester Department of Computer Science, Manchester, 2000.

[14] Ian Pratt-Hartmann. A two-variable fragment of English. *Journal of Logic, Language and Information*, forthcoming.

[15] William C. Purdy. A logic for natural language. *Notre Dame Journal of Formal Logic*, 32(3):409–425, 1991.

[16] Fred Sommers. *The Logic of Natural Language*. Clarendon Press, Oxford, 1982.

[17] Patrick Suppes. Logical inference in English: a preliminary analysis. *Studia Logica*, 38:375–391, 1979.

Received 8th April, 2002