Logics for Two Fragments Beyond the Syllogistic Boundary

Lawrence S. Moss
Department of Mathematics, Indiana University, Bloomington IN 47405 USA

This is a draft of a paper to be submitted to a festschrift volume for Yuri Gurevich. The unusual introduction is based on dialogs that are a regular part of Yuri's columns in the *Bulletin of the EATCS*. I may change this introduction, or drop it altogether. Concerning the rest of the paper: aside from typos and unclear points, the technical content should be stable. **I welcome your comments.** I'm a little bit late with the draft, so if you get a chance to look in the next month, I would be especially happy.

Abstract

This paper is a contribution to *natural logic*, the study of logical systems for linguistic reasoning. We construct a system with the following properties: its syntax is closer to that of a natural language than is first-order logic; it can faithfully represent simple sentences with standard quantifiers, subject relative clauses (a recursive construct), and negation on nouns and verbs. We also give a proof system which is complete and has the finite model property. We go further by adding comparative adjective phrases, assuming interpretations by transitive relations. This last system has all the previously-mentioned properties as well.

The paper was written for theoretical computer scientists and logicians interested in areas such as decidability questions for fragments of first-order logic, modal logic, and natural deduction.

Contents

1 Introduction

Q: Hello, I'm wondering if you can help me.

A: Sure, but who *are* you? Your manner is familiar, but I don't recognize your face. I think I need an Introduction.

Q: Well, my name is Quisani, and ...

A: Quisani! It's a pleasure to meet you. I almost feel that I know you already, since I've been following your discussions with Yuri Gurevich for so many years. What brings you here?

Q: I have heard that there is a resurgence of interest in logical systems that deal with inference in natural language. Is this so? And can you explain a bit of it to me, an open-minded theoretical computer scientist?

A: This is a tall order of business for a short chat, but I'll try. It is true that there has been activity in the last few years that could be of interest. Once area coming from AI and natural language processing has been concerned with getting programs to carry out the inferences from text that people do, or usually do, when they read some text. This is an active area with connections to Information Extraction. But there is not much logic there, at least not yet. Another area, one probably closer to your interests, has been an exploration of fragments of natural language that correspond to complexity classes. This work has been carried out by Ian Pratt-Hartmann [14, 15]. I think it could be interesting to complexity theorists. And then there is the matter of giving logical systems in which one can carry out as much simple reasoning in language as possible. This has been going on for some time: see van Benthem [1] for example, for one early reference close to the area of work that I am thinking about.

Q: But surely the general problem is undecidable. Actually, when I think about matters like vagueness, incomplete sentences, failures of reference, figurative language, and more and more problems ..., I'm not even sure it makes much sense to talk about what you call "simple reasoning in language".

A: To quote your teacher [8] on the Entscheidungsproblem,

> The ambitious attempt to mechanize mathematics via a decision algorithm for first-order logic failed. Does this mean that the field should be abandoned? Of course not. One should try to see what can be done. It is natural to try to isolate special cases of interest where mechanization is possible. There are many ways to define syntactic complexity of logic formulas. It turns out that, with respect to some natural definitions, sentences of low syntactic complexity suffice to express many mathematical problems. For example, many mathematical problems can be formulated with very few quantifier alternations. The decision problem for such classes of sentences is of interest.

If we replace "mathematics" by "language", and "mathematical problems" by "simple natural language arguments," you'll get my point.

Q: All right, I would like to know more about this. But first, could you please give me an example of some non-trivial inference carried out in natural language?

A: Sure, I'd be happy to. Let's say we're talking about a big table full of fruit. (*He writes on*

the board.)

$$\frac{\begin{array}{l}\text{Every sweet fruit is bigger than every ripe fruit}\\ \text{Every pineapple is bigger than every kumquat}\\ \text{Every non-pineapple is bigger than every unripe fruit}\end{array}}{\text{Every fruit bigger than some sweet fruit is bigger than every kumquat}} \quad (1)$$

I say that if we assume (or believe, or know) all the sentences above the line, then we should do the same for the sentence below the line.

Q: "non-pineapple"?! I thought this was supposed to be natural language.

A: Take it as a shorthand for "piece of fruit which is not a pineapple".

Q: Ok, I get it. But is everything either a pineapple or a non-pineapple?

A: You bet. You'll need to use this when you convince yourself that (1) is valid.

Q: Do I need to use anything else?

A: Yes, you need to know that bigger than is a transitive relation.

Q: I'll have to think about the reasoning. But anyhow, why don't you just type (1) into a theorem prover? It is in first-order logic. More precisely, it translates easily into FOL.

A: I'd rather translate to a decidable system, such as one of the ones in The Book [2]. The full expressive power of first-order logic does not seem appropriate either to modeling how people reason, or to getting a computer to carry out the reasoning in examples like (1).

Q: Well, looking more carefully, I can see that your argument can be translated into the two-variable fragment FO^2. Then it really would fit into a decidable logic, by Mortimer's Theorem [12].

A: Not so fast. Although what I wrote translates to FO^2, the fact that bigger than is transitive means that we go beyond it. As you know, three variables is enough to have undecidability. You might try to study FO^2 on transitive relations, but this too is undecidable [7].

Q: But are there other logics that are on the one hand big enough to express the sentences in (1) and yet are decidable?

A: Yes, there is at least one that I can think of: *Boolean modal logic*, again with the stipulation that the semantics lives on transitive relations. This is known to be decidable [10].

Q: I thought that modal logic was about possible worlds, that sort of thing. It's hard to see why it's rearing its ugly head here. But again, I ask: if this Boolean modal logic is so great, why not translate (1) into it, and be done with it?

A: If I was only interested in the complexity of the example, or similar ones, then I'd agree. But with my logical baggage, I mean background, I feel that I should be looking for general principles and not just algorithms and complexity results. Think about Aristotle. As it happens, his system can be reformulated a little, and it turns out to be *complete*. But do you think he would have been happy if someone came and told him to forget the syllogisms because they had an algorithm to tell whether a conclusion followed, without telling him the reasons?

Q: Aristotle, ..., hmm. You know, I once heard that the three greatest logicians of all time

are Aristotle, Frege, and Gödel. I know a bit about the last two, but I have almost no idea what Aristotle did. Something about *all men are mortal*?

A: Aristotle raised the matter of *inference* in the first place, with no precedent. And then for the fragment he was concerned with he provided a complete answer. I can't imagine a bigger project.

Q: I can see that you're really taken with Aristotle. But okay, let's go back to what's on the blackboard (1). Do you have a logical system in which we can prove (1), and which is still decidable?

A: Yes. Want to see it?

Q: Sure, but before you start in, I have a last question. You started out mentioning that there is some interest in natural logic from people in much more applied areas of computer science. Are they going to be interested in logical systems for cases like (1), the kind of thing that you are going to show me?

A: Probably not. Presumably they would be much more interested in an algorithm that worked quickly and correctly on 90% of the real-world inferences that come up in practice, than on a logical system that was complete in the logician's sense but was only good for a small amount of real-world inference. On the other hand, knowing what a complete system looked like could be an inspiration, or at least a comfort.

Q: Can you give me an example? Something that a logical system is not likely to get, but which is an *inference from text* in the sense of current work in computational linguistics?

A: (*Again writing on the board.*)

$$\frac{\textsf{Frege's favorite food was sushi}}{\textsf{Frege ate sushi at least once}} \tag{2}$$

Q: Are you sure you don't mean Russell?

A: Nearly everyone would agree to (2), but almost nobody will get (1) on their own.

Q: Yes, but real-world knowledge ... this is just not my cup of tea. Actually, formal systems and completeness proofs are not really my cup of tea either, but I'm curious enough to want to see yours.

A: Speaking of tea, can I show you the logical system for (1) and related matters over a cup?

Q: Sure, but now that you bring up eating, I'd prefer sushi and pineapples.

2 Logic for a Fragment $\mathcal{L}$ with Set Terms and Negation

There are only two languages in this paper (actually they are families of languages parameterized by sets of basic symbols): the language $\mathcal{L}$ of this section, and extended $\mathcal{L}(adj)$ studied in Section 3. $\mathcal{L}$ is based on three pairwise disjoint sets called $\mathbf{P}$, $\mathbf{R}$, and $\mathbf{K}$. These are called *predicate symbols*, *relation symbols*, and *constant symbols*.

Expression	Variables	Syntax
unary atom	p, q	
binary atom	s	
constant	j, k	
unary literal	l	$p \mid \bar{p}$
binary literal	r	$s \mid \bar{s}$
set term	b, c, d	$l \mid \exists(c,r) \mid \forall(c,r)$
sentence	φ, ψ	$\forall(c,d) \mid \exists(c,d) \mid c(j) \mid r(j,k)$

Figure 1: Syntax of sentences of $\mathfrak{L}$.

2.1 Syntax and semantics

We present the syntax of $\mathfrak{L}$ in Figure 1. Sentences are built from constant symbols, unary and binary atoms using an involutive symbol for negation, a formation of set terms, and also a form of quantification. The second column indicates the variables that we shall use in order to refer to the objects of the various syntactic categories. Because the syntax is not standard, it will be worthwhile to go through it slowly and to provide glosses in English for expressions of various types.

One might think of the constant symbols as proper names such as John and Mary. The predicate symbols may be glossed as one-place predictates such as boys, girls, etc. And the relation symbols correspond to transitive verbs (that is, verbs which take a direct object) such as likes, sees, etc. They also correspond to comparative adjective phrases such as is bigger than. (However, later on in Section 3, we introduce a new syntactic primitive for the adjectives.)

Unary atoms *appear to be* one-place relation symbols, especially because we shall form sentences of the form $p(j)$. However, we do not have sentences $p(x)$, since we have no variables at this point in the first place. Similar remarks apply to binary atoms and two-place relation symbols. So we chose to change the terminology from *relation symbols* to *atoms*.

We form unary and binary *literals* using the bar notation. We think of this as expressing classical negation. So we take it to be involutive, so that $\bar{\bar{p}} = p$ and $\bar{\bar{s}} = s$.

The set terms in this language are the only recursive construct. If b is read as boys and s as sees, then one should read $\forall(b,s)$ as sees all boys, and $\exists(b,s)$ as sees some boys. Hence these set terms correspond to simple verb phrases. We also allow negation on the atoms, so we have $\forall(b,\bar{s})$; this can be read as fails to see all boys, or (better) sees no boys or doesn't see any boys. We also have $\exists(b,\bar{s})$, fails to see some boys. But the recursion allows us to embed set terms, and so we have set terms like

$$\exists(\forall(\forall(b,\bar{s}),h),a)$$

which may be taken to symbolize a verb phrase such as admires someone who hates everyone who does not see any boy.

We should note that the relative clauses (RCs) which can be obtained in this way are all subject RCs, not object RCs. The language is too poor to express predicates like λx.all boys see x.

The main sentences in the language are of the form $\forall(b,c)$ and $\exists(b,c)$; they can be read as statements of the inclusion of one set term extension in another, and of the non-empty intersection. We also have sentences using the constants, such as $\forall(g,s)(m)$, corresponding

to Mary sees all girls. But we are not able to say all girls see Mary; the syntax again is too weak. (However, in our Conclusion we shall see how to extend our system to handle this.) This weakness in expressive power corresponds to a less complex decidability result, as we shall see.

Semantics A *structure* (for this language $\mathcal{L}$) is a pair $\mathcal{M} = \langle M, [\![\ ]\!] \rangle$, where M is a non-empty set, $[\![p]\!] \subseteq M$ for all $p \in \mathbf{P}$, $[\![r]\!] \subseteq M^2$ for all $r \in \mathbf{R}$, and $[\![j]\!] \in M$ for all $j \in \mathbf{K}$.

Given a model $\mathcal{M}$, we extend the interpretation function $[\![\ ]\!]$ to the rest of the language by setting

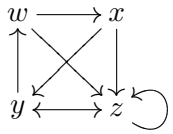
$$\begin{array}{lcl} [\![\overline{p}]\!] & = & M \setminus [\![p]\!] \\ [\![\overline{s}]\!] & = & M^2 \setminus [\![s]\!] \\ [\![\exists(l,t)]\!] & = & \{x \in M : \text{ for some } y \text{ such that } [\![l]\!](y),\ [\![t]\!](x,y)\} \\ [\![\forall(l,t)]\!] & = & \{x \in M : \text{ for all } y \text{ such that } [\![l]\!](y),\ [\![t]\!](x,y)\} \end{array}$$

We define the truth relation $\models$ between models and sentences by:

$$\begin{array}{lll} \mathcal{M} \models \forall(c,d) & \text{iff} & [\![c]\!] \subseteq [\![d]\!] \\ \mathcal{M} \models \exists(b,c) & \text{iff} & [\![c]\!] \cap [\![d]\!] \neq \emptyset \\ \mathcal{M} \models c(j) & \text{iff} & [\![c]\!]([\![j]\!]) \\ \mathcal{M} \models r(j,k) & \text{iff} & [\![r]\!]([\![j]\!],[\![k]\!]) \end{array}$$

If Γ is a set of formulas, we write $\mathcal{M} \models \Gamma$ if for all $\varphi \in \Gamma$, $\mathcal{M} \models \varphi$.

Example 1 We consider a simple case, with one unary atom p, one binary atom s, and two constants j and k. Consider the following model. We set $M = \{w, x, y, z\}$, and $[\![p]\!] = \{w, x, y\}$. For the relation symbol, s, we take the arrows below:



For example, $[\![\overline{p}]\!] = \{z\}$, $[\![\forall(p,s)]\!] = \emptyset$, $[\![\exists(\overline{p},s)]\!] = M$, and $[\![\exists(\forall(p,\overline{s}),s)]\!] = \emptyset$. Here are two $\mathcal{L}$-sentences true in $\mathcal{M}$: $\forall(p, \exists(\overline{p}, s))$ and $\forall(\exists(\forall(p,\overline{s}),s),\overline{p})$.

Now set $[\![j]\!] = w$ and $[\![k]\!] = x$. We get additional sentences true in $\mathcal{M}$ such as $s(j,k)$, $\overline{s}(k,j)$, and $\exists(\overline{p},s)(k)$.

Here is a point that will be important later. For all terms c, $\mathcal{M} \models c(j)$ iff $\mathcal{M} \models c(k)$. (The easiest way to check this is to show that for all set terms c, $[\![c]\!]$ is one of the following four sets: $\emptyset, M, \{w,x,y\},$ or $\{z\}$.) However, $\mathcal{M} \models s(j,k)$ and $\mathcal{M} \models \overline{s}(k,j)$.

Satisfiability A sentence φ is *satisfiable* if there exists $\mathcal{M}$ such that $\mathcal{M} \models \varphi$; satisfiability of a set of formulas Γ is defined similarly. We write $\Gamma \models \varphi$ to mean that every model of every sentence in Γ is also a model of φ.

The satisfiability problem for the language is decidable for a very easy reason. The language $\mathcal{L}$ translates to the *two-variable fragment* FO^2 of first-order logic. Thus we have the finite model property (Mortimer [12]) and decidability of satisfiability in non-deterministic exponential time (Grädel et al [6]). It might therefore be interesting to ask whether the smaller fragment $\mathcal{L}$ is of a lower complexity. As it happens, it is. Pratt-Hartmann [15] showed that the satisfiability problem for a certain fragment of FO^2 can be decided in EXPTIME in the length of the input Γ, and his fragment was essentially the same as the one in this paper.

The bar notation We have already seen that our unary and binary atoms come with negative forms. We extend this notation to all sentences in the following ways: $\overline{\overline{p}} = p$, $\overline{\overline{s}} = s$, $\overline{\exists(l,r)} = \forall(l,\overline{r})$, $\overline{\forall(l,r)} = \exists(l,\overline{r})$, $\overline{\forall(c,d)} = \exists(c,\overline{d})$, $\overline{\exists(c,d)} = \forall(c,\overline{d})$, $\overline{c(j)} = \overline{c}(j)$, and $\overline{r(j,k)} = \overline{r}(j,k)$.

Translation of the syllogistic into $\mathfrak{L}$ We indicate briefly a few translations to orient the reader. First, the classical syllogistic translates into $\mathfrak{L}$:

All p are q	$\mapsto$	$\forall(p,q)$	No p are q	$\mapsto$	$\forall(p,\overline{q})$
Some p are q	$\mapsto$	$\exists(p,q)$	Some p aren't q	$\mapsto$	$\exists(p,\overline{q})$

The same is true of the relational syllogistic; cf. [16]. In the other direction, we translate $\mathfrak{L}$ to FO^2, the fragment of first order logic using only the variables x and w. We do this by mapping the set terms two ways, called $c \mapsto \varphi_{c,x}$ and $c \mapsto \varphi_{c,y}$. Here are the recursion equations for $c \mapsto \varphi_{c,x}$:

$$\begin{array}{lcllcl} p & \mapsto & P(x) & \forall(c,r) & \mapsto & (\forall y)(\varphi_{c,y}(y) \to r(x,y)) \\ \overline{p} & \mapsto & \neg P(x) & \exists(c,r) & \mapsto & (\exists y)(\varphi_{c,y}(y) \wedge r(x,y)) \end{array}$$

The equations for $c \mapsto \varphi_{c,y}$ are similar. Then the translation of the sentences into FO^2 follows easily.

Translation of $\mathfrak{L}$ into Boolean modal logic We shall write $\hat{\mathfrak{L}}$ for the logic obtained by taking the logic of Humberstone [9] and using many pairs of operators instead of just one. $\hat{\mathfrak{L}}$ has each $p \in \mathbf{P}$ as an *atomic proposition*, and it has two *modal operators*, $\Box_s$ and $\blacksquare_s$, one for each $s \in \mathbf{R}$. The syntax of $\hat{\mathfrak{L}}$ is given by

$$\varphi \quad := \quad p \mid \neg\varphi \mid \varphi \wedge \psi \mid \Box_s\varphi \mid \blacksquare_s\varphi$$

The language is interpreted on the same kind of structures that we have been using for $\mathfrak{L}$. Then $[\![p]\!]$ is given for all atoms p, and we also set $[\![\neg\varphi]\!] = M \setminus [\![\varphi]\!]$, $[\![\varphi \wedge \psi]\!] = [\![\varphi]\!] \cap [\![\psi]\!]$, and

$$\begin{array}{lcl} [\![\Box_s\varphi]\!] & = & \{x \in M : \text{for all } y \text{ such that } [\![s]\!](x,y),\ y \in [\![\varphi]\!]\} \\ [\![\blacksquare_s\varphi]\!] & = & \{x \in M : \text{for all } y \text{ such that not } [\![s]\!](x,y),\ y \in [\![\varphi]\!]\} \end{array}$$

We write $\Gamma \models \varphi$ to mean that for all structures $\mathcal{M}$ and all $x \in M$, if $x \in [\![\psi]\!]$ for all $\psi \in \Gamma$, then again $x \in [\![\varphi]\!]$.

Let $\mathfrak{L}_0$ be the set of sentences of $\mathfrak{L}$ which do not involve constants. We translate $\mathfrak{L}_0$ into $\hat{\mathfrak{L}}$. First, for each set term c, we define a sentence c^* of $\hat{\mathfrak{L}}$. The definition is: $p^* = p$, $\overline{p}^* = \neg p$,

$$\begin{array}{lcllcl} \forall(c,s)^* & = & \blacksquare_s\neg c^* & \forall(c,\overline{s})^* & = & \Box_s\neg c^* \\ \exists(c,s)^* & = & \neg\Box_s\neg c^* & \exists(c,\overline{s})^* & = & \neg\blacksquare_s\neg c^* \end{array}$$

An easy induction shows that $[\![c]\!] = [\![c^*]\!]$ for all set terms c. Then we translate $\forall(c,d)$ to $\forall(c,d)^*$ and $\exists(c,d)$ to $\exists(c,d)^*$:

$$\begin{array}{lcl} \forall(c,d)^* & = & \Box_s(c^* \to d^*) \wedge \blacksquare_s(c^* \to d^*) \\ \exists(c,d)^* & = & \neg\forall(c,\overline{d})^* \end{array}$$

where s is an arbitrary element of $\mathbf{R}$. Then for all sentences φ of $\mathfrak{L}_0$, and all models $\mathcal{M}$,

$$\mathcal{M} \models \varphi \quad \text{iff} \quad [\![\varphi^*]\!] = M \quad \text{iff} \quad [\![\varphi^*]\!] \neq \emptyset.$$

Expression	Variables	Syntax
individual variable	x, y	
individual term	t, u	$x \mid j$
general sentence	α	$\varphi \mid c(t) \mid r(t,u) \mid \bot$

Figure 2: Syntax of general sentences of $\mathfrak{L}$, with φ ranging over sentences.

It follows easily from this that for $\Gamma \cup \{\varphi\}$ a set of $\mathfrak{L}_0$ sentences, $\Gamma \models \varphi$ in the semantics of $\mathfrak{L}$ iff $\Gamma^* \models \varphi^*$ in the semantics of $\hat{\mathfrak{L}}$.

It is more common nowadays to view a system such as $\hat{\mathfrak{L}}$ as a fragment of *Boolean modal logic*.

2.2 Proof system

Pratt-Hartmann and Moss [16] investigated several logical systems for the *relational syllogistic* and asked whether they were axiomatizable in a purely syllogistic fashion. We shall not enter into their definition of *syllogistic proof system* except to say that it is an adequate formalization of the concept. It comes in two flavors, depending on whether one permits *reductio ad absurdum* or not. It turns out that the consequence relation for some some logical languages can be captured by syllogistic systems even without *reductio*, some can be captured with *reductio* but (provably) not without it, and some are so strong that they cannot be captured even with *reductio*. The language of this paper would be one example of this latter phenomenon. Theorem 6.12 of [16] shows that for the language $\mathfrak{L}$ of this paper (but without constant symbols), there indeed is no finite complete syllogistic system. It is therefore of interest to build a proof system which goes beyond syllogistic logic. This is the main technical goal of this paper.

We present our system in natural-deduction style in Figure 3. It makes use of *introduction* and *elimination* rules, and more critically of *variables*. For a textbook account of a proof system for first-order logic presented in this way, see van Dalen [3].

General sentences in this fragment are what usually are called *formulas*. We prefer to change the standard terminology to make the point that here, sentences are not built from formulas by quantification. In fact, sentences in our sense do not have variable occurrences. But general sentences do include *variables*. They are only used in our proof theory.

The syntax of general sentences is given in Figure 2. What we are calling *individual terms* are just variables and constant symbols. (There are no function symbols here.) Using terms allows us to shorten the statements of our rules, but this is the only reason to have terms.

An additional note: we don't need general sentences of the form $r(j,x)$ or $r(x,j)$. In larger fragments, we would expect to see general sentences of these forms, but our proof theory will not need these.

A note on variables We have been silent on whether the basic set of variables in the system is finite or infinite. It is natural to formulate the system using an infinite set of variables. However, every derivation in our current system may be recast to use *only two variables*. This special property is lost in our extended system of Section 3.

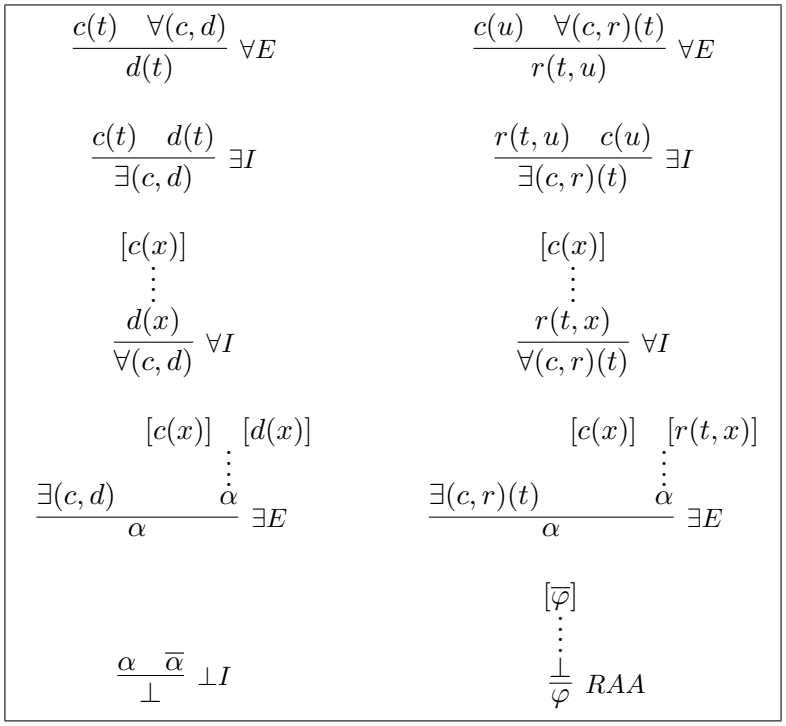


Figure 3: Proof rules. See the text for the side conditions in the $(\forall I)$ and $(\exists E)$ rules.

The bar notation, again We have already seen the bar notation $\overline{c}$ for set terms c, and $\overline{\varphi}$ for sentences φ. We extend this to formulas $\overline{b(x)} = \overline{b}(x)$, $\overline{r(x,y)} = \overline{r}(x,y)$. We technically have a general sentence $\overline{\bot}$, but this plays no role in the proof theory.

We write $\Gamma \vdash \varphi$ if there is a proof tree conforming to the rules of the system with root labeled φ and whose axioms are labeled by elements of Γ. (Frequently we shall be sloppy about the labeling and just speak, e.g, of the root as if it *were* a sentence instead of being *labeled by* one.) Instead of giving a precise definition here, we shall content ourselves with a series of examples in Section 2.3 just below.

The system has two rules called $(\forall E)$, one for deriving general sentences of the form $c(x)$ or $c(j)$, and one for deriving general sentences $r(x,y)$ or $r(j,k)$. (Other rules are doubled as well, of course.) It surely looks like these should be unified, and the system would of course be more elegant if they were. But given the way we are presenting the syntax, there is no way to do this. That is, we do not have a concept of *substitution*, and so rules like $(\forall E)$ cannot be formulated in the usual way. Returning to the two rules with the same name, we could have chosen to use different names, say $(\forall E1)$ and $(\forall E2)$. But the result would have been a more cluttered notation, and it is always clear from context which rule is being used.

Although we are speaking of trees, we don't distinguish left from right. This is especially the case with the $(\exists E)$ rules, where the canceled hypotheses may occur in either order.

Side Conditions As with every natural deduction system using variables, there are some side conditions which are needed in order to have a *sound* system.

In $(\forall I)$, x must not occur free in any uncanceled hypothesis. For example, in the version

whose root is $\forall(c,d)$, one must cancel all occurrences of $c(x)$ in the leaves, and x must not appear free in any other leaf.

In $(\exists E)$, the variable x must not occur free in the conclusion α or in any uncanceled hypothesis in the subderivation of α.

In contrast to usual first-order natural deduction systems, there are *no side conditions* on the rules $(\forall E)$ and $(\exists I)$. The usual side conditions are phrased in terms of concepts such as *free substitution*, and the syntax here has no substitution to begin with. To be sure on this point, one should check the soundness result of Lemma 2.1.

Formal proofs in the Fitch style Textbook presentations of logic overwhelmingly use natural deduction. Pelletier [13] discusses the history of this in texts used in philosophy classes. In such books, by far the most common style of presentation is via Fitch diagrams. Pelletier was not concerned with books for computer science students, and here the situation is more mixed, I believe. I have chosen to present the system in a more "classical" Gentzen-style format. But the system may easily be re-formatted to look more like a Fitch system, as we shall see in Example 3 and Figure 5. These examples might give the impression that we have merely re-presented Fitch-style natural deduction proofs. The difference is that our syntax is not a special case of the syntax of first-order logic. Corresponding to this, our proof rules are rather restrictive, and the system cannot be used for much of anything beyond the language $\mathcal{L}$. However, the fact that our Fitch-style proofs *look like* familiar formal proofs is a virtue: for example, it means that one could teach logic using this material.

2.3 Examples

We present a few examples of the proof system at work, along with comments pertaining to the side conditions. Many of these are taken from the proof system $\mathbf{R}^*$ for the language $\mathcal{R}^*$ of [16]. That system $\mathbf{R}^*$ is among the strongest of the known syllogistic systems, and so it is of interest to check the current proof system is at least as strong.

Example 2 Here is a proof of the classical syllogism *Darii*: $\forall(b,d), \exists(c,b) \vdash \exists(c,d)$:

$$\dfrac{\exists(c,b) \qquad \dfrac{\dfrac{[b(x)]^1 \quad \forall(b,d)}{d(x)}\,\forall E \qquad [c(x)]^1}{\exists(c,d)}\,\exists I}{\exists(c,d)}\,\exists E^1$$

Example 3 Next we study a principle called (K) in [16]. Intuitively, if all watches are expensive items, then everyone who owns a watch owns an expensive item. The formal statement in our language is $\forall(c,d) \vdash \forall(\exists(c,r), \exists(d,r))$. See Figure 4. We present a Fitch-style proof on the left and the corresponding one in our formalism on the right. One aspect of the Fitch-style system is that $(\exists E)$ gives two lines; see lines 3 and 4 on the left above.

1	$\forall(c,d)$	hyp	$\dfrac{\dfrac{\dfrac{[\exists(c,r)(x)]^2 \quad \dfrac{\dfrac{[r(x,y)]^1 \quad \dfrac{[c(y)]^1 \quad \forall(c,d)}{d(y)}\,\forall E}{\exists(d,r)(x)}\,\exists I}{\exists(d,r)(x)}\,\exists E^1}{\forall(\exists(c,r),\exists(d,r))}\,\forall I^2$
2	$x \mid \exists(c,r)(x)$	hyp	
3	$\quad c(y)$	$\exists E$, 2	
4	$\quad r(x,y)$	$\exists E$, 2	
5	$\quad d(y)$	$\forall E$, 1, 3	
6	$\quad \exists(d,r)(x)$	$\exists I$, 4, 5	
7	$\forall(\exists(c,r),\exists(d,r))$	$\forall I$, 1–6	

Figure 4: Derivations in Example 3

Example 4 Here is an example of a derivation using (RAA). It shows $\forall(c,\overline{c}) \vdash \forall(d,\forall(c,r))$.

$$\dfrac{\dfrac{[d(x)]^2 \quad \dfrac{\dfrac{\dfrac{[c(y)]^1 \quad \forall(c,\overline{c})}{\overline{c}(y)}\,\forall E \quad [c(y)]^1}{\bot}\,\bot I}{r(x,y)}\,RAA}{\forall(c,r)(x)}\,\forall I^1}{\forall(d,\forall(c,r))}\,\forall I^2$$

Example 5 Here is a statement of the *rule of proof by cases*: If $\Gamma + \varphi \vdash \psi$ and $\Gamma + \overline{\varphi} \vdash \psi$, then $\Gamma \vdash \psi$. (Here and below, $\Gamma + \varphi$ denotes $\Gamma \cup \{\varphi\}$.) Instead of giving a derivation, we only indicate the ideas. Since $\Gamma + \varphi \vdash \psi$, we have $\Gamma + \varphi + \overline{\psi} \vdash \bot$ using $(\bot I)$. From this and (RAA), $\Gamma, \overline{\psi} \vdash \overline{\varphi}$. Take a derivation showing $\Gamma + \overline{\varphi} \vdash \psi$, and replace the labeled $\overline{\varphi}$ with derivations from $\Gamma + \overline{\psi}$. We thus see that $\Gamma + \overline{\psi} \vdash \psi$. Using $(\bot I)$, $\Gamma + \overline{\psi} \vdash \bot$. And then using (RAA) again, $\Gamma \vdash \psi$. (This point is from [16].)

Example 6 The example at the beginning of this paper cannot be formalized in this fragment because the correct reasoning uses the transitivity of is bigger than. However, we can prove a result which may itself be used in a formal proof of (1):

$$\dfrac{\begin{array}{l}\textsf{Every sweet fruit is bigger than every ripe fruit}\\ \textsf{Every pineapple is bigger than every kumquat}\\ \textsf{Every non-pineapple is bigger than every unripe fruit}\end{array}}{\textsf{Every sweet fruit is bigger than every kumquat}} \tag{3}$$

To discuss this, we take the set $\mathbf{P}$ of predicate symbols to be

$$\mathbf{P} \quad = \quad \{\textsf{sweet}, \textsf{ripe}, \textsf{pineapple}, \textsf{kumquat}\}.$$

We also take $\mathbf{R} = \{\textsf{bigger}\}$ and $\mathbf{K} = \emptyset$. Figure 5 contains a derivation showing (3), done in the manner of Fitch [4]. The main way in which we have bent the English in the direction

1	Every sweet fruit is bigger than every ripe fruit	hyp
2	Every pineapple is bigger than every kumquat	hyp
3	Every $\overline{\text{pineapple}}$ is bigger than every $\overline{\text{ripe fruit}}$	hyp
4	x | x is a sweet fruit	hyp
5	x is bigger than every ripe fruit	$\forall E$, 1, 4
6	x is a pineapple	hyp
7	x is bigger than every kumquat	$\forall E$, 2, 6
8	x is a $\overline{\text{pineapple}}$	hyp
9	x is bigger than every $\overline{\text{ripe fruit}}$	$\forall E$, 3, 8
10	y | y is a kumquat	hyp
11	y is a ripe fruit	hyp
12	x is bigger than y	$\forall E$, 5, 11
13	y is a $\overline{\text{ripe fruit}}$	hyp
14	x is bigger than y	$\forall E$, 9, 13
15	x is bigger than y	cases, 13–14, 11–12
16	x is bigger than every kumquat	$\forall I$, 10–15
17	x is bigger than every kumquat	cases, 6–7, 8–16
18	Every sweet fruit is bigger than every kumquat	$\forall I$, 4–17

Figure 5: A derivation corresponding to the argument in (3).

of our formalism is to use the bar notation on the nouns. The main reason for presenting the derivation as a Fitch diagram is that the derivation given as a tree (as demanded by our definitions) would not fit on a page. This is because the cases rule is not a first-class rule in the system, it is a derived rule (see Example 5 above). Our Fitch diagram pretends that the system has a rule of cases. Another reason to present the derivation as in Figure 5 is to make the point that the treatment in this paper is a beginning of a formalization of the work that Fitch was doing.

2.4 Soundness

Before presenting a soundness result, it might be good to see an improper derivation. Here is one, purporting to infer some men see some men from some men see some women:

$$\dfrac{\exists(m,\exists(w,s)) \qquad \dfrac{[\exists(w,s)(x)]^2 \qquad \dfrac{\dfrac{[s(x,x)]^1 \quad [m(x)]^2}{\exists(s,m)(x)}\,\exists I \qquad [m(x)]^2}{\exists(m,\exists(m,s))}\,\exists E^1}{\exists(m,\exists(m,s))}\,\exists E^2$$

The specific problem here is that when $[s(x,x)]$ is withdrawn in the application of $\exists I^1$, the variable x is free in the as-yet-uncanceled leaves labeled $m(x)$.

To state a result pertaining to the soundness of our system, we need to define the truth value of a general sentence under a variable assignment. First, a *variable assignment* in a model $\mathcal{M}$ is a function $v : V \to M$, where V is the set of variable symbols and M is the universe of $\mathcal{M}$. We need to define $\mathcal{M} \models \alpha[v]$ for general sentences α. If α is a sentence, then $\mathcal{M} \models \alpha[v]$ iff $\mathcal{M} \models \alpha$ in our earlier sense. If α is $b(x)$, then $\mathcal{M} \models \alpha[v]$ iff $[\![b]\!](v(x))$. If α is $r(x, y)$, then $\mathcal{M} \models \alpha[v]$ iff $[\![r]\!](v(x), v(y))$. If α is $\bot$, then $\mathcal{M} \not\models \bot$ for all models $\mathcal{M}$.

Lemma 2.1 *Let Π be any proof tree for this fragment all of whose nodes are labeled with $\mathcal{L}$-formulas, let φ be the root of Π, let $\mathcal{M}$ be a structure, let $v : X \to M$ be a variable assignment, and assume that for all uncanceled leaves ψ of Π, $\mathcal{M} \models \psi[v]$. Then also $\mathcal{M} \models \varphi[v]$.*

Proof By induction on Π. We shall only go into details concerning two cases. First, consider the case when the root of Π is

$$\dfrac{\exists(c,r)(t) \qquad \begin{array}{c}[c(x)] \quad [r(t,x)] \\ \vdots \\ \alpha\end{array}}{\alpha}\ \exists E$$

To simplify matters further, let us assume that t is a variable. Let v be a variable assignment making true all of the leaves of the tree, except possibly $c(x)$ and $r(t, x)$. By induction hypothesis, $\mathcal{M} \models \exists(c, r)(t)[v]$. Let $a \in A$ witness this assertion. In the obvious notation, $[\![c]\!](a)$ and $[\![r]\!](t^{\mathcal{M},v}, a)$. Let w be the same variable assignment as v, except that $w(x) = a$. Then since x is not free in any leaves except those labeled $c(x)$ and $r(t, x)$, we have $\mathcal{M} \models \psi[w]$ for all those ψ. And so $\mathcal{M} \models \alpha[w]$, using the induction hypothesis applied to the subtree on the right. And since x is not free in the conclusion α, we also have And so $\mathcal{M} \models \alpha[v]$, as desired. above $\exists(c, r)(t)$,

Second, let us consider the case when the root is

$$\dfrac{c(y) \quad \forall(c,r)(x)}{r(x,y)}\ \forall E$$

(That is, we are considering an instance of $(\forall E)$ when the terms t and u are variables.) The variables x and y might well be the same. Let $\mathcal{M}$ be a structure, and v be a variable assignment making true the leaves of the tree. By induction hypothesis, $[\![c]\!](v(y))$ and also $[\![r]\!](v(x), m))$ for all $m \in [\![c]\!]$. In particular, $[\![r]\!](v(x), v(y))$.

The remaining cases are similar. $\dashv$

2.5 The Henkin property

The completeness of the logic parallels the Henkin-style completeness result for first-order logic. Given a consistent theory Γ, we get a model of Γ in the following way: (1) take the underlying language $\mathcal{L}$, add constant symbols to the language to witness existential sentences, also extend Γ to a maximal consistent set in the larger language; and then (3) use the set of constant symbols as the carrier of a model in a canonical way. In the setting of this paper, the work is in some ways easier than in the standard setting, and in some ways harder. There are more details to check, since the language has more basic constructs. But one doesn't need to take a quotient by equivalence classes, and in other ways the work here is easier.

Given two languages $\mathcal{L}$ and $\mathcal{L}'$, we say that $\mathcal{L}' \supseteq \mathcal{L}$ if every symbol (of any type) in $\mathcal{L}$ is also a symbol (of the same type) in $\mathcal{L}'$. In this paper, the main case is when $\mathbf{P}(\mathcal{L}) = \mathbf{P}(\mathcal{L}')$, $\mathbf{R}(\mathcal{L}) = \mathbf{R}(\mathcal{L}')$, and $\mathbf{K}(\mathcal{L}) \subseteq \mathbf{K}(\mathcal{L}')$; that is, $\mathcal{L}'$ arises by adding constants to $\mathcal{L}$.

A *theory* in a language is just a set of sentences in it. Given a theory Γ in a language $\mathcal{L}$, and a theory Γ^* in an extension $\mathcal{L}' \supseteq \mathcal{L}$, we say that Γ^* is a *conservative extension* of Γ if for every $\varphi \in \mathcal{L}$, and $\Gamma^* \vdash \varphi$, then $\Gamma \vdash \varphi$.

Lemma 2.2 *Let Γ be a consistent $\mathcal{L}$-theory, and let $j \notin \mathbf{K}(\mathcal{L})$.*

1. *If $\exists(c,d) \in \Gamma$, then $\Gamma + c(j) + d(j)$ is a conservative extension of Γ.*
2. *If $\exists(c,r)(k) \in \Gamma$, then $\Gamma + r(k,j) + c(j)$ is a conservative extension of Γ.*

Proof For (1), suppose that Γ contains $\exists(c,d)$ and that $\Gamma + c(j) + d(j) \vdash \varphi$. Let Π be a derivation tree. Replace the constant j by an individual variable x which does not occur in Π. The result is still a derivation tree, except that the leaves are not labeled by *sentences*. (The reason is that our proof system has no rules specifically for constants, only for terms which might be constants and also might be individual variables.) Call the resulting tree Π'. Now the following proof tree shows that $\Gamma \vdash \varphi$:

$$\dfrac{\exists(c,d) \qquad \begin{array}{c}[c(x)] \quad [d(x)] \\ \vdots \\ \varphi\end{array}}{\varphi}\ \exists E$$

The subtree on the right is Π'. The point is that the occurrences of $c(x)$ and $d(x)$ have been canceled by the use of $\exists E$ at the root.

This completes the proof of the first assertion, and the proof of the second is similar. ⊣

Definition An $\mathcal{L}$-*theory* Γ has the *Henkin property* if the following hold:

1. If $\exists(c,d) \in \Gamma$, then for some constant j, $c(j)$ and $d(j)$ belong to Γ.
2. If r is a literal of $\mathcal{L}$ and $\exists(c,r)(j) \in \Gamma$, then for some constant k, $r(j,k)$ and $c(k)$ belong to Γ.

Lemma 2.3 *Let Γ be a consistent $\mathcal{L}$-theory. Then there is some $\mathcal{L}^* \supset \mathcal{L}$ and some $\mathcal{L}^*$-theory Γ^* such that Γ^* is a maximal consistent Henkin theory. Moreover, if $s \in \mathbf{R}(\mathcal{L})$, $j \in \mathbf{K}(\mathcal{L}^*)$ and $k \in \mathbf{K}(\mathcal{L})$, and if $s(j,k) \in \Gamma^*$, then $j \in \mathbf{K}(\mathcal{L})$.*

Proof This is a routine argument, using Lemma 2.2. One dovetails the addition of constants which is needed for the Henkin property together with the addition of sentences needed to insure maximal consistency. The formal details would use Lemma 2.2 for steps of the first kind, and for the second kind we need to know that if Γ is consistent, then for all φ, either $\Gamma + \varphi$ or $\Gamma + \overline{\varphi}$ is consistent. This follows from the derivable rule of proof by cases; see Example 5 in Section 2.3.

The last point states a technical property that will be useful in Section 3.1. ⊣

It might be worthwhile noting that the extensions produced by Lemma 2.3 add *infinitely many constants* to the language.

2.6 Completeness via canonical models

In this section, fix a language $\mathfrak{L}$ and a maximal consistent Henkin $\mathfrak{L}$-theory Γ. We construct a *canonical model* $\mathcal{M} = \mathcal{M}(\Gamma)$ as follows: $M = \mathbf{K}(\mathfrak{L})$; $[\![p]\!](j)$ iff $p(j) \in \Gamma$; $[\![s]\!](j,k)$ iff $s(j,k) \in \Gamma$; and $[\![j]\!] = j$. That is, we take the constant symbols of the language to be the points of the model, and the interpretations of the atoms are the natural ones. Each constant symbol is interpreted by itself.

Lemma 2.4 *For all set terms c, $[\![c]\!] = \{j : c(j) \in \Gamma\}$.*

Proof By induction on c. The base case of unary atoms p is by definition of $\mathcal{M}$.

Before we turn to the induction proper, here is a preliminary point. Assuming that $[\![c]\!] = \{j : c(j) \in \Gamma\}$, we check that $[\![\overline{c}]\!] = \{j : \overline{c}(j) \in \Gamma\}$:

$$j \in [\![\overline{c}]\!] \quad \text{iff} \quad j \notin [\![c]\!] \quad \text{iff} \quad c(j) \notin \Gamma \quad \text{iff} \quad \overline{c}(j) \in \Gamma.$$

The last point uses the maximal consistency of Γ.

Turning to the inductive steps, assume our result for c; we establish it for $\forall(c,s)$ and $\exists(c,s)$; it then follows from the preliminary point that we have the same fact for $\forall(c,\overline{s})$ and $\exists(c,\overline{s})$.

Let $j \in [\![\forall(c,s)]\!]$. We claim that $\forall(c,s)(j) \in \Gamma$. For if not, then $\exists(c,\overline{s})(j) \in \Gamma$. By the Henkin condition, let k be such that Γ contains $c(k)$ and $\overline{s}(j,k)$. By the induction hypothesis, $k \in [\![c]\!]$, and by the definition of $\mathcal{M}$, $[\![s]\!](j,k)$ is false. Thus $j \notin [\![\forall(c,s)]\!]$. This is a contradiction.

In the other direction, assume that $\forall(c,s)(j) \in \Gamma$; this time we claim that $j \in [\![\forall(c,s)]\!]$. Let $k \in [\![c]\!]$. By induction hypothesis, Γ contains $c(k)$. Using $(\forall E)$, we see that $\Gamma \vdash s(j,k)$. Hence Γ contains $s(j,k)$. So in $\mathcal{M}$, $[\![s]\!](j,k)$. Since k was arbitrary, we see that indeed $j \in [\![\forall(c,s)]\!]$.

The other induction step is for $\exists(c,s)$. Let $j \in [\![\exists(c,s)]\!]$. We thus have some $k \in [\![c]\!]$ such that $[\![s]\!](j,k)$. That is, $s(j,k) \in \Gamma$. Using $(\exists I)$, we have $\Gamma \vdash \exists(c,s)(j)$; from this we see that $\exists(c,s)(j) \in \Gamma$, as desired.

Finally, assume that $\exists(c,s)(j) \in \Gamma$. By the Henkin condition, let k be such that Γ contains $c(k)$ and $s(j,k)$. Using the derivation above, we have the desired conclusion that $j \in [\![\exists(c,s)]\!]$.

This concludes the proof. $\dashv$

Lemma 2.5 $\mathcal{M} \models \Gamma$.

Proof We check the sentence types in turn. Throughout the proof, we shall use Lemma 2.4 without mention.

First, let Γ contain the sentence $\forall(c,d)$. Let $j \in [\![c]\!]$, so that $c(j) \in \Gamma$. We have $d(j) \in \Gamma$ using $(\forall E)$. This for all j shows that $\mathcal{M} \models \forall(c,d)$.

Second, let $\exists(c,d) \in \Gamma$. By the Henkin condition, let j be such that both $c(j)$ and $d(j)$ belong to Γ. This element j shows that $[\![c]\!] \cap [\![d]\!] \neq \emptyset$. That is, $\mathcal{M} \models \exists(c,d)$.

Continuing, consider a sentence $c(j) \in \Gamma$. Then $j \in [\![c]\!]$, so that $\mathcal{M} \models b(j)$.

Finally, the case of sentences $r(j,k) \in \Gamma$ is immediate from the structure of the model. $\dashv$

Theorem 2.6 *If $\Gamma \models \varphi$, then $\Gamma \vdash \varphi$.*

Proof We rehearse the standard argument. Due to the classical negation, we need only show

that consistent sets Γ are satisfiable. Let $\mathfrak{L}$ be the language of Γ, Let $\mathfrak{L}' \supseteq \mathfrak{L}$ be an extension of $\mathfrak{L}$, and let $\Gamma^* \supseteq \Gamma$ be a maximal consistent theory in $\mathfrak{L}'$ with the Henkin property (see Lemma 2.3). Consider the canonical model $\mathcal{M}(\Gamma^*)$ as defined in this section. By Lemma 2.5, $\mathcal{M}(\Gamma^*) \models \Gamma^*$. Thus Γ^* is satisfiable, and hence so is Γ. $\dashv$

2.7 The finite model property

Let Γ be a consistent finite theory in some language $\mathfrak{L}$. As we now know, Γ has a model. Specifically, we have seen that there is some $\Gamma^* \supseteq \Gamma$ which is a maximal consistent theory with the Henkin property in an extended language $\mathfrak{L}^* \supseteq \mathfrak{L}$. Then we may take the set of constant symbols of $\mathfrak{L}^*$ to be the carrier of a model of Γ^*, hence of Γ. The model obtained in this way is infinite. It is of interest to build a finite model, so in this section Γ must be finite. The easiest way to see that Γ has a finite model is to recall that our overall language is a sub-language of the two variable fragment FO^2 of first-order logic. And FO^2 has the finite model property by Mortimer's Theorem [12].

However, it is possible to give a direct argument for the finite model property, along the lines of filtration in modal logic (but with some differences). We sketch the result here because we shall the same method in Section 3.1 below to prove a finite model property for our second logical system $\mathfrak{L}(adj)$ with respect to its natural semantics; that result does not follow from others in the literature.

Let $\mathcal{M} = \mathcal{M}(\Gamma^*)$ be the canonical model as defined in Section 2.6. Let $Sub(\Gamma)$ be the collection of set terms occurring in any sentence in the original finite theory Γ. So $Sub(\Gamma)$ is finite, and if $\forall(c, r) \in Sub(\Gamma)$ or $\exists(c, r) \in Sub(\Gamma)$, then also $c \in Sub(\Gamma)$. For constant symbols j and k of $\mathfrak{L}^*$, write $j \equiv k$ iff the following conditions hold:

1. If j is a constant of $\mathfrak{L}$, then $k = j$.
2. For all $c \in Sub(\Gamma)$, $c(j) \in \Gamma$ iff $c(k) \in \Gamma$.

Remark The equivalence relation $\equiv$ may be defined on any structure. It is not necessarily a congruence, as Example 1 shows. Specifically, we had constant symbols j and k such that $j \equiv k$, and yet in our structure $s(j, k)$ and $\overline{s}(k, j)$. In the case of $\mathcal{M}(\Gamma^*)$, we have no reason to think that $\equiv$ is a congruence. That is, the construction in Section 2.6 did not arrange for this.

Let $N = \{[k] : k \in \mathbf{K}(\mathfrak{L})\} \times \{\forall, \exists\}$. (We use $\forall$ and $\exists$ as tags to give two copies of the quotient $\mathbf{K}/\equiv$.) We endow N with an $\mathfrak{L}$-structure as follows:

$[\![p]\!] = \{([j], Q) : p(j) \in \Gamma^* \text{ and } Q \in \{\forall, \exists\}\}$.

$[\![s]\!](([j], Q), ([k], Q'))$ iff one of the following two conditions holds:

1. There is a set term c such that Γ^* contains $c(k)$ and $\forall(c, s)(j)$.
2. $Q' = \exists$, and for some $j_* \equiv j$ and $k_* \equiv k$, Γ^* contains $s(j_*, k_*)$.

For a constant j of $\mathfrak{L}$, $[\![j]\!] = ([j], \exists)$. (Of course, $[j]$ is the singleton set $\{j\}$.)

Before going on, we note that the first alternative in the definition of $[\![s]\!](([j], Q), ([k], Q'))$ is independent of the choice of representatives of equivalence classes. And clearly so is the second alternative.

We shall write $\mathcal{N}$ for the resulting $\mathfrak{L}$-structure, hiding the dependence on Γ and Γ^*.

Lemma 2.7 *For all* $c \in \mathrm{Sub}(\Gamma)$, $[\![c]\!] = \{([j], Q) : c(j) \in \Gamma^*$ *and* $Q \in \{\forall, \exists\}\}$.

Proof By induction on set terms c. We are not going to present any of the details here because in Lemma 3.3 below, we shall see all the details on a more involved result. ⊣

Lemma 2.8 $\mathcal{N} \models \Gamma$.

Proof Again we are only highlighting a few details, since the full account is similar to what we saw in Lemma 2.5, and to what we shall see in Lemma 3.4. One would check the sentence types in turn, using Lemma 2.7 frequently. We want to go into details concerning sentences in Γ of the form $s(j, k)$ or $\overline{s}(j, k)$. Recall that we are dealing in this result with sentences of $\mathfrak{L}$, and so j and k are constant symbols of that language. Also recall that $[\![j]\!] = ([j], \exists)$, and similarly for k.

First, consider sentences in Γ of the form $s(j, k)$. By the definition of $[\![s]\!]$, we have

$$[\![s]\!](([j], \exists), ([k], \exists)).$$

By the way binary atoms and constants are interpreted in $\mathcal{N}$, we have $\mathcal{N} \models s(j, k)$, as desired.

We conclude with the consideration of a sentence in Γ of the form $\overline{s}(j, k)$. We wish to show that $\mathcal{N} \models \overline{s}(j, k)$. Suppose towards a contradiction that $\mathcal{N} \models s(j, k)$. Then we have $[\![s]\!](([j], \exists), ([k], \exists))$. There are two possibilities, corresponding to the alternatives in the semantics of s. The first is when there is a set term c such that Γ^* contains $c(k)$ and $\forall(c, s)(j)$. Using $(\forall E)$, Γ^* then contains $s(j, k)$. But recall that Γ contains $\overline{s}(j, k)$. So in this alternative, $\Gamma^* \supseteq \Gamma$ is inconsistent. In the second alternative, there are $j_* \equiv j$ and $k_* \equiv k$ such that $s(j_*, k_*) \in \Gamma^*$. But recall that the equivalence classes of constant symbols from the base language $\mathfrak{L}$ are singletons. Thus in this alternative, $j_* = j$ and $k_* = k$; hence $s(j, k) \in \Gamma^*$. But then again Γ^* is inconsistent, a contradiction. ⊣

Theorem 2.9 (Finite Model Property) *If* Γ *is consistent, then* Γ *has a model of size at most* 2^{2n}*, where* n *is the number of set terms in* Γ.

Complexity notes Theorem 2.9 implies that the satisfiability problem for our language is in NExpTime. We can improve this to an ExpTime-completeness result by quoting the work of others. Pratt-Hartmann [15] define a certain logic $\mathcal{E}_2$ and showed that the complexity of its satisfiability problem is ExpTime-complete. $\mathcal{E}_2$ corresponds to a fragment of first-order logic, and it is somewhat bigger than the language $\mathfrak{L}$. (It would correspond to adding converses to the binary atoms in $\mathfrak{L}$, as we mention at the very end of this paper.) Since satisfiability for $\mathcal{E}_2$ is ExpTime-complete, the same problem for $\mathfrak{L}$ is in ExpTime.

A different way to obtain this upper bound is via the embedding into Boolean modal logic which we saw in Section 2.1. For this, see Theorem 7 of Lutz and Sattler [10]. We shall use an extension of that result below in connection with an extension $\mathfrak{L}(adj)$ of $\mathfrak{L}$.

The ExpTime-hardness for $\mathfrak{L}$ follows from Lemma 6.1 in [16]. That result dealt with a language called $\mathcal{R}^\dagger$, and $\mathcal{R}^\dagger$ is a sub-language of $\mathfrak{L}$.

3 Adding Transitivity: the language $\mathfrak{L}(adj)$

Before going further, let us briefly recapitulate the overall problem of this paper and point out where we are and what remains to be done. We aim to formalize a fragment of first-order logic in which one may represent arguments as complex as that in (1) in the Introduction. We are especially interested in decidable systems, and so the systems must be weaker than first-order logic. We presented in Section 2 a language $\mathfrak{L}$ and a proof system for it. Validity in the logic cannot be captured by a purely syllogistic proof system, and so our proof system uses variables. But the use is very special and restricted. The proof system is complete and decidable in exponential time. To our knowledge, it is the first system with these properties.

There are a number of ways in which one can go further. In this paper, we want to explore one such way, connected to our example in (1). One key feature of this example is that comparative adjectives such as bigger than are transitive. This is true for all comparative adjectives. (Another point of interest is that comparative adjectives are typically irreflexive. We are going to ignore that in the present paper.)

We extend our language $\mathfrak{L}$ to a language $\mathfrak{L}(adj)$ by taking a basic set $\mathbf{A}$ of comparative adjective phrases in the base. The proof system simply extends the one we have already seen with a rule corresponding to the transitivity of comparatives. Our completeness result, Theorem 2.6, extends to the new setting. The next section does this. The decidability of the language is a more delicate matter than before, since it does not follow from Mortimer's Theorem [12] on the finite model property for FO^2. Indeed, adding transitivity statements to FO^2 renders the logic undecidable, as shown in Grädel, Otto, and Rosen [7]. Instead, one could use Theorem 12 of Lutz and Sattler [10] on the decidability of a variant on Boolean modal logic in which some of the relations are taken to be transitive. This would indeed give the ExpTime-compleneness of $\mathfrak{L}(adj)$ with our semantics. However, we have decided to present a direct proof for several reasons. First, Lutz and Sattler's results does not give a finite model property, and our result does do this. Second, our argument is shorter. Finally, our treatment connects to modal filtration arguments and is therefore different; [10] uses automata on infinite trees and is based on Vardi and Wolper [17].

I do not wish to treat the transitivity of comparison with adjectives as an enthymeme (missing premise) because the transitivity seems more fundamental, more 'logical' somehow. Hence it should be treated on a deeper level. The decidability considerations give a supporting argument: if we took the transitivity to be a *meaning postulate*, then it would seem that the underlying language would have to be rich enough to state transitivity. This requires three universal quantifiers. For other reasons, we want our languages to be closed under negation. It thus seems very likely that any logical system with these properties is going to be undecidable. The upshot is a system in which the transitivity turns out to be a *proof postulate* rather than a *meaning postulate*. We turn to the system itself.

Syntax and semantics We start with four pairwise disjoint sets, $\mathbf{A}$ (for *comparative adjective phrases*) and the three that we saw before: $\mathbf{P}$, $\mathbf{R}$, and $\mathbf{K}$. We use a as a variable to range over $\mathbf{A}$ in our statement of the syntax and the rules.

For the syntax, we take elements $a \in \mathbf{A}$ to be binary atoms, just as the elements $s \in \mathbf{R}$ are. Thus, the binary literals are the expressions of the form s, $\overline{s}$, a, or $\overline{a}$.

The syntax is the same as before, except that we allow the binary atoms to be elements of

A in addition to elements of **R**. So in a sense, we have the same syntax is before, except that some of the binary atoms are taken to render transitive verbs, and some are taken to render comparative adjective phrases. The only difference is in the semantics. Here, we require that (in every model $\mathcal{M}$) for an adjective $a \in \mathbf{A}$, $[\![a]\!]$ must be a transitive relation.

Proof system We adopt the same proof system as in Figure 3, but with one addition. This addition is the rule for transitivity:

$$\frac{a(t_1, t_2) \quad a(t_2, t_3)}{a(t_1, t_3)} \text{ trans}$$

This rule is added for all $a \in \mathbf{A}$.

Example 7 We have seen an informal example in (1) at the beginning of this paper. At this point, we can check that our system does indeed have a derivation corresponding to this. We need to check that $\Gamma \vdash \varphi$, where Γ contains

$$\forall(\mathsf{sweet}, \forall(\mathsf{ripe}, \mathsf{bigger})), \forall(\mathsf{pineapple}, \forall(\mathsf{kumquat}, \mathsf{bigger})), \forall(\overline{\mathsf{pineapple}}, \forall(\overline{\mathsf{ripe}}, \mathsf{bigger})),$$

and φ is

$$\forall(\exists(\mathsf{sweet}, \mathsf{bigger}), \forall(\mathsf{kumquat}, \mathsf{bigger})).$$

In Example 6, we saw that $\Gamma \vdash \forall(\mathsf{sweet}, \forall(\mathsf{kumquat}, \mathsf{bigger}))$. (Recall that we saw this with a derivation in a different format, in Figure 5. This could be converted to our official format of natural deduction trees.) That work dealt used $\mathbf{R} = \{\mathsf{bigger}\}$, but here we want $\mathbf{R} = \emptyset$ and $\mathbf{A} = \{\mathsf{bigger}\}$. The same derivation works, of course. Transitivity enables us to obtain a derivation for (1):

$$\dfrac{\dfrac{\dfrac{[\exists(\mathsf{sweet}, \mathsf{bigger})(x)]^3 \qquad \dfrac{\dfrac{[\mathsf{bigger}(x,y)]^2 \qquad \dfrac{[\mathsf{kumquat}(z)]^1 \qquad \dfrac{\vdots}{\dfrac{[\mathsf{sweet}(y)]^2 \quad \forall(\mathsf{sweet}, \forall(\mathsf{kumquat}, \mathsf{bigger}))}{\forall(\mathsf{kumquat}, \mathsf{bigger})(y)} \forall E}}{\mathsf{bigger}(y,z)} \forall E}{\mathsf{bigger}(x,z)} \text{ trans}}{\forall(\mathsf{kumquat}, \mathsf{bigger})(x)} \forall I^1}{\forall(\mathsf{kumquat}, \mathsf{bigger})(x)} \exists E^2}{\forall(\exists(\mathsf{sweet}, \mathsf{bigger}), \forall(\mathsf{kumquat}, \mathsf{bigger}))} \forall I^3$$

Adding the transitivity rule gives a sound and complete proof system for the semantic consequence relation $\Gamma \models \varphi$. The soundness is easy, and so we only sketch the completeness. We must show that a set Γ which is consistent in the new logic has a transitive model. The canonical model $\mathcal{M}(\Gamma)$ as defined in Section 2.6 is automatically transitive; this is immediate from the transitivity rule. And as we know, it satisfies Γ.

3.1 $\mathcal{L}(adj)$ has the finite model property

Our final result is that $\mathcal{L}(adj)$ has the finite model property. We extend the work in Section 2.7. The inspiration for our definitions comes from the technique of *filtration* in modal logic, but we shall not refer explicitly to this area.

We again assume that Γ is consistent, and Γ^* has the properties of Lemma 2.3.

Definition For $a \in \mathbf{A}$, we say that j *reaches* k (*by a chain of* $\equiv$ *and* a *statements*) if there is a sequence

$$j = j_0 \ \equiv \ k_0, \quad j_1 \ \equiv \ k_1, \quad \ldots, \quad j_n \ \equiv \ k_n = k \tag{4}$$

such that $n \geq 1$, and Γ^* contains $a(k_0, j_1)$, ..., $a(k_{n-1}, j_n)$.

Lemma 3.1 *Assume that j reaches k by a chain of $\equiv$ and a statements.*

1. *If $c(k) \in \Gamma^*$, then Γ^* contains $\exists(c, a)(j)$.*
2. *If $j, k \in \mathbf{K}(\mathcal{L})$, then Γ^* contains $a(j, k)$.*

Proof By induction on $n \geq 1$ in (4). For $n = 1$, we have essentially seen the argument as a step in Lemma 2.7. Here it is again. Since $c(k_1)$ and $j_1 \equiv k_1$, we see that $c(j_1)$. Together with $s(k_0, j_1)$, we have $\exists(c, a)(k_0)$. And as $j_0 \equiv k_0$, we see that $\exists(c, a)(j_0)$.

Assume our result for n, and now consider a chain as in (4) of length $n + 1$. The induction hypothesis applies to

$$j = j_1 \ \equiv \ k_1, \quad j_2 \ \equiv \ k_2, \quad \ldots, \quad j_{n+1} \ \equiv \ k_{n+1} = k$$

and so we have $\exists(c, a)(j_1)$. Since $a(k_0, j_1)$, we easily have $\exists(c, a)(k_0)$ by transitivity. And as $j_0 \equiv k_0$, we have $\exists(c, a)(j_0)$.

The second assertion is also proved by induction on $n \geq 1$. For $n = 1$, we have $j = j_0 \equiv k_0$, Γ^* contains $a(k_0, j_1)$; and $j_1 \equiv k_1 = k$. Then since the $\equiv$ is the identity on $\mathbf{K}(\mathcal{L})$, $j = j_0 = k_0$, and $j_1 = k_1 = k$. Hence Γ^* contains $s(j, k)$. Assuming our result for n, we again consider a chain as in (4) of length $n + 1$. Just as before, $j = j_0 = k_0$, and so Γ^* contains $a(j, j_1)$. By induction hypothesis, Γ^* contains $a(j_1, k)$. By transitivity, Γ^* contains $a(j, k)$. $\dashv$

We endow N with an $\mathcal{L}$-structure as follows:

$[\![p]\!] = \{([j], Q) : p(j) \in \Gamma^* \text{ and } Q \in \{\forall, \exists\}\}$.

$[\![s]\!](([j], Q), ([k], Q'))$ iff one of the following two conditions holds:

1. There is a set term c such that Γ^* contains $c(k)$ and $\forall(c, s)(j)$.
2. $Q' = \exists$, and for some $j_* \equiv j$ and $k_* \equiv k$, Γ^* contains $s(j_*, k_*)$.

$[\![a]\!](([j], Q), ([k], Q'))$ iff

1. If $\forall(c, a)(k) \in \Gamma^*$, then also $\forall(c, a)(j) \in \Gamma^*$.
2. In addition, either (a) or (b) below holds:

(a) There is a set term c such that Γ^* contains $c(k)$ and $\forall(c,a)(j)$.
(b) $Q' = \exists$, and j reaches k by a chain of $\equiv$ and a statements.

For a constant j of $\mathcal{L}$, $[\![j]\!] = ([j], \exists)$.

Once again, we suppress Γ and Γ^* and simply write $\mathcal{N}$ for the resulting $\mathcal{L}$-structure.

Lemma 3.2 *For $a \in \mathbf{A}$, each relation $[\![a]\!]$ is transitive in $\mathcal{N}$.*

Proof In this proof and the next, we are going to use l to stand for a constant symbol, even though earlier in the paper we used it for a literal. Assume that

$$([j], Q)\ [\![a]\!]\ ([k], Q')\ [\![a]\!]\ ([l], Q''). \tag{5}$$

Clearly we have the first requirement concerning $[\![a]\!]$: if $\forall(c,a)(l) \in \Gamma^*$, then also $\forall(c,a)(j) \in \Gamma^*$.

We have four cases, depending on the reasons for the two assertions in (5).

Case 1 There is a set term b such that Γ^* contains $b(k)$ and $\forall(b,a)(j)$, and there is also a set term c such that Γ^* contains $c(l)$ and $\forall(c,a)(k)$. By (1), Γ^* contains $c(l)$ and $\forall(c,a)(j)$. And so we have requirement (2a) concerning $[\![a]\!]$ for $([j], Q)$ and $([l], Q'')$.

Case 2 There is a set term b such that Γ^* contains $b(k)$ and $\forall(b,a)(j)$, and k reaches l. Note that $a(j,k)$. So j reaches l.

Case 3 j reaches k by a chain of $\equiv$ and a statements, and there is a set term c such that Γ^* contains $c(l)$ and $\forall(c,a)(k)$. Then $a(k,l)$. And so j reaches l.

Case 4 j reaches k, and k reaches l. Then concatenating the chains shows that j reaches l.

$\dashv$

Lemma 3.3 *For all $c \in \mathrm{Sub}(\Gamma)$, $[\![c]\!] = \{([j], Q) : c(j) \in \Gamma^*$ and $Q \in \{\forall, \exists\}\}$.*

Proof We argue by induction on c. Much of the proof is as in Lemma 2.7, For c a unary atom, the result is obvious. Also, assuming that $[\![c]\!] = \{([j], Q) : c(j) \in \Gamma^*\}$ we easily have the same result for $\overline{c}$ using the maximal consistency of Γ^*:

$$([j], Q) \in [\![\overline{c}]\!] \quad \text{iff} \quad ([j], Q) \notin [\![c]\!] \quad \text{iff} \quad c(j) \notin \Gamma^* \quad \text{iff} \quad \overline{c}(j) \in \Gamma^*.$$

Assume about c that if $c \in \mathit{Sub}(\Gamma)$, then $[\![c]\!] = \{([j], Q) : c(j) \in \Gamma^*\}$. In view of what we just saw, we only need to check the same result for $\forall(c,s)$, $\exists(c,s)$, $\forall(c,a)$, and $\exists(c,\overline{a})$.

$\underline{\forall(c,s)}$ Suppose that $\forall(c,s) \in \mathit{Sub}(\Gamma)$, so that $c \in \mathit{Sub}(\Gamma)$ as well. We prove that

$$[\![\forall(c,s)]\!] \quad = \quad \{([j], Q) : \forall(c,s)(j) \in \Gamma^*\}.$$

Let $([j], Q) \in [\![\forall(c,s)]\!]$. We shall show that $\forall(c,s)(j) \in \Gamma^*$. If not, then by maximal consistency, $\exists(c,\overline{s})(j) \in \Gamma^*$. By the Henkin property, let k be such that Γ^* contains $c(k)$ and $\overline{s}(j,k)$. By induction hypothesis, $([k], \forall) \in [\![c]\!]$. And so $([j], \forall)[\![s]\!]([k], \forall)$. Thus there is a set term b such that Γ^* contains $b(k)$ and $\forall(b,s)(j)$. From these, Γ^* contains $s(j,k)$. And thus Γ^* is inconsistent. This contradiction shows that indeed $\forall(c,s)(j) \in \Gamma^*$.

In the other direction, suppose that $([j], Q)$ is such that $\forall(c,s)(j) \in \Gamma^*$. Let $([k], Q') \in [\![c]\!]$, so by induction hypothesis, $c(k) \in \Gamma^*$. By the way we interpret binary relations in $\mathcal{N}$, $[\![s]\!](([j], Q), ([k], Q'))$. This for all $([k], Q') \in [\![c]\!]$ shows that $([j], Q) \in [\![\forall(c,s)]\!]$.

$\underline{\exists(c,s)}$ Suppose that $\exists(c,s) \in Sub(\Gamma)$, so that $c \in Sub(\Gamma)$ as well. Let $([j],Q) \in [\![\exists(c,s)]\!]$. Let k and Q' be such that $[\![c]\!]([k],Q')$ and $[\![s]\!](([j],Q),([k],Q'))$. By induction hypothesis, $c(k) \in \Gamma^*$. First, let us consider the case when $Q' = \forall$. Let b be such that Γ^* contains $b(k)$ and $\forall(b,s)(j)$. Using $(\forall E)$, we have $\Gamma^* \vdash \exists(c,s)(j)$. And as Γ^* is closed under deduction, $\exists(c,s)(j) \in \Gamma^*$ as desired. The more interesting case is when $Q' = \exists$, so that for some $j_* \equiv j$ and $k_* \equiv k$, Γ^* contains $s(j_*,k_*)$. Since $c(k)$ and $k \equiv k_*$, we have $c(k_*) \in \Gamma^*$. Then using $(\exists I)$, we see that $\exists(c,s)(j_*) \in \Gamma^*$. Since $j \equiv j_*$, once again we have $\exists(c,s)(j) \in \Gamma^*$.

Conversely, suppose that $\exists(c,s)(j) \in \Gamma^*$. By the Henkin property, let k be such that $c(k)$ and $s(j,k)$ belong to Γ^*. Then $[\![s]\!](([j],Q),([k],\exists))$, and by induction hypothesis, $[\![c]\!](k)$. Hence $([j],Q) \in [\![\exists(c,s)]\!]$.

$\underline{\forall(c,a)}$ Suppose that $\forall(c,a) \in Sub(\Gamma)$, so that $c \in Sub(\Gamma)$ as well. We prove that

$$[\![\forall(c,a)]\!] \quad = \quad \{([j],Q) : \forall(c,a)(j) \in \Gamma^*\}.$$

The first part argument is the left-to-right inclusion. It is exactly the same as what we saw above for the sentences of the form $\forall(c,s)$.

In the other direction, suppose that $\forall(c,a)(j) \in \Gamma^*$; we show that $([j],Q) \in [\![\forall(c,a)]\!]$. For this, let $([k],Q') \in [\![c]\!]$. By induction hypothesis, $c(k) \in \Gamma^*$. We must verify that if $\forall(b,a)(k) \in \Gamma^*$, then also $\forall(b,a)(j) \in \Gamma^*$. This is shown in the derivation below:

$$\dfrac{\dfrac{\dfrac{c(k) \quad \forall(c,a)(j)}{a(j,k)}\,\forall E \quad \dfrac{[b(x)]^1 \quad \forall(b,a)(k)}{a(k,x)}\,\forall E}{a(j,x)}\,\text{trans}}{\forall(b,a)(j)}\,\forall I^1$$

Since Γ^* is closed under deduction, we see that indeed $\forall(b,a)(j) \in \Gamma^*$. Going on, we see from the structure of $\mathcal{N}$ that $[\![s]\!](([j],Q),([k],Q'))$. This for all $([k],Q') \in [\![c]\!]$ shows that $([j],Q) \in [\![\forall(c,a)]\!]$.

$\underline{\exists(c,a)}$ Suppose that $\exists(c,a) \in Sub(\Gamma)$, so that $c \in Sub(\Gamma)$ as well.

Let $([j],Q) \in [\![\exists(c,a)]\!]$. Let k and Q' be such that $[\![a]\!](([k],Q')$ and $[\![a]\!](([j],Q),([k],Q'))$. By induction hypothesis, $c(k) \in \Gamma^*$. There are two cases depending on whether $Q' = \forall$ or $Q' = \exists$. The argument for $Q' = \forall$ is the same as the one we saw in our work on sentences $\exists(c,s)$ above. The more interesting case is when $Q' = \exists$. This time, j reaches k. By Lemma 3.1, $\exists(c,a)(j) \in \Gamma^*$.

Conversely, suppose that $\exists(c,a)(j) \in \Gamma^*$. By the Henkin property, let k be such that $c(k)$ and $a(j,k)$ belong to Γ^*. The derivation below shows that if $\forall(d,a)(k) \in \Gamma^*$, then $\forall(d,a)(j) \in \Gamma^*$ as well:

$$\dfrac{\dfrac{a(j,k) \quad \dfrac{[d(x)]^1 \quad \forall(d,a)(k)}{a(k,x)}\,\forall E}{a(j,x)}\,\text{trans}}{\forall(d,a)(j)}\,\forall I^1$$

So $[\![a]\!](([j],Q),([k],\exists))$, and by induction hypothesis, $[\![c]\!](k)$. Hence $([j],Q) \in [\![\exists(c,a)]\!]$.

This completes the induction. $\dashv$

Lemma 3.4 $\mathcal{N} \models \Gamma$.

Proof We check the sentence types in turn, using Lemma 3.3 without mention.

First, let Γ contain the sentence $\forall(b, c)$. Then b and c belong to $Sub(\Gamma)$. Let $([j], Q) \in [\![b]\!]$, so that $b(j) \in \Gamma^*$. We have $d(j) \in \Gamma^*$ using $(\forall E)$. This for all $([j], Q)$ shows that $\mathcal{N} \models \forall(b, c)$.

Second, let $\exists(c, d) \in \Gamma$. By the Henkin condition, let j be such that both $c(j)$ and $d(j)$ belong to Γ^*. The element $([j], \forall)$ shows that $[\![c]\!] \cap [\![d]\!] \neq \emptyset$. That is, $\mathcal{N} \models \exists(c, d)$.

Continuing, consider a sentence $b(j) \in \Gamma$. As $b \in Sub(\Gamma)$, we have $([j], \exists) \in [\![b]\!]$, so that $\mathcal{N} \models b(j)$.

The work for sentences of the forms $s(j, k)$ and $\overline{s}(j, k)$ was done in Lemma 2.8.

The most intricate part of this proof concerns sentences $a(j, k), \overline{a}(j, k) \in \Gamma$. Recall that we are dealing in this result with sentences of $\mathcal{L}$, and so j and k are constant symbols of that language. Also recall that $[\![j]\!] = ([j], \exists)$, and similarly for k.

Consider sentences in Γ of the form $a(j, k)$. It is easy to see that if $\exists(c, a)(k)$ belongs to Γ, then so does $\exists(c, a)(j)$. (See the $\exists(c, a)$ case in Lemma 3.3.) From this it follows easily that $[\![a]\!]([\![j]\!], [\![k]\!])$. And so $\mathcal{N} \models a(j, k)$ in this case.

We conclude with the consideration of a sentence in Γ of the form $\overline{a}(j, k)$. We wish to show that $\mathcal{N} \models \overline{a}(j, k)$. Suppose towards a contradiction that $\mathcal{N} \models a(j, k)$. Then we have $[\![a]\!](([j], \exists), ([k], \exists))$. There are two possibilities, corresponding to the alternatives in the semantics of a. The first is when there is a set term c such that Γ^* contains $c(k)$ and $\forall(c, a)(j)$. Using $(\forall E)$, Γ^* then contains $a(j, k)$. But recall that Γ contains $\overline{a}(j, k)$. So in this alternative, $\Gamma^* \supseteq \Gamma$ is inconsistent. In the second alternative, j reaches k by a chain of $\equiv$ and a statements. By Lemma 3.1, $a(j, k) \in \Gamma^*$. So Γ^* is inconsistent, and we have our contradiction. $\dashv$

Once again, this gives us the finite model property for $\mathcal{L}(adj)$. The result is not interesting from a complexity-theoretic point of view, since we already could see from Lutz and Sattler [10] that the logic had an ExpTime satisfiability problem.

4 Conclusion and Future Work

This paper has provided two logical systems, $\mathcal{L}$ and $\mathcal{L}(adj)$, along with semantics. We presented proof systems in the format of natural deduction, and in both cases we have completeness theorems and the finite model property. The semantics of the language allows us to translate some natural language sentences into the languages faithfully.

Set terms in this sense of this paper come from McAllester and Givan [11], where they are called *class terms*. That paper was probably the first to present an infinite fragment relevant to natural language and to study its logical complexity. Their work concernes two fragments: a full fragment equivalent to first-order logic, and a "quantifier-free" sub-fragment. ("Quantifier-free" in [11] means that the fragment lacks the lambda terms present in the full version; the smaller system still has quantifiers `some` and `every` in the same manner as the present paper.) A main result on the quantifier-free fragment is that its satisfiability problem is NP-complete. The quantifier-free fragment is essentially the language $\mathcal{R}^*$ of Pratt-Hartmann and Moss [16]. This language $\mathcal{R}^*$ is extended in [16] to a language called $\mathcal{R}^{*\dagger}$ by permitting full noun-level negation. $\mathcal{R}^{*\dagger}$ is equivalent in most respects to the language $\mathcal{L}$ of this paper, but there are two small differences. First, $\mathcal{L}$ has constant symbols. In addition to making the system more

expressive, constants allowed us to present a Henkin-style completeness proof in Sections 2.5. The other change is that $\mathcal{R}^{*\dagger}$ does not allow recursively defined set terms, only "flat" terms. However, from the point of view of decidability and complexity, this change is really minor: one may add new symbols to flatten a sentence, at the small cost of adding new sentences. The flat version is also essentially the same as the language $\mathcal{E}_2$ of Pratt-Hartmann [15].

The decidability of $\mathcal{L}(adj)$ follows from known results on Boolean modal logics [10], but the finite model property appears to be new here.

Proof systems for fragments that are weaker than $\mathcal{L}$ appear in [16]. These proof systems are syllogistic; there are no variables or what we have in this paper called general sentences. Modulo complexity hypotheses, the proof systems of this paper are the first ones which are complete and go beyond the capabilities of syllogistic proof systems. At the same time, they are decidable and useable. For example, we have seen how the inference in (1) in the Introduction is handled in our system; see Examples 6 and 7.

The use of natural deduction proofs in connection with natural language is very old, going back to Fitch [4]. Fitch's paper does not deal with a formalized fragment, and so it is not possible to even ask about questions like completeness and decidability. Also, the phenomena of interest in the paper went beyond what we covered here. We would like to think that the methods of this paper could eventually revive interest in Fitch's proposal by giving it a foundation.

Francez and Dyckhoff [5] propose a *proof-theoretic semantics* for natural language. Their far-ranging proposal goes beyond what we can discuss in this paper. We only want to mention that our proof rules bear some similarity to theirs. Their system had no recursive constructs and also no negative determiners, but it went beyond ours in covering both readings of scope-ambiguous simple sentences. Since our motivation was not proof-theoretic in this paper, we did not investigate proof-theoretic properties of our system beyond the two-variable result noted in Section 2.2. But it would be interesting to do so.

It is of interest to go further in order to render more of natural language inference in complete and decidable logical systems. One next step would be to add converses to the binary atoms in order to express simple sentences beyond what we have seen. For example, writing sees^{-1} for the inverse of see, we could render $\mathsf{Every\ girl\ sees\ Mary}$ as $\forall(\mathsf{girl}, \mathsf{see}^{-1})(\mathsf{Mary})$. It is possible to extend our work in such a way as to incorporate these converses. The logic would be axiomatized on top of $\mathcal{L}$ by adding the rule deriving $r^{-1}(t, u)$ from $r(u, t)$. But this is only one of the many things to do.

Acknowledgement

My thanks to Ian Pratt-Hartmann for his comments on an earlier draft.

References

[1] Johan van Benthem. *Essays in Logical Semantics*. Reidel, Dordrecht, 1986,

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

[3] Dirk van Dalen. **Logic and Structure** (Fourth edition). Universitext, Springer-Verlag, Berlin, 2004.

[4] Frederic B. Fitch. Natural Deduction Rules for English. *Philosophical Studies*, 24:2 (1973), 89–104.

[5] Nissim Francez and Roy Dyckhoff. Proof-Theoretic Semantics for a Natural Language Fragment. To appear in the *Journal of Logic, Language, and Information*.

[6] Erich Grädel, Phokion Kolaitis, and Moshe Vardi. On the Decision Problem for Two-Variable First-Order Logic. *Bulletin of Symbolic Logic*, 3:1 (1997), 53–69.

[7] Erich Grädel, Martin Otto, and Eric Rosen. Undecidability Results on Two-Variable Logics. Archive for Mathematical Logic, vol. 38, pp. 313–354, 1999.

[8] Yuri Gurevich. On the Classical Decision Problem. Logic in Computer Science Column, the *Bulletin of the European Association for Theoretical Computer Science*, October 1990.

[9] I. L. Humberstone. Inaccessible Worlds. *Notre Dame J. Formal Logic*, Volume 24, Number 3, 346–352, 1983.

[10] Carsten Lutz and Ulrike Sattler. The Complexity of Reasoning with Boolean Modal Logics. In F. Wolter et al (eds.), *Advances in Modal Logics*, Vol. 3. CSLI Publications, Stanford, 2001.

[11] David McAllester and Robert Givan. Natural Language Syntax and First Order Inference. *Artificial Intelligence* vol. 56, no. 1, 1992, 1–20.

[12] Michael Mortimer. On Languages with Two Variables. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 21, 135–140, 1975.

[13] Francis Jeffry Pelletier. A Brief History of Natural Deduction. *History and Philosophy of Logic* 20(1), 1–31, 1999.

[14] Ian Pratt-Hartmann. A Two-Variable Fragment of English. *Journal of Logic, Language and Information*, 12(1), 2003, pp. 13–45.

[15] Ian Pratt-Hartmann. Fragments of Language. *Journal of Logic, Language and Information*, 13:207–223, 2004.

[16] Ian Pratt-Hartmann and Lawrence S. Moss. Logics for the Relational Syllogistic. To appear in the *Review of Symbolic Logic* Vol. 2, No. 3, 37 pp, 2009.

[17] Moshe Y. Vardi and Pierre Wolper. Automata-Theoretic Techniques for Modal Logics of Programs. *Journal of Computer and System Sciences*, Volume 32, Issue 2 (1986), 183–221.