ISSN 1361 - 6161

THE UNIVERSITY of MANCHESTER

Computer Science
University of Manchester

On the Semantic Complexity of some Fragments of English

Ian Pratt-Hartmann

Department of Computer Science

University of Manchester

Technical Report Series

UMCS-00-5-1

On the Semantic Complexity of some Fragments of English*

Ian Pratt-Hartmann

Department of Computer Science
University of Manchester
Oxford Road, Manchester, UK.
ipratt@cs.man.ac.uk

28.6.00

*Refereed by: Enrico Franconi

Abstract

The goal of this paper is to develop our understanding of the relationship between fragments of English grammar and the fragments of logic which they give rise to. Our strategy is to present the grammar of a minimal controlled language, called RFE, together with some of its variants. The grammar of RFE is deliberately kept as simple as possible, in order to clarify the logical issues involved. Since the elimination of ambiguity is thought important in controlled languages, we consider the variant RFE^{-}, which incorporates into RFE a a very simple standardized pronoun-resolution strategy. In addition, we consider the variants RFEN and $\mathrm{RFE}^{-}\mathrm{N}$, where various forms of negation are added. We show that, despite the minimal grammatical resources involved, the problems of determining entailment in RFE and satisfiability in RFEN are undecidable. We also show that the fragment $\mathrm{RFE}^{-}\mathrm{N}$ corresponds to the two-variable fragment of first-order logic, so that satisfiability in $\mathrm{RFE}^{-}\mathrm{N}$ is decidable, but only with nondeterministic exponential time complexity.

1 Introduction

Controlled languages are regimented fragments of natural language designed to make the processing of natural language, by both humans and computers, more efficient and reliable. Although work on controlled languages was originally motivated by the need to produce uniform and easily translatable technical documentation, attention has recently turned to their possible application to system requirements specifications (Fantechi *et al.* [3], Holt and Klein [6], Holt *et al.* [7], Macias and Pulman [9], Nelken and Francez [11]). This interest in natural specification languages is motivated by the fact that many design engineers and programmers find formal specification languages—usually, some variety of logic—alien and hard to understand. The hope is that, by selecting a regimented subset of natural language, the precision and clarity offered by formal specification languages can be combined with the familiarity and ease-of-understanding associated with natural language.

In recent years, a new factor has arisen with potentially important implications for the use of controlled languages—the explosion of research into decidable fragments of logic. The picture emerging from this research is that inference engines can be designed for many fragments of logic, combining provable completeness with realistic running times in commonly encountered cases. Known decidable fragments include various prefix classes (Börger, Grädel and Gurevich [1]), the guarded fragment (Grädel [4]) and, most importantly for our purposes, the two-variable fragment (Grädel and Otto [5]). The relevance of this work for controlled languages is clear. If a controlled language is mapped into a fragment of first-order logic, the question naturally arises as to whether that fragment enjoys good computational properties. Conversely, given a fragment of logic for which efficient theorem-provers have been developed, it would be useful to identify a controlled language which maps to it.

The goal of this paper is to develop our understanding of the relationship between fragments of English grammar and the fragments of logic which they give rise to. This is not the first time such work has been attempted. In particular, McAllester and Givan [10] describe a logic which is motivated by the structure of simple English sentences, and for which they provide decision-procedures and complexity analyses. However, McAllester and Givan's logic consciously 'over-generates', in that it allows the construction of formulas which do not naturally correspond to English sentences, and at the same time, has such limited expressive power that its usefulness must be open to question. Our strategy in this paper, by contrast, is to present the grammar of a minimal controlled language, called RFE, together with some of its variants. The grammar of RFE is deliberately kept as simple as possible, in order to clarify the logical issues involved. Thus, while RFE would not constitute a useful controlled language in its own right, any useful controlled language must, we claim, incorporate at least the linguistic resources of RFE. Since the elimination of ambiguity is thought important in controlled languages, we consider the variant RFE^{-}, which incorporates into RFE a a very simple standardized pronoun-resolution strategy. In addition, we consider the variants RFEN and $\mathrm{RFE}^{-}\mathrm{N}$, where various forms of negation are added. We show that, despite the minimal grammatical resources involved, the problems of determining entailment in RFE and satisfiability in RFEN are undecidable. We also show that the fragment $\mathrm{RFE}^{-}\mathrm{N}$ corresponds to the two-variable fragment of first-order logic, so that satisfiability in $\mathrm{RFE}^{-}\mathrm{N}$ is decidable, but only with nondeterministic exponential time complexity.

2 Four fragments of English

In this section, we present the basic fragments of English whose complexity we analyse.

2.1 The syntax of RFE and RFEN

The fragment RFE (for *restricted fragment of English*) is defined by a *grammar* and a *special lexicon*. The grammar consists of the definite clause grammar rules

S → NP(i), VP(i)
VP(i) → V(i,j), NP(j)
NP(i) → Det(i), N′(i)
N′(i) → N(i)
N′(i) → N(i), CP(i)
CP(i) → NP(i), S

and the wh-movement rule

> In a phrase CP(i)→ NP(i),S, the NP(i) is filled by movement of a phrase [$_{\mathrm{NP}}$ which] from the S; moreover, every occurrence of [$_{\mathrm{NP}}$ which] must so move.

The special lexicon consists of the entries

Det(i) → every
Det(i) → a(n)
NP(i) → it
NP(i) → itself
NP(i) → which.

A *vocabulary* for RFE consists of indefinitely many entries for the open-class word categories (N and V) of the forms

N(i) → artist
N(i) → beekeeper

V(i,j) → admires
V(i,j) → berates.

These rules and lexica generate sentences by successive expansion of nonterminal nodes under unification of arguments, in the usual way. As usual, the words it, itself and which will be referred to as *pronouns*, *reflexives* and *relative pronouns*, respectively. All other NPs will be called *full* NPs. We modify the gender and case of reflexives, pronouns and relative pronouns as appropriate, without troubling to incorporate this detail into our grammar. Thus, the following are acceptable sentences of RFE.

(1) Every artist admires himself

(2) Every artist despises every beekeeper

(3) Every beekeeper whom an artist despises admires him

Figure 1 shows the relevant phrase structures.

The arguments i and j in the above grammar rules will be referred to as *indices*. Indices are required to satisfy additional *indexing rules*, which serve to regulate the use of pronouns and reflexives, and to delimit their potential referents. In the fragments of English considered in this paper, all anaphora is assumed to be intra-sentential: that is, pronouns are assumed to take referents within the sentences in which they occur.

Two phrases are said to be *coindexed* if their indices are the same. An *antecedent* of a pronoun or reflexive is a *full* NP occurring before it in the lexical ordering of the sentence. If α and β are distinct antecedents of a given pronoun or reflexive γ, we say that β is *closer to* γ than α is, if β lies on the (unique) path between α and γ in the phrase-structure tree. We can then state a preliminary version of the indexing rules as follows.

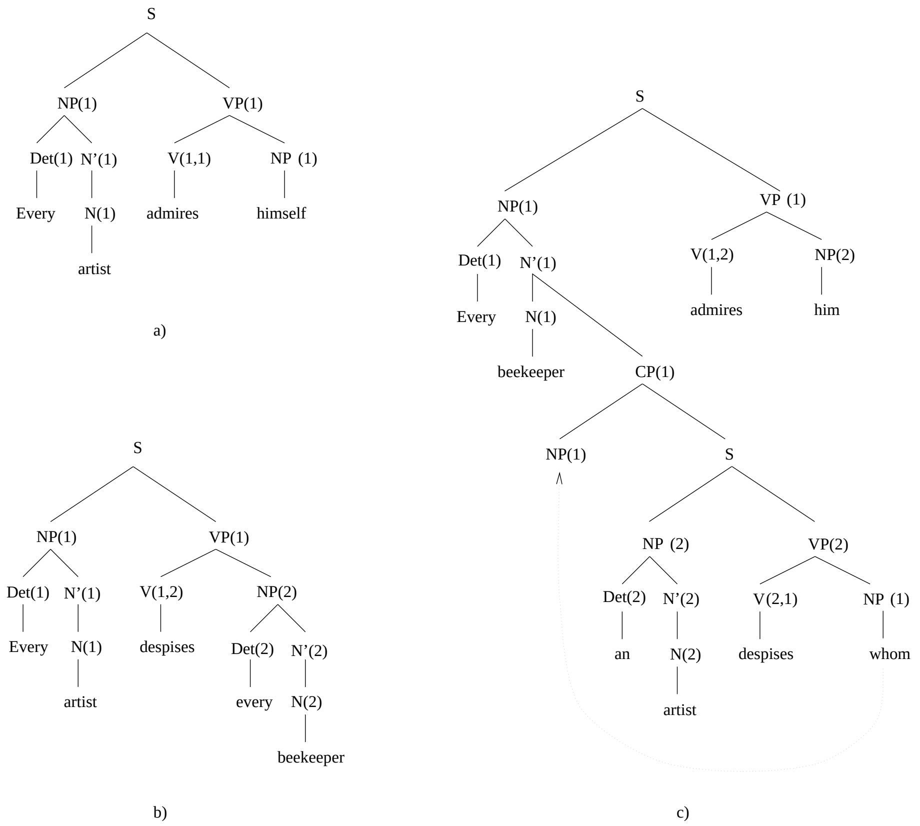

Figure 1: Phrase-structures of the sentences (1)–(3).

I1: Every reflexive must be coindexed with its closest antecedent, which must lie under the immediate S-node containing that reflexive.

I2′: Every pronoun must be coindexed with some antecedent, but not with its closest antecedent.

I3: No two full NPs may be coindexed.

We shall modify rule **I2′** slightly below.

The integers in parentheses in figure 1 indicate, up to re-labelling, the only possible patterns of indexation for sentences (1)–(3) obeying these indexing rules. Thus, in figure 1a) rule **I1** forces the two indices of admires to be identical; in figure 1b), rule **I3** forces the two indices of despises to be distinct; and in figure 1c), rule **I2′** forces him to coindex with artist and not with beekeeper.

Readers familiar with the standard approach of Government and Binding Theory (e.g. Cowper [2]) will have expected a different set of indexing rules, namely the following.

G1: Every reflexive must be A-bound within its governing category.

G2: No pronoun may be A-bound within its governing category.

G3: Every R-expression must be A-free.

In fact, the two sets of rules are equivalent, given our prohibition on inter-sentential anaphora. Specifically, we have

Observation 1. For any sentence accepted by the above grammar rules, if the indexing pattern satisfies **I1**, **I2′** and **I3**, then it also satisfies **G1**, **G2** and **G3**. Conversely, if the indexing pattern satisfies **G1**, **G2** and **G3**, and all pronouns and reflexives are coindexed with antecedents in the sentence itself, then the sentence can be re-indexed so as to satisfy **I1**, **I2′** and **I3**.

Proof. Routine. □

The advantage of our **I**-rules over the more familiar **G**-rules is that the former permit a more perspicuous treatment of the relationship between the various fragments discussed in this paper, as shown below.

There is a further restriction on pronoun anaphora which we mention for the sake of faithfulness to ordinary usage, though it plays no significant role in the technical results reported here. Consider again sentence (3), an example of so-called donkey-anaphora. Here the anaphora can be resolved intra-sententially, with the pronoun taking an artist as the antecedent with which it coindexes. However, this interpretation requires that quantification over artists be existential, and not universal. For example, consider the sentence:

(4) Every beekeeper whom every artist despises admires him.

Although this is a grammatical sentence of English, it is excluded from the fragment RFE, since the pronoun cannot be resolved intra-sententially: the individual(s) admired must have been introduced in some earlier sentence in whatever discourse sentence (4) occurs in.

In such cases, we might say that the antecedent every artist is *inaccessible* to the pronoun in sentence (4). And we can formalize this notion of accessibility for our fragment as follows. Suppose that a phrase-structure tree has been computed for a given sentence. Let us say that the path from a node α to a distinct node β in that tree is *open* if every upward move along that path from a full NP (including the starting point) satisfies either of following two conditions: (i) the determiner of the NP is a(n); (ii) the mother of the NP is the top node (so that the NP is the sentence subject). An NP α is said to be *accessible* to a later node β if the path from α to β is open. We can now state the modification of **I2′** to take account of this phenomenon. It is:

I2: Every pronoun must be coindexed with some accessible antecedent, but not with its closest antecedent.

We say that a string of words is an *RFE-sentence* over a given vocabulary if it is classified as a sentence by the above grammar in accordance with rules **I1**–**I3**. An RFE parser for a given vocabulary is easily implemented in the form of a definite clause grammar (for example, in Prolog). In particular, wh-movement and rules **I1**–**I3** can be incorporated into such a grammar using well-known argument-passing techniques. We note that any given string of words has at most one parse-tree, though the indexing rules may permit several patterns of indexation up to re-labelling.

Having defined the fragment RFE, the fragment RFEN can be defined in exactly the same way except for the following addition to the special lexicon:

$\text{Det(i)} \rightarrow \textsf{no}$.

For example, the following are RFEN-sentences:

(5) No artist admires any beekeeper

(6) No beekeeper admires an artist who despises no beekeeper.

Strictly speaking, the grammar of RFEN should be complicated to include the negative polarity determiner any; but we shall ignore this complication in the sequel for the sake of simplicity: the few mildly awkward sentences that result could easily be removed at the cost of complicating the grammar rules somewhat. At first sight, the possibilities for negation provided by the determiner no seem very limited. As we shall see in section 3, however, this appearance is deceptive.

2.2 The semantics of RFE and RFEN

The interpretation of the fragments just described proceeds according to a set of translation rules which serve to map any (parsed and indexed) sentence of those fragments into a formula of first-order logic. Translation proceeds phrase-by-phrase and is specified by induction on the phrase-structure tree (before wh-movement). If α is a phrase, denote its translation by $T(\alpha)$. Denote by $\mathcal{L}$ the set of formulas of first-order logic over some suitable signature (which we leave implicit). If X and Y are sets, denote by $X \rightarrow Y$ the set of functions from X to Y. We identify indices in a parsed sentence with variables of first-order logic; however, for readability we shall often write x_i instead of i when we think of i as a variable of first-order logic. For brevity of exposition, we admit well-formed formulas of the form $\exists\bar{x}\phi$ and $\forall\bar{x}\phi$, where $\bar{x}$ is a *set* of variables, with the obvious semantics. We allow that $\bar{x}$ be the empty set, in which case the quantification is vacuous.

If α is an N, V, N′, VP, S or CP, then $T(\alpha) \in \mathcal{L}$ and is given by the rules:

T1: If $\alpha = [_{\text{N}} w]$ where w is a noun, and i is the index of α, then $T(\alpha) = w(i)$.

T2: If $\alpha = [_{\text{V}} w]$ where w is a verb, and i, j are the indices of α, $T(\alpha) = w(i, j)$.

T3: If $\alpha = [_{\text{N}'}[_{\text{N}}\beta]]$ then $T(\alpha) = T(\beta)$.

T4: If $\alpha = [_{\text{N}'}[_{\text{N}}\beta][_{\text{CP}}\gamma]]$ then $T(\alpha) = T(\beta) \wedge T(\gamma)$.

T5: If $\alpha = [_{\text{VP}}[_{\text{V}}\beta][_{\text{NP}}\gamma]]$ then $T(\alpha) = T(\gamma)(T(\beta))$.

T6: If $\alpha = [_{\text{S}}[_{\text{NP}}\beta][_{\text{VP}}\gamma]]$ then $T(\alpha) = T(\beta)(T(\gamma))$.

T7: If $\alpha = [_{\text{CP}}[_{\text{NP}}][_{\text{S}}\beta]]$ then $T(\alpha) = T(\beta)$.

If α is an NP then $T(\alpha) \in \mathcal{L} \rightarrow \mathcal{L}$ and is given by the rules:

T8: if α is one of $[_{\text{NP}}\textsf{it}]$, $[_{\text{NP}}\textsf{itself}]$ or $[_{\text{NP}}\textsf{which}]$, then $T(\alpha)$ is the identity function on $\mathcal{L}$.

T9: If $\alpha = [_{\text{NP}}[_{\text{Det}}\beta][_{\text{N}'}\gamma]]$ then $T(\alpha) = T(\beta)(T(\gamma))$.

If α is a Det then $T(\alpha) \in \mathcal{L} \rightarrow (\mathcal{L} \rightarrow \mathcal{L})$ and is given by the rules:

T10: If $\alpha = [_{\text{Det}}\textsf{every}]$ then $T(\alpha) = \lambda P[\lambda Q[\forall \text{Var}(\alpha)(P \rightarrow Q)]]$.

T11: If $\alpha = [_{\text{Det}}\mathsf{no}]$ then $T(\alpha) = \lambda P[\lambda Q[\forall \text{Var}(\alpha)(P \rightarrow \neg Q)]]$.

T12: If $\alpha = [_{\text{Det}}\mathsf{a(n)}]$ then $T(\alpha) = \lambda P[\lambda Q[\exists \text{Var}(\alpha)(P \wedge Q)]]$.

We use meta-variables P and Q to range over formulas of $\mathcal{L}$ in the obvious way. Here, Var is a function which, when applied to a determiner α, returns a set of (between zero and two) variables to be bound by the quantifier associated with α. We shall give a full definition of Var presently, but in most cases, if α is a determiner, then Var(α) is simply the singleton consisting of the index of α.

Taking this explanation of Var(α) on trust, we are in a position to translate sentences (1) and (2). Referring to figure 1a), we see that $[_{\text{VP}}$ admires himself$]$ translates to admire(x_1, x_1) by rules **T2** and **T8**; $[_{\text{NP}}$ every artist$]$ translates to $\lambda Q[\forall\{x_1\}(\text{artist}(x_1) \rightarrow Q)]$ by rules **T1**, **T3**, **T9** and **T10**; and the whole of sentence (1) (after tidying up $\forall\{x_1\}$ to $\forall x_1$) translates by rule **T6** to the formula

(7) $\forall x_1(\text{artist}(x_1) \rightarrow \text{admire}(x_1, x_1))$.

An exactly similar calculation maps sentence (2) to the formula

(8) $\forall x_1(\text{artist}(x_1) \rightarrow \forall x_2(\text{beekeeper}(x_2) \rightarrow \text{despise}(x_1, x_2)))$.

To translate sentence (3), we must attend to the proper definition of the function Var(α). The grammar rules of RFE ensure that every determiner will have a unique sister node (an N$'$) and a unique grandmother node (either an S or a VP).

Definition 1. If the phrase α is a Det in a sentence of RFEN, define Var(α) to be the set of those free variables in the translation of the sister of α which do not occur (as indices) outside the grandmother of α.

Referring to figure 1c), let α be the determiner an. Since the pronoun him coindexes with artist, every free variable in the translation of the sister of α occurs outside the grandmother of α, so that Var(α) is empty. Hence, omitting the vacuous quantifier, $[_{\text{NP}}$ an artist $]$ translates to $\lambda Q[(\text{artist}(x_2) \wedge Q)]$, and $[_{\text{N}'}$ beekeeper whom an artist despises $]$ translates to beekeeper$(x_1) \wedge (\text{artist}(x_2) \wedge \text{despise}(x_2, x_1))$. Now let α be the determiner every. Since the sister of α has free variables $\{x_2, x_1\}$ and the grandmother of α is the top node, we have Var$(\alpha) = \{x_1, x_2\}$. Hence $[_{\text{NP}}$ Every beekeeper whom an artist despises$]$, translates to $\lambda Q[\forall\{x_1, x_2\}(\text{beekeeper}(x_1) \wedge \text{artist}(x_2) \wedge \text{despise}(x_2, x_1) \rightarrow Q)]$. Again, re-writing $\forall\{x_1, x_2\}$ in more familiar form, the final translation of sentence (3) is:

(9) $\forall x_1 \forall x_2(\text{beekeeper}(x_1) \wedge \text{artist}(x_2) \wedge \text{despise}(x_2, x_1) \rightarrow \text{admire}(x_1, x_2))$.

An alternative and more general approach to the semantics of anaphoric constructions found in RFEN is provided by Discourse Representation Theory or DRT (Kamp and Reyle [8]). Like the semantics presented above, DRT in effect employs a two-stage approach in translating from English to $\mathcal{L}$. First, English sentences are mapped to so-called Discourse Representation Structures (DRSs)—in effect a formal representation language with nonstandard mechanisms for quantification—and these DRSs are then translated into formulas of $\mathcal{L}$ via the standard semantics for the DRS language. This two-stage approach allows DRT to give an account of the variable-binding encountered in sentence 3 very similar to that given above.

When the expressive power of RFEN is at issue however, our translation scheme is more useful than that offered by DRT. The main difference concerns the point at which quantifiers corresponding to indefinite articles are introduced. The strategy adopted by DRT is that the determiner a(n) does not introduce an existential quantifier: instead, 'left-over' variables corresponding to the indices of these determiners are gathered up by quantifiers introduced in the interpretation of DRSs. The effect of this strategy is—simplifying somewhat—to quantify these variables *as late as possible*. Thus, for example, the sentence

(10) Every artist who admires a beekeeper who berates a carpenter castigates himself

will be translated to

(11) $\forall x \forall y \forall z((\text{artist}(x) \wedge \text{beekeeper}(y) \wedge \text{admire}(x,y) \wedge$
$\quad\quad \text{carpenter}(z) \wedge \text{berate}(y,z)) \rightarrow$
$\quad \text{castigate}(x,x))$

with the quantifiers taking maximal scope. Our approach, by contrast, quantifies early wherever possible. Thus, sentence (10) would be translated not as (11), but rather as

(12) $\forall x((\text{artist}(x) \wedge$
$\quad \exists y(\text{beekeeper}(y) \wedge \exists z(\text{carpenter}(z) \wedge \text{berate}(y,z)) \wedge \text{admire}(x,y))) \rightarrow$
$\quad \text{castigate}(x,x))$.

Of course, these two translations are logically equivalent. However, our strategy of early quantification makes the expressiveness of RFEN and its subfragments easier to determine. Thus, in proposing the above semantics for RFEN, we are not taking issue with DRT; rather, we are simply presenting the semnatics in a form which is more convenient for the issues at hand.

One respect in which our semantics for RFEN may fail to do justice to native English speakers' intuitions however is quantifier rescoping. It is sometimes claimed that the sentence

(13) Every artist despises a beekeeper

is ambiguous between two choices for the scoping of the quantifiers. By contrast, the above semantics for RFE unambiguously assigns wide scope to the universal quantifier. We do not consider quantifier rescoping in this paper. Generally, proposals for controlled natural languages attempt to eliminate such ambiguities by stipulating that quantifiers scope in a particular way; and this is the approach that we shall take. Very roughly, the effect of the above grammar rules is that the quantifier introduced by the subject determiner scopes over the quantifier introduced by the object determiner, and that the quantifier introduced by any NP determiner scopes over those introduced in any relative clauses within that NP. This policy seems the most sensible default in those cases where a choice has to be made.

Finally, it will help to align our semantics for RFE(N) with the familiar apparatus of model theory. Any RFE(N) vocabulary associates a unary predicate with every noun mentioned in the vocabulary and a binary predicate with every verb mentioned in the vocabulary. Thus, for example, the above vocabulary associates the predicate $\text{artist}(x)$ with the noun artist and the predicate $\text{admire}(x,y)$ with the verb admires. Hence, we may regard a structure $\mathfrak{A}$ over these predicates as a structure interpreting the RFE(N) vocabulary itself. We say that an RFE sentence e over this vocabulary is *true* in $\mathfrak{A}$, written $\mathfrak{A} \models e$, just in case $\mathfrak{A} \models T(e)$. The standard semantic relations of satisfiability and entailment are then defined as usual: a set of sentences E is *satisfiable* if some structure makes every sentence in E true; and E *entails* a sentence e if every structure which makes every member of E true also makes e true.

At this point, we have a sufficiently precise specification of the syntax an semantics of RFE and RFEN to determine their expressive power and computational complexity. These questions will be addressed in sections 3 and 4. Our next task is to define a restriction of RFEN which, as we shall see later, has a dramatic effect on expressiveness and complexity.

The fragments RFE^- and RFE^-N

The absence of quantifier rescoping and the restricted syntax of RFE mean that the only source of ambiguity in RFE is pronoun resolution. A typical ambiguity of this kind is

(14) Every artist who despises a beekeeper who admires a carpenter berates him.

The rules **I1–I3** permit the pronoun to be coindexed with either beekeeper or carpenter (of which the former is the closer to the pronoun). The resulting translations are, respectively:

(15) $\forall x_1 \forall x_2((\text{artist}(x_1) \wedge$
$\quad\quad \text{beekeeper}(x_2) \wedge$
$\quad\quad \exists x_3(\text{carpenter}(x_3) \wedge \text{admire}(x_2,x_3)) \wedge \text{despise}(x_1,x_2)) \rightarrow$
$\quad \text{berate}(x_1,x_2))$

(16) $\forall x_1 \forall x_3((\text{artist}(x_1) \wedge$
$\exists x_2(\text{beekeeper}(x_2) \wedge \text{carpenter}(x_3) \wedge \text{admire}(x_2, x_3)) \wedge$
$\text{despise}(x_1, x_2)) \rightarrow$
$\text{berate}(x_1, x_3))$.

Since, as we noted in the introduction, controlled natural languages standardly aim to avoid ambiguity, the question naturally arises as to how to respond to this sort of situation. The simplest solution is to introduce an additional coindexation rule artificially stipulating which of the permitted antecedents a pronoun may be coindexed with. And the simplest such rule is the following strengthening of **I2**:

I2$^-$: Every pronoun must be coindexed with its second closest accessible antecedent.

Again, "second closest" means "second closest in the syntax tree", not "second closest in the lexical ordering". It is easy to verify that the above grammar rules ensure that, for any pronoun (or reflexive) which has two accessible antecedents, it must have a unique second closest accessible antecedent, which is in fact the second closest antecedent.

It is important understand the special status of rule **I2$^-$**. The previous rules reflect the observed facts of English usage: dropping any of them would admit ungrammatical sentences or sentences in which pronouns must be resolved extrasententially. However, violations of **I2$^-$** occur with grammatical sentences under intuitively admissible interpretations. An example is sentence (14): the interpretation (16), which is acceptable to native English speakers, is ruled out by **I2$^-$**. Linuistically artificial restrictions in syntax or semantics seem to be the inevitable price of ironing out ambiguities in controlled languages.

Let us call the English fragment formed by replacing rule **I2** in RFE with rule **I2$^-$**, RFE$^-$. Clearly, RFE and RFE$^-$ admit the same set of sentences, though RFE allows more interpretations of these sentences than does RFE$^-$. In fact, from the foregoing discussion, we conclude:

Observation 2. RFE$^-$ is unambiguious.

Although everything expressible in RFE$^-$ is expressible in RFE, the converse is not true. For example, the RFE sentence

(17) Every thing$_1$ ays every thing which some thing which it$_1$ ays ays.

translates (under the indicated indexing) to

(18) $\forall x_1(\text{thing}(x_1) \rightarrow$
$\forall x_2(\text{thing}(x_2) \wedge \exists x_3(\text{thing}(x_3) \wedge \text{ay}(x_1, x_3) \wedge \text{ay}(x_3, x_2)) \rightarrow$
$\text{ay}(x_1, x_2)))$,

which states that the verb ays expresses a transitive relation over the domain of things. This indexing pattern is not available in RFE$^-$, since the second occurrence of everything is closer to the pronoun than the first. That transitivity is in fact not expressible in RFE$^-$ follows from the results proved below.

Finally, let us turn to the fragment RFE$^-$N, which adds negation to RFE$^-$. The restrictions on coindexing in RFE$^-$ mean that special measures are needed to deal with negation: for it turns out that simply adding the determiner no does not give us the same power as it does with full RFE. Therefore, we take RFE$^-$N to be defined in the same way as RFE$^-$, except for the following three additions. First, we add to the grammar the rules

S $\rightarrow$ NP(i), NEGP(i)
NEGP(i) $\rightarrow$ NEG, VP(i),

and to the special lexicon the entry

NEG $\rightarrow$ does not.

(We ignore the treatment of verb-inflections in the grammar presented here.) Second, we modify the definition of the relation of accessibility by adding the requirement that an *open* path in the phrase-structure tree contain no node labelled NEGP. This modification is necessary to conform to linguistic intuition regarding donkey anaphora in negative contexts; an illustrative example is given below. Third, we add the translation rule:

T13: If $\alpha = [_{\text{NEGP}}$ does not $[_{\text{VP}}\beta]]$, then $T(\alpha) = \neg T(\beta)$.

These additions translate the sentences

(19) Every beekeeper despises every artist whom he does not admire

(20) Every artist does not admire a beekeeper

into the formulas

(21) $\forall x_1(\text{beekeeper}(x_1) \rightarrow \forall x_2((\text{artist}(x_1) \wedge \neg\text{admire}(x_1, x_2)) \rightarrow \text{despise}(x_1, x_2)))$,

(22) $\forall x(\text{artist}(x_1) \rightarrow \neg\exists\text{beekeeper}(x_2) \wedge \text{admire}(x_1, x_2))$

respectively. As sentence (20) illustrates, the introduction of negation into RFE$^-$ raises additional scoping issues, and additional contexts where ordinary usage insists on negative polarity items. Again, however, we do not pause to investigate these details here.

To be sure, the treatment of negation in RFE$^-$N leaves much to be desired from a linguistic perspective. A more sensitive (and standard) alternative would be to render sentences as IPs, with the verb moving to the empty I position when no auxiliary such as does is present. However, tidying up the above grammar in this way would not affect the logical issues discussed below, and would render the proofs in section 5 unnecessarily cumbersome. In the interests of clarity, therefore, we shall leave things as they stand.

Let us return briefly to the modification of the notion of accessibility in RFE$^-$N. As with RFEN, issues of accessibility do not affect the logical and computational properties of the fragment; we include them merely for the sake of maintaining faithfulness to English usage. Consider the sentence (23) and its translation (24):

(23) Every artist who despises a beekeeper who does not admire a carpenter berates him

(24) $\forall x_1 \forall x_2((\text{artist}(x_1) \wedge$
$\text{beekeeper}(x_2) \wedge$
$\neg\exists x_3(\text{carpenter}(x_3) \wedge \text{admire}(x_2, x_3)) \wedge \text{despise}(x_1, x_2))) \rightarrow$
$\text{berate}(x_1, x_2))$.

Given the new definition of open path, the noun-phrase a beekeeper is not accessible to the pronoun in (23), so that no other pattern of indexing is possible for this sentence. This seems intuitively reasonable.

One final modification is necessary to complete the definition of RFE$^-$N. In the fragment RFEN, we did not single out any verb to denote the identity relation. This is because the identity relation, though not definable in RFEN, can at least be 'simulated', in a sense which we make precise below. In RFE$^-$N, however, the restrictions on coindexing render this simulation impossible. Accordingly, we declare a special verb—eqs—to denote the identity relation. That is, in any interpretation $\mathfrak{A}$, the binary predicate corresponding to eqs is interpreted as the identity relation over A. This concludes our definition of the fragment RFE$^-$N.

Now that we have defined the fragments RFEN and RFE$^-$N, we can answer the questions posed in the introduction: what can we express in these fragments, and what is the computational complexity of reasoning within them? Intuitively, we should expect that RFEN is strictly more expressive than RFE$^-$N, and that reasoning within RFEN is correspondingly more difficult than reasoning in RFE$^-$N. Just how much more expressive and how much more difficult, we are about to find out.

3 Identity and negation in RFEN

The fragment RFEN contains no predicate constrained to denote the identity relation, and indeed it is not difficult to show that identity cannot be defined using the logical resources of RFEN. Furthermore, although the determiner no in RFEN contributes a negation symbol, it does not allow this negation symbol to be inserted at any point in a formula. However, it transpires that both identity and negation can be reproduced indirectly in RFEN, and the task of this section is to establish this fact. The following results are therefore largely technical and may be omitted on first reading. The statement of theorem 1 contains all that is needed for later sections.

We begin with a simple observation concerning quantification over the whole domain of a structure.

Definition 2. We call a RFEN vocabulary which contains the single noun thing and a set of verbs τ a *verb-vocabulary*, denoted simply by τ.

Recalling that we can regard an RFEN vocabulary τ as a signature for first-order logic, and noting that all quantification in RFEN sentences over a verb-vocabulary is restricted to the denotation of thing, the following result is obvious.

Lemma 1. *Let $\mathfrak{A}$ be any structure interpreting a verb-vocabulary τ, and let $\mathfrak{B}$ be the restriction of $\mathfrak{A}$ to the set $\mathsf{thing}^{\mathfrak{A}}$. Then $\mathfrak{A}$ and $\mathfrak{B}$ make the same set of RFEN sentences over τ true.*

Thus, given any structure over a verb-vocabulary, we can always chop away those elements not in the extension of thing without affecting anything expressible in RFEN. In the sequel, we contract every thing to everything and a thing to something, noting that, provided we are operating with a verb vocabulary, these words can always be interpreted so that quantification ranges over the whole domain. Occasionally, in the sequel, we will omit the translation of thing from translations, translating, for example, Everything ays something as $\forall x_1 \exists x_2 \text{ay}(x_1, x_2)$ rather than the officially correct $\forall x_1(\text{thing}(x_1) \rightarrow \exists x_2(\text{thing}(x_2) \wedge \text{ay}(x_1, x_2)))$.

We now turn to the representation of the identity relation in RFE. Consider the following three RFE-sentences.

Everything eqs something
Everything eqs everything which eqs it
Everything$_1$ eqs everything which something which it$_1$ eqs eqs

A structure $\mathfrak{A}$ makes these sentences true if and only if eqs denotes an equivalence relation over $\mathsf{thing}^{\mathfrak{A}}$. (The first of these sentences could equivalently be replaced by Everything eqs itself.) Notice that the anaphora in the third of these sentences is 'long-range' and thus banned in RFE$^-$. Henceforth, then, we can write Eqs is an equivalence relation as a sentence of RFE, understanding that it can be expanded to the three sentences above.

Definition 3. If the structure $\mathfrak{A}$ interprets the verb eqs as the identity relation, then we say that $\mathfrak{A}$ is *normal.*

Definition 4. If τ is a verb-vocabulary, we denote by EQ_τ the following set of RFE-sentences:

{ Everything$_1$ α everything which eqs something which it$_1$ α |α a verb in τ } $\cup$
{ Everything$_1$ α everything which something which it$_1$ eqs α |α a verb in τ } $\cup$
{ Eqs is an equivalence relation }

Here, eqs may or may not be in τ.

Lemma 2. *Let $\mathfrak{A}$ be a structure over the verb-vocabulary $\tau \cup \{\mathsf{eqs}\}$. If $\mathfrak{A}$ is normal, then $\mathfrak{A} \models \text{EQ}_\tau$. Conversely, if $\mathfrak{A} \models \text{EQ}_\tau$, then there exists a normal model $\mathfrak{B}$ over $\tau \cup \{\mathsf{eqs}\}$ such that $\mathfrak{A}$ and $\mathfrak{B}$ make the same RFEN sentences true.*

Proof. The first part of the lemma is obvious. For the second part, Let $\mathfrak{A}'$ be result of restricting $\mathfrak{A}$ to $\mathsf{thing}^{\mathfrak{A}}$. By lemma 1, $\mathfrak{A}' \models \text{EQ}_\tau$, so that $\mathsf{eqs}^{\mathfrak{A}'}$ is an equivalence relation on A'. Let $B = A'/\mathsf{eqs}^{\mathfrak{A}'}$ and define $\alpha^{\mathfrak{B}}$ for $\alpha \in \tau \cup \{\mathsf{eqs}\}$ by

$$\alpha^{\mathfrak{B}} = \{([a], [b]) | (a, b) \in \alpha^{\mathfrak{A}'}\}.$$

Since $\mathfrak{A}' \models \text{EQ}_\tau$, $\mathfrak{B}$ is well-defined; moreover, $\mathfrak{B}$ is certainly normal. Given that we can regard $\tau \cup \{\mathsf{eqs}\}$ as a signature of binary relations, a simple induction shows that $\mathfrak{A}'$ and $\mathfrak{B}$ make the same closed formulas over this signature true. Hence $\mathfrak{A}$ and $\mathfrak{B}$ make the same RFEN sentences true. $\square$

Thus, although identity is not definable in RFEN, any consistent set E of sentences over a verb-vocabulary τ has a normal model provided $\mathrm{EQ}_\tau \subseteq E$.

Finally, we show how sentence-negations can be simulated in RFEN over a verb-vocabulary τ. Our strategy is to introduce, for the main verb β in the sentence, a new verb $\bar{\beta}$, and to write RFEN sentences constraining the interpretations of these two verbs to be complementary.

Definition 5. If α and β are verbs, denote by $N_{\alpha,\beta}$ the pair of RFE sentences:

Nothing β something it α
Everything$_1$ α everything which eqs nothing it$_1$ β.

(Of course, in a more adequate grammar, the first of these sentences would be replaced by Nothing β anything it α.) Notice that the anaphora in the second of these sentences violates **I2**$^-$ and is thus excluded from RFE$^-$. These sentences translate to

$$\forall x_1 \forall x_2 (a(x_1, x_2) \rightarrow \neg b(x_1, x_2))$$
$$\forall x_1 \forall x_2 (\forall x_3 (b(x_1, x_3) \rightarrow \neg \mathrm{eq}(x_2, x_3)) \rightarrow a(x_1, x_2)),$$

where $a(x, y)$ and $b(x, y)$ are the predicates associated with α and β, respectively, whence:

Lemma 3. *Let $\mathfrak{A}$ be a normal structure interpreting α and β. Then $\mathfrak{A} \models N_{\alpha,\beta}$ if and only if* $\mathsf{thing}^{\mathfrak{A}} \times \mathsf{thing}^{\mathfrak{A}}$ *is the disjoint union of $\alpha^{\mathfrak{A}}$ and $\beta^{\mathfrak{A}}$.*

Proof. Immediate. □

Definition 6. Let e be an RFE-sentence. We define $\bar{e}$ to be the RFE-sentence constructed as follows:

Replace the main verb β in e by the new verb $\bar{\beta}$.

In the subject of e (which will be a full noun-phrase), switch its determiner (so that a(n) becomes every and every becomes a(n)).

If the object of e is a full noun-phrase, switch its determiner (so that a(n) becomes every and every becomes a(n)).

For example, if e is Every artist despises a beekeeper, then $\bar{e}$ is An artist $\overline{\text{despises}}$ every beekeeper; and if e is Every artist who admires a beekeeper despises himself, then $\bar{e}$ is An artist who admires a beekeeper $\overline{\text{despises}}$ himself.

The change from e to $\bar{e}$ does not violate any of the rules **I1**–**I3**, since no antecedents that were previously accessible to a pronoun become inaccessible. Moreover, if $\mathfrak{A}$ is any structure in which the interpretations of the verbs β and $\bar{\beta}$ are complementary, then, visibly, $\mathfrak{A} \models \bar{e}$ if and only if $\mathfrak{A} \not\models e$. Hence we have proved:

Lemma 4. *Let e be an RFE sentence over a verb-vocabulary, and let the main verb of e be β. If $\mathfrak{A}$ is normal and $\mathfrak{A} \models \{\bar{e}\} \cup N_{\beta,\bar{\beta}}$ then $\mathfrak{A} \not\models e$. Conversely, if $\mathfrak{A}$ is normal, $\mathfrak{A} \not\models e$, and $\bar{\beta}$ is a verb not interpreted in $\mathfrak{A}$, then $\mathfrak{A}$ can be expanded to a structure $\mathfrak{A}'$ such that $\mathfrak{A}' \models \{\bar{e}\} \cup N_{\beta,\bar{\beta}}$.*

This lemma gives us enough flexibility in the use of negation to reduce entailment in RFE to satisfiability in RFEN.

Theorem 1. *The problem of deciding validity in RFE over a verb-vocabulary τ can be effectively reduced to the problem of deciding unsatisfiability in RFEN.*

Proof. Suppose E is a set of RFE-sentences over τ, and e an RFE-sentence over τ. Without loss of generality, we may assume that τ does not contain the verb eqs or the verb $\bar{\beta}$, where β is the main verb in e. We show that $E \models e$ if and only if the set of RFEN sentences $E \cup \mathrm{EQ}_{\tau \cup \{\bar{\beta}\}} \cup \{\bar{e}\} \cup N_{\beta,\bar{\beta}}$ is unsatisfiable.

If $E \models e$ then by lemma 4, $E \cup \{\bar{e}\} \cup N_{\beta,\bar{\beta}}$ has no normal model. By lemma 2, $E \cup \mathrm{EQ}_{\tau\cup\{\bar{\beta}\}} \cup \{\bar{e}\} \cup N_{\beta,\bar{\beta}}$ has no model at all.

Conversely, if $E \not\models e$, we can find a structure $\mathfrak{A}$ such that $\mathfrak{A} \models E$ but $\mathfrak{A} \not\models e$. Since τ does not contain eqs, $\mathfrak{A}$ can trivially be made normal. By lemma 4, we can expand $\mathfrak{A}$ to a structure $\mathfrak{A}' \models \{\bar{e}\} \cup N_{\beta,\bar{\beta}}$. Since this is an expansion, $\mathfrak{A}' \models E$, and $\mathfrak{A}'$ is still normal. Hence, by lemma 2, $\mathfrak{A}' \models \mathrm{EQ}_{\tau\cup\{\bar{\beta}\}}$. □

Theorem 1 is important given that the problem of determining entailment in RFE is undecidable. It is to this matter that we turn in the next section.

4 The undecidability of Validity in RFE

The task of this section is to establish that entailment in RFE is undecidable. In the light of the previous section, it follows that satisfiability in RFEN is also undecidable. Our strategy is to reduce subsumption in NIKL′, a slight variant of the description logic NIKL (Patel-Schneider [12]), to RFE-validity.

Expressions in NIKL′ are either *concepts*, *roles* or *role-chains*. Concepts express unary relations, while roles and role-chains express binary relations. We use the variables C, D, etc. to range over concepts, R, S, etc. to range over roles and role-chains, and L to range over NIKL′ expressions generally. Expressions in NIKL′ are built up from *atomic concepts* and *atomic roles* according to the following rules.

If C is an atomic concept then C is a concept
If R is an atomic role then R is a role
If C and D are concepts, and R is a role then $(C \wedge D)$,
  $(\exists R)$ and $(\exists_{=1} R)$ are concepts, and $(R|C)$ and R^u are roles
If R and S are role-chains then $(\forall R, S)$ is a concept
If R is a role then R and SELF are role chains
If R is a role and S a role-chain, then $R \circ S$ is a role chain.

Let $\mathfrak{A}$ be a structure assigning to any atomic concept C a subset $C^{\mathfrak{A}}$ of A and to any atomic role R a subset $R^{\mathfrak{A}}$ of $A \times A$. Then $\mathfrak{A}$ assigns to any NIKL′ expression L an interpretation $L^{\mathfrak{A}}$ according to the following rules.

$$(C \wedge D)^{\mathfrak{A}} = C^{\mathfrak{A}} \cap D^{\mathfrak{A}}$$
$$(\exists R)^{\mathfrak{A}} = \{a \in A : (a,b) \in R^{\mathfrak{A}}, \text{ for some } b \in A\}$$
$$(\exists_{=1} R)^{\mathfrak{A}} = \{a \in A : (a,b) \in R^{\mathfrak{A}}, \text{ for exactly one } b \in A\}$$
$$(\forall R, S)^{\mathfrak{A}} = \{a \in A : (a,b) \in R^{\mathfrak{A}} \text{ implies}(a,b) \in S^{\mathfrak{A}}, \text{ for all } b \in A\}$$
$$(R|C)^{\mathfrak{A}} = \{(a,b) \in A \times A : (a,b) \in R^{\mathfrak{A}} \text{ and } b \in C^{\mathfrak{A}}\}$$
$$(R^u)^{\mathfrak{A}} = \{(a,b) \in A \times A : (b,a) \in R^{\mathfrak{A}}\}$$
$$\textsc{self}^{\mathfrak{A}} = \{(a,a) \in A \times A : a \in A\}$$
$$(R \circ S)^{\mathfrak{A}} = \{(a,b) \in A \times A : (a,c) \in R^{\mathfrak{A}} \text{ and } (c,b) \in S^{\mathfrak{A}}, \text{ for some } c \in A\}$$

Given two concepts C and D, we say that C is *subsumed* by D if, for all $\mathfrak{A}$ interpreting the primitives of C and D, $C^{\mathfrak{A}} \subseteq D^{\mathfrak{A}}$. This completes the definition of the language NIKL′.

We have

Theorem 2 (Patel-Schneider [12], theorem 3.5)). *Subsumption in NIKL′ is r.e.-complete.*

In his proof of the above theorem, which proceeds by effective reduction of the subsumption problem to the Post correspondence problem, Patel-Schneider actually uses the construct $(\exists_{\leq 1} R)$ where we have used $(\exists_{=1} R)$, with the obvious corresponding changes to the semantics. However, in his proof of the above theorem, $(\exists_{\leq 1} R)$ only ever appears conjoined with $(\exists R)$, so that the proof would go through with this conjunction replaced by our $(\exists_{=1} R)$.

We now proceed to the reduction of subsumption in NIKL′ to entailment in RFE over a verb-vocabulary.

Definition 7. Let L be a NIKL$'$ expression (either a concept or role or role-chain), and let $\mathfrak{A}$ be a structure interpreting the primitives of L. For every subexpression M of L except (SELF), let α_M be a new verb. Let $\alpha_{(\mathrm{SELF})}$ be the distinct verb eqs. Let τ_L be the set of all verbs thus obtained, together with the noun thing. Thus, τ_L is a verb-vocabulary. We say that a structure $\mathfrak{B}$ interpreting τ_L is *compatible* with $\mathfrak{A}$ if (i) $\mathfrak{A}$ and $\mathfrak{B}$ have the same domain A, the interpretation of thing in $\mathfrak{B}$ is the whole of that domain, (iii) for every atomic role S mentioned in C, $\alpha_S^{\mathfrak{B}} = S^{\mathfrak{A}}$, and (iv) for every atomic concept D mentioned in C, $\alpha_D^{\mathfrak{B}} = D^{\mathfrak{A}} \times A$.

Thus, if $\mathfrak{B}$ is compatible with $\mathfrak{A}$, then every verb α_S, where S is a atomic role, has the same interpretation under $\mathfrak{B}$ as S has under $\mathfrak{A}$; and every verb α_D, where D is a atomic concept, has, as its interpretation under $\mathfrak{B}$, the cylindrification of the interpretation of D under $\mathfrak{A}$.

The following definition makes clear how NIKL$'$ can be reproduced in RFE.

Definition 8. Let L be a NIKL$'$ expression. We define the set of RFE-sentences E_L over the verb-vocabulary τ_L recursively as follows.

(i) if L is an atomic concept, then $E_L =$
{Everything which α_L something α_L everything}
(ii) if L is an atomic role, then $E_L = \emptyset$
(iii) If L is $D \wedge E$, then $E_L = E_D \cup E_E \cup$
{Everything which α_D something it α_E α_L it,
Everything α_D everything it α_L,
Everything α_E everything it α_L}
(iv) If L is $(\exists R)$ then $E_L = E_R \cup$
{Everything which α_R something α_L everything,
Everything which α_L something α_R something}
(v) If L is $(\exists_{=1} R)$ then $E_L = E_R \cup$
{Everything$_1$ which α_R something which everything it$_1$ α_R eqs α_L everything,
Everything which α_L something α_R something,
Everything which something$_1$ which α_L something α_R eqs everything it$_1$ α_R }
(vi) If L is $(\forall R, S)$ then $E_L = E_R \cup E_S \cup$
{Everything which α_S everything it α_R α_L everything,
Everything which α_L something α_S everything it α_R}
(vii) If L is $(R|C)$ then $E_L = E_R \cup E_C \cup$
{Everything which α_R something$_1$ which α_C something α_L it$_1$,
Everything which α_L something α_R it,
Everything which something α_L α_C something}
(viii) If L is (SELF), then $E_L = \emptyset$
(ix) If L is $(R \circ S)$ then $E_L = E_R \cup E_S \cup$
{Everything which α_R something which α_S something$_1$ α_L it$_1$,
Everything which α_L something$_1$ α_R something which α_S it$_1$}

Lemma 5. *Let $\mathfrak{A}$ be a structure over the primitives in some NIKL$'$ concept C. Then there exists a normal structure $\mathfrak{B}$ compatible with $\mathfrak{A}$ such that $\mathfrak{B} \models E_C$. Moreover, any such $\mathfrak{B}$ satisfies $\alpha_C^{\mathfrak{B}} = C^{\mathfrak{A}} \times A$.*

Proof. The first part of the lemma follows by setting, for every concept expression D occuring as a subexpression of C, $\alpha_D^{\mathfrak{B}} = D^{\mathfrak{A}} \times A$, and for every role expression S occuring as a subexpression of C, $\alpha_S^{\mathfrak{B}} = D^{\mathfrak{A}}$, and finally by setting $\mathsf{eq}^{\mathfrak{B}}$ to be the identity relation and $\mathsf{thing}^{\mathfrak{B}}$ to be the whole domain. A routine induction on the structure of C establishes that $\mathfrak{B} \models E_C$.

Conversely, suppose $\mathfrak{B}$ is a normal structure interpreting the verb-vocabulary τ_C, compatible with $\mathfrak{A}$ and satisfying $\mathfrak{B} \models E_C$. A routine structural induction shows that, for any subexpression L of C, if L is a role or role-chain, then $\alpha_L^{\mathfrak{B}} = L^{\mathfrak{A}}$, and that, if L is a concept, then $\alpha_L^{\mathfrak{B}} = L^{\mathfrak{A}} \times A$. In particular, $\alpha_C^{\mathfrak{B}} = C^{\mathfrak{A}} \times A$. □

Armed with this lemma, we can establish the sought-after reduction:

Theorem 3. *Given NIKL′ concepts C, D, we can effectively find a set of RFE-sentences $E_{C,D}$ and an RFE-sentence $e_{C,D}$ such that C is subsumed by D if and only if $E_{C,D} \models e_{C,D}$.*

Proof. Let $E_{C,D} = E_C \cup E_D \cup \mathrm{EQ}_{\tau_C \cup \tau_D}$, and let $e_{C,D}$ be

Everything which α_C something α_D it.

Suppose $E_{C,D} \models e_{C,D}$. Let $\mathfrak{A}$ be any structure interpreting the primitives of C and D. By lemma 5, let $\mathfrak{B}$ be a normal structure interpreting the verb-vocabulary $\tau_C \cup \tau_D$, compatible with $\mathfrak{A}$, such that $\mathfrak{B} \models E_C \cup E_D$, whence $\alpha_C^{\mathfrak{B}} = C^{\mathfrak{A}} \times A$ and $\alpha_D^{\mathfrak{B}} = D^{\mathfrak{A}} \times A$. Moreover, since $\mathfrak{B}$ is normal, $\mathfrak{B} \models \mathrm{EQ}_{\tau_C \cup \tau_D}$, so that $\mathfrak{B} \models E_{C,D}$. But since we are supposing $E_{C,D} \models e_{C,D}$, we have $\mathfrak{B} \models e_{C,D}$, so $\alpha_C^{\mathfrak{B}} \subseteq \alpha_D^{\mathfrak{B}}$, so $C^{\mathfrak{A}} \subseteq D^{\mathfrak{A}}$. Thus, for any structure $\mathfrak{A}$ interpreting the primitives in C and D, $C^{\mathfrak{A}} \subseteq D^{\mathfrak{A}}$. In other words, C is subsumed by D.

Conversely, suppose that C is subsumed by D. Let $\mathfrak{A}' \models E_{C,D}$. Since $\mathfrak{A}' \models \mathrm{EQ}_{\tau_C \cup \tau_D}$, lemmas 1 and 2 imply the existence of a normal structure $\mathfrak{B}$ with domain $B = \mathsf{thing}^{\mathfrak{B}}$ such that $\mathfrak{A}'$ and $\mathfrak{B}$ make the same RFEN sentences over τ_L true. Now define a structure $\mathfrak{A}$ interpreting the NIKL′ primitives in C and D as follows. Let the domain of $\mathfrak{A}$ be B. For every atomic concept D' mentioned in C or D, set $D'^{\mathfrak{A}}$ to be the projection of $\alpha_{D'}^{\mathfrak{B}'}$ onto its first component, i.e.

$$D'^{\mathfrak{A}} = \{a \in A | (a,b) \in \alpha_{D'}^{\mathfrak{B}'}, \text{ some } b \}.$$

For every atomic role S mentioned in C or D, set $S^{\mathfrak{A}}$ to be $\alpha_S^{\mathfrak{B}'}$. Since $\mathfrak{B} \models E_{C,D}$, and since $E_{C,D}$ contains the sentence Everything which $\alpha_{D'}$ something $\alpha_{D'}$ everything for every atomic concept D' mentioned in C or D, it is easily seen that $\mathfrak{B}$ is compatible with $\mathfrak{A}$, so that, by lemma 5, $\alpha_C^{\mathfrak{B}} = C^{\mathfrak{A}} \times B$ and $\alpha_D^{\mathfrak{B}} = D^{\mathfrak{A}} \times B$. Since by hypothesis C is subsumed by D, we have $C^{\mathfrak{A}} \subseteq D^{\mathfrak{A}}$, so $\alpha_C^{\mathfrak{B}} \subseteq \alpha_D^{\mathfrak{B}}$, so $\mathfrak{B} \models e_{C,D}$, so $\mathfrak{A}' \models e_{C,D}$. Thus, $\mathfrak{A}' \models E_{C,D}$ implies $\mathfrak{A}' \models e_{C,D}$. In other words, $E_{C,D} \models e_{C,D}$. □

Corollary 1. *The problem of determining entailment in RFE is r.e.-complete.*

Proof. By theorem 2 □

Corollary 2. *The problem of determining satisfiability in RFEN is co-r.e.-complete.*

Proof. By theorem 1. □

Corollary 3. *There is an effective process for transforming every formula ϕ of $\mathcal{L}$ into a set E of RFEN sentences such that ϕ is satisfiable if and only if E is satisfiable.*

Proof. The set of unsatisfiable formulas of $\mathcal{L}$ is r.e. □

5 The expressiveness of RFE$^-$N: upper bound

Our task in this section is to establish an upper bound on the expressiveness of RFEN$^-$; this upper expressiveness bound automatically gives us an upper complexity bound for the satisfiability problem.

As usual, we denote by $\mathcal{L}$ the set of formulas of first-order logic, and by $\mathcal{L}^2$ the subset of $\mathcal{L}$ involving only the variables x_1 and x_2. The fragment $\mathcal{L}^2$ is well-understood. In particular, we have the following well-known result.

Theorem 4 (Börger, Gurevich and Grädel [1] corollary 8.1.5). *The problem of deciding the satisfiability of a sentence in $\mathcal{L}^2$ is in NEXPTIME.*

Before we come to the main theorem, we prove two useful observations. The first gives us an alternative characterization of $\mathcal{L}^2$.

Definition 9. If $\phi \in \mathcal{L}$, we say that ϕ is *two-variable compatible* (for short: 2vc) if no proper or improper subformula of ϕ contains more than two free variables.

Lemma 6. *Every 2vc formula is equivalent to a formula of* $\mathcal{L}^2$.

Proof. We replace successively complex subformulas of ϕ with equivalent formulas in $\mathcal{L}^2$, renaming variables in the rest of ϕ if necessary. The details are routine. □

The second observation reports an effect of the more restrictive coindexation rule **I2$^-$**.

Definition 10. Let e be a sentence of RFEN. We say that e is *donkey* if the object of e contains (properly or improperly) a pronoun which is coindexed with an NP properly contained in the subject of e. Let α be a CP in some sentence e of RFEN. We say that α is *donkey* if e is a donkey sentence, and α is the relative clause attached to the subject of e. Similarly, Let α be an N$'$ in some sentence of e of RFEN. We say that α is *donkey* if e is a donkey sentence, and α contains a donkey CP.

For instance, in the donkey sentence Every farmer who owns a donkey beats it, the phrases $[_{\mathrm{CP}}$who owns a donkey$]$ and $[_{\mathrm{N}'}$farmer who owns a donkey$]$ are donkey. It is obvious that a donkey sentence contains exactly one donkey CP and exactly one donkey N$'$.

In the sequel, if α is a phrase in an RFEN sentence such that α has exactly one index, we denote that index by $\mathrm{Ind}(\alpha)$.

Lemma 7. *If α is a* CP *occurring in a sentence e of* RFE$^-$N*, such that α is not donkey, and contains the* N$'$ *γ, then* $\mathrm{Ind}(\gamma)$ *does not occur in e outside the phrase α.*

Proof. Suppose for contradiction that $\mathrm{Ind}(\gamma)$ does occur outside the phrase α. The grammar rules do not pass N$'$-indices up out of a CP, and rule **I3** prohibits the coindexing of full NPs. Therefore, the NP immediately containing γ must coindex with a pronoun or reflexive outside α. By rule **I1**, every reflexive is coindexed with the subject of its immediately containing S; hence no reflexive outside α can coindex with γ. So let β be a pronoun outside α which takes γ as antecedent. Now α must attach to some NP, which therefore intervenes between β and γ. By rule **I2$^-$**, no other NP can lie on the path between β and γ. Therefore, β itself is not contained in a CP, and so must be the object of e. Moreover, again by **I2$^-$**, α must be attached to the subject of e. This contradicts the assumption that α is not donkey. □

We note that lemma 7 concerns only patterns of indexing in RFE$^-$N; in particular, the translation rules **T1–T13** do not figure in the proof.

The main theorem of this section establishes that the semantics given above translates RFE$^-$N sentences to 2vc formulas.

The following notation will be useful in the ensuing proof.

Definition 11. Let α be a CP in a sentence of RFEN. Then α contains exactly two maximal NPs, of which one is occupied by a relative pronoun moved to the empty CP-specifier position. We call the other maximal NP in α the *main* NP of α. We call α *pronomial* if its main NP is a pronoun; otherwise, we call α *standard*. Similarly, if α is an N$'$ in a sentence of RFEN, we call α *pronomial* if it is of the form $[_{\mathrm{N}}\gamma][_{\mathrm{CP}}\gamma]$ with γ pronomial; otherwise, we call α *standard*.

Theorem 5. *If e is an* RFE$^-$N*-sentence, then $T(e)$ is a closed 2vc formula.*

Proof. In the sequel, we use the translation rules **T1–T13** freely without mention. We take the Roman $a(x)$, $b(x,y)$. $c(x,y)$ etc. to denote atomic formulas and the Greek $\phi(x)$, $\psi(x,y)$, $\pi(x,y)$ etc. to denote 2vc formulas. In addition, all free variables are explicitly listed. Thus, for example, $\phi(x,y)$ denotes a 2vc formula whose *only* free variables are x and y.

We show first that, if α is a CP or an N$'$ in e, and α is non-donkey and standard, then $T(\alpha)$ is a 2vc formula with $\mathrm{Ind}(\alpha)$ as its only free variable. By induction, we suppose that the result holds for all nodes strictly below α in the phrase-structure tree for the sentence.

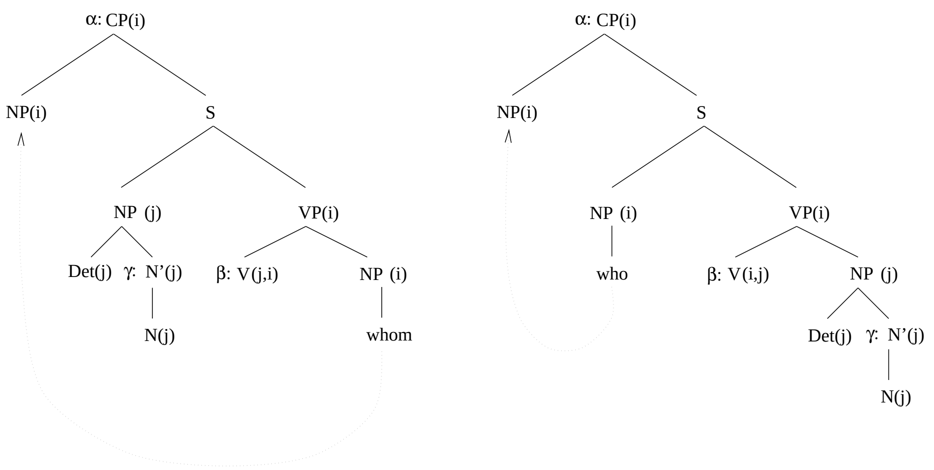

Figure 2: The structure of a CP

Let α be a non-donkey, standard N$'$. If α consists of a single N, let $T(\alpha) = a(x)$ with $x = \text{Ind}(\alpha)$. If, on the other hand, α is of the form $[_{\text{N}}\beta][_{\text{CP}}\gamma]$ then γ must be non-donkey and standard. Let $T(\beta) = b(x)$ and, by inductive hypothesis, let $T(\gamma) = \psi(x)$ with $x = \text{Ind}(\gamma) = \text{Ind}(\alpha)$. Then $T(\alpha) = b(x) \wedge \psi(x)$ with $x = \text{Ind}(\alpha)$.

Let α be a non-donkey, standard CP. If the main NP of α is a reflexive, let $T(\alpha) = a(x, x)$ with $x = \text{Ind}(\alpha)$. If, on the other hand, the main NP of α is a full NP, let the main V and N$'$ of α be β and γ respectively, as shown in figure 2. (The phrases β, γ and δ are identified in exactly the same way in negative relative clauses, of course.) Since γ is contained within a CP, it is certainly non-donkey. We consider separately the cases where γ is standard or pronomial. If γ is standard, by inductive hypothesis, let $T(\gamma) = \psi(y)$ with $y = \text{Ind}(\gamma)$. If, on the other hand, γ is pronomial, let $T(\gamma) = c(x'', y)$ with $y = \text{Ind}(\gamma)$, and x'' the index of the pronoun in γ. Now the CP α is attached to some NP, with which it coindexes. And that NP is the closest antecedent but one of the pronoun contained in γ. By rule **I2$^-$** then, α coindexes with the pronoun in γ, so that $T(\gamma) = c(x, y)$ with $x = \text{Ind}(\alpha)$ and $y = \text{Ind}(\gamma)$. Now let $T(\beta) = b(x', y')$, with $\{x', y'\} = \{x, y\}$. It follows that $T(\alpha) = Q\bar{z}\pi(x, y)$, where Q is a quantifier ($\forall$ or $\exists$), $\bar{z}$ is Var(δ) and $\pi(x, y)$ is a Boolean combination either of $\phi(x', y')$ and $\psi(y)$, or of $\phi(x', y')$ and $c(x, y)$. Certainly, Ind(α) appears outside the grandmother of δ, but, by lemma 7, since α is not donkey, Ind(γ) does not. Hence $\bar{z}$ is the 1-tuple consisting of y, so that $T(\alpha) = Qy\pi(x, y)$ with $x = \text{Ind}(\alpha)$, as required. This establishes our claim that, if α is a CP or an N$'$ in e, and α is non-donkey and standard, then $T(\alpha)$ is a 2vc formula with Ind(α) as its only free variable.

We are now in a position to complete our proof of theorem 5. Suppose first that e is a donkey sentence. Let α be the donkey CP in e, and let γ be the main N' contained in α, again as shown in figure 2. By identical reasoning to that of the last paragraph, we have $T(\alpha) = Q\bar{z}\pi(x, y)$, with $x = \text{Ind}(\alpha)$ and $y = \text{Ind}(\gamma)$. This time, however, α is donkey, so that both x and y occur outside the grandmother of δ, whence $\bar{z}$ is empty. Ignoring the vacuous quantification, we can write $T(\alpha) = \pi(x, y)$. If β is the top-level VP in e then we have $T(\beta) = b(x, y)$ with $x = \text{Ind}(\alpha)$ and (by **I2$^-$**) $y = \text{Ind}(\gamma)$. Hence $T(e) = Q\{x, y\}\phi(x, y)$, where Q is a quantifier and $\phi(x, y)$ is a Boolean combination of $\pi(x, y)$ and $b(x, y)$.

Finally, suppose that e is not a donkey sentence. Let α be the subject of e. Since α is a full NP with a standard, non-donkey N$'$, let $T(\alpha) = \phi(x)$, where $x = \text{Ind}(\alpha)$. Let β be the main VP in e. If the object of β is a reflexive, we can write $T(\gamma) = b(x, x)$, whence $T(e) = Q\{x\}\pi(x)$ where Q is a quantifier and $\pi(x)$ is a Boolean combination of $\phi(x)$ and $b(x, x)$. Suppose, then, that the object of β is a full NP containing an N$'$ γ. If γ is standard, then the result established above ensures that $T(\gamma) = \psi(y)$ with $y = \text{Ind}(\gamma)$. If, on the other hand, γ is pronomial, then it is easy to see that $T(\gamma) = \psi(x', y)$, where $y = \text{Ind}(\gamma)$, and x' the index of the pronoun in γ. By **I2$^-$**, this pronoun must coindex with α, so we

$\forall x \forall y(a(x,y) \rightarrow b(y,x))$	Everything bees everything which ays it
$\forall x(\exists y a(x,y) \rightarrow b(x,x))$	Everything which ays something bees itself
$\forall x(b(x,x) \rightarrow \forall y a(x,y))$	Everything which bees itself ays everything
$\forall x(\exists y a(x,y) \rightarrow b(x))$	Everything which ays something eqs a bee
$\forall x(b(x) \rightarrow \forall y a(x,y))$	Every bee ays everything
$\forall x \forall y(a(x,y) \rightarrow \neg b(x,y))$	Nothing bees something which it ays
$\forall x \forall y(\neg b(x,y) \rightarrow a(x,y))$	Everything ays everything which it does not bee
$\forall x \forall y((b(x,y) \wedge c(x,y)) \rightarrow a(x,y))$	Everything which bees something which it cees ays it
$\forall x \forall y(a(x,y) \rightarrow b(x,y))$	Everything bees everything which it ays
$\forall x \forall y a(x,y)$	Everything ays everything
$\forall x \exists y a(x,y)$	Everything ays something
$\exists x \forall y a(x,y)$	Something ays everything
$\exists x \exists y a(x,y)$	Something ays something

Table 1: Some formulas of $\mathcal{L}^2$ and their RFE$^-$N-translations

may write $T(\gamma) = \psi(x,y)$ with $x = \text{Ind}(\alpha)$. Let the main verb of the sentence translate to $b(x,y)$. Then $T(\beta) = Q_1\{y\}\pi(x,y)$ where $\pi(x,y)$ is a Boolean combination either of $b(x,y)$ and $\psi(x)$ or of $b(x,y)$ and $\psi(x,y)$. Hence $T(e) = Q_2\{x\}\theta(x)$, where θ is a Boolean combination of $\phi(x)$ and $Q_1\{y\}\pi(x,y)$. ☐

Corollary 4. *Every sentence of RFE$^-$N is equivalent to a formula in $\mathcal{L}^2$.*

Corollary 5. *The problem of determining the satisfiability of a set of RFE$^-$N sentences is in NEXPTIME.*

6 The expressiveness of RFE$^-$N: lower bound

Our task in this section is to establish a lower bound on the expressiveness of RFEN$^-$; this lower expressiveness bound automatically gives us a lower complexity bound for the satisfiability problem.

Our strategy will be to show that, for any closed formula ϕ in the two-variable fragment, a set E of RFE$^-$N sentences can be found such that E and ϕ are satisfiable over the same domains. To this end, we define:

Definition 12. Let $\phi, \psi \in \mathcal{L}$. We say that ψ is an *expansion* of ϕ if $\psi \models \phi$ and, for any structure $\mathfrak{A}$ such that $\mathfrak{A} \models \phi$, there is an expansion (in the usual model-theoretic sense) $\mathfrak{B}$ of $\mathfrak{A}$ such that $\mathfrak{B} \models \psi$.

Thus an expansion ψ of ϕ is satisfiable if and only if ϕ is satisfiable.

Lemma 8. *There is a function $f : \mathcal{L}^2 \rightarrow \mathcal{L}^2$, computable in polynomial time and linear space, such that, for all $\phi \in \mathcal{L}^2$, $f(\phi)$ is an expansion of ϕ consisting of a conjunction of formulas of the forms found in the left-hand column of table 1, where the a, b and c are relation-symbols of the indicated arity.*

Proof. We give a recipe for computing $f(\phi)$. By a well-known result, there is a function $f' : \mathcal{L}^2 \rightarrow \mathcal{L}^2$, computable in polynomial time and linear space, such that, for all $\phi \in \mathcal{L}^2$, $f'(\phi)$ is an expansion of ϕ of the form

$$\forall x \forall y \chi_0 \wedge \bigwedge_{1 \leq i \leq m} \forall x \exists y \chi_i \tag{1}$$

where the χ_i $(0 \leq i \leq m)$ are quantifier-free. So we first obtain the formula $f'(\phi)$.

All atoms in $f'(\phi)$ are of the forms $b(x,y)$, $b(y,x)$, $b(x,x)$, $b(y,y)$, $b(x)$, and $b(y)$. We process $f'(\phi)$ so that all atoms appearing in the χ_i $(0 \leq i \leq m)$ are of the form $b(x,y)$. (If b is the equality relation, the necessary manipulations are trivial, so we may ignore this case in what follows.) We proceed by replacing any atom not in the desired form with an atom $a(x, y)$, where a is a new binary relation, and by adding conjuncts as specified in the following table:

Form of replaced atom	Added conjuncts
$b(y, x)$	$\forall x \forall y(a(x, y) \to b(y, x))$, $\forall x \forall y(b(x, y) \to a(y, x))$
$b(x, x)$	$\forall x(\exists y a(x, y) \to b(x, x))$, $\forall x(b(x, x) \to \forall y a(x, y))$
$b(x)$	$\forall x(\exists y a(x, y) \to b(x))$, $\forall x(b(x) \to \forall y b(x, y))$

The remaining possibilities, namely $b(y)$ and $b(y, y)$, are handled similarly. Let the result of this replacement process be denoted by $f''(\phi)$. It is clear that f'' is computable in polynomial time and linear space, and that $f''(\phi)$ is an expansion of $f'(\phi)$ and hence of ϕ.

We next replace the smallest non-atomic subformulas appearing in $f''(\phi)$ by atoms of the form $a(x, y)$, where a is a new binary relation symbol; and we add conjuncts according to the table as follows:

Form of subformula	Added conjuncts
$\neg b(x, y)$	$\forall x \forall y(a(x, y) \to \neg b(x, y))$, $\forall x \forall y(\neg b(x, y) \to a(x, y))$
$b(x, y) \wedge c(x, y)$	$\forall x \forall y((b(x, y) \wedge c(x, y)) \to a(x, y))$, $\forall x \forall y(a(x, y) \to b(x, y))$, $\forall x \forall y(a(x, y) \to c(x, y))$

We then simply take $f(\phi)$ to be the result of applying the above replacement process to exhaustion. It is clear that f is computable in polynomial time and linear space, that $f(\phi)$ is an expansion of $f''(\phi)$ and hence of ϕ and moreover that $f(\phi)$ is of the form required by the lemma. □

Theorem 6. *Let ϕ be in $\mathcal{L}^2$. Then we can compute, in polynomial time and linear space, a set E of sentences in RFE⁻N, such that E is satisfiable if and only if ϕ is.*

Proof. Compute $f(\phi)$ as in lemma 8, and read the RFE$^-$N translations of each conjunct in $f(\phi)$ from table 1, taking care to translate $=$ using the verb eqs. □

We note in passing that we have used the does not-construction only in one very restricted case, where no scoping ambiguities arise. Hence the above proof would work even if modifications were made to the scoping conventions regarding negation in RFE$^-$N.

The following result gives us a lower complexity bound for RFE$^-$N.

Theorem 7 (Börger, Gurevich and Grädel [1] corollary 6.2.14). *The problem of deciding the satisfiability of a sentence in $\mathcal{L}^2$ has a lower complexity bound* NTIME($2^{cn/\log n}$) *for some constant c.*

Corollary 6. *The problem of determining the satisfiability of a set E of RFE$^-$N sentences has a lower complexity bound* NTIME($2^{cn/\log n}$) *for some constant c.*

7 Conclusions

We began this paper with a precise characterization of the syntax and semantics of the English fragment RFEN. The grammar and special vocabulary of RFEN are restricted to certain minimal logical resources grammar which, it is reasonable to suppose, every useful controlled natural language must include. The only respect in which the grammatical resources of RFE and RFEN may outstrip those of practical controlled languages concerns their free use of pronouns, which most controlled languages are likely to curb in order to eliminate ambiguity. Accordingly, we defined the subfragment RFE$^-$N in which pronouns are artificially constrained to refer to the syntactically closest available antecedent. We take it that the disambiguation strategy adopted in RFE$^-$N is the most natural strategy for which useful results on expressive power are available. The main result of this paper is that this pronoun restriction strategy yields a fragment for which the satisfiability problem is decidable. Less positively, we showed that the entailment problem for sets of RFE sentences and the satisfiability problem for sets of RFEN sentences are both undecidable. In addition, we showed that the RFE$^-$N is weakly equivalent to the two-variable fragment of first-order logic, and hence that the problem of determining satisfiability in RFE$^-$N has nondeterministic exponential time complexity. Thus, all controlled natural languages can be expected to have satisfiability problems with high worst-case complexity.

However, such lower complexity bounds are unavoidable in logical systems of non-trivial expressive power, and we should not be discouraged by them. The syntax and semantics of RFEN outlined in section 2 can be immediately implemented as a Prolog program which translates RFE$^-$N sentences into formulas of the two-variable fragment. (All the examples given in this paper were checked with such a program.) Moreover, efficient and practical decision procedures based on ordered resolution are known for the two-variable fragment. Two directions for future research therefore suggest themselves. First, in what ways can the grammar be extended without leaving the two-variable fragment (or some other, decidable fragment) behind? For example, the addition of intransitive verbs, attributive and predicative adjectives, and coordination using and and or would increase the naturalness and usability of the language without altering its fundamental logical or computational properties. Second, what is the practical usefulness of RFE$^-$N and its various extensions? Specifically, does it really let us specify the sorts of behaviours which designers of real systems are interested in? Is it really easier to use than a formal specification language? And do our best theorem-provers for the relevant fragments of logic really produce answers within an acceptable time? We contend that the above results at least establish that these are questions worthy of serious attention.

References

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

[2] Elizabeth A. Cowper. *A consice introduction to syntactic theory.* University of Chicago Press, Chicago, 1992.

[3] A. Fantechi, S. Gnesi, G. Ristori, M. Carenini, M. Vanocchi, and P. Moreschini. Assisting requirement formalization by means of natural language translation. *Formal Methods in System Design*, 4:243–263, 1994.

[4] Erich Grädel. On the restraining power of guards. *Journal of Symbolic logic*, 1999.

[5] Erich Grädel and Martin Otto. On logics with two variables. *Theoretical computer science*, 1999. Forthcoming.

[6] Alexander Holt and Ewan Klein. A semantically-derived subset of english for hardware verification. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 451–456. Association for Computational Linguistics, June 1999.

[7] Alexander Holt, Ewan Klein, and Claire Grover. Natural language for hardware verification: semantic interpretation and model checking. In *ICoS-1 Inference in Computational Semantics: Institute for Logic, Language and Computation (ILLC)*, pages 133–137. ILLC, University of Amsterdam, August 1999.

[8] Hans Kamp and Uwe Reyle. *From discourse to logic: introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory.* Kluwer Academic Publishers, London, Boston, Dordrecht, 1993. Studies in Linguistics and Philosophy, Volume 42.

[9] Benjamin Macias and Stephen Pulman. A method for controlling the production of specifications in natural language. *The Computer Journal*, 38(4):310–318, 1995.

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

[11] Rani Nelken and Nissim Francez. Translating natural language system specifications into temporal logic via DRT. Technical Report LCL-96-2, Laboratory for Computational Linguistics, Department of Computer Science, Technion, Israel Institute of Technology, 1996.

[12] Peter Patel-Schneider. Undecidability of subsumption in NIKL. *Artificial Intelligence*, 39:263–272, 1989.