Semantic Complexity in Natural Language

Ian Pratt-Hartmann[1]

School of Computer Science, University of Manchester, Manchester, M13 9PL, UK.
ipratt@cs.man.ac.uk

A draft chapter for the Wiley-Blackwell *Handbook of Contemporary Semantics — second edition*, edited by Shalom Lappin and Chris Fox. This draft formatted on 19th October 2015.

1 Introduction

That sentences in natural language exhibit logical entailments was recognized in antiquity. For example, the argument

(1)
Every artist is a beekeeper
Some artist is a carpenter
No carpenter is a dentist
____
Some beekeeper is not a dentist,

is evidently valid: every possible situation in which the premises are true is one in which the conclusion is true. Likewise valid, but less evidently so, is the argument

(2)
Some artist admires no beekeeper
Every beekeeper admires some artist
____
Some artist is not a beekeeper

Indeed, consider any artist, a, who admires no beekeeper. If he is not a beekeeper himself, the conclusion is certainly true. On the other hand, if a is a beekeeper, the second premise guarantees the existence of an artist, b, whom a admires. But then b cannot be a beekeeper, since otherwise, a—who by assumption admires no beekeeper—would not admire him, whence the conclusion is again true. Note that we assume no quantifier re-scoping in (2).

Argument (1) features only the language of the Classical syllogistic—i.e. the determiners *every*, *some* and *no* together with the (possibly negated) copula construction. Argument (2), by contrast, relies crucially on the relational information expressed by transitive verbs. This leads us to the natural question: how does the complexity of determining logical relationships between sentences vary with the syntactic constructions they feature? Is the language of Argument (2) really harder to reason in than the the language of Argument (1)? Would arguments involving *ditransitive* verbs be harder still? Would the availability of relative clauses, for example in the evidently valid

(3)
Every artist who is not a beekeeper is a carpenter
No beekeeper is a dentist
No carpenter is a dentist
____
No artist is a dentist

affect the complexity of inference? What of anaphora, passives, quantifier rescoping, numerical determiners, . . . ? The purpose of this Chapter is to outline what is known in this area.

Our approach is inspired by recent developments in computational logic, and in particular, by the enormous strides that have been made in characterizing the computational complexity of various fragments of first-order logic. It has been known since the work of Turing that first-order logic is undecidable: no computer program can determine whether an arbitrary argument formulated in that language is valid. On the other hand, the existence of *fragments* of first-order logic for which such algorithms do exist antedate even Turing's negative result; and in the years since, a great many such decidable fragments have been found. Moreover, since the emergence of computational complexity theory in the 1970s, it has been possible to characterize, in mathematical terms, the relative difficulty of determining entailments in these fragments. This development has fuelled a surge of interest in Computer Science, particularly in the area of so-called terminological logics.

The idea of tailoring logical systems to (fragments of) natural languages is certainly not new. Early investigations in this direction include Fitch (1973) and Suppes (1979). Of particular note is the use of polarity marking to detect entailments based on set-inclusions (Fyodorov *et al.*, 2003; Zamansky *et al.*, 2006; Moss, 2012; Icard, 2012), a strategy which has recently been employed in the development of robust systems for textual entailment (MacCartney & Manning, 2008). However, these treatments do not aim at proof-theoretical completeness, and certainly do not provide a complexity-theoretic analysis of the underlying inferential problems. The work reported in this Chapter aims to establish a systematic programme for investigating the logic of natural language. That programme is to characterize the complexity of determining entailments in fragments of natural languages, along the lines familiar from contemporary research in computational logic.

The work described here has no connection with an influential tradition of research in psycholinguistics, according to which the study of semantics is centrally concerned with the data-structures used in human cognition, and the study of inference with the algorithms used to manipulate those data-structures. Within that tradition, it is mental representations and mental processes that are to the fore: issues connected with the objective meanings of those representations, or with the validity of the inferential processes applied to them, are viewed as secondary, or even spurious (Jackendoff, 1987). While we do not reject cognitive processes as a legitimate object of investigation, we do reject the claim that such an investigation would be the end of the story, or even that it would help answer the questions we are interested in here. The subject of our investigation is the logical content that natural language constructions put at our disposal, not its mode of (re)presentation. That is, fragments of natural language are to be understood *purely extensionally*, and independently of any representation scheme used to describe them. How to articulate this view, and what results it makes possible, will emerge in the course of the Chapter.

The Chapter is structured as follows. Sec. 2 presents the technical framework which we use to define fragments of natural languages and formulate questions as to their semantic complexity. Sec. 3 reviews the necessary technical background in logic and complexity theory. Sec. 4 makes an excursion into the study of the Classical syllogistic and its extensions. Secs. 5–9 analyse the semantic complexity of various salient fragments of English. We shall show, *inter alia*, that the language of Argument (2), featuring transitive verbs, is in an objective sense inferentially no more complex than the language of Classical syllogisms exemplified by Argument (1); indeed, the analogous extension featuring ditransitive verbs involves only a modest increase in complexity. On the other hand, the language of Argument (3), which adds relative clauses to the Classical syllogistic, entails a greater complexity-theoretic cost, a pattern which is repeated in the presence of transitive or ditransitive verbs. Finally, we investigate the effect of noun-level negation (*non-artist*, *non-beekeeper*), as well as numerical determiner-phrases (*at most 1*, *more than 5*). Our results demonstrate that techniques previously employed in the complexity-theoretic investigation of formal logic can be effectively applied to the domain of natural language.

2 Fragments of Language

By a *fragment* of a natural language, we understand a collection of sentences forming a naturally delineated subset of that language, and equipped with a truth-conditional semantics commanding the general assent of its native speakers. To explain what this means in practice, we begin by defining some simple fragments of English.

Our first fragment is the language of Classical syllogisms, namely, the collection of English sentences having the following forms, with semantics given by the associated first-order formulas.

Every p is a q	$\forall x(p(x) \rightarrow q(x))$
Some p is a q	$\exists x(p(x) \wedge q(x))$
No p is a q	$\forall x(p(x) \rightarrow \neg q(x))$
Some p is not a q	$\exists x(p(x) \wedge \neg q(x))$.

Here, p and q are to be substituted by common (count) nouns in the English sentence-forms, and by corresponding unary predicates in the logical translations. This fragment, which we shall call *Syl*, can be used to formulate Argument (1), above. The lexicon p, q, ... of common nouns is assumed to be countably infinite: that is, although the number of sentence-*forms* in Syl is finite, the number of its sentence*s* is infinite. This assumption of course reflects the linguistic difference between the open category of common nouns on the one hand, and the closed category of determiners and the copula *is* on the other. According to the logical translations proposed here, universally quantified sentences do not have existential import: if no artists exist, then *All artists are beekeepers* is true. This lack of existential import does not restrict the fragment's expressive power; and of course it would be a simple matter to re-instate it if we wished.

The linguistic salience of Syl becomes more perspicuous if, instead of simply enumerating its sentence-forms, we define it using a context-free grammar whose productions are annotated with expressions in the simply-typed λ-calculus. Our grammar for Syl features the following productions.

$\mathrm{S}/\varphi(\psi) \rightarrow \mathrm{NP}/\varphi,\ \mathrm{VP}/\psi$	$\mathrm{Det}/\lambda p \lambda q[\exists x(p(x) \wedge q(x))] \rightarrow$ some
$\mathrm{VP}/\varphi \rightarrow$ is a N'/φ	$\mathrm{Det}/\lambda p \lambda q[\forall x(p(x) \rightarrow q(x))] \rightarrow$ every
$\mathrm{VP}/\lambda x[\neg\varphi(x)] \rightarrow$ is not a N'/φ	$\mathrm{Det}/\lambda p \lambda q[\forall x(p(x) \rightarrow \neg q(x))] \rightarrow$ no
$\mathrm{NP}/\varphi(\psi) \rightarrow \mathrm{Det}/\varphi,\ \mathrm{N}'/\psi$	
$\mathrm{N}'/\varphi \rightarrow \mathrm{N}/\varphi$.	$\mathrm{N}/\mathrm{artst} \rightarrow$ artist
	$\mathrm{N}/\mathrm{bkpr} \rightarrow$ beekeeper
	...

Sentence-semantics are computed by combining the semantic annotations as specified by the productions, and applying the usual simplification rules of the simply-typed λ-calculus. The process is illustrated in Fig. 1. To reduce notational clutter, we have indicated the types of variables informally by the choice of variable names. Thus, x, y range over objects (i.e. have type e), while p, q range over unary predicates (i.e. have type $e \to t$). All non-logical constants involved arise from common nouns, and are unary predicates.

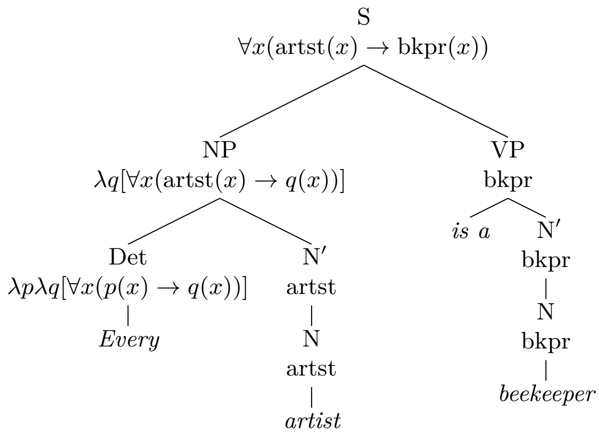

Figure 1. Meaning derivation in Syl

Note that the alternation between *a* and *an* is ignored for simplicity; sentences in examples will be silently corrected as required. Our grammar additionally generates the two rather awkward sentence-forms

Every p is a not a q	$\forall x(p(x) \to \neg q(x))$
No p is not a q	$\forall x(p(x) \to \neg\neg q(x))$,

associating them with the indicated first-order formulas. However, these additional forms evidently do not increase the fragment's expressive power; and of course it would be a simple matter to eliminate them if we wished.

It has long been remarked that the Classical syllogistic cannot formulate inferences that essentially depend on relational information, such as Argument (2), above. We therefore define an extension of Syl featuring transitive verbs, for example:

Every p rs some q	$\forall x(p(x) \to \exists y(q(y) \wedge r(x,y))$
Every p rs every q	$\forall x(p(x) \to \forall y(q(y) \to r(x,y))$
Some p does not r every q	$\exists x(p(x) \wedge \exists y(q(y) \to \neg r(x,y))$
Some p rs no q	$\exists x(p(x) \wedge \forall y(q(y) \to \neg r(x,y))$.

Such a fragment may again conveniently be presented using a semantically annotated context-free grammar. Accordingly, we take the fragment TV to be defined by the productions of Syl together with:

$$\begin{array}{l} \mathrm{VP}/\varphi(\psi) \rightarrow \mathrm{TV}/\varphi,\ \mathrm{NP}/\psi \\ \mathrm{VP}/\lambda x[\neg\varphi(\psi)(x)] \rightarrow \text{does not},\ \mathrm{TV}/\varphi,\ \mathrm{NP}/\psi \\ \\ \mathrm{TV}/\lambda u \lambda x[u(\lambda y[\mathrm{admr}(x,y)])] \rightarrow \text{admire} \\ \cdots \end{array}$$

Again, it is assumed that there are countably many lexical entries for transitive verbs, all similar to the above entry for *admire*. Typing of variables follows the same conventions as above, with u ranging over functions from predicates to truth-values (i.e. having type $(e \rightarrow t) \rightarrow t$). Third-person singular inflections, as well as the occasional need for the negative polarity determiner *any* in place of *some*, have been ignored for simplicity; we will silently correct these defects in examples as required. Fig. 2 shows a sample derivation in this grammar.

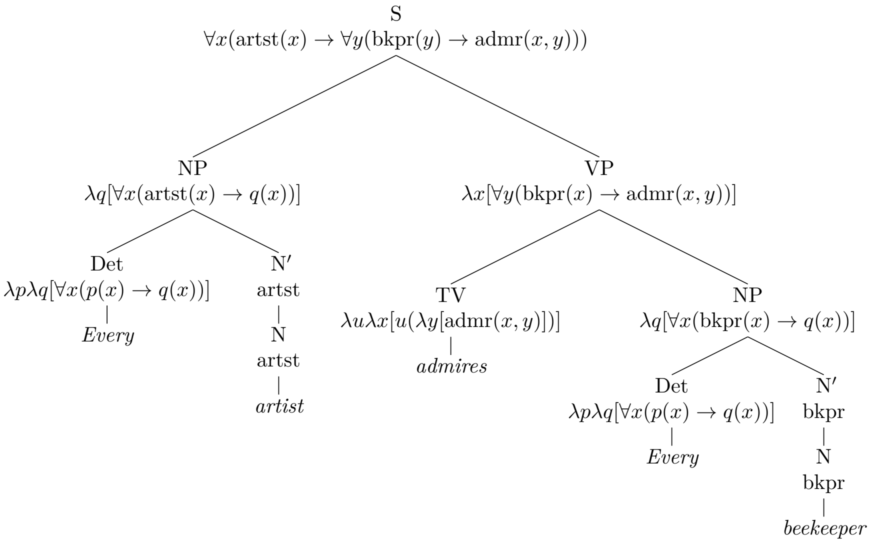

Figure 2. Meaning derivation in TV.

The first-order translations produced by the above grammar are, up to logical equivalence, exactly those of the forms

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

where $\pm\psi$ stands for either ψ or $\neg\psi$. As with Syl, so too with TV, while the number of sentence-*forms* is finite, the number of sentence*s* is infinite. Again, the reader may have observed that TV contains some rather strained and unnatural sentences, and assigns them truth-conditions equivalent to those from the above list, for example:

No p does not r no q $\quad$ $\forall x(p(x) \to \neg\neg\forall y(q(y) \to \neg r(x,y))) \equiv$
$\forall x(p(x) \to \forall y(q(y) \to \neg r(x,y)))$.

However, such sentences do not increase the fragment's expressive power, and their elimination would anyway be routine. More significantly, the above grammar makes specific scoping decisions: subjects outscope direct objects; and negation outscopes object quantifiers, but not subject quantifiers.

Ditransitive verbs may be treated in an analogous way. Let the fragment *DTV* be the result of extending TV with the productions

$$\mathrm{VP}/\varphi(\psi)(\pi) \to \mathrm{DTV}/\varphi,\ \mathrm{NP}/\psi,\ \text{to},\ \mathrm{NP}/\pi$$
$$\mathrm{VP}/\neg\varphi(\psi)(\pi) \to \text{does, not},\ \mathrm{TV}/\varphi,\ \mathrm{NP}/\psi,\ \text{to},\ \mathrm{NP}/\pi$$

$$\mathrm{DTV}/\lambda u \lambda v \lambda x[u(\lambda y[v(\lambda z[\mathrm{rcmnd}(x,y,z)])])] \to \text{recommend},$$
$$\ldots$$

where v has the same type as u, namely $(e \to t) \to t$. Again, the open class of ditransitive verbs is assumed here to be countably infinite, even though such verbs are actually quite infrequent in English. Straightforward calculation analogous to that illustrated above shows that DTV contains, for example, the following sentence, and associates it to the indicated first-order formula.

No artist recommends every beekeeper to some carpenter
$$\forall x(\mathrm{artst}(x) \to \neg\forall y(\mathrm{bkpr}(y) \to \exists z(\mathrm{crpntr}(z) \wedge \mathrm{rcmnd}(x,y,z)))). \quad (1)$$

Again, DTV contains only finitely many sentence-forms, but infinitely many sentences. Similar remarks regarding the elimination of unnatural sentences and quantifier scoping apply as for TV. Subjects outscope direct objects, which in turn outscope indirect objects; and negation outscopes quantifiers in objects (direct or indirect), but not subject quantifiers.

Let us now extend Syl in a different direction. The Classical syllogistic makes no provision for sentences with relative clauses, and thus cannot formulate Argument (3), above. It is natural, then, to consider a fragment which can. Let Syl + Rel, extend Syl with the productions

$$\mathrm{N}'/\varphi(\psi) \to \mathrm{N}/\psi,\ \mathrm{CP}/\varphi \qquad \mathrm{CSpec}_t/\lambda q\lambda p\lambda x[p(x) \wedge q(x)] \to$$
$$\mathrm{CP}/\varphi(\psi) \to \mathrm{CSpec}_t/\varphi,\ \mathrm{C}'_t/\psi \qquad \mathrm{C} \to$$
$$\mathrm{C}'_t/\lambda t[\varphi] \to \mathrm{C},\ \mathrm{S}/\varphi \qquad \mathrm{RelPro} \to \text{who}$$
$$\mathrm{NP}/\varphi \to \mathrm{RelPro}/\varphi.$$

In addition, we assume that, following generation of an S by these productions, 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 S 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 (new) trace t, which contributes the semantic value $\lambda p[p(t)]$. We denote by Syl+Rel the language defined by the above productions and rule of wh-movement. Again, for the sake of clarity, we have ignored the issue of agreement of relative pronouns with their antecedents (animate or inanimate).

The semantic information with which the above rules are augmented can then be understood as for our previous fragments, with meanings computed *before* wh-movement. Fig. 3 illustrates a typical derivation in Syl + Rel, with the arrow indicating wh-movement in the obvious way.

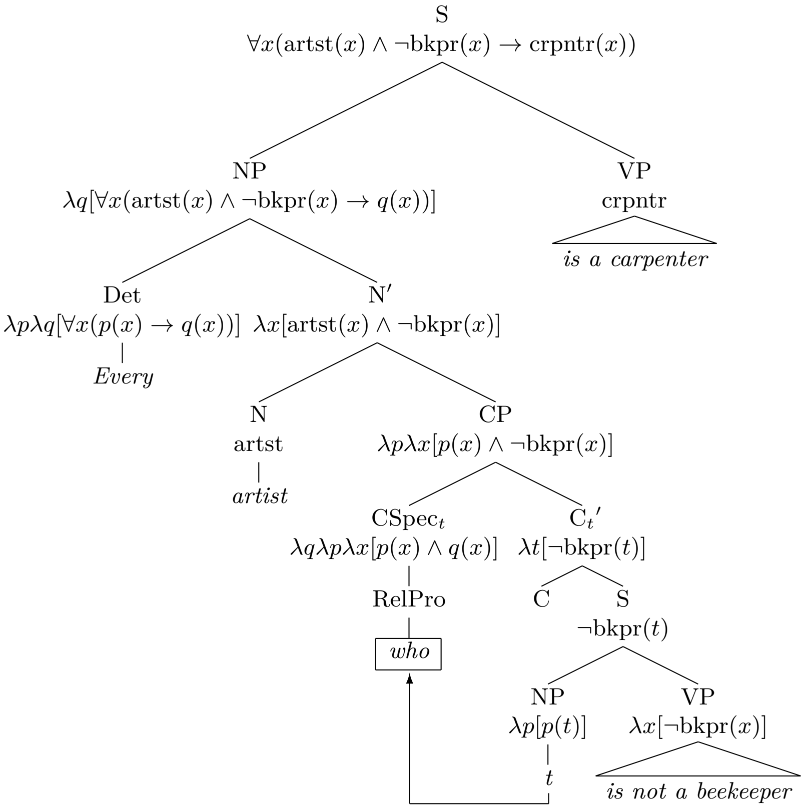

Figure 3. Meaning derivation in Syl + Rel.

The above rules for relative clauses can be unproblematically added to the collections of productions defining TV and DTV. Let the resulting fragments be denoted *TV+Rel* and *DTV+Rel*, respectively. The reader may easily verify that these productions yield the expected translations, as shown, for example, in Fig. 4. Note that, in this example, the source of wh-movement is the verb-

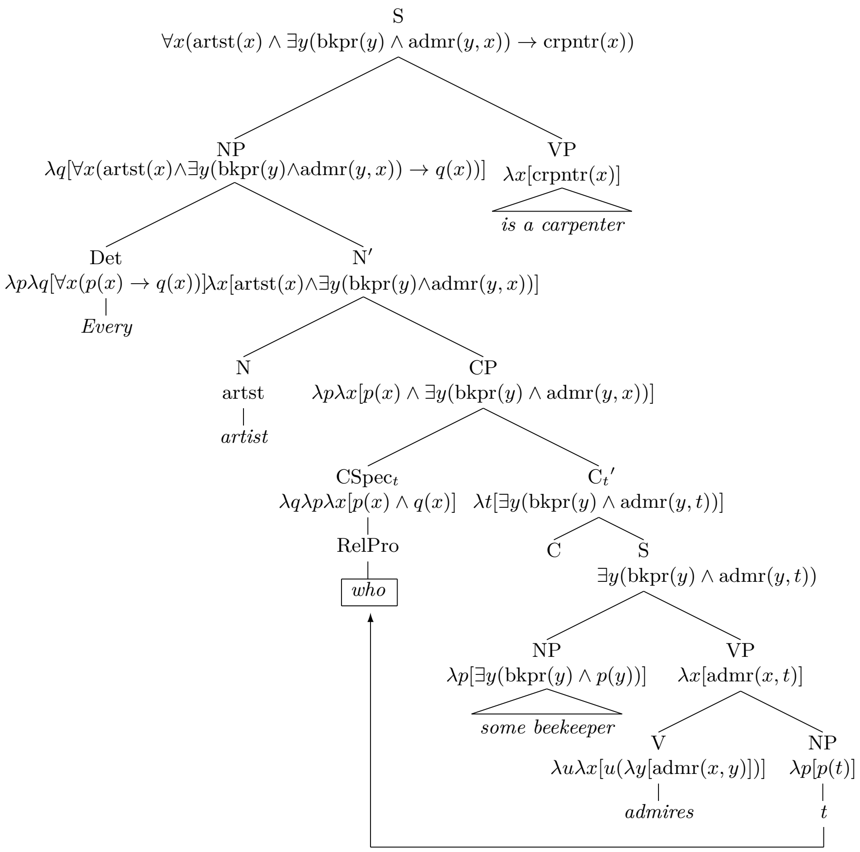

Figure 4. Meaning derivation in TV+Rel.

object; movement from the subject position works similarly. In DTV + Rel, movement is allowed from any of subject, direct-object or indirect-object positions. These grammars make no attempt to ban centre-embedded sentences. Thus, TV+Rel accepts

Every $[_{\text{N}'}$ artist who some
$[_{\text{N}'}$ beekeeper who some carpenter admires] despises]
hates some dentist,

and assigns it the meaning

$$\forall x(\text{artst}(x) \wedge \exists y(\text{bkpr}(y) \wedge \exists z(\text{crpntr}(z) \wedge \text{admr}(z, y)) \wedge \text{dspse}(y, x)) \rightarrow \exists y(\text{dntst}(y) \wedge \text{hate}(x, y))),$$

with similar remarks applying to DTV + Rel. We return to this matter in the sequel.

With these examples at our disposal, it is time to generalize. We take the syntax of a fragment $\mathcal{E}$ of some natural language to consist of a set of sentence forms of that language, but with certain open class categories replaced by countably infinite lexica, whose elements we regard as non-logical constants of some appropriate type. We take the semantics of the fragment to be a function that associates, to each sentence s in the fragment, a set of structures interpreting the non-logical primitives corresponding to the open-class lexical items occurring in s. If $\mathfrak{A}$ is one of the structures thus associated to s, we say that $\mathfrak{A}$ *satisfies* s, or that s is *true* in $\mathfrak{A}$, and write $\mathfrak{A} \models s$.

We must remove a potential source of misunderstanding at this point. In the foregoing examples, the class of structures associated with any sentence of the fragments we defined was specified by a formula of first-order logic. But this was purely a convenience: in associating to some sentence s a formula φ, our real intention was to associate to s a *class of structures* interpreting the relevant non-logical primitives—viz, the class of structures $\mathfrak{A}$ such that φ is true in $\mathfrak{A}$ according to the standard semantics of first-order logic. Thus, fragments of natural language are, for us, to be understood *purely extensionally*—they are simply sets of sentences together with a mapping taking each of these sentences to a class of models. In particular, the use of first-order logic does not embody any particular methodological assumption. Furthermore, the complexity-theoretic results reported below on fragments of English depend *only* on the extensions of those fragments, and not on any representation scheme used to describe them. Our approach thus contrasts with the psycholinguistic tradition referred to in Sec. 1, where the emphasis is on inferring the kinds of the kinds of mental representations from experimental data on performance in reasoning tasks.

With this in mind, let us define some of the key semantic concepts to be used in the sequel. A structure $\mathfrak{A}$ *satisfies* a set of $\mathcal{E}$-sentences S if it satisfies every element of S. An $\mathcal{E}$-sentence or set of $\mathcal{E}$-sentences is *satisfiable* if there exists a structure satisfying it. A set S of $\mathcal{E}$-sentences is taken to *entail* an $\mathcal{E}$-sentence s if every structure satisfying S also satisfies s. If S entails s, we write $S \models s$. It is uncontentious that, when applied to the fragments discussed in this Chapter, this notion of entailment adequately reconstructs the intuitive notion of validity of arguments. Finally, the fragment Syl contains sentences of the form *Some p is not a p*, which are unsatisfiable—i.e., false in every structure. We refer to any sentence having this form as an *absurdity.* (Since all the fragments we are concerned with include Syl, this notion of *absurdity* is as general as we require.)

If $\mathcal{E}$ is a fragment of some natural language, the main question we address is the complexity of the *satisfiability problem* for $\mathcal{E}$, denoted $\text{Sat}(\mathcal{E})$:

GIVEN: A finite set, S, of $\mathcal{E}$-sentences
OUTPUT: Yes, if S is satisfiable; No otherwise.

Closely related to $\text{Sat}(\mathcal{E})$ is the corresponding *entailment problem*:

GIVEN: A finite set, S, of $\mathcal{E}$-sentences and an $\mathcal{E}$-sentence s
OUTPUT: Yes, if $S \models s$; No otherwise.

The fragments defined in this Chapter all have an obvious notion of negation: if s is a sentence of any of these fragments, there is a sentence $\bar{s}$ such that $\mathfrak{A} \models s$ if and only if $\mathfrak{A} \not\models \bar{s}$. For such fragments, the satisfiability and entailment problems are dual in the usual sense: $S \models s$ if and only if $S \cup \{\bar{s}\}$ is unsatisfiable. Hence, knowing the complexity of either one gives us the complexity of the other. We concentrate in the sequel on $\text{Sat}(\mathcal{E})$.

3 Technical background

This section reviews the principal technical concepts we shall encounter in the sequel, and establishes notation.

We employ basic ideas from computational complexity theory. In this context, a *problem* is simply a set $\mathcal{P}$ of strings over some fixed alphabet Σ. Intuitively, we think of $\mathcal{P}$ as the task of deciding whether a given string over Σ is an element of the subset $\mathcal{P}$. A Turing machine (possibly nondeterministic) *recognizes* $\mathcal{P}$ if, for every string x over the relevant alphabet, it has a terminating run with input x and output 'Yes' just in case $x \in \mathcal{P}$. A problem $\mathcal{P}$ is *decidable* if it is recognized by a Turing machine which always terminates.

Decidable problems may be classified according to the computational resources required to decide them. Important complexity classes are: NLogSpace, the set of problems recognized by Turing machines using at most logarithmic working memory; NPTime, the set of problems recognized by Turing machines using at most polynomial time; and NExpTime, the set of problems recognized by Turing machines using at most exponential time. The complexity classes PTime and ExpTime are defined as for NPTime and NExpTime, but with the restriction that the Turing machine in question be deterministic. There are of course many other well-known complexity classes, but we shall not need them in this Chapter. We have

$$\text{NLogSpace} \subseteq \text{PTime} \subseteq \text{NPTime} \subseteq \text{ExpTime} \subseteq \text{NExpTime}.$$

Moreover, PTime is known to be a strict subset of ExpTime, and similarly for their non-deterministic counterparts.

One problem $\mathcal{P}$ over alphabet Σ can be *reduced to* a problem $\mathcal{P}'$ over alphabet Σ' if there is a function g, computable using logarithmically bounded working memory, mapping strings over Σ to strings over Σ', such that $x \in \mathcal{P}$ if and only if $g(x) \in \mathcal{P}'$. (Intuitively: any quick method for solving $\mathcal{P}'$ gives us a quick method for solving $\mathcal{P}$.) If $\mathcal{C}$ is a complexity class, we say that a problem is $\mathcal{C}$-*hard* if any problem in $\mathcal{C}$ can be reduced to it. A problem is $\mathcal{C}$-*complete* if it is in $\mathcal{C}$ and $\mathcal{C}$-*hard*. We may regard $\mathcal{C}$-complete problems as the hardest problems in $\mathcal{C}$. In practice, to show that a problem $\mathcal{P}$ is $\mathcal{C}$-hard, one usually takes a known $\mathcal{C}$-hard problem $\mathcal{P}'$, and reduces $\mathcal{P}'$ to $\mathcal{P}$. We note that all of the complexity classes considered in this Chapter are closed under reductions in the following sense: if a problem $\mathcal{P}$ is known to be in a class $\mathcal{C}$, and $\mathcal{P}'$ reduces to $\mathcal{P}$, then $\mathcal{P}'$ is in $\mathcal{C}$. This fact is often useful for establishing membership of problems in complexity classes.

In a similar way, we can establish that a problem $\mathcal{P}$ is undecidable by reducing a known undecidable problem $\mathcal{P}'$ to it. (This time, the reduction just needs to be computable—not necessarily computable using logarithmically bounded working memory.) One useful such undecidable problem $\mathcal{P}'$ is the *infinite tiling problem*. We are given a set of colours C and two binary relations H and V on C. The given instance $\langle C, H, V\rangle$ is positive just in case there exists a function $f : \mathbb{N} \times \mathbb{N} \to C$, called a *tiling*, such that for all $i, j \in \mathbb{N}$: (i) $\langle f(i,j), f(i+1,j) \rangle \in H$, and (ii) $\langle f(i,j), f(i,j+1) \rangle \in V$. Intuitively, we think of f as a colouring of the points of the infinite grid $\mathbb{N} \times \mathbb{N}$ with the colours C: the binary relation H tells us which colours may be placed immediately to the right of which others; the binary relation V tells us which colours may be placed immediately above which others.

We assume general familiarity with the syntax and semantics of propositional and first-order logic. A *fragment* of first-order logic is simply a set $\mathcal{L}$ of first-order formulas (usually infinite). For example, if $k \geqslant 1$, the *k-variable fragment*, denoted $\mathcal{L}^k$, is the set of first-order formulas featuring only the variables $x_1, \ldots, x_k$. The *satisfiability problem for* $\mathcal{L}$, denoted $\text{Sat}(\mathcal{L})$, is defined analogously to the satisfiability problem for fragments of English. The fragment $\mathcal{L}^1$ is not interestingly different from propositional logic, and its satisfiability problem is easily seen to be NPTime-complete. The satisfiability problem for $\mathcal{L}^2$ (with equality) was shown to be decidable by Mortimer (1975), and in fact to be NExpTime-complete by Grädel *et al.* (1997). For all larger k, $\text{Sat}(\mathcal{L}^k)$ is undecidable.

It will be convenient to extend the formalism of first-order logic with the so-called *counting quantifiers* $\exists_{>C}$ ("there exist more than C ...") and $\exists_{\leqslant C}$ ("there exists at most C ..."), where C is a bit string representing a natural number in the standard way. (In the sequel, we take the liberty of silently translating from binary into decimal notation for readability.) Within the context of first-order logic, counting quantifiers are always eliminable in favour of the standard quantifiers $\exists$ and $\forall$. Thus, for instance $\exists_{\leqslant 2} x.\varphi(x)$ is equivalent to $\forall x_1 \forall x_2 \forall x_3 (\varphi(x_1) \wedge \varphi(x_2) \wedge \varphi(x_3) \to (x_1 = x_2 \vee x_1 = x_3 \vee x_2 = x_3))$; and so on. Notice that these translations increase the number of variables used in the respective formulas. We denote the extensions of the fragments $\mathcal{L}^1$ and $\mathcal{L}^2$ to include counting quantifiers by $\mathcal{C}^1$ and $\mathcal{C}^2$, respectively. It is easy to show that the satisfiability problem for $\mathcal{C}^1$ is decidable; it was shown by Kuncak & Rinard (2007) that it is in fact NPTime-complete. (Membership in NPTime is by no means trivial, and relies on an interesting combinatorial argument due to Eisenbrand & Shmonin 2006.) The satisfiability problem for $\mathcal{C}^2$ was shown to be decidable by Pacholski *et al.* (1997) and Grädel *et al.* (1997), and in fact to be NExpTime-complete by Pratt-Hartmann (2005). Many other decidable fragments of first-order logic are known. However, the only other fragment that concerns us here is the so-called *fluted fragment*, introduced by Quine (1960). The definition of this fragment is too involved to reproduce here. Its satisfiability problem was shown to be decidable by Purdy (1996). (In fact, Purdy 2002 claims to present a proof that the problem is in NExpTime.)

A *literal* is an atomic formula (i.e. a predicate applied to the requisite number of arguments) or the negation of an atomic formula. A *clause* is a disjunction of literals. (The falsum $\bot$ counts as a clause, because we may regard it as the disjunction of the empty set of literals.) The *universal closure* of a clause γ is the formula $\forall x_1 \cdots \forall x_n.\gamma$, where $x_1, \ldots, x_n$ are the free variables of γ in some order. Given a finite set of first-order formulas Φ, we can compute (using at most logarithmically bounded working memory) a set of clauses Γ, such that Φ is satisfiable if and only if the universal closure of Γ is satisfiable. We call Γ the *clause-form* of Φ. Sets of clauses may be tested for unsatisfiability by the technique of *resolution theorem-proving*, which allows clauses to be derived from other clauses: Γ is unsatisfiable if and only if there is a resolution proof of $\bot$ from Γ. As we say: resolution theorem-proving is (*sound* and) *refutation-complete*. The reader is referred to Leitsch (1997) for a readable introduction to the resolution calculus and its refinements.

Resolution-theorem proving always terminates if the clauses to which it is applied are propositional, but not, in general, if they contain variables. *Ordered resolution* is a variant of this technique in which only certain clauses are allowed to combine to create new clauses. In certain cases it can be shown that ordering does not compromise refutation-completeness. At the same time, for sets of clauses with certain special properties—in particular those obtained from certain fragments of first-order logic—ordered resolution theorem-proving can be shown to terminate, and indeed to do so within a time-bound that can be computed in advance. Thus, for example, de Nivelle & Pratt-Hartmann (2001) presents an alternative proof that Sat($\mathcal{L}^2$) is in NExpTime, using this method. Such resolution-based arguments turn out to be particularly useful in obtaining upper complexity bounds for fragments of English, viz, Theorems 5, 7, 8, 10 and 11 below.

4 Syllogistic proof systems

In times past, the connection between logic and natural language must have seemed too transparent to merit discussion. The forms of the Classical Syllogistic (the language that we are calling Syl) are evidently linguistically inspired: the *Classical syllogisms* are the set of valid, two-premise argument-forms in this language. They include, for example,

$$\begin{array}{l} \text{Every } p \text{ is a } q \\ \text{Every } o \text{ is a } p \\ \hline \text{Every } o \text{ is a } q, \end{array} \qquad \begin{array}{l} \text{Every } p \text{ is a } q \\ \text{Some } o \text{ is a } p \\ \hline \text{Some } o \text{ is a } q, \end{array} \qquad \begin{array}{l} \text{No } p \text{ is a } q \\ \text{Some } o \text{ is a } p \\ \hline \text{Some } o \text{ is not a } q, \end{array}$$

which are sometimes known by their Mediæval mnemonics *Barbara*, *Darii* and *Ferio* respectively. Likewise, the following 1-premise arguments are traditionally known as *conversion rules*:

$$\frac{\text{Some } p \text{ is a } q}{\text{Some } q \text{ is a } p}, \qquad \frac{\text{No } p \text{ is a } q}{\text{No } q \text{ is a } p}.$$

But of course, a simple list of rules on its own is of little interest: their power comes from the possibility of chaining them together to demonstrate the validity of infinitely many valid arguments in the fragment Syl. Thus, for example, Argument (1) can be shown to be valid by means of the following derivation employing Darii, Ferio and conversion:

(4)
$$\cfrac{\cfrac{\cfrac{\cfrac{\text{Some artist is a carpenter}}{\text{Some carpenter is a artist}} \quad \text{Every artist is a beekeeper}}{\text{Some carpenter is a beekeeper}}}{\text{Some beekeeper is a carpenter}} \quad \text{No carpenter is a dentist}}{\text{Some beekeeper is not a dentist}}\ .$$

When logicians of later epochs attempted to overcome the obvious expressive poverty of this system, they naturally tried to mimic the Classical syllogisms for richer fragments of natural languages. This was to some extent evident in Mediæval logic, but particularly noticeable among various pre-Fregean logicians of the Nineteenth Century, such as Boole, De Morgan and Jevons. The technical apparatus at our disposal allows us to to complete the tasks that these writers set themselves—or, in some cases, to show that they cannot be completed. Our analysis will prove useful for deriving some of the complexity-theoretic results encountered below.

Let $\mathcal{E}$ be any fragment of a natural language. By a *syllogistic rule* in $\mathcal{E}$, we understand a pair S/s, where S is a finite set (possibly empty) of $\mathcal{E}$-sentences, and s an $\mathcal{E}$-sentence. We call S the *antecedents* of the rule, and s its *consequent*. All the Classical syllogisms are syllogistic rules in this sense, as we can see from the above presentations of Barbara, Darii and Ferio, where a horizontal line divides the antecedents from the consequent. A syllogistic rule is *valid* if its antecedents entail its consequent.

Let $\mathcal{E}$ be any fragment of a natural language, and X a set of syllogistic rules in $\mathcal{E}$; and denote by $\mathbb{P}(\mathcal{E})$ the set of subsets of $\mathcal{E}$. By a *substitution* for $\mathcal{E}$ we understand a function g which, for any category C of open class words in $\mathcal{E}$, (for example, nouns or transitive verbs), maps C to C. Substitutions are extended to $\mathcal{E}$-sentences in the obvious way: if g is a substitution and s an $\mathcal{E}$-sentence, $g(s)$ is the result of replacing any open-class word d in s by $g(d)$. We assume that categories are chosen such that $g(s)$ is guaranteed to be an $\mathcal{E}$-sentence. Substitutions are extended to sets of $\mathcal{E}$-sentences similarly. An *instance* of a syllogistic rule S/s is the syllogistic rule $g(S)/g(s)$, where g is a substitution. Formally, we define the *direct syllogistic derivation relation* $\vdash_{\mathsf{X}}$ to be the smallest relation on $\mathbb{P}(\mathcal{E}) \times \mathcal{E}$ satisfying:

(1) if $s \in S$, then $S \vdash_{\mathsf{X}} s$;
(2) if $\{s_1, \ldots, s_n\}/s$ is a syllogistic rule in X, g a substitution, $S = S_1 \cup \cdots \cup S_n$, and $S_i \vdash_{\mathsf{X}} g(s_i)$ for all i $(1 \leqslant i \leqslant n)$, then $S \vdash_{\mathsf{X}} g(s)$.

Thus, $S \vdash_{\mathsf{X}} s$ formalizes the existence of a derivation from premises S to conclusion s. We typically contract 'syllogistic rule' to 'rule'. The syllogistic derivation relation $\vdash_{\mathsf{X}}$ is said to be *sound* if $\Theta \vdash_{\mathsf{X}} \theta$ implies $\Theta \models \theta$. It is obvious that, for any set of rules X, $\vdash_{\mathsf{X}}$ is sound if and only if every rule in X is valid. Since the rules Darii and Ferio are clearly valid, Derivation (4) thus guarantees that the premises of Argument (1) entail the conclusion.

The syllogistic derivation relation $\vdash_{\mathsf{X}}$ is said to be *complete* if $\Theta \models \theta$ implies $\Theta \vdash_{\mathsf{X}} \theta$. Intuitively, this means that the rules defining it suffice for the entire fragment in question: any additional rules must be either derivable in terms of them, or invalid. Showing completeness is almost always more difficult than showing soundness, but is certainly possible, as we shall see below. For the purposes of this Chapter, it is helpful to use the following weakening of completeness. A set S of sentences is *inconsistent* (*with respect to* $\vdash_{\mathsf{X}}$) if $S \vdash_{\mathsf{X}} \bot$ for some absurdity $\bot$; otherwise, *consistent*. A derivation relation $\vdash_{\mathsf{X}}$ is *refutation-complete* if any unsatisfiable set S is inconsistent with respect to $\vdash_{\mathsf{X}}$. (In fact, we encountered this notion in the context of resolution theorem-proving in Section 3.) Completeness trivially implies refutation-completeness, but not conversely.

The semantic framework outlined in the previous section dates from the first half of the Twentieth Century; and the question of whether the Classical syllogisms are complete for the fragment Syl (in the sense of the previous paragraph) could not therefore have been formulated by logicians of earlier epochs. A moment's thought shows that at least some additional rules are needed—for example, a rule enabling us to infer *Some p is a p* from *Some p is a q*. But these can easily be provided. It was shown by Corcoran (1972) and Smiley (1973) that there exist finite sets of rules for Syl that are sound and refutation-complete; this was later slightly strengthened to full completeness by Pratt-Hartmann & Moss (2009):

Theorem 1. *There is a finite set of syllogism-like rules* X *in Syl such that* $\vdash_{\mathsf{X}}$ *is sound and complete.*

We omit the proof, which is technical (but not difficult). It is important to understand that Theorem 1 makes a statement about the infinity of valid arguments in Syl. As such, it constitutes an essential step forward from earlier accounts of the Classical syllogism, which could do little more than list the apparently valid two-premise argument forms (of which there is of course only a finite number).

The situation with TV is more complicated. Pratt-Hartmann & Moss (2009) go on to show:

Theorem 2. *There is a finite set of syllogism-like rules* X *in TV such that* $\vdash_{\mathsf{X}}$ *is sound and refutation-complete. However, there is no finite set of syllogism-like rules* X *in TV such that* $\vdash_{\mathsf{X}}$ *is sound and complete.*

We omit the proofs, which are technical. Again, it is important to understand that Theorem 2 makes a statement about the infinity of valid arguments in TV. As such, it is an essential advance on simply listing the (for example) valid two-premise argument forms in TV (Keane, 1969).

We mention in passing a variant of the Classical syllogistic—less obviously a counterpart of natural language, but nevertheless of historical interest. Łukasiewicz (1939) and Słupecki (1949) showed the completeness of a logic in which the sentence-forms of the the Classical syllogistic are embedded in the propositional calculus. (See also Łukasiewicz 1957.) Łukasiewicz claimed that this larger system represented Aristotle's actual conception of the syllogistic. Whatever the merits of this claim, this work probably represents the first serious completeness proof for anything resembling the syllogistic. The approach taken by Łukasiewicz and Słupecki is rather idiosyncratic—a more modern style of completeness proof for the same system is given by Shepherdson (1956). A version of the relational syllogistic similarly embedded in propositional logic is investigated by Nishihara *et al.* (1990); see also Ivanov & Vakarelov (2012). Curiously, Leibniz attempted to give a numerical semantics for the Classical syllogistic—a project which does turn out to be realizable (see Sotirov 2012); no interesting computational consequences result, however. To the author's knowledge, no one has attempted to provide a sound and (refutation-) complete system of rules for DTV.

5 Basic syllogistic fragments: complexity

In this section we analyse the complexity of satisfiability for the fragments Syl, TV and DTV, defined in Sec. 2.

Let us begin with some very simple upper complexity bounds. Recall from Sec. 3 the one- and two-variable fragments of first-order logic, $\mathcal{L}^1$ and $\mathcal{L}^2$: the satisfiability problems for these logics are NPTIME- and NEXPTIME-complete, respectively. Since Syl evidently translates into $\mathcal{L}^1$, Sat(Syl) is in NPTIME; and since TV evidently translates into $\mathcal{L}^2$, Sat(TV) is in NEXPTIME. Finally, although DTV-sentences require three variables, this fragment can nevertheless be shown to translate into the *fluted fragment* of first-order logic, whose satisfiability problem, as we remarked, is decidable. On the other hand, Syl, TV and DTV by no means exhaust the expressive power of the first-order fragments mentioned above, and it seems likely that these upper bounds can be improved on.

It will come as no surprise that the satisfiability problem for Syl has very low complexity.

Theorem 3. *The problem* Sat(Syl) *is* NLOGSPACE-*complete.*

Proof: To establish the lower bound, we reduce the problem of *un*-reachability in directed graphs to Sat(Syl). Let G be a directed graph with vertices V listed as $v_1, \ldots, v_n$, and edges E. The unreachability problem asks whether, given such a G, it is impossible to find a path from v_1 to v_n. This problem is known to be NLOGSPACE-complete. Taking the vertices in V to be common nouns, define S_G to be the set of (Syl)-sentences:

$$\{\text{Every } u \text{ is a } v \mid (u, v) \in E\} \cup \{\text{Some } v_1 \text{ is a } v_1, \text{No } v_n \text{ is a } v_n\}.$$

Let $\mathfrak{A}$ be the model over the 1-element domain $A = \{a\}$, and, for any $v \in V$, set $v^{\mathfrak{A}} = \{a\}$ if there is a path in G from v_1 to v, and $p^{\mathfrak{A}} = \emptyset$ otherwise. It is easy to verify that, if there is no path in G from v_1 to v_n, then $\mathfrak{A} \models S_G$. Conversely, it is obvious that, if there is a path in G from v_1 to v_n, then S_G cannot have a model, since, if v_i is reachable from v_1, then S_G entails that some v_is are v_is. Thus, G is a positive instance of directed graph unreachability if and only if S_G is satisfiable. This completes the reduction.

We consider next the upper bound. In the context of propositional logic, define a clause to be *Krom* if it contains at most two literals. The problem KROMSAT is defined as follows: given a set Γ of Krom clauses, determine whether Γ is satisfiable. It is known that KROMSAT is in NLOGSPACE. We reduce Sat(Syl) to it. Let S be a given set of Syl-sentences. For every sentence of the form *Some p is* (*not*) *a q*, write the corresponding pair of 1-literal clauses $p(a)$ and $\pm q(a)$, where a is a fresh constant; and for every sentence of the form *Every* (*No*) *p is a q*, write the corresponding set of 2-literal clauses $\neg p(a) \vee \pm q(a)$, for all constants a. Clearly, this transformation requires only logarithmic space, and the resulting set of Krom-clauses is satisfiable if and only if S is. □

Comparison of Arguments (1) and (2) above suggests that determining entailments in TV may be harder than in Syl. Our next result shows that, at least from the point of view of standard complexity classes, this is not the case: we retain the NLogSpace upper bound.

Theorem 4. *The problem* Sat(TV) *is* NLogSpace*-complete.*

Sketch proof: The lower bound is secured by Theorem 3. The matching upper bound can be established as follows. Recall that, by the first statement of Theorem 2, there exists a finite set of syllogistic rules X in TV such that $\vdash_{\mathsf{X}}$ is sound and refutation-complete. The proof of this fact in Pratt-Hartmann & Moss (2009) proceeds by constructing, from any unsatisfiable set S of TV-sentences, a derivation of an absurdity, using the rules of X. However, the derivation in question can be seen to have a special form. Let us say that a *B-chain* is a left-branching derivation involving only the rule Barbara, i.e. a derivation having the form

$$\frac{\begin{array}{c}\dfrac{\dfrac{\text{every } p_1 \text{ is a } p_2 \quad \text{every } p_2 \text{ is a } p_3}{\text{every } p_1 \text{ is a } p_3} \quad \text{every } p_3 \text{ is a } p_4}{\text{every } p_1 \text{ is a } p_4} \\ \ddots \\ \text{every } p_1 \text{ is a } p_{n-1}\end{array} \qquad \text{every } p_{n-1} \text{ is a } p_n}{\text{every } p_1 \text{ is a } p_n} .$$

It is shown that, if S is unsatisfiable, then there is a derivation of an absurdity from S featuring at most two B-chains, together with a fixed number of additional inference steps. Now let G be a directed graph whose vertices are the common nouns occurring in S and whose edges are those ordered pairs (p, q) for which *Every* p *is a* q is a sentence of S. It is easy to see that a B-chain connecting a pair of common nouns in S is simply a path connecting the corresponding pair of vertices in G. The required complexity bound then follows from the fact that the (un)reachability problem for directed graphs is in NLogSpace. □

The above argument thus shows that, while relational principles are certainly required to deal with arguments such as (2), they do not, from a complexity-theoretic point of view, make inference more difficult. The apparently greater difficulty of arguments such as (2) as compared to (1) is purely psychological.

Extending TV with ditransitive verbs, however, yields a modest increase in complexity:

Theorem 5. *The problem* Sat(DTV) *is* PTime*-complete.*

Proof sketch: The lower bound is relatively straightforward. In the context of propositional logic, define a clause to be *Horn* if it contains at most one negative literal. The problem HornSat, which is defined analogously to KromSat, is well-known to be PTIME-complete. We reduce this problem to Sat(DTV). Let Γ be a set of propositional Horn-clauses. We may without loss of generality assume that all clauses in Γ are of the forms in the left-hand column of Table 1. Further, we take all proposition letters in Γ to be common nouns, and we take d to be a ditransitive verb. Now replace each clause having any

$\neg o \vee \neg p \vee q$	Every o ds every p to some q
$\neg p$	No p is a p
p	Some p is a p

Table 1. Encoding Horn-clause satisfiability in DTV

of these forms with the corresponding DTV-sentence given in the right-hand column of Table 1. Let the resulting set of DTV-sentences be S_Γ. It is routine to show that Γ has a satisfying truth-value assignment if and only if S_Γ is satisfiable.

The upper bound is more complicated, and we can only indicate the broad strategy here. Let S be a given set of sentences in DTV. We compute, in polynomial time, the set Φ of first-order translations of S as defined by the semantics for DTV, convert to a set Γ of clauses, and apply resolution theorem-proving to try to derive a contradiction. The body of the proof involves showing that, by using a particular form of ordered resolution, we can ensure that the process terminates in polynomial time, and that, moreover, the clause $\perp$ (i.e. a contradiction) is obtained if and only if Φ—and hence S—is unsatisfiable. We remark that a simpler application of this strategy is used to prove Theorem 7 below, where it is possible to outline the details more fully. □

The above results on the Classical syllogistic make no reference to the long history of psychological research in this area, from the earliest investigations of Störing (1908); Woodworth & Sells (1935); Chapman & Chapman (1959), through to the influential Johnson-Laird (1983), and the long, many-sided debate it has generated. Psychological research in this area is dominated by the issue of which of a finite set of valid (or invalid) syllogistic forms human subjects to are likely to accept as valid. Of course, such facts can be of no relevance for us: all problems with finite domains are, from a complexity-theoretic viewpoint, decidable in constant time and space.

6 Relative clauses

Adding relative clauses to fragments of English which lack them frequently increases the complexity of determining satisfiability.

Theorem 6. *The problem* Sat(Syl + Rel) *is* NPTIME-*complete.*

Proof: Membership in NPTIME is instant from that fact that Syl + Rel translates into the 1-variable fragment of first-order logic, $\mathcal{L}^1$. We have therefore only to show NPTIME-hardness. In the context of propositional logic, the problem 3-SAT is defined as follows: given a set Γ of (propositional) clauses each of which has at most three literals, determine whether Γ is satisfiable. This problem is known to be NPTIME-complete; we reduce it to Sat(Syl+Rel). Let Γ be a set of formulas of propositional logic each of which has at most three literals. It is easily seen not to compromise NPTIME-hardness if we assume every $\gamma \in \Gamma$ to have one of the forms $p \vee q$, $\neg p \vee \neg q$ or $\neg o \vee \neg p \vee q$. We take the proposition letters of Γ to be common nouns, and take *element* to be a common noun. We then map each clause in Γ to a sentence of Syl + Rel as follows:

$p \vee q$	Every element which is not a q is a p
$\neg p \vee \neg q$	No p is a q
$\neg o \vee \neg p \vee q$	Every o which is a p is a q,

and finally add the Syl + Rel-sentence *Some element is an element.* Let the resulting set of Syl + Rel-sentences be S_Γ. It is routine to transform any satisfying truth-value assignment for Γ into a structure satisfying S_Γ, and vice versa. This completes the reduction. □

Theorem 7. *The problem* Sat(TV+Rel) *is* EXPTIME-*complete.*

Proof sketch: To show membership in EXPTIME, we describe a procedure to solve Sat(TV+Rel), and show that it runs in exponential time. That procedure makes use of the apparatus of *ordered resolution* theorem-proving for first-order logic, discussed briefly in Sec. 3. Let S be a finite set of TV+Rel-sentences, and let Φ be the set of their translations into first-order logic. Define a *special* formula recursively as follows: (i) if p is a unary predicate and x a variable, then $p(x)$ is a special formula; (ii) if p is a unary predicate, r a binary predicate, x, y variables and $\pi(x)$ a special formula, then

$$p(x) \wedge \pi(x) \qquad \neg\pi(x)$$
$$\exists y(\pi(y) \wedge r(x,y)) \qquad \exists y(\pi(y) \wedge r(y,x))$$
$$\forall y(\pi(y) \rightarrow r(x,y)) \qquad \forall y(\pi(y) \rightarrow r(y,x))$$

are special formulas.

A simple induction on the phrase-structures of TV+Rel-sentences shows that every N′ contributes a meaning of the form $\lambda x[\psi(x)]$, where ψ is (modulo trivial logical manipulations) a special 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

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

where o, p and q 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 exponential time.

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. This enables us to use ordered resolution on Γ, not to try to derive the absurdity $\bot$, but rather, to eliminate all these binary predicates in polynomial time. Specifically, we use ordered resolution on Γ to derive a set of clauses Γ', such that: (i) the universal closure of Γ has a model if and only if the universal closure of Γ' has; (ii) Γ' features only unary (not binary) predicates and only unary function-symbols; and (iii) Γ' is computed in polynomial time.

Since Γ' contains only unary predicates and only unary function-symbols, any clause in $\gamma(x,y) \in \Gamma'$ featuring two variables can be written as a disjunction $\gamma(x,y) = \gamma_1(x) \vee \gamma_2(y)$, where γ_1 and γ_2 each feature a single variable. According to the *splitting rule*, we may replace each such γ nondeterministically by either γ_1 or γ_2: the universal closure of Γ' is satisfiable if and only if, for some way of performing this splitting, a set of clauses results whose universal closure is satisfiable. By (i), Φ' is satisfiable if and only if the universal closure of Γ' is. By (ii), we can apply the splitting rule to every clause in Γ' to obtain clauses involving only one variable; to test the satisfiability of Φ', it therefore suffices to consider all possible choices for applying the splitting rule to clauses in Γ', and to determine whether the universal closure of at least one of the resulting clause sets is satisfiable. By (iii), $|\Gamma'|$ is at most polynomial in $|\Gamma|$, whence the number of choices generated by the splitting rule is at most exponential in $|\Phi'|$. The satisfiability of the universal closure of clause sets in which clauses feature just one variable can be decided using the technique of ordered resolution. Specifically, there exists a process of ordered resolution that, when applied to such a set of clauses, is guaranteed to reach saturation (no more clauses can be derived) after at most exponentially many steps. The universal closure of that clause set is then satisfiable if and only if the clause $\bot$ has not been derived by this point. To test the satisfiability of Φ, then, we first generate Φ, and then Γ and Γ', and simply check, for each clause set Δ obtained by applying the splitting rule to Γ', whether the universal closure of

Δ is satisfiable, reporting that Φ is satisfiable if we get at least one positive result. This concludes the description of our procedure to solve Sat(TV+Rel).

We now turn to ExpTime-hardness, taking as our point of departure the satisfiability problem for propositional modal logic with a universal modality, K^U, which is known to be ExpTime-complete. Essentially, K^U can be thought of as the set of first-order formulas of the following forms:

$$\forall x(\pm p(x) \rightarrow \pm q(x)) \qquad \forall x(\pm p(x) \rightarrow \forall y(\pm q(y) \rightarrow \neg r(x,y))) \tag{2}$$

$$\forall x(p(x) \wedge q(x) \rightarrow o(x)) \qquad \forall x(\pm p(x) \rightarrow \exists y(\pm q(y) \wedge r(x,y))), \tag{3}$$

where p and q range over all unary predicates, and r is a binary predicate. (We may assume that r is fixed: i.e. K^U features at most one binary predicate.)

We transform any such set of formulas, Φ, into a set of TV+Rel-sentences, S_Φ. The common nouns occurring in S_Φ will be all the unary predicates occurring in Φ together with the additional noun *element*; and the single transitive verb occurring in S_Φ will be r. We illustrate the transformation in a few cases only, as the generalization should be obvious:

$\forall x(p(x) \rightarrow q(x))$	Every p is a q
$\forall x(\neg p(x) \rightarrow q(x))$	Every element which is not a p is a q
$\forall x(p(x) \rightarrow \forall y(\neg q(y) \rightarrow \neg r(x,y)))$	No p rs any element which is not a q
$\forall x(p(x) \wedge q(x) \rightarrow o(x))$	Every p which is a q is an o. ...

In addition, we add to S_Φ the sentences

$$\{\text{Every } p \text{ is an element} \mid p \text{ occurring in } \Phi\} \cup \{\text{Some element is an element}\}.$$

Suppose $\mathfrak{A} \models \Phi$. Expand $\mathfrak{A}$ to a structure $\mathfrak{A}'$ by taking the denotation of a new unary predicate elmnt (translating the noun *element*) to be the entire domain. Then we obtain a structure satisfying S_Φ. Conversely, suppose $\mathfrak{A} \models S_\Phi$. Let $\mathfrak{A}'$ be the restriction of $\mathfrak{A}$ to the extension of the unary predicate elmnt. Then $\mathfrak{A} \models \Phi$. This completes the reduction of Sat(K^U) to Sat(TV+Rel). □

Let us now return to the issue raised briefly in Sec. 2, that our definition of TV+Rel allows centre-embedded sentences. Can we be sure that banning such sentences does not change the complexity of the fragment in question? Yes we can. Trivially, restricting the fragment cannot affect the upper complexity bound of its satisfiability problem, so we need only worry about establishing ExpTime-hardness. But all sentences in the set $T_\varphi \cup S_\varphi$ in the proof of Theorem 4 are grammatically unobjectionable, and in particular exhibit no centre-embedding. (In fact, they do not involve multiple relative clauses at all.) It follows that no linguistically motivated tightening of the fragment TV+Rel could possibly invalidate Theorem 7. As an aside, we remark that none of the sentences in $T_\varphi \cup S_\varphi$ involves object-relative clauses. Thus, determining logical relationships between TV+Rel-sentences with just subject-relative clauses is no easier than the general case.

For the fragment DTV, we have still higher complexity.

Theorem 8. *The problem* Sat(DTV + Rel) *is* NExpTime-*complete.*

The proof strategy is essentially the same as for Theorem 7: the upper complexity bound is established using a modified form of resolution theorem-proving; the lower bound is established by means of a reduction from tiling problems for exponential grids. Both proofs are quite involved, and the reader is referred to Pratt-Hartmann & Third (2006), Lemmas 4.5 and 4.7 for details.

7 Noun-level-negation

The Classical syllogistic, as commonly understood, does not include the sentence forms

Every non-p is a q	$\forall x(\neg p(x) \rightarrow q(x))$
Some non-p is not a q	$\exists x(\neg p(x) \wedge \neg q(x))$.

We take the fragment *Syl + Non* to be defined by the productions of Syl together with

$$\mathrm{N}'/\lambda x[\neg\varphi(x)] \rightarrow \text{non-}, \mathrm{N}/\varphi, \tag{4}$$

thus admitting the above sentences, with the given semantics. (Again, certain awkward sentence-forms such as *No p is a non-q* are also admitted; but these do not affect expressive power, and we do not trouble to filter them out.)

It is easy to see that this extension of Syl has no effect on the complexity of satisfiability, since Sat(Syl + Non) is evidently reducible to the satisfiability problem for Krom clauses in exactly the same way as Sat(Syl). Thus:

Theorem 9. *The problem* Sat(Syl + Non) *is* NLogSpace*-complete.*

Now let *TV+Non* be the fragment of English defined by the productions of TV together with (4), thus admitting sentences such as

Every non-artist admires some non-beekeeper
$\forall x(\neg\mathrm{artst}(x) \rightarrow \exists y(\neg\mathrm{bkpr}(y) \wedge \mathrm{admr}(x, y)))$.

This time, allowing noun-level negation results in a significant complexity jump. In fact, the *non*-construction is, in complexity theoretic terms, as harmful as relative clauses.

Theorem 10. *The problem* Sat(TV+Non) *is* ExpTime*-complete.*

Proof: Membership in ExpTime follows by exactly the same argument as for Theorem 7. For ExpTime-hardness, we again proceed by reduction from Sat(K^U); however, we no longer have relative clauses at our disposal to express K^U-formulas, and so must adopt a slightly different transformation scheme. Again, we illustrate with examples only: the generalization should be obvious:

$\forall x(p(x) \rightarrow q(x))$	Every p is a q
$\forall x(\neg p(x) \rightarrow q(x))$	Every non-p is a q
$\forall x(p(x) \rightarrow \forall y(\neg q(y) \rightarrow \neg r(x, y)))$	No p rs any non-q
...	

The only problematic case is formulas of the form

$$\forall x(p(x) \wedge q(x) \rightarrow o(x)), \tag{5}$$

which are essential for the ExpTime-completeness of K^U, and yet seem to require relative clauses for their expression. Let o^* be a new unary predicate.

For $\theta \in \Phi$ of the form (5), let r_θ be a new binary predicate, and define Θ_θ to be the set of formulas

$$\forall x(\neg o(x) \rightarrow \exists z(o^*(z) \wedge r_\theta(x,z))) \tag{6}$$
$$\forall x(p(x) \rightarrow \forall z(\neg p(z) \rightarrow \neg r_\theta(x,z))) \tag{7}$$
$$\forall x(q(x) \rightarrow \forall z(p(z) \rightarrow \neg r_\theta(x,z))), \tag{8}$$

which can all be translated by TV+Non-sentences. It is easy to check that $\Theta_\theta \models \theta$. For suppose (for contradiction) that $\mathfrak{A} \models \Theta_\theta$ and a satisfies p and q but not o in $\mathfrak{A}$. By (6), there exists b such that $\mathfrak{A} \models r_\theta[a,b]$. If $\mathfrak{A} \not\models p[b]$, then (7) is false in $\mathfrak{A}$; on the other hand, if $\mathfrak{A} \models p[b]$, then (8) is false in $\mathfrak{A}$. Thus, $\Theta_\theta \models \theta$ as claimed. Conversely, if $\mathfrak{A} \models \theta$, we can expand $\mathfrak{A}$ to a structure $\mathfrak{A}'$ by interpreting o^* and r_θ as follows:

$$\begin{aligned}(o^*)^{\mathfrak{A}} &= A\\ r_\theta^{\mathfrak{A}} &= \{\langle a,a\rangle \mid \mathfrak{A} \not\models o[a]\}.\end{aligned}$$

We check that $\mathfrak{A}' \models \Theta_\theta$. Formula (6) is true, because $\mathfrak{A}' \not\models o[a]$ implies $\mathfrak{A}' \models r_\theta[a,a]$. Formula (7) is true, because $\mathfrak{A}' \models r_\theta[a,b]$ implies $a = b$. To see that Formula (8) is true, suppose $\mathfrak{A}' \models q[a]$ and $\mathfrak{A}' \models p[b]$. If $a = b$, then $\mathfrak{A} \models o[a]$ (since $\mathfrak{A}' \models \theta$); that is, either $a \neq b$ or $\mathfrak{A} \models o[a]$. By construction, then, $\mathfrak{A}' \not\models r_\theta[a,b]$.

Now let Φ^* be the result of replacing all formulas θ in Φ of form (5) with the corresponding trio Θ_θ. (The binary predicates r_θ for the various θ are assumed to be distinct; however, the same unary predicate o^* can be used for all θ.) By the previous paragraph, Φ^* is satisfiable if and only if Φ is satisfiable. But Φ^* can evidently be translated into a set of TV+Non-sentences satisfied in exactly the same structures. This completes the reduction of K^U to TV+Non. □

Now let *DTV+Non* be the fragment of English defined by the productions of DTV together with (4), thus admitting sentences such as

Every non-artist recommends some non-beekeeper to some non-carpenter
$\forall x(\neg \text{artst}(x) \rightarrow \exists y(\neg \text{bkpr}(y) \wedge \exists z(\neg \text{crpntr}(z) \wedge \text{rcmnds}(x,y))))$.

The effect is just as dramatic as with TV+Non:

Theorem 11. *The problem* Sat(DTV + Non) *is* NExpTime*-complete.*

Proof sketch: The lower bound is obtained using the same strategy as with Theorem 10: the *non*-construction is used to duplicate the effect of relative clauses. The upper bound follows using a similar strategy to that employed for DTV + Rel (Theorem 8). □

8 Numerical determiners

Replacing the determiners *some* and *no* in the fragments Syl and TV with the phrases *more than C* and *at most C*, allows us to express arguments with a combinatorial flavour. Consider, for example:

$$\frac{\begin{array}{l}\text{More than 12 artists are beekeepers}\\ \text{At most 3 beekeepers are carpenters}\\ \text{At most 4 dentists are not carpenters}\end{array}}{\text{More than 5 artists are not dentists.}} \tag{9}$$

Argument (9) is evidently valid. Indeed, suppose the premises are true: take any collection of thirteen artists who are beekeepers; since at most three of these may be carpenters, at least ten must be non-carpenters; and since, of these ten, at most four may be dentists, at least six must be non-dentists.

Considerably more thought shows the argument

$$\frac{\begin{array}{l}\text{At most 1 artist admires at most 7 beekeepers}\\ \text{At most 2 carpenters admire at most 8 dentists}\\ \text{At most 3 artists admire more than 6 electricians}\\ \text{At most 4 beekeepers are not electricians}\\ \text{At most 5 dentists are not electricians}\\ \text{At most 1 beekeeper is a dentist}\end{array}}{\text{At most 6 artists are carpenters}} \tag{10}$$

to be likewise valid (assuming, that is, that the quantified subjects in these sentences scope over their respective objects). Indeed, suppose to the contrary that its premises are true, but its conclusion false. By the negation of the conclusion, take any collection of seven artists who are carpenters; by the first two premises, at least four of these seven must admire eight or more beekeepers and nine or more dentists; and by the third premise, at least one of these four satisfies the additional property of admiring at most six electricians. Let a be such an artist, then, and consider any set of eight beekeepers and any set of nine dentists admired by a: by the fourth and fifth premises, respectively, at least four of these beekeepers and four of these dentists must be electricians. But since a admires only six electricians altogether, these sets of four beekeepers and four dentists must overlap by at least two, which contradicts the final premise.

These observations suggest adding productions to our fragments to handle numerical determiners. Using the notation of counting quantifiers from Sec. 3, we define the fragments *Syl+ Num*, *TV+Num* and *DTV+ Num* by adding to the respective grammars for Syl, TV and DTV the infinite set of productions

$$\begin{array}{l}\text{Det}/\lambda p\lambda q[\exists_{>C}x(p(x) \wedge q(x))] \rightarrow \text{more than } C\\ \text{Det}/\lambda p\lambda q[\exists_{\leqslant C}x(p(x) \wedge q(x))] \rightarrow \text{at most } C,\end{array}$$

where C ranges over all (finite) bit strings. (We continue to translate from binary into decimal notation for readability.) These productions yield the expected translations, for example:

More than 12 artists are beekeepers
$\exists_{>12}(\text{artst}(x) \wedge \text{bkpr}(x))$

At most 1 artist admires at most 7 beekeepers
$\exists_{\leqslant 1}(\text{artst}(x) \wedge \exists_{\leqslant 7} y(\text{bkpr}(y) \wedge \text{admr}(x, y)))$.

For k positive, we define *Syl + Num$_k$* to be the fragment of Syl_k in which all numbers in determiners are bounded by k; and similarly for *TV+Num$_k$* and *DTV + Num$_k$*. Thus, *More than 12 artists are beekeepers* is in $\text{Syl} + \text{Num}_{12}$ but not in $\text{Syl} + \text{Num}_{11}$.

Some readers may wonder whether the incorporation of determiners featuring bit strings is really a fragment of English, but we could easily replace them with familiar number words. (The fact that this language is not context-free is not a problem: there is no methodological commitment to specifying fragments of languages by means of context-free grammars.) Moreover, most of the results we report below for Syl + Num, TV+Num and DTV + Num hold for their finite-form variants $\text{Syl} + \text{Num}_k$, TV+Num_k and $\text{DTV} + \text{Num}_k$ for *all* positive k. And surely $\text{Syl} + \text{Num}_1$, TV+Num_1 and $\text{DTV} + \text{Num}_1$ are fragments of English. Finally, in view of the obvious logical equivalences $\exists_{>0} x.\varphi \equiv \exists x.\varphi$ and $\exists_{\leqslant 0} x.\varphi \equiv \forall x.\neg\varphi$ we may henceforth ignore the standard determiners *some*, *all* and *no* in our subsequent discussion of fragments with numerical determiners.

It is easy to see that Syl + Num translates into $\mathcal{C}^1$, the 1-variable fragment of first-order logic with counting quantifiers. Since, as we observed in Sec. 3, $\text{Sat}(\mathcal{C}^1)$ is in NPTime, so too is Sat(Syl+Num). Likewise, TV+Num translates into $\mathcal{C}^2$, whence Sat(TV) is in NExpTime. On the other hand, since Syl+Num, and TV+Num by no means exhaust the expressive power of $\mathcal{C}^1$ or $\mathcal{C}^2$, the question arises as to whether these upper bounds can be improved on. The next theorems show that they cannot.

Theorem 12. *The problems* Sat(Syl + Num) *and* $\text{Sat}(\text{Syl} + \text{Num}_k)$*, for all positive k, are* NPTime*-complete.*

Proof: The upper bound is immediate from the fact that Syl + Num translates into $\mathcal{C}^1$. We establish a matching lower bound. If G is an undirected graph, a 3-*colouring* of G is a function t mapping the vertices of G to the set $\{0, 1, 2\}$ such that no edge of G joins two vertices mapped to the same value. We say that G is 3-*colourable* if a 3-colouring of G exists. The problem of deciding whether a given graph G is 3-colourable is well-known to be NP-hard. We first reduce it to $\text{Sat}(\text{Syl} + \text{Num}_3)$.

Let the vertices of G be $\{1, \ldots, n\}$. Let p be a common noun, and, for all i ($1 \leqslant i \leqslant n$) and k ($0 \leqslant k < 3$), let p_i^k be a fresh common noun. Think of p as denoting a 'selected' colouring of G, and think of p_i^k as denoting a 'selected' colouring of G in which vertex i has colour k. (The trick will be to consider a universe containing three 'selected' colourings.) Now let S_G be the set of Syl + Num-sentences consisting of

$$\text{At most 3 } p \text{ are } p \tag{11}$$

$$\{\text{At most 0 } p_i^j \text{ are } p_i^k\text{s} \mid 1 \leqslant i \leqslant n, 0 \leqslant j < k < 3\} \tag{12}$$

$$\{\text{At least 1 } p_i^k \text{ is a } p \mid 1 \leqslant i \leqslant n, 0 \leqslant k < 3\} \tag{13}$$

$$\{\text{At most 0 } p_i^k \text{ are } p_j^k\text{s} \mid (i,j) \text{ is an edge of } G, 0 \leqslant k < 3\} \tag{14}$$

We prove that S_G is satisfiable if and only if G is 3-colourable.

Suppose $\mathfrak{A} \models S_G$. By (11), $|p^{\mathfrak{A}}| \leqslant 3$. Fix any i ($1 \leqslant i \leqslant n$). No $a \in p^{\mathfrak{A}}$ satisfies any two of the predicates p_i^0, p_i^1, p_i^2, by (12); on the other hand, each of these predicates is satisfied by at least one element of $p^{\mathfrak{A}}$, by (13); therefore, $|p^{\mathfrak{A}}| = 3$, and each element a of $p^{\mathfrak{A}}$ satisfies exactly one of the predicates p_i^0, p_i^1, p_i^2. Now fix any $a \in p^{\mathfrak{A}}$, and, for all i ($1 \leqslant i \leqslant n$), define $t_a(i)$ to be the unique k ($1 \leqslant k < 3$) such that $\mathfrak{A} \models p_i^k[a]$, by the above argument. The formulas (14) then ensure that t_a defines a colouring of G. Conversely, suppose that $t : \{1, \ldots, n\} \to \{0, 1, 2\}$ defines a colouring of G. Let $\mathfrak{A}$ be a structure with domain $A = \{0, 1, 2\}$; let all three elements satisfy p; and, for all $k \in A$, let p_i^k be satisfied by the single element $k + t(i)$ (where the addition is modulo 3). It is routine to verify that $\mathfrak{A} \models S_G$. This completes the reduction of graph 3-colourability to Sat(Syl + Num_3).

The next step is to reduce Sat(Syl + Num_3) to Sat(Syl + Num_1). More precisely, inspection of (11)–(14) shows that we we may restrict attention to the subset of Syl + Num_3 in which the only sentences lying outside Syl + Num_1 are those of the form *At most 3 p are p*. Let S be any such set of Syl + Num_3-sentences, then. For any sentence s = *At most 3 p are p*, let o, o' be new common nouns, and replace s by the Syl + Num_1-sentences

At most 1 p is not o
At most 1 o is o'
At most 1 o is not o'.

Let the resulting set of Syl + Num_1-sentences be T. Evidently, T entails every sentence of S; conversely, any structure $\mathfrak{A}$ such that $\mathfrak{A} \models S$ can easily be expanded to a structure $\mathfrak{B}$ such that $\mathfrak{B} \models T$. This completes the reduction, and the proof that Sat(Syl + Num_1) is NPTime-hard. It follows, of course that every Sat(Syl + Num_k) for k positive, and indeed Sat(Syl + Num) are all NPTime-hard. □

Theorem 13. *The problem* Sat(TV+Num) *is* NExpTime*-complete.*

Proof: The upper bound is immediate from the fact that TV+Num translates into $\mathcal{C}^2$. The matching lower bound is obtained by a relatively straightforward reduction of exponential tiling problems to Sat(TV+Num). Details may be found in Pratt-Hartmann (2008), Theorem 3. □

We remark that TV+Num is certainly not the whole of $\mathcal{C}^2$. In particular, it is also shown in Pratt-Hartmann (2008), Lemma 5, that TV+Num has the *finite model property*: if a finite set of TV+Num-sentences is satisfiable, then it is satisfiable in a finite structure. This is easily seen not to be the case for $\mathcal{C}^2$.

Tight complexity bounds for the problems TV+Num$_k$, DTV + Num and DTV + Num$_k$, for k positive, are currently not known. Indeed, it is not known whether DTV + Num$_k$, for k positive, is even decidable.

We round this section off by stating a further negative result on the existence of syllogism-like proof-systems for fragments of English with numerical determiners (Pratt-Hartmann, 2013).

Theorem 14. *There is no finite set of syllogism-like rules* X *in Syl + Num such that the indirect proof-system* $\vdash_{\mathsf{X}}$ *is sound and refutation-complete. If* $k > 0$*, then there is no finite set of syllogism-like rules* X *in Syl + Num$_k$ such that the direct proof-system* $\vdash_{\mathsf{X}}$ *is sound and refutation-complete.*

Again, the proof is technical, and we omit it.

9 Bound-variable anaphora

There are still many simple arguments that cannot be captured by the fragments considered above. Here is one:

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

(Of course, we are assuming here that the pronouns above are resolved intrasententially.) The next question is what happens to the computational complexity of the satisfiability problem when pronouns and reflexives are admitted.

The syntax of such a fragment may be defined by adding to the following productions to those of of TV+Rel:

Syntax	Formal lexicon	
NP → Reflexive	Reflexive → himself	(15)
NP → Pronoun	Pronoun → him	

For simplicity, we have suppressed the semantic annotations required to produce the standard translations into first-order logic, since these would require tedious and inessential modifications to the productions already given for TV+Rel. As the correct translations are anyway not in dispute, we leave the issue of their formal specification to the interested reader. For one possibility, see (Pratt-Hartmann, 2003).

Two semantic issues, however, do require clarification before we proceed. First of all, we always take pronouns and reflexives to have antecedents in the sentences in which they occur. That is to say: all anaphora is intra-sentential. We further assume the selection of such antecedents to be subject to the usual rules of binding theory, which we need not rehearse here. Of course there is nothing wrong with sentences in which *he* refers to an object identified in some earlier sentence; however, referential pronouns are, from a logical point of view, equivalent to proper nouns.

Secondly, the above productions generate sentences featuring anaphoric ambiguities. Thus, for example, in

Every artist who admires a beekeeper hates
every carpenter who despises him, (16)

the pronoun may take as antecedent either the NP headed by *artist* or the NP headed by *beekeeper*. (The NP headed by *carpenter* is not available as a pronoun antecedent here according to binding theory.) These two indexation patterns correspond, respectively, to the first-order translations

$$\forall x(\text{artst}(x) \wedge \exists y(\text{bkpr}(y) \wedge \text{admr}(x,y)) \rightarrow \forall z(\text{crpntr}(z) \wedge \text{dsps}(z,x) \rightarrow \text{hte}(x,z))) \quad (17)$$

$$\forall x \forall y(\text{artst}(x) \wedge \text{bkpr}(y) \wedge \text{admr}(x,y) \rightarrow \forall z(\text{crpntr}(z) \wedge \text{dsps}(z,y) \rightarrow \text{hte}(x,z))). \quad (18)$$

In defining fragments of English equipped with anaphora, therefore, we must decide how to treat ambiguities.

Two options present themselves. The first is to adopt a general method of resolving ambiguities by artificial stipulation; the second is to decorate nouns and pronouns in these sentences with indices specifying which pronouns take which antecedents. Considering the former option, let the semantics assigned to each sentence of our fragment incorporate the artificial stipulation that *pronouns must take their closest allowed antecedents*. Here, *closest* means "closest measured along edges of the phrase-structure" and *allowed* means "allowed by the principles of binding theory". (We ignore case and gender agreement.) Fig. 5 illustrates this restriction for sentence (16). Evidently, the NP headed by *artist* is closer, in the relevant sense, to the pronoun *him* than is the NP headed by *beekeeper*. Since co-indexing the pronoun with the NP headed by *artist* corresponds to the sentence-meaning captured by formula (17), we take this to be the meaning of (16). Let us denote the resulting fragment of English by *TV+Rel+RA* (transitive verbs, relative clauses and *restricted* anaphora).

Turning our attention now to the latter option for dealing with anaphoric ambiguity, denote by TV+Rel+GA the same fragment as TV+Rel+RA, except that anaphoric antecedents are indicated by co-indexing, subject only to the rules of binding theory (transitive verbs, relative clauses and *general* anaphora). Again, we omit a specification of the semantics which yield the generally accepted translations of indexed sentences; but this can be provided as a matter of routine. Strictly speaking, TV+Rel+GA-sentences are not English sentences, but rather, English sentences with pronomial antecedents explicitly indicated. In particular, sentence (16) corresponds to two essentially distinct indexed sentences of TV+Rel+GA, depending on which NP the pronoun takes as its antecedent. One of these indexed sentences translates to the formula (17), the other, to the formula (18). We remark that the computational complexity of recovering the possible antecedents of anaphors in TV+Rel+RA and TV+Rel+GA (and of producing the corresponding first-order translations) is so low that we may ignore it in the sequel.

Having defined the fragments TV+Rel+RA and TV+Rel+GA, we turn to the complexity of their respective satisfiability problems. Recall from Sec. 1 that the satisfiability problem for $\mathcal{L}^2$, the two-variable fragment of first-order logic, is NExpTime-complete. It transpires that TV+Rel+RA corresponds closely to $\mathcal{L}^2$. More precisely, the following result is shown in Pratt-Hartmann (2003).

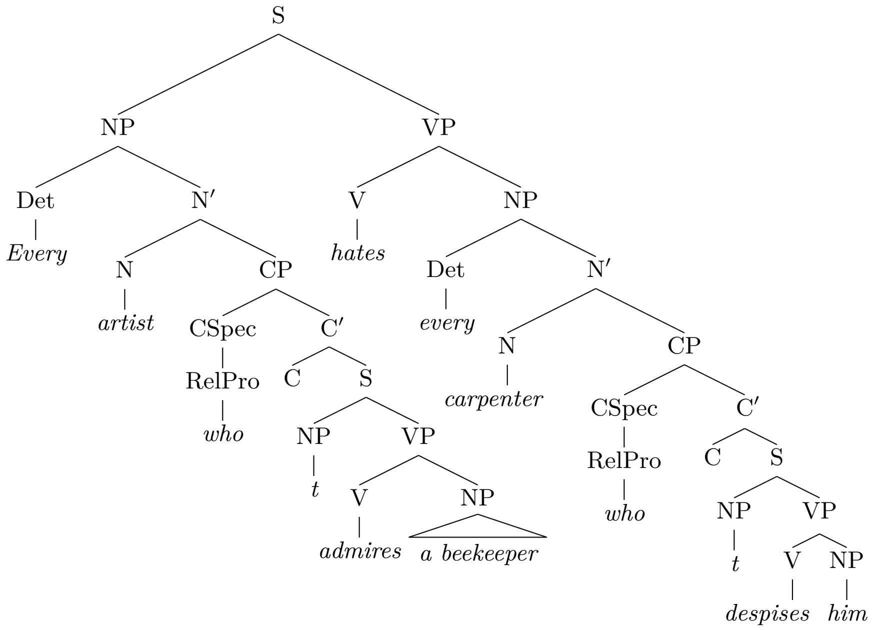

Figure 5. Parsing in TV+Rel+RA.

Theorem 15. *The problem* Sat(TV+Rel+RA) *is* NExpTime*-complete.*

Proof sketch: Using the semantics of TV+Rel+RA, it is possible to show that any first-order formula obtained as the translation of a TV+Rel+RA-sentence has the property that none of its subformulas contains more than two free-variables. Any such formula is easily seen to be equivalent to a formula of $\mathcal{L}^2$. This guarantees that Sat(TV+Rel+RA) is in NExpTime.

To show that Sat(TV+Rel+RA) is NExpTime-hard, we show that, for any $\mathcal{L}^2$-formula φ, we can compute, in logarithmic space, a set S_φ of TV+Rel+RA-sentences, over a suitable lexicon of common nouns and transitive verbs, with the properties that: (i) any structure in which the sentences of S_φ are all true is one in which φ is true; (ii) any structure in which φ is true can be expanded to a structure in which the sentences of S_φ are all true. Thus, we have reduced the satisfiability problem for $\mathcal{L}^2$ to Sat(TV+Rel+RA). □

The next theorem shows that abandoning the restriction on anaphora in TV+Rel+RA leads to undecidability (Pratt-Hartmann, 2003).

Theorem 16. *The problem* Sat(TV+Rel+GA) *is undecidable.*

Proof: Consider the following TV+Rel+GA-sentences, together with their first-order translations. We have gathered them into groups for ease of understanding. The first group asserts that a *vertex* exists, and that every vertex is related to vertices by a *horizontal* relation, h, and by a *vertical* relation, v:

Some vertex is a vertex
$\exists x(\text{vertex}(x) \wedge \text{vertex}(x))$

Every vertex aiches some vertex
$\forall x(\text{vertex}(x) \rightarrow \exists y(\text{vertex}(y) \wedge h(x,y)))$

Every vertex vees some vertex
$\forall x(\text{vertex}(x) \rightarrow \exists y(\text{vertex}(y) \wedge v(x,y)))$.

We will also employ a *diagonal* relation d. The second group sets up the converses of the relations h, v and d:

Every vertex$_1$ anti-aiches every vertex which aiches it$_1$
$\forall x(\text{vertex}(x) \rightarrow \forall y(\text{vertex}(y) \wedge h(y,x)) \rightarrow h^{-1}(x,y))$

Every vertex$_1$ anti-vees every vertex which vees it$_1$
$\forall x(\text{vertex}(x) \rightarrow \forall y(\text{vertex}(y) \wedge v(y,x)) \rightarrow v^{-1}(x,y))$

Every vertex$_1$ anti-dees every vertex which dees it$_1$
$\forall x(\text{vertex}(x) \rightarrow \forall y(\text{vertex}(y) \wedge d(y,x)) \rightarrow d^{-1}(x,y))$.

Finally, using the diagonal relation, we write sentences ensuring that that the vertical successor of a horizontal successor of any vertex is also a horizontal successor of a vertical successor of that vertex:

Every vertex$_1$ dees every vertex which anti-vees some vertex which it$_1$ aiches
$\forall x(\text{vertex}(x) \rightarrow \forall z(\text{vertex}(z) \wedge \exists y(\text{vertex}(y) \wedge h(x,y) \wedge v^{-1}(z,y)) \rightarrow d(x,z)))$

Every vertex$_1$ aiches every vertex which anti-dees some vertex which vees it$_1$
$\forall x(\text{vertex}(x) \rightarrow \forall z(\text{vertex}(z) \wedge \exists y(\text{vertex}(y) \wedge v(y,x) \wedge d^{-1}(z,y)) \rightarrow h(x,z)))$.

These sentences are evidently true in the infinite structure $\mathfrak{A}$ with domain $A = \mathbb{N}\times\mathbb{N}$, $\text{vertex}^{\mathfrak{A}} = A$, $h^{\mathfrak{A}} = \{\langle(i,j),(i{+}1,j)\rangle \mid i,j,\in\mathbb{N}\}$, $v^{\mathfrak{A}} = \{\langle(i,j),(i,j{+}1)\rangle \mid i,j,\in\mathbb{N}\}$ and $d^{\mathfrak{A}} = \{\langle(i,j),(i{+}1,j{+}1)\rangle \mid i,j,\in\mathbb{N}\}$, and with h^{-1}, v^{-1} and d^{-1} the converses of h, v and d, respectively. On the other hand, if $\mathfrak{B}$ is a structure making these sentences true, then it is straightforward to show that $\mathfrak{A}$ can be homomorphically embedded in $\mathfrak{B}$. It is then a routine matter to encode the infinite tiling problem (see Sec. 3) as a collection of TV+Rel+GA-sentences. This establishes the undecidability of Sat(TV+Rel+GA). □

Finally, we consider the addition of bound-variable anaphora to DTV+Rel, which proceeds exactly as for TV+Rel. Denote the resulting fragments by *DTV+ Rel+ RA* (anaphora restricted to the closest available antecedent) and *DTV+ Rel+ GA* (anaphoric references explicitly indicated). We know from Theorem 16 that the latter must have an undecidable satisfiability problem. It is shown in Pratt-Hartmann & Third (2006) that the former has too.

Theorem 17. *The problem* $\mathrm{Sat}(\mathrm{DTV} + \mathrm{Rel} + \mathrm{RA})$ *is undecidable.*

Proof: Similar to Theorem 16. □

References

Chapman, L. & J. Chapman (1959), Atmosphere effect re-examined, *Journal of Experimental Psychology* 58:220–226.

Corcoran, J. (1972), Completeness of an ancient logic, *Journal of Symbolic Logic* 37(4):696–702.

Eisenbrand, F. & G. Shmonin (2006), Carathéodory bounds for integer cones, *Operations Research Letters* 34(5):564–568.

Fitch, F. (1973), Natural deduction rules for English, *Philosophical Studies* 24:89–104.

Fyodorov, Y., Y. Winter, & N. Francez (2003), Order-based inference in natural logic, *Logic Journal of the IGPL* 11(4):385—-417.

Grädel, E., P. Kolaitis, & M. Vardi (1997), On the decision problem for two-variable first-order logic, *Bulletin of Symbolic Logic* 3(1):53–69.

Grädel, E., M. Otto, & E. Rosen (1997), Two-variable logic with counting is decidable, in *Logic in Computer Science*, IEEE, (306–317).

Icard, T. (2012), Inclusion and exclusion in natural language, *Studia Logica* 100(4):705–725.

Ivanov, N. & D. Vakarelov (2012), A system of relational syllogistic incorporating full Boolean reasoning, *Journal of Logic, Language and Information* 21:433–459.

Jackendoff, R. (1987), *Consciousness and the Computational Mind*, MIT Press, Cambridge, MA.

Johnson-Laird, P. (1983), *Mental Models*, Cambridge University Press, Cambridge.

Keane, G. (1969), *The Relational Syllogism: A Systematic Approach*, University of Exeter Printing Unit, Exeter.

Kuncak, V. & M. Rinard (2007), Towards efficient satisfiability checking for Boolean algebra with Presburger arithmetic, in F. Pfenning (ed.), *Proceedings, 21st International Conference on Automated Deduction (CADE-21)*, Springer, Berlin, volume 4603 of *Lecture Notes in Computer Science*, (215–230).

Leitsch, A. (1997), *The Resolution Calculus*, Springer.

Łukasiewicz, J. (1939), O sylogistyce Arystotelesa, *Sprawozdania z Czynności i Posiedeń Polskiej Akademii Umiejętności* 44(6):220–227.

Łukasiewicz, J. (1957), *Aristotle's Syllogistic*, Clarendon Press, Oxford, 2nd edition.

MacCartney, B. & C. Manning (2008), Modeling semantic containment and exclusion in natural language inference, in *Proceedings of COLING*, ACL, (521–528).

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

Moss, L. (2012), The soundness of internalized polarity marking, *Studia Logica* 100(4):683–704.

Nishihara, N., K. Morita, & S. Iwata (1990), An extended syllogistic system with verbs and proper nouns, and its completeness proof, *Systems and Computers in Japan* 21(1):760–771.

de Nivelle, H. & I. Pratt-Hartmann (2001), A resolution-based decision procedure of the two-variable fragment with equality, in R. Gore, A. Leitsch, & T. Nipkow (eds.), *Automated Reasoning*, Springer, Berlin, volume 2083 of *Lecture Notes in Computer Science*, (211–255).

Pacholski, L., W. Szwast, & L. Tendera (1997), Complexity of two-variable logic with counting, in *Logic in Computer Science*, IEEE, (318–327).

Pratt-Hartmann, I. (2003), A two-variable fragment of English, *Journal of Logic, Language and Information* 12(1):13–45.
Pratt-Hartmann, I. (2005), Complexity of the two-variable fragment with counting quantifiers, *Journal of Logic, Language and Information* 14:369–395.
Pratt-Hartmann, I. (2008), On the computational complexity of the numerically definite syllogistic and related logics, *Bulletin of Symbolic Logic* 14(1):1–28.
Pratt-Hartmann, I. (2013), The syllogistic with unity, *Journal of Philosophical Logic* 42(2):391–407.
Pratt-Hartmann, I. & L. Moss (2009), Logics for the relational syllogistic, *Review of Symbolic Logic* 2(4):647–683.
Pratt-Hartmann, I. & A. Third (2006), Fragments of language: the case of ditransitive verbs, *Notre Dame Journal of Formal Logic* 47(2):151–177.
Purdy, W. (1996), Fluted formulas and the limits of decidability, *Journal of Symbolic Logic* 61(2):608–620.
Purdy, W. (2002), The complexity and nicety of fluted logic, *Studia Logica* 71:177–198.
Quine, W. (1960), Variables explained away, *Proceedings of the American Philosophical Society* 104:343–347.
Shepherdson, J. (1956), On the interpretation of Aristotelian syllogistic, *Journal of Symbolic Logic* 21:137–147.
Słupecki, J. (1949), Z badań nad sylogistyką Arystotelesa, *Travaux de la Société des Sciences et des Lettres de Wrocław* B(6):5–30.
Smiley, T. (1973), What is a syllogism?, *Journal of Philosophical Logic* 2:135–154.
Sotirov, V. (2012), Arithmetizations of syllogistic à la Leibniz, *Journal of Applied Non-Classical Logics* 9(2–3):387–405.
Störing, G. (1908), Experimentelle Untersuchungen über einfache Schlußprozesse, *Archiv für die gesamte Psychologie* 11:1–127.
Suppes, P. (1979), Logical inference in English: a preliminary analysis, *Studia Logica* 38:375–391.
Woodworth, R. & S. Sells (1935), An atmosphere effect in formal syllogistic reasoning, *Journal of Experimental Psychology* 18:451–460.
Zamansky, A., N. Francez, & Y. Winter (2006), A 'natural logic' inference system using the Lambek calculus, *Journal of Logic, Language and Information* 15(3):273–295.