UCLA Working Papers in Linguistics, Theories of Everything
Volume 17, Article 31:261-273, 2012

Inference in a Boolean Fragment

Lawrence S. Moss

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, intransitive and transitive verbs, converses (for passive sentences), subject relative clauses (a recursive construct), and conjunction and negation on nouns and verbs. We also give a proof system which is complete and is decidable due to the the finite model property. The fragment itself was inspired by Ivanov and Vakarelov [2]. Our logical system differs from theirs in several respects and is an extension of our system in [6]. At the time of this writing, I know of no strictly larger system than the one in this paper and in [2] which is complete and decidable and which is capable of representing interesting in natural language.

Introduction

I was a student at UCLA from 1976 to 1984, mainly studying logic, mathematics, and linguistics. After taking a few undergraduate linguistics courses, I wandered into a class or seminar on *Boolean Semantics for Natural Language*. Way before computers, the class would have read a draft version of what was published some years later as [3]. I was immediately attracted to formal semantics; it seemed to have so much of what I sought in life. This kind of attraction has stuck with me, even though I never pursued the subject full force. Getting back to to my days as an undergraduate and later a graduate student at UCLA, I occasionally attended the seminars that Ed ran on topics like generalized quantifiers. I later worked with him on two papers on GQ theory. And in recent years we've been working on a textbook for his course on Mathematical Structures in Language. I always appreciate Ed's wide-ranging knowledge of linguistics, languages, and Language, and also his visionary applications of mathematical ideas. On top of all that, he's a real mensch. It's a pleasure to dedicate this paper to you, Ed.

Page 2 of *Boolean Semantics for Natural Language* proclaims "The fundamental relation we, and others, endeavor to represent is the *entailment (=logical implication, logical consequence)* relation." Semanticists usually *are* interested in facts of entailment, and they frequently study small fragments of language. But for the most part, they have not done what we aim to do in this paper: to present a small fragment of language with both syntax and semantics, and then to completely characterize the semantic entailment relation in terms of a deductive system of the sort one finds in logic.

We shall present a logical system in which one can carry out inferences in natural language, such as the following

$$\frac{\textit{Some dog sees some cat}}{\textit{Some cat is seen by some dog}} \tag{1}$$

$$\frac{\textit{Bao is seen and heard by every student} \quad \textit{Amina is a student}}{\textit{Amina sees Bao}} \tag{2}$$

$$\frac{\textit{All skunks are mammals}}{\textit{All who fear all who respect all skunks fear all who respect all mammals}} \tag{3}$$

I take all of these to be valid inferences in the sense that a competent speaker who accepts the premisses (above the line) will accept the conclusion. (1) involves the passive, as does (2). The latter also has conjunction in the VP. (3) is a complicated example of iterated subject relative clauses. In my experience with this example during talks, most people cannot see that (3) is a valid inference. I mention this to point out that fragments which are syntactically very simple might still host non-trivial inferences.

This paper provides a logical system in which one may formally derive inferences corresponding to these examples, and many others. One might think that the simplest way to achieve this goal would be to simply use first-order logic (FOL). However, FOL is *undecidable*. We aim for a decidable system; indeed, we aim for a system of low computational complexity. It is currently open to devise logical systems which are decidable but at the same time are as expressive in the sense of being able to represent as much natural language inference as possible. This paper represents work in this direction.

Previous work The language introduced in this paper was based on the language in Ivanov and Vakarelov [2]. That paper goes beyond systems of natural logic in having relational inverses and the boolean operators on sentences. Specifically, the language in our paper [6] does not have these features because it lacks boolean connectives on one-place and two-place relations. The connectives are present in Ivanov and Vakarelov [2]; that paper also has full boolean connectives on sentences. However, the proof system in [2] is a Hilbert-style system, and so most people would find it more difficult to use. In addition, we believe that our completeness proof is substantially easier than the proof in [2]. However, this paper presents only a sketch and so it will not be persuasive on this point.

To summarize: the main technical contribution in this paper is a natural deduction-style logical system related to that of [2] and with an easier completeness proof.

1 The Language $\mathcal{L}$

There is only one language in this paper, called $\mathcal{L}$. $\mathcal{L}$ is based on three pairwise disjoint sets called **P**, **R**, and **K**. These are called *unary atoms*, *binary atoms*, and *constant symbols*.

The unary atoms correspond *predicate symbols*, and the binary atoms *relation symbols*, and this is the reason for our notations **P** and **R**. However, there is a significant difference having to do with variables, as we shall see shortly.

We speak of $\mathcal{L}$ as a language, but actually it is a family of languages parameterized by sets of basic symbols.

1.1 Syntax and semantics

The syntax of $\mathcal{L}$ is in Figure 1. Relational terms are built from unary and binary atoms using a negation, conjunction, and inverse on binary relations. Set terms (that is, terms whose denotations are subsets of the universe) are defined from relational terms using and quantification turning binary relational terms and a form of quantification; the semantics of that form of quantification is given below. Sentences are formed from set terms using either quantification or else by plugging the constants into relational terms. The second column in Figure 1 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.

Constant symbols correspond to proper names, unary atoms to one-place predicates which we gloss as plural nouns or intransitive verbs, and relation symbols to transitive verbs.

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. We do not take it to be involutive, so that $\overline{\overline{p}}$ and p are technically different symbols. Of course, the negation operation is semantically involutive, and it will turn out that in the proof theory two involutively related expressions will be provably equivalent.

The set terms in this language are the one and only recursive construct in the language. 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 $\forall(b,\overline{s})$; this can be read as fails to see all boys, or (better) sees no boys or doesn't see any boys. We also have $\exists(b,\overline{s})$, fails to see some boys. But the recursion allows us to embed set terms, and so we have set terms like

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

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

Sentences allow quantification as in $\forall(b,c)$ and $\exists(b,c)$; the semantics renders these in the obvious way, using inclusion and disjointness of the extensions of the set terms. We also have sentences using the constants, such as $\forall(g,s)(m)$, corresponding to Mary sees all girls. Using relational inverses, we can also say $\forall(g,s^{-1})(m)$, corresponding to Mary is seen by all girls. We should note that the relative clauses which can be obtained in this way are all “missing the subject”, never “missing the object”. The language is too poor to express predicates like $\lambda x.$all boys see x.

Expression	Variables	Syntax
unary atom	p, q	
binary atom	b	
constant	j, k	
unary relational term	l, m	$p \mid \bar{l} \mid l \wedge m$
binary relational term	r, s	$b \mid r^{-1} \mid \bar{r} \mid r \wedge s$
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 terms and sentences of $\mathcal{L}$.

Some decisions We chose to keep the language small for this paper. So we did not include disjunction on the terms, and we also did not include boolean connectives on the level of sentences. It would not be hard to add those and still obtain a complete and decidable system.

Semantics A *structure* (for this language $\mathcal{L}$) is a pair $\mathcal{M} = \langle \mathcal{M}, [\![\ ]\!] \rangle$, 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

$$
\begin{array}{lcl}
[\![\bar{l}]\!] & = & M \setminus [\![l]\!] \\
[\![l \wedge m]\!] & = & [\![l]\!] \cap [\![m]\!] \\
[\![\bar{r}]\!] & = & M^2 \setminus [\![r]\!] \\
[\![r^{-1}]\!] & = & [\![r]\!]^{-1} \\
[\![r \wedge s]\!] & = & [\![r]\!] \cap [\![s]\!] \\
[\![\exists(l,t)]\!] & = & \{x \in M : \text{ for some } y \in [\![l]\!], [\![t]\!](x,y)\} \\
[\![\forall(l,t)]\!] & = & \{x \in M : \text{ for all } y \in [\![l]\!], [\![t]\!](x,y)\}
\end{array}
$$

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

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

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

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.

Since $\mathcal{L}$ translates into the *two-variable fragment* FO^2 of first-order logic, the satisfiability problem for $\mathcal{L}$ is decidable. $\mathcal{L}$ also has the finite model property (Mortimer [5]): every sentence with a model has a finite model.

The bar notation on all syntactic items Note that every syntactic item has a negation: for relational terms, this is immediate from the definition of the syntax (no abbreviations are needed). For set terms and sentences: $\overline{\exists(l,r)} = \forall(l,\bar{r})$, $\overline{\forall(l,r)} = \exists(l,\bar{r})$, $\overline{\forall(c,d)} = \exists(c,\overline{d})$, $\overline{\exists(c,d)} = \forall(c,\overline{d})$, $\overline{c(j)} = \overline{c}(j)$, and $\overline{r(j,k)} = \overline{r}(j,k)$.

Rendering our examples in $\mathcal{L}$ Returning to (1), (2), and (3), we translate them into our language as follows:

(4)
$$\frac{\exists(\mathsf{dog}, \exists(\mathsf{cat}, \mathsf{see}))}{\exists(\mathsf{cat}, \exists(\mathsf{dog}, \mathsf{see}^{-1}))}$$

(5)
$$\frac{\forall(\mathsf{student}, \mathsf{see}^{-1} \wedge \mathsf{hear}^{-1})(\mathsf{Bao}) \quad \mathsf{student}(\mathsf{Amina})}{\mathsf{see}(\mathsf{Amina}, \mathsf{Bao})}$$

(6)
$$\frac{\forall(\mathsf{skunk}, \mathsf{mammal})}{\forall(\forall(\forall(\mathsf{skunk}, \mathsf{respect}), \mathsf{fear}), \forall(\forall(\mathsf{mammal}, \mathsf{respect}), \mathsf{fear}))}$$

We shall see a proof system for these inferences in the next section.

We also translate $\mathcal{L}$ to FO^2, the fragment of first order logic using only the variables x and y. 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 $c \mapsto \varphi_{c,x}$:

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

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

1.2 Proof system

We present our system in natural-deduction style in Figure 3. It makes use of *introduction* and *elimination* rules, and more critically of *variables*.

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.

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

Figure 2: Syntax of general sentences of $\mathcal{L}$, with φ ranging over sentences, c over set terms, and r over relational terms.

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 1.3 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.

Two-way rules The rules $\wedge$ and inv are two-way rules, going up and down. For example: from $(r \wedge s)(j,k)$, we may derive $r(j,k)$ and also $s(j,k)$. From both $r(j,k)$ and $s(j,k)$, we derive $(r \wedge s)(j,k)$.

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.

Formal proofs in the Fitch style Textbook presentations of logic overwhelmingly use natural deduction instead of Hilbert-style systems because the latter approach requires one to use complicated propositional tautologies at every step, and it also lacks the facility to use sub-proofs with temporary assumptions. I have chosen to present the system of this paper in

$$\frac{c(t) \quad \forall(c,d)}{d(t)}\ \forall E \qquad\qquad \frac{c(u) \quad \forall(c,r)(t)}{r(t,u)}\ \forall E$$

$$\frac{c(t) \quad d(t)}{\exists(c,d)}\ \exists I \qquad\qquad \frac{r(t,u) \quad c(u)}{\exists(c,r)(t)}\ \exists I$$

$$\frac{\begin{matrix}[c(x)]\\ \vdots \\ d(x)\end{matrix}}{\forall(c,d)}\ \forall I \qquad\qquad \frac{\begin{matrix}[c(x)]\\ \vdots \\ r(t,x)\end{matrix}}{\forall(c,r)(t)}\ \forall I$$

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

$$\frac{r(j,k) \quad s(j,k)}{(r \wedge s)(j,k)}\ \wedge \qquad\qquad \frac{r^{-1}(k,j)}{r(j,k)}\ \mathrm{inv}$$

$$\frac{\alpha \quad \overline{\alpha}}{\perp}\ \perp I \qquad\qquad \frac{\begin{matrix}[\overline{\varphi}]\\ \vdots \\ \perp\end{matrix}}{\varphi}\ RAA$$

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

a "classical" Gentzen-style format. But the system may easily be re-formatted to look more like a Fitch system. For more on this, see [6].

1.3 Examples

We present a few examples of the proof system at work, along with comments pertaining to the side conditions. Some of these are taken from the proof system $\mathbf{R}^*$ for the language $\mathcal{R}^*$ of [7]. 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 1. Here is a formal derivation of (4):

$$\frac{\exists(\mathsf{dog},\exists(\mathsf{cat},\mathsf{see})) \qquad \dfrac{[\exists(\mathsf{cat},\mathsf{see})(x)]^2 \qquad \dfrac{[\mathsf{cat}(y)]^1 \qquad \dfrac{[\mathsf{dog}(x)]^2 \qquad \dfrac{[\mathsf{see}(x,y)]^1}{\mathsf{see}^{-1}(y,x)}}{\exists(\mathsf{dog},\mathsf{see}^{-1})(y)}\ \exists I}{\exists(\mathsf{cat},\exists(\mathsf{dog},\mathsf{see}^{-1}))}\ \exists I}{\exists(\mathsf{cat},\exists(\mathsf{dog},\mathsf{see}^{-1}))}\ \exists E^1}{\exists(\mathsf{cat},\exists(\mathsf{dog},\mathsf{see}^{-1}))}\ \exists E^2$$

Example 2. Here is a formal derivation of (5):

$$\dfrac{\dfrac{\dfrac{\forall(\mathsf{student},\mathsf{see}^{-1}\wedge\mathsf{hear}^{-1})(\mathsf{Bao}) \quad \mathsf{student}(\mathsf{Amina})}{(\mathsf{see}\wedge\mathsf{hear})^{-1}(\mathsf{Bao},\mathsf{Amina})}\,\forall E}{(\mathsf{see}\wedge\mathsf{hear})(\mathsf{Amina},\mathsf{Bao})}\,\mathrm{inv}}{\mathsf{see}(\mathsf{Amina},\mathsf{Bao})}\,\wedge$$

Example 3. Here is a a formal derivation that $\forall(c,d) \vdash \forall(\forall(d,r),\forall(c,r))$:

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

In the first application of $(\forall I)$, x is not free in any uncanceled hypothesis. (That is, it is free in $c(x)$, but this is canceled in the application.) The other variable y is free at that point, and it is quantified away by the second application of $(\forall I)$.

Example 4. We use the previous example to give a derivation (in shortened form) for (6):

$$\dfrac{\dfrac{\forall(\mathsf{skunk},\mathsf{mammal})}{\forall(\forall(\mathsf{mammal},\mathsf{respect}),\forall(\mathsf{skunk},\mathsf{respect}))}}{\forall(\forall(\forall(\mathsf{skunk},\mathsf{respect}),\mathsf{fear}),\forall(\forall(\mathsf{mammal},\mathsf{respect}),\mathsf{fear}))}$$

Note that we used Example 3 twice and that the inference is *antitone* each time: skunk and mammal have switched positions.

Example 5. Here is an example of a derivation using (RAA). It shows $\forall(c,\overline{c}) \vdash \forall(d,\forall(c,r))$.

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

Example 6. 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 [7].)

Example 7. *RAA* gives the rule of double negation:

$$\dfrac{\dfrac{\overline{\overline{\varphi}} \quad [\overline{\varphi}]}{\bot}}{\varphi}$$

Example 8. For binary relations, inverse and complement commute: $(\overline{R})^{-1} = \overline{R^{-1}}$. Here are the reflections of this fact in the proof system:

$$\dfrac{\dfrac{\dfrac{\overline{r}^{-1}(j,k)}{\overline{r}(k,j)} \qquad \dfrac{[r^{-1}(j,k)]}{r(k,j)}}{\bot}}{\overline{\overline{r^{-1}}}(j,k)} \qquad\qquad \dfrac{\dfrac{\dfrac{\overline{r^{-1}}(j,k) \qquad \dfrac{[r(k,j)]}{r^{-1}(j,k)}}{\bot}}{\overline{r}(k,j)}}{\overline{r}^{-1}(j,k)}$$

1.4 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:

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

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 $m(x)$.

Lemma 1. *The proof system is sound: if $\Gamma \vdash \varphi$, then $\Gamma \models \varphi$.*

See [6] for the proof in a smaller fragment. The argument there easily extends to the current language; the point is that the side conditions on the quantifier rules insure that the proof system is sound.

1.5 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; (2) 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 $\varphi \in \mathcal{L}$, if $\Gamma^* \vdash \varphi$, then $\Gamma \vdash \varphi$. (Notice that if Γ is consistent and $\Gamma^* \supseteq \Gamma$ is a conservative extension, then Γ^* is also consistent.)

Lemma 2. *Let* Γ *be a consistent* $\mathcal{L}$*-theory, and let* $j,k \notin \mathbf{K}(\mathcal{L})$.

1. *If* $\exists(c,d) \in \Gamma$*, then* $\Gamma + c(j) + d(j)$ *is a conservative extension of* Γ.
2. *If* $\exists(c,r)(j) \in \Gamma$*, then* $\Gamma + r(j,k) + c(k)$ *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 3. 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 4. *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.*

Proof. This is a routine argument, using Lemma 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 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 6 in Section 1.3. □

It might be worthwhile noting that the extensions produced by Lemma 4 add *infinitely many constants* to the language.

1.6 Completeness via canonical models

In this section, fix a language $\mathcal{L}$ and a maximal consistent Henkin $\mathcal{L}$-theory Γ. We construct a *canonical model* $\mathcal{M} = \mathcal{M}(\Gamma)$ as follows: $M = \mathbf{K}(\mathcal{L})$; $[\![p]\!](j)$ 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: there is no need of taking quotients as in the parallel argument for FOL completeness.

Lemma 5. *For all binary relation terms r, and all constants j and k,*

$$[\![r]\!](j,k) \textit{ in } \mathcal{M} \quad \textit{iff} \quad r(j,k) \in \Gamma \tag{7}$$

Proof. By induction on r. For r atomic, (7) is the definition of the semantics in our canonical model.

Assume that (7) holds for r; we show it for r^{-1}:

$$\Gamma \vdash r^{-1}(j,k) \quad \text{iff} \quad \Gamma \vdash r(k,j) \quad \text{iff} \quad r(k,j) \in \Gamma \quad \text{iff} \quad r^{-1}(j,k) \in \Gamma$$

The middle step uses the induction hypothesis.

Assume that (7) holds for r; then (7) holds for $\overline{r}$ using maximal consistency of Γ.

Assume that (7) holds for r and s. Then due to the $\wedge$ rules of the system we see that

$$[\![r \wedge s]\!](j,k) \quad \text{implies} \quad r(j,k) \text{ and } s(j,k) \text{ belong to } \Gamma$$

Conversely, if both $r(j,k)$ and $s(j,k)$ belong to Γ, then by the last rule above, $[\![r \wedge s]\!](j,k)$ belongs. ☐

Lemma 6. *For all set terms c, $[\![c]\!] = \{j : c(j) \in \Gamma\}$.*

Proof. By induction on c, using the Henkin property in the step for set terms of the form $\exists(c,s)(j)$. ☐

Lemma 7. $\mathcal{M} \models \Gamma$.

Proof. We check the sentence types in turn. Throughout the proof, we shall use Lemma 6 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 $\mathcal{M} \models c(j)$.

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

Theorem 8. *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 $\mathcal{L}$ be the language of Γ, Let $\mathcal{L}' \supseteq \mathcal{L}$ be an extension of $\mathcal{L}$, and let $\Gamma^* \supseteq \Gamma$ be a maximal consistent theory in $\mathcal{L}'$ with the Henkin property (see Lemma 4). Consider the canonical model $\mathcal{M}(\Gamma^*)$ as defined in this section. By Lemma 7, $\mathcal{M}(\Gamma^*) \models \Gamma^*$. Thus Γ^* is satisfiable, and hence so is Γ. ☐

1.7 The finite model property

Let Γ be a consistent finite theory in some language $\mathcal{L}$. 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 $\mathcal{L}^* \supseteq \mathcal{L}$. Then we may take the set of constant symbols of $\mathcal{L}^*$ 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 [5].

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 shall not go into the details on this, since they are essentially the same as in [6].

Theorem 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.

Conclusion

This paper has provided a logical system, defined in terms of a semantics, and then presented a proof system in the format of natural deduction, and finally proved the completeness theorem and the finite model property. The semantics of the language allows us to translate some natural language sentences into the languages faithfully. This particular fragment goes beyond other work in the area of natural logic in the sense of being complete and decidable, and also having the capability to represent inference. It is possible to go even further, and future papers will take this up.

Here is a bit more on this history of this topic. Set terms originate in McAllester and Givan [4], 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. The language of [4] did not have negation, and the paper shows that satisfiability problem is NP-complete. The language of [4] is included in the language $\mathcal{R}^*$ of Pratt-Hartmann and Moss [7]; the difference is that $\mathcal{R}^*$ has "a small amount" of negation. Yet more negation is found in the language $\mathcal{R}^{*\dagger}$ of [7]. This fragment has binary and unary atoms and negation. It is fairly close to being a sublanguage of the language $\mathcal{L}$ of this paper, but there are two small differences. First, here we have added constant symbols. In addition to making the system more expressive, the reason for adding constants is in order to present the Henkin-style completeness proof in Sections 1.5. A second difference 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. Finally, $\mathcal{R}^{*\dagger}$ lacks the conjunction and negation operations that this paper has. The first paper to tackle the fragment in this paper was Ivanov and Vakarelov [2].

The proof systems in [7] are syllogistic; there are no variables or what we have in this paper called general sentences. The proof systems in [2] also do not have variables, but there is a sense in which the syntax of that fragment is farther from natural language than ours. For example, the subject wide-scope reading of *Every man likes some animal* in their system is rendered as

$$\forall\exists(\mathsf{man},\mathsf{animal})[\mathsf{likes}]$$

in their system and

$$\forall(\mathsf{man},\exists(\mathsf{animal},\mathsf{likes}))$$

in ours.

The use of natural deduction proofs in connection with natural language is very old, going back to Fitch [1]. 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. For that matter, one could look open a classic like *Boolean Semantics for Natural Language* and aim for a perspicuous proof theory to match its sophisticated semantics.

References

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

[2] Nikolay Ivanov and Dimiter Vakarelov. A System of Relational Syllogistic Incorporating Full Boolean Reasoning. *Journal of Logic, Language, and Information*. vol. 21, no. 4, 433–459, 2012.

[3] Edward L. Keenan and Leonard M. Faltz. *Boolean Semantics for Natural Language*, Reidel, 1985.

[4] David McAllester and Robert Givan. Natural Language Syntax and First Order Inference. *Artificial Intelligence*, vol. 56, 1992.

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

[6] Lawrence S. Moss. Logics for Two Fragments Beyond the Syllogistic Boundary. In *Fields of Logic and Computation: Essays Dedicated to Yuri Gurevich on the Occasion of His 70th Birthday*. A. Blass, N. Dershowitz, and W. Reisig (eds.), LNCS, vol. 6300, Springer-Verlag, 538–563, 2010.

[7] Ian Pratt-Hartmann and Lawrence S. Moss. Logics for the Relational Syllogistic. *Review of Symbolic Logic*, Vol. 2, No. 4, 647–683, 2009.

Affliation

Lawrence S. Moss
Department of Mathematics
Indiana University
lsm@cs.indiana.edu