A Logic of Arbitrary and Indefinite Objects*

Stuart C. Shapiro
Department of Computer Science and Engineering
and Center for Cognitive Science
201 Bell Hall
University at Buffalo, The State University of New York
Buffalo, NY 14260-2000
shapiro@cse.buffalo.edu

Abstract

A Logic of Arbitrary and Indefinite Objects, $\mathcal{L}_A$, has been developed as the logic for knowledge representation and reasoning systems designed to support natural language understanding and generation, and commonsense reasoning. The motivations for the design of $\mathcal{L}_A$ are given, along with an informal introduction to the theory of arbitrary and indefinite objects, and to $\mathcal{L}_A$ itself. $\mathcal{L}_A$ is then formally defined by presenting its syntax, proof theory, and semantics, which are given via a translation scheme between $\mathcal{L}_A$ and the standard classical First-Order Predicate Logic. Soundness is proved. The completeness theorem for $\mathcal{L}_A$ is stated, and its proof is sketched. $\mathcal{L}_A$ is being implemented as the logic of SNePS 3, the latest member of the SNePS family of Knowledge Representation and Reasoning systems.

Introduction

Motivations

In this paper, we present a logic of arbitrary and indefinite objects, $\mathcal{L}_A$, which we have developed as the logic for knowledge representation and reasoning systems designed to support natural language understanding and generation, and commonsense reasoning. $\mathcal{L}_A$ is based on, but is different from, both the logic of arbitrary objects of Fine (1983; 1985a; 1985b) and the ANALOG logic of Ali and Shapiro (Ali 1993; 1994; Ali & Shapiro 1993). $\mathcal{L}_A$ is being implemented as the logic of SNePS 3 (Shapiro 2000a), the latest member of the SNePS family (Shapiro & Rapaport 1992; Shapiro 2000c; Shapiro & The SNePS Implementation Group 2002). SNePS is a propositional semantic network in which every well-formed subexpression is a node in a network whose labeled arcs indicate the argument position of the node at the head of the arc within the functional term which is the node at the tail end of the arc. In fact, since we view propositions as first-class members of the domain, every node, including nodes that denote propositions, is a term (Shapiro 1993). Nevertheless, we feel that the first step in developing $\mathcal{L}_A$ as a logic for SNePS 3 is to specify its syntax, semantics, and proof theory, and the best way to do that is with a "normal" linear syntax. We hope that others will find this version of $\mathcal{L}_A$ interesting and useful on its own, outside of its intended home in SNePS 3.

In the remainder of this section, we will give some motivations for the development of $\mathcal{L}_A$, and compare it with some previously developed logics. This will necessitate displaying some formulas of $\mathcal{L}_A$ before its syntax has been defined.

One motivation is the desire for a uniform syntax in the representation of differently quantified sentences. We would like sentences of the form *x is white* to uniformly be represented by formulas of the form *White(x)* for ease of retrieval from a knowledge base. For example, for the question *What is white?* it should be as easy to retrieve *every sheep* or *some sheep* as it is to retrieve *Dolly*. The standard classical first-order logic, henceforth referred to as $\mathcal{L}_S$, does not provide a uniform representation. A logic with restricted quantifiers ($\mathcal{L}_R$) does use a uniform representation for universally and existentially quantified sentences, but it is not the same representation as for ground sentences. This is illustrated in Table 1

A second motivation[1] is the desire to simplify the problem of translating between natural language sentences and sentences of a formal knowledge representation, by maintaining the locality of natural language phrases. In $\mathcal{L}_S$ and $\mathcal{L}_R$, natural language phrases are split into several pieces, as they are in Minimal Recursion Semantics (MRS) (Copestake *et al.* 1999). However, in $\mathcal{L}_A$ as well as in logics using "logical form" (Allen 1995, p. 228) or "complex-terms" (Jurafsky & Martin 2000, p. 555) ($\mathcal{L}_C$) they are left intact. For example, in Table 2, pieces of formulas derived from the phrase *a trunk* are underlined. Even though $\mathcal{L}_C$ maintains locality of phrases, it is an intermediate form, leaving quantifier scope undecided, which is appropriate for its intended use representing the semantics of isolated sentences, whereas $\mathcal{L}_A$ is a final form with quantifier scope indicated, which is appropriate for its intended use in a knowledge base representing the integrated beliefs of some natural-language understanding agent. MRS provides a representation both for ambiguous, but constrained, quantifier scopes, and for fully-scoped sentences. The MRS entry in Table 2 is a fully-scoped version. Of course, the importance of this motivating factor depends on the details of the parsing and generation techniques used.

[1] As pointed out by an anonymous reviewer, the first and second motivations are very similar to those of Richard Montague for his treatment of quantification (*see* (Thomason 1974)). The author is happy to acknowledge his intellectual debt to Montague.

English	$\mathcal{L}_S$	$\mathcal{L}_R$	$\mathcal{L}_A$
Dolly is White.	$White(Dolly)$	$White(Dolly)$	$White(Dolly)$
Every sheep is white.	$\forall x(Sheep(x) \Rightarrow White(x))$	$\forall x_{Sheep}\, White(x)$	$White(any\ x\ Sheep(x))$
Some sheep is white.	$\exists x(Sheep(x) \land White(x))$	$\exists x_{Sheep}\, White(x)$	$White(some\ x\ ()\ Sheep(x))$

Table 1: Some English sentences and their formalizations in several logics, illustrating uniform and non-uniform syntax for different quantifiers. See Table 3 for the abbreviations of the logics.

Language	Sentence
$\mathcal{L}_S$	$\forall x(Elephant(x) \Rightarrow \underline{\exists y(Trunk(y)} \land Has(x, \underline{y})))$
$\mathcal{L}_R$	$\forall x_{Elephant}\underline{\exists y_{Trunk}}Has(x, \underline{y})$
MRS	*h0*: every(*x*,*h1*,*h2*), *h1*: elephant(*x*), *h2*: <u>some(*y*,h3,h4)</u>, *h3*: <u>trunk(*y*)</u>, *h4*: has(x,<u>y</u>)
$\mathcal{L}_C$	$Has(\langle \forall x Elephant(x)\rangle, \underline{\langle \exists y Trunk(y)\rangle})$
$\mathcal{L}_A$	$Has(any\ x\ Elephant(x), \underline{some\ y\ (x)\ Trunk(y)})$

Table 2: *Every elephant has a trunk* in several logical languages. The underlined portion of each entry is the translation of *a trunk*.

A third motivation is the prospect of representing and reasoning about generalized quantifiers such as *many, most, few,* and *both* (Barwise & Cooper 1981), which is not done in $\mathcal{L}_S$ or $\mathcal{L}_R$, but can be done in MRS, $\mathcal{L}_C$, or $\mathcal{L}_A$[2]

A fourth motivation is the use of structure sharing, in which terms that occur in multiple places in one sentence, or, more importantly, in multiple sentences in one knowledge base, are stored only once (Referred to as the "Uniqueness Principle" in, for example, (Shapiro & Rapaport 1992, §3.4). See also (Sekar, Ramakrishnan, & Voronkov 2001) for the importance of such structure sharing in automated deduction). Since quantified terms in $\mathcal{L}_A$ are "conceptually complete" (Ali & Shapiro 1993); they can be shared without losing their identity. Such complete terms do not exist in $\mathcal{L}_S$ or $\mathcal{L}_R$, nor even in $\mathcal{L}_C$ where non-disambiguated scoping information is in the context surrounding the occurrences of complex-terms. It has been suggested[3] that MRS provides for structure sharing. However, MRS is defined so that "the MRS structure forms a tree ..., with a single root that dominates every other node, and no nodes having more than one parent" (Copestake *et al.* 1999, p. 8). Indeed, the MRS technique of implicit conjunction of nodes with the same handle would make it impossible to share only one conjunct of a group. Nevertheless, MRS may be seen as a notational variant of $\mathcal{L}_R$ plus generalized quantifiers, and only a few small syntactic changes are required to change MRS into a similar notational variant of $\mathcal{L}_A$. These changes are outlined in the section "MRS and $\mathcal{L}_A$" near the end of this paper.

A fifth motivation is the desire to use a simple form of subsumption reasoning (Woods 1991) among terms of the logic. Subsumption reasoning is at the heart of description logics (see, *e.g.,* (Woods & Schmolze 1992)). However, in description logics subsumption holds among concepts, which "are understood as unary predicates" (Brachman & Levesque 2004, p. 161). In $\mathcal{L}_A$, the subsumption relation may hold between quantified terms such as $(any\ x\ Elephant(x))$ and $(any\ x\ Albino(x) \land Elephant(x))$.

These motivations, and which of the logics satisfy them, are summarized in Table 3.

The differences between $\mathcal{L}_A$ and Fine's logic of arbitrary objects are discussed in the next section. ANALOG (Ali 1993; 1994; Ali & Shapiro 1993) could not distinguish between formulas structured like $\forall x \neg A(x)$ and $\neg \forall x A(x)$. See the section, "An Informal Introduction to $\mathcal{L}_A$", to see how this is corrected in $\mathcal{L}_A$.

Arbitrary Objects

A theory of arbitrary objects has been developed and defended by Fine (1983; 1985a; 1985b), "upon the basis of which a satisfactory explanation of the rule of universal generalization could be given" (Fine 1985b, p. vii). "An arbitrary object has those properties common to the individual objects in its range" (Fine 1985b, p. 5). The idea of using such arbitrary objects has also been tried by researchers in knowledge representation under the rubric "typical member," most notably by Fahlman [1979]. The general approach was analyzed in (Shapiro 1980), along with its problems and difficulties. The advantage of Fine's approach over the previous KR approaches is its firm logical foundation.

Fine distinguishes two classes of arbitrary objects, independent and dependent (Fine 1985b, p. 18). An independent arbitrary object is characterized only by its range of values. For example, a might be the arbitrary real number. Its range of values is all the individual real numbers. Dependent arbitrary objects are characterized by their range and the other arbitrary objects they are dependent on. For example, one dependent arbitrary object is a^3, which "assumes the value j when and only when a assumes the value $\sqrt[3]{a}$" (Fine 1985b, p. 17). The phrase "characterized (only) by" is made more precise by identity criteria. If a and b are independent arbitrary objects, "we say that $a = b$ iff their ranges are the same ... [if] a and b are dependent [arbitrary] objects... we shall say that $a = b$ iff two conditions are satisfied. The first is that they should depend upon the same arbitrary objects ... The second is that they should depend upon these objects

[2] However, the inclusion of generalized quantifiers in $\mathcal{L}_A$ will not be discussed in this paper.

[3] by an anonymous reviewer of a version of this paper

	$\mathcal{L}_S$	$\mathcal{L}_R$	MRS	$\mathcal{L}_C$	$\mathcal{L}_A$
Uniform syntax	No	No	No	Yes	Yes
Locality of phrases	No	No	No	Yes	Yes
Final form, including quantifier scoping	Yes	Yes	Yes	No	Yes
Potential for generalized quantifiers	No	No	Yes	Yes	Yes
Possibility of structure sharing	No	No	No	No	Yes
Subsumption relation holds on terms	No	No	No	No	Yes

Table 3: A summary of which logics satisfy which motivations. $\mathcal{L}_S$= Standard classical first-order logic; $\mathcal{L}_R$= Logic with restricted quantifiers; MRS = Minimal Recursion Semantics; $\mathcal{L}_C$= Logic with complex terms; $\mathcal{L}_A$= Logic of arbitrary and indefinite objects.

in the same way" (Fine 1985b, p. 18).

In some ways, independent arbitrary objects correspond to universally quantified variables in classical logics, and dependent arbitrary objects to existentially quantified variables: "Consider the sentence $\forall x \exists y Fxy$. It is true ... if, for a an unrestricted A-object [*i.e.* an arbitrary object whose range is all the members of the domain], we can find a totally defined A-object b dependent upon a alone for which Fab is true" (Fine 1985b, p. 46).

We, however, want to detach the notion of dependency from the notion of correspondence to existentially quantified variables in classical logics, so that we can have objects that correspond to existentially quantified variables in classical logics that yet are not dependent on any other arbitrary object. We want this so that, among other benefits, we can distinguish among the formalizations of the following sentences.

1. *Every sheep is white.*
2. *Some sheep is white.*
3. *Every sheep is not white.*
4. *Some sheep is not white.*
5. *It is not the case that every sheep is white.*
6. *It is not the case that some sheep is white.*

We will use "arbitrary object" for the terms that correspond to universally quantified variables in classical logics, whether or not they are dependent on other arbitrary objects. Arbitrary objects will be used to formalize sentences (1), (3), and (5) above. We will use "indefinite object" for the terms that correspond to existentially quantified variables in classical logics, whether or not they are dependent on any arbitrary objects. (An indefinite object will never be dependent on any indefinite objects.) Indefinite objects will be used to formalize sentences (2), (4), and (6).

Our arbitrary and indefinite objects will have restrictions to specify their ranges and the other arbitrary objects they depend on. In addition, each indefinite object will have a set (perhaps empty) of arbitrary objects on which it explicitly depends, called its "supporting variables". Similarly to Fine's criteria, we will consider two arbitrary objects to be identical if they have the same restrictions, and these involve the same other arbitrary objects. However, an indefinite object occurring in one sentence (This use of "sentence" will be defined below.) will never be considered to be identical to an indefinite object occurring in another sentence. This is to make invalid such arguments as

Lucy saw some dog.
Some dog is white.
Therefore, Lucy saw some white dog.

The next section contains an informal description of some unusual features of $\mathcal{L}_A$. The section after that contains a formal definition.

An Informal Introduction to $\mathcal{L}_A$

Arbitrary Objects, Binding, Capture, and Closure

We will express an **arbitrary object**, x, with the restriction $\mathcal{R}(x)$, as $(any\ x\ \mathcal{R}(x))$, so *every sheep is white* is formalized[4] as $White((any\ x\ Sheep(x)))$. We will allow the omission of redundant parentheses, so this becomes $White(any\ x\ Sheep(x))$.

Similarly, *every sheep is a mammal* is formalized as $Mammal(any\ x\ Sheep(x))$. In accord with the motivation of using $\mathcal{L}_A$ in an integrated knowledge base, if these two wffs are combined into $White(any\ x\ Sheep(x)) \wedge Mammal(any\ x\ Sheep(x))$ the two occurrences of $(any\ x\ Sheep(x))$ are literally two occurrences of the same variable, and denote the same arbitrary object. The arbitrary terms $(any\ x\ Sheep(x))$ and $(any\ x\ Raven(x))$, however, we deem to be **incompatible**. Therefore, $White(any\ x\ Sheep(x))$ and $Black(any\ x\ Raven(x))$ cannot be combined into a well-formed formula unless one of the variables is renamed. For example, $White(any\ x\ Sheep(x)) \wedge Black(any\ y\ Raven(y))$ is well-formed, and means *every sheep is white and every raven is black.*

When a wff with a **bound** variable, such as $White(any\ x\ Sheep(x))$, is combined with a wff with the same variable **free**, such as $Mammal(x)$, the binding of the bound variable **captures** the free one. So $White(any\ x\ Sheep(x)) \wedge Mammal(x)$ is the same wff as $White(any\ x\ Sheep(x)) \wedge Mammal(any\ x\ Sheep(x))$. If you want to avoid this, rename all occurrences of x in one of the wffs. The wff $White(any\ x\ Sheep(x)) \wedge Mammal(y)$

[4]We realize that a natural language sentence cannot be represented independently of context. Nevertheless, the practice of associating one isolated natural language sentence with one sentence of a logic is common when defining a logic, and we shall follow that practice in this paper.

contains a bound occurrence of the variable x and a free occurrence of the variable y.

Since bound occurrences of a variable capture free occurrences of the same variable, and all occurrences of a variable in a well-formed formula are compatible, all but one occurrence of $(any\ x\ \mathcal{R}(x))$ may be abbreviated as x. Thus, $White(any\ x\ Sheep(x)) \wedge Mammal(x)$ is officially an abbreviation of $White(any\ x\ Sheep(x)) \wedge Mammal(any\ x\ Sheep(x))$.

Since the bindings of bound variables take wide scope over the wff, $\neg White(any\ x\ Sheep(x))$ means that *every sheep is not white*. To keep variable bindings from rising too far, we use the **closure** operator $\lfloor_x \ldots \rfloor$, which contains the scope of the variable x. Thus $\neg\lfloor_x White(any\ x\ Sheep(x))\rfloor$ means *it is not the case that every sheep is white.*

The binding of a closed, bound variable does not capture free occurrences of the same variable outside its closure. So the wff[5] $Odd(any\ x\ Number(x)) \vee Even(x)$ which is an abbreviation of

$$Odd(any\ x\ Number(x)) \vee Even(any\ x\ Number(x))$$

means *every number is odd or even*, while

$$\begin{array}{l}\lfloor_x Odd(any\ x\ Number(x))\rfloor \\ \vee\lfloor_x Even(any\ x\ Number(x))\rfloor\end{array}$$

means *either every number is odd or every number is even*.

Indefinite Objects

We will express an **indefinite object**, x, dependent on the arbitrary objects $y_1, \ldots, y_n$, with the restriction $\mathcal{R}(x)$, as $(some\ x\ (y_1, \ldots, y_n)\ \mathcal{R}(x))$, where $n \geq 0$. Issues of abbreviations, binding, capture, and closure apply to indefinite variables in the same way as they do to arbitrary variables. So a few examples should suffice for introducing indefinite variables.

$Has(any\ x\ Elephant(x), some\ y\ (x)\ Trunk(y))$ means *every elephant has a (its own) trunk.* Note that the occurrence of x in the list of variables that y depends on is an abbreviated form of $(any\ x\ Elephant(x))$, and is the same variable as is the first argument of *Has*.

$\neg White(some\ x\ ()\ Sheep(x))$ means *some sheep is not white.* $\neg\lfloor_y White(some\ y\ ()\ Sheep(y))\rfloor$ means *it is not the case that some sheep is white.*

$(any\ x\ Number(x)) < (some\ y\ (x)\ Number(y))$ means *every number has some number bigger than it*, or, in $\mathcal{L}_S, \forall x \exists y (x < y)$.

$(any\ x\ Number(x)) < (some\ y\ ()\ Number(y))$, where y has no supporting variables, means *some number is bigger than every number*, or, in $\mathcal{L}_S, \exists y \forall x (x < y)$.

The list of arbitrary objects an indefinite object depends on is directly related to the arguments of Skolem functions, and is similar to the dependency links used in several previous propositional semantic network formalisms (Ali 1993; 1994; Ali & Shapiro 1993; Kay 1973; Schubert 1976; Schubert, Goebel, & Cercone 1979)

[5] These examples are based on a discussion in (Fine 1985b, p. 9ff).

Tricky Sentences

There are several classes of sentences that have been discussed in the literature as being particularly difficult to represent in $\mathcal{L}_S$. The most natural apparent representation of the donkey sentence, *Every farmer who owns a donkey beats it* (Geach 1962) in $\mathcal{L}_S$ is

$$\begin{array}{l}\forall x[(Farmer(x) \wedge \exists y(Donkey(y) \wedge Owns(x,y))) \\ \quad \Rightarrow Beats(x,y)]\end{array}$$

However, the occurrence of y in the consequent is outside the scope of its quantifier. The only logically correct representation of the donkey sentence in $\mathcal{L}_S$ is

$$\begin{array}{l}\forall x \forall y[(Farmer(x) \wedge Donkey(y) \wedge Owns(x,y)) \\ \quad\quad \Rightarrow Beats(x,y)]\end{array}$$

but this has been objected to because it quantifies over all farmers and all donkeys, rather than just donkey-owning farmers. The variable capturing feature of $\mathcal{L}_A$, which is associated with its intended structure-sharing representation in SNePS 3 captures the sense of the donkey sentence correctly as

$$\begin{array}{l}Beats(any\ x\ Farmer(x) \\ \quad\quad\quad\quad \wedge\ Owns(x, some\ y\ (x)\ Donkey(y)), \\ \quad\ y)\end{array}$$

Another tricky class of sentences are those that require branching quantifiers, for example *Some relative of each villager and some relative of each townsman hate each other.* (*See* (McCawley 1981, p. 449).) An attempt to represent this sentence in $\mathcal{L}_S$ is

$$\begin{array}{l}\forall v \exists x \forall w \exists y[(Villager(v) \wedge Relative(x,v) \\ \quad\quad\quad \wedge Townsman(w) \wedge Relative(y,w)) \\ \quad\quad\quad \Rightarrow Hates(x,y)]\end{array}$$

However, there is no way to express this without making at least one of x or y dependent on both v and w, which does not seem warranted. In $\mathcal{L}_A$, this sentence can be represented correctly as

$$\begin{array}{l}Hate(some\ x\ (any\ v\ Villager(v))\ Relative(x,v), \\ \quad\quad some\ y\ (any\ w\ Townsman(w))\ Relative(y,w))\end{array}$$

The tricky sentences discussed in this subsection are represented the same way in ANALOG (Ali 1993; 1994; Ali & Shapiro 1993) as they are in $\mathcal{L}_A$, but ANALOG does not have the closure operator, and so cannot distinguish *some sheep is not white* from *it is not the case that some sheep is white* nor *every number is odd or even* from *either every number is odd or every number is even*.

Nested Beliefs (An Aside)

There are several techniques in the KR literature for representing nested beliefs, including sentences (***e.g.**, see* (Davis 1990)) and reified propositions (*e.g, see* (McCarthy 1979; Shapiro 1993) and (Copestake *et al.* 1999) for an example from the NL processing literature). We prefer the latter, which requires a reinterpretation of the representation logic so that predicates, logical connectives, and quantifiers are proposition-forming rather than sentence-forming. We do not do that reinterpretation in the bulk of this paper in order to present $\mathcal{L}_A$ as a variety of a classical logic. However, we do use that reinterpretation in SNePS (*e.g.,* (Shapiro 1979; Shapiro & Rapaport 1991; 1992; Rapaport, Shapiro, & Wiebe 1997)), and representing nested beliefs was one of our motivations for the closure operator. In this version of $\mathcal{L}_A$, $Believes(Mike, Spy(some\ x\ ()\ Person(x)))$ means *there is someone whom Mike believes is a spy*, $Believes(Mike, \neg Spy(some\ x\ ()\ Person(x)))$ means *there is someone whom Mike believes is not a spy*, $Believes(Mike, \lfloor_x Spy(some\ x\ ()\ Person(x))\rfloor)$ means *Mike believes that there is someone who is a spy*, $Believes(Mike, \lfloor_x \neg Spy(some\ x\ ()\ Person(x))\rfloor)$ means *Mike believes that there is someone who is not a spy*, and $Believes(Mike, \neg\lfloor_x Spy(some\ x\ ()\ Person(x))\rfloor)$ means *Mike believes that it is not the case that there is someone who is a spy.*

$\mathcal{L}_A$, The Logic of Arbitrary and Indefinite Objects

Syntax of $\mathcal{L}_A$

Atomic symbols The following sets of atomic symbols should be disjoint:

Individual constants Such as $John$, $Clyde$, and $savanna$.

Variables Such as x, y, and z, possibly with subscripts, such as x_1, and x_2.

Determiners any and $some$.

Function symbols Such as $sonOf$ and $successor$, each with some arity, which may be shown as a superscript, such as $sonOf^2$ and $successor^1$.

Predicate symbols Such as $Elephant$ and On, each with some arity, which may be shown as a superscript, such as $Elephant^1$ and On^2.

Quantified terms

1. If x is a variable and $\mathcal{A}(x)$ is a formula containing one or more occurrences of x open and free, then $(any\ x)$ and $(any\ x\ \mathcal{A}(x))$ are arbitrary terms. x is called the variable of those arbitrary terms. any is called their determiner. $\mathcal{A}(x)$ is called the restriction of the arbitrary term $(any\ x\ \mathcal{A}(x))$, and of the variable x.

   All open occurrences of the variable of an arbitrary term are bound in the arbitrary term, all closed occurrences remain closed, and all occurrences of other variables that are free (bound, open, closed) in $\mathcal{A}(x)$ are free (bound, open, closed) in $(any\ x\ \mathcal{A}(x))$.

2. If x is a variable, $q_1, \ldots, q_n$ are variables or arbitrary terms, and $\mathcal{A}(x)$ is a formula containing one or more occurrences of x open and free, then $(some\ x\ ())$, $(some\ x\ ()\ \mathcal{A}(x))$, $(some\ x\ (q_1, \ldots, q_n))$, and $(some\ x\ (q_1, \ldots, q_n)\ \mathcal{A}(x))$ are indefinite terms. x is called the variable of those indefinite terms. $some$ is called their determiner. $\mathcal{A}(x)$, where included, is called the restriction of the indefinite term and of the variable, and $q_1, \ldots, q_n$, where included, are called the supporting variables of the indefinite term, and of the variable.

   All open occurrences of the variable in an indefinite term are bound, All closed occurrences remain closed, and all occurrences of other variables that are free (bound, open, closed) in the supporting variables or the restriction of an indefinite term are free (bound, open, closed) in the indefinite term.

3. arbitrary terms and indefinite terms are quantified terms, and nothing else is.

4. The quantified terms in a set of quantified terms are *compatible* if either

   (a) No two terms have the same variable, or

   (b) whenever two terms have the same variable, then: i) they have the same determiner; ii) they have the same restriction; and iii) if they are indefinite terms they also have the same supporting variables.

   Otherwise, they are called *incompatible*.

   We will say that the set, α, of quantified terms is compatible with the set, β, of quantified terms if and only if the quantified terms in $\alpha \cup \beta$ are compatible.

Terms

1. Every individual constant is a term.

2. Every variable is a term. For every variable, x, the occurrence of x in the term x is free and open.

3. Every quantified term is a term.

4. If f^n is a function symbol of arity n, $t_1, \ldots, t_n$ are terms, and all open quantified terms in $t_1, \ldots, t_n$ are compatible, then $f^n(t_1, \ldots, t_n)$ is a term.

   Any open variables that are bound in any $t_i, 1 \leq i \leq n$, are open and bound in $f^n(t_1, \ldots, t_n)$, any other variables that are free in any $t_i, 1 \leq i \leq n$ are free in $f^n(t_1, \ldots, t_n)$, and all occurrences of variables that are open (closed) in $t_i, 1 \leq i \leq n$, remain open (closed) in $f^n(t_1, \ldots, t_n)$.

5. Nothing else is a term.

Atomic formulas If P^n is a predicate symbol of arity n, $t_1, \ldots, t_n$ are terms, and all open quantified terms in $t_1, \ldots, t_n$ are compatible, then $P^n(t_1, \ldots, t_n)$ is an atomic formula.

Any open variables that are bound in any $t_i, 1 \leq i \leq n$, are open and bound in $P^n(t_1, \ldots, t_n)$, any other variables that are free in any $t_i, 1 \leq i \leq n$ are free in $P^n(t_1, \ldots, t_n)$, and all occurrences of variables that are open (closed) in $t_i, 1 \leq i \leq n$, remain open (closed) in $P^n(t_1, \ldots, t_n)$.

Well-formed formulas (wffs)

1. Every atomic formula is a well-formed formula.

2. If $\mathcal{A}(x)$ is a wff containing open bound occurrences of the variable x, then $\lfloor_x \mathcal{A}(x)\rfloor$ is a wff, called the closure of $\mathcal{A}(x)$ with respect to x.

   All open occurrences of the variable x in $\mathcal{A}(x)$ are closed in $\lfloor_x \mathcal{A}(x)\rfloor$. Other open (closed, bound, free) (occurrences of) variables in $\mathcal{A}(x)$ remain so in $\lfloor_x \mathcal{A}(x)\rfloor$. Note that every closed occurrence of a variable is of a bound variable.

3. If $\mathcal{A}$ is a wff, then $(\neg\mathcal{A})$ is a wff.

   Any (occurrences of) variables that are free (bound, open, closed) in $\mathcal{A}$ are free (bound, open, closed) in $(\neg\mathcal{A})$.

4. If $\mathcal{A}$ and $\mathcal{B}$ are wffs, and the set of open quantified terms in $\mathcal{A}$ is compatible with the set of open quantified terms in $\mathcal{B}$, then $(\mathcal{A} \circ \mathcal{B})$ is a wff, where $\circ$ is one of $\wedge$, $\vee$, $\Rightarrow$, or $\Leftrightarrow$.

   Any open variables that are bound in $\mathcal{A}$ or $\mathcal{B}$ are open and bound in $(\mathcal{A} \circ \mathcal{B})$, any other variables that are free in $\mathcal{A}$ or $\mathcal{B}$ are free in $(\mathcal{A} \circ \mathcal{B})$, and all occurrences of variables that are open (closed) in $\mathcal{A}$ or $\mathcal{B}$ remain so in $(\mathcal{A} \circ \mathcal{B})$.

5. Nothing else is a well-formed formula.

Sentences

1. Any wff all of whose variables are bound is a sentence.
2. A sentence containing an open occurrence of a variable is a generic sentence.
3. A sentence containing no open occurrence of a variable is a non-generic sentence.

Theorem 1 *If σ is the set of open quantified terms of any sentence of $\mathcal{L}_A$, or of any well-formed subformula of a sentence of $\mathcal{L}_A$, or of any term occurring in a sentence of $\mathcal{L}_A$, then the quantified terms in σ are compatible.*

Proof: By the definition of a sentence, all the variables in a sentence are bound. The rest of the proof is by structural induction on the formation of terms, atomic formulas, and wffs.

The significance of Theorem 1 is that all the open quantified terms in a sentence or in a subformula of a sentence that share a variable are occurrences of the same term. The scope of the variable of a quantified term in a sentence in which the variable is open is the entire sentence, but the scope of x in a closed formula $\lfloor_x \mathcal{A}(x)\rfloor$ is limited to that closed formula. This observation will be reinforced by the translation rules in the section, "Translation between $\mathcal{L}_A$ and $\mathcal{L}_S$".

Ground Expressions Any term, wff, or sentence that contains no variables is called a ground term, wff, or sentence, respectively.

Abbreviations

1. In any wff containing a set of bound quantified terms with the same variable, all but one of the terms may be abbreviated as just the variable.

2. Parentheses may be omitted wherever they are not needed. In particular, the outer parentheses of a quantified term may be omitted whenever no confusion results.

3. Any quantified term of the form $(any\ x\ \mathcal{A}_1(x) \wedge \cdots \wedge \mathcal{A}_n(x))$ or $(some\ x\ \phi\ \mathcal{A}_1(x) \wedge \cdots \wedge \mathcal{A}_n(x))$ may be abbreviated as $(any\ x\ \mathcal{A}_1(x)\ \ldots\ \mathcal{A}_n(x))$ or $(some\ x\ \phi\ \mathcal{A}_1(x)\ \ldots\ \mathcal{A}_n(x))$, respectively.

Examples See the section, "An Informal Introduction to $\mathcal{L}_A$" for examples of sentences of $\mathcal{L}_A$.

Translation between $\mathcal{L}_A$ and $\mathcal{L}_S$

To show that the language of $\mathcal{L}_A$ includes $\mathcal{L}_S$, the standard, classical first-order logic, and to show the correctness of the rules presented in the section, "Proof Theory of $\mathcal{L}_A$", we provide translations between the wffs of $\mathcal{L}_A$ and those of $\mathcal{L}_S$.

Translation from $\mathcal{L}_A$ to $\mathcal{L}_S$ The translation from $\mathcal{L}_A$ to $\mathcal{L}_S$ is to be done top-down starting with sentences. No higher numbered rule is to be applied unless no lower numbered rule can apply.

1. If no bound quantified term occurs in $\mathcal{A}$, then $tr_{\mathcal{AS}}(\mathcal{A}) = \mathcal{A}$.

2. $tr_{\mathcal{AS}}(\lfloor_x \mathcal{A}\rfloor) = tr_{\mathcal{AS}}(\mathcal{A})$, for any variable x.

3. If no open bound quantified term occurs in $\neg\mathcal{A}$, then $tr_{\mathcal{AS}}(\neg\mathcal{A}) = \neg tr_{\mathcal{AS}}(\mathcal{A})$.

4. If no open bound quantified term occurs in both $\mathcal{A}(x)$ and $\mathcal{B}(y)$, then $tr(\mathcal{A}(x) \circ \mathcal{B}(y)) = tr(\mathcal{A}(x)) \circ tr(\mathcal{B}(y))$, where $\circ$ is one of $\wedge$, $\vee$, $\Rightarrow$, or $\Leftrightarrow$.

5. $$\begin{aligned} &tr_{\mathcal{AS}}(\mathcal{A}((some\ x\ (q_1, \ldots, (any\ q_i\ \mathcal{C}(q_i)), \ldots, q_n) \\ &\qquad\qquad \mathcal{B}(x, q_1, \ldots, q_n)))) \\ &= \forall q_i\, tr_{\mathcal{AS}}(\mathcal{C}(q_i) \\ &\qquad\qquad \Rightarrow \mathcal{A}((some\ x\ (q_1, \ldots, q_{i-1}, q_{i+1}, \ldots, q_n) \\ &\qquad\qquad\qquad \mathcal{B}(x, q_1, \ldots, q_n)))), \end{aligned}$$

   whether or not $\mathcal{B}(x, q_1, \ldots, q_n)$ actually occurs, and where the right hand side is obtained by replacing all open occurrences of $(any\ q_i\ \mathcal{C}(q_i))$ in $\mathcal{A}$ by q_i.

6. $tr_{\mathcal{AS}}(\mathcal{A}((some\ x\ ()))) = \exists x\ tr_{\mathcal{AS}}(\mathcal{A}(x))$, where the right hand side is obtained by replacing all open occurrences of $(some\ x\ ())$ in $\mathcal{A}$ by x.

7. $tr_{\mathcal{AS}}(\mathcal{A}((some\ x\ ()\ \mathcal{B}(x)))) = \exists x\ tr_{\mathcal{AS}}(\mathcal{B}(x) \wedge \mathcal{A}(x))$, where the right hand side is obtained by replacing all open occurrences of $(some\ x\ ()\ \mathcal{B}(x))$ in $\mathcal{A}$ by x.

8. $tr_{\mathcal{AS}}(\mathcal{A}((any\ x))) = \forall x\ tr_{\mathcal{AS}}(\mathcal{A}(x))$, where the right hand side is obtained by replacing all open occurrences of $(any\ x)$ in $\mathcal{A}$ by x.

9. $tr_{\mathcal{AS}}(\mathcal{A}((any\ x\ \mathcal{B}(x)))) = \forall x\ tr_{\mathcal{AS}}(\mathcal{B}(x) \Rightarrow \mathcal{A}(x))$, where the right hand side is obtained by replacing all open occurrences of $(any\ x\ \mathcal{B}(x))$ in $\mathcal{A}$ by x.

Example Table 4 shows $tr_{\mathcal{AS}}$ applied to the branching quantifier sentence in the section, "Tricky Sentences", assuming that each rule applies to the left-most place possible.

Translation from $\mathcal{L}_S$ to $\mathcal{L}_A$ To translate from a wff, $\mathcal{F}$, of $\mathcal{L}_{\mathcal{S}}$ to one, $tr_{\mathcal{SA}}(\mathcal{F})$, of $\mathcal{L}_{\mathcal{A}}$, follow the following steps, in order.

1. Rename the variables apart so that no two quantifiers govern the same variable.

2. Change every subformula $\mathcal{F}$ of the form $\exists x\mathcal{A}(x)$ to $\exists x\mathcal{A}((some\ x\ (v_1, \ldots, v_n)))$, where $(v_1, \ldots, v_n)$ is a list of all the universally quantified variables within whose scope $\mathcal{F}$ is.

Rules	Result
	$tr_{\mathcal{AS}}(\mathit{Hate}(\mathit{some}\ x\ (\mathit{any}\ v\ \mathit{Villager}(v))\ \mathit{Relative}(x,v), \mathit{some}\ y\ (\mathit{any}\ w\ \mathit{Townsman}(w))\ \mathit{Relative}(y,w)))$
5	$\forall v\ tr_{\mathcal{AS}}(\mathit{Villager}(v) \Rightarrow \mathit{Hate}(\mathit{some}\ x\ ()\ \mathit{Relative}(x,v), \mathit{some}\ y\ (\mathit{any}\ w\ \mathit{Townsman}(w))\ \mathit{Relative}(y,w)))$
4, 1	$\forall v(\mathit{Villager}(v) \Rightarrow tr_{\mathcal{AS}}(\mathit{Hate}(\mathit{some}\ x\ ()\ \mathit{Relative}(x,v), \mathit{some}\ y\ (\mathit{any}\ w\ \mathit{Townsman}(w))\ \mathit{Relative}(y,w))))$
5, 4, 1	$\forall v(\mathit{Villager}(v) \Rightarrow \forall w(\mathit{Townsman}(w) \Rightarrow tr_{\mathcal{AS}}(\mathit{Hate}(\mathit{some}\ x\ ()\ \mathit{Relative}(x,v), \mathit{some}\ y\ ()\ \mathit{Relative}(y,w)))))$
7, 4, 1	$\forall v(\mathit{Villager}(v) \Rightarrow \forall w(\mathit{Townsman}(w) \Rightarrow \exists x(\mathit{Relative}(x,v) \wedge tr_{\mathcal{AS}}(\mathit{Hate}(x, \mathit{some}\ y\ ()\ \mathit{Relative}(y,w))))))$
7, 4, 1, 1	$\forall v(\mathit{Villager}(v) \Rightarrow \forall w(\mathit{Townsman}(w) \Rightarrow \exists x(\mathit{Relative}(x,v) \wedge \exists y(\mathit{Relative}(y,w) \wedge \mathit{Hate}(x,y)))))$

Table 4: Translation of the branching quantifier sentence from the section, "Tricky Sentences", showing the order of $tr_{\mathcal{AS}}$ rule application, assuming that each rule applies to the left-most place possible.

3. Change every subformula of the form

   $$\exists x(\mathcal{A}((\mathit{some}\ x\ (v_1,\ldots,v_n))) \wedge \mathcal{B}((\mathit{some}\ x\ (v_1,\ldots,v_n))))$$

   to $\exists x \mathcal{B}((\mathit{some}\ x\ (v_1,\ldots,v_n)\ \mathcal{A}(x)))$.
4. Change every subformula of the form $\exists x \mathcal{A}$ to $\lfloor_x \mathcal{A}\rfloor$.
5. Change every subformula of the form $\forall x \mathcal{A}(x)$ to $\forall x \mathcal{A}((\mathit{any}\ x))$.
6. Change every subformula of the form $\forall x(\mathcal{A}((\mathit{any}\ x)) \Rightarrow \mathcal{B}((\mathit{any}\ x)))$ to $\forall x \mathcal{B}((\mathit{any}\ x\ \mathcal{A}(x)))$.
7. Change every subformula of the form $\forall x \mathcal{A}$ to $\lfloor_x \mathcal{A}\rfloor$.

$\mathcal{L}_\mathcal{A}$ More Expressive Than $\mathcal{L}_\mathcal{S}$ $\mathcal{L}_\mathcal{A}$ is more expressive than $\mathcal{L}_\mathcal{S}$ in the sense that $tr_{\mathcal{AS}}$ translates several formulas of $\mathcal{L}_\mathcal{A}$ into the same formula of $\mathcal{L}_\mathcal{S}$, and there are wffs of $\mathcal{L}_\mathcal{A}$ that are not in the range of $tr_{\mathcal{SA}}$. For example, the result of the translation shown in Table 4 is also the translation of

$$\mathit{Hate}(\mathit{some}\ x\ (\mathit{any}\ v\ \mathit{Villager}(v), w)\ \mathit{Relative}(x,v), \mathit{some}\ y\ (v, \mathit{any}\ w\ \mathit{Townsman}(w))\ \mathit{Relative}(y,w))$$

in which both x and y depend on both v and w. This formula of $\mathcal{L}_\mathcal{A}$, is the value of $tr_{\mathcal{SA}}$ applied to the last line of Table 4. On the other hand, there is no formula of $\mathcal{L}_\mathcal{S}$ which $tr_{\mathcal{SA}}$ translates into the first line of Table 4.

Semantics of $\mathcal{L}_\mathcal{A}$

Specifying an independent semantics for $\mathcal{L}_\mathcal{A}$ is yet to be done. For now, we will take the meaning of any wff $\mathcal{A}$ of $\mathcal{L}_\mathcal{A}$ to be the meaning in $\mathcal{L}_\mathcal{S}$ of $tr_{\mathcal{AS}}(\mathcal{A})$.

Proof Theory of $\mathcal{L}_A$

In this section, we present the rules of inference of $\mathcal{L}_A$.[6]

(axiom) $\Gamma, \mathcal{A} \vdash_{\mathcal{L}_A} \mathcal{A}$.

(hyp) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}$, then $\Gamma, \mathcal{B} \vdash_{\mathcal{L}_A} \mathcal{A}$.

(cut) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}$, and $\Gamma, \mathcal{A} \vdash_{\mathcal{L}_A} \mathcal{B}$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}$.

(closureI) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}$, then $\Gamma \vdash_{\mathcal{L}_A} \lfloor_x \mathcal{A}\rfloor$, for any variable x.

(closureE) If $\Gamma \vdash_{\mathcal{L}_A} \lfloor_x \mathcal{A}\rfloor$, for some variable x, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}$.

Notice that $\mathcal{L}_A$ does not have the subformula property that if $\mathcal{A} \Leftrightarrow \mathcal{B}$, and $\mathcal{C}$ contains $\mathcal{A}$ as a subformula, and $\mathcal{D}$ is exactly like $\mathcal{C}$ except for having $\mathcal{B}$ where $\mathcal{C}$ has $\mathcal{A}$, then $\mathcal{C} \Leftrightarrow \mathcal{D}$, at least not if $\mathcal{A}$ is a closure of $\mathcal{B}$ or *vice versa.*

($\neg$**I**) If $\Gamma, A \vdash_{\mathcal{L}_A} B$ and $\Gamma, A \vdash_{\mathcal{L}_A} \neg B$ then $\Gamma, \vdash_{\mathcal{L}_A} \neg A$.

($\neg$**E**) If $\Gamma \vdash_{\mathcal{L}_A} \neg\neg A$ then $\Gamma \vdash_{\mathcal{L}_A} A$.

($\wedge$**I**) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}$ and $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}$, and if $\mathcal{A}$ and $\mathcal{B}$ have no incompatible open arbitrary terms, and if there is no open indefinite term in $\mathcal{B}$ with the same variable as an open indefinite term in $\mathcal{A}$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A} \wedge \mathcal{B}$. To make this rule applicable, all the occurrences of some variable in $\mathcal{A}$ or $\mathcal{B}$ may be renamed to a variable that occurs in neither $\mathcal{A}$ nor in $\mathcal{B}$.

An example of a situation the conditions on $\mathcal{A}$ and $\mathcal{B}$ are designed to prevent is the putative inference

from $\Gamma \vdash_{\mathcal{L}_A} \mathit{Saw}(\mathit{Lucy}, \mathit{some}\ x\ ()\ \mathit{Dog}(x))$
and $\Gamma \vdash_{\mathcal{L}_A} \mathit{White}(\mathit{some}\ x\ ()\ \mathit{Dog}(x))$
to $\Gamma \vdash_{\mathcal{L}_A} \mathit{Saw}(\mathit{Lucy}, \mathit{some}\ x\ ()\ \mathit{Dog}(x)) \wedge \mathit{White}(\mathit{some}\ x\ ()\ \mathit{Dog}(x))$.

(See the section, "Arbitrary Objects", above.)

($\wedge$**E**) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A} \wedge \mathcal{B}$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}$ and $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}$.

($\vee$**I**) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}$, and if all open quantified terms in $\mathcal{A}$ and $\mathcal{B}$ are compatible, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A} \vee \mathcal{B}$ and $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B} \vee \mathcal{A}$.

($\vee$**E**) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A} \vee \mathcal{B}$ and $\Gamma, \mathcal{A} \vdash_{\mathcal{L}_A} \mathcal{C}$ and $\Gamma, \mathcal{B} \vdash_{\mathcal{L}_A} \mathcal{C}$,then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{C}$.

($\Rightarrow$ **I**) If $\Gamma, \mathcal{A} \vdash_{\mathcal{L}_A} \mathcal{B}$, and if $\mathcal{A}$ and $\mathcal{B}$ have no incompatible open arbitrary terms, and if there is no open indefinite term in $\mathcal{B}$ with the same variable as an open indefinite term in $\mathcal{A}$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A} \Rightarrow \mathcal{B}$

($\Rightarrow$ **E**) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}$, and $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A} \Rightarrow \mathcal{B}$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}$.

($\Leftrightarrow$ **I**) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A} \Rightarrow \mathcal{B}$, and $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B} \Rightarrow \mathcal{A}$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A} \Leftrightarrow \mathcal{B}$.

($\Leftrightarrow$ **E**) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A} \Leftrightarrow \mathcal{B}$, and $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}$, and if $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A} \Leftrightarrow \mathcal{B}$, and $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}$.

(**anyI$_1$**) If $\Gamma, \mathcal{A}(a) \vdash_{\mathcal{L}_A} \mathcal{B}(a)$ and a is a term that does not occur in Γ, and x is a variable that does not occur open in $\mathcal{A}(a)$ or in $\mathcal{B}(a)$, nor does a occur within $\mathcal{A}(a)$ or $\mathcal{B}(a)$ inside the scope of $\lfloor_x \ldots\rfloor$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(\mathit{any}\ x\ \mathcal{A}(x))$, where $\mathcal{B}(\mathit{any}\ x\ \mathcal{A}(x))$ and $\mathcal{A}(x)$ are derived from $\mathcal{B}(a)$ and $\mathcal{A}(a)$, respectively, by replacing all occurrences of a with $(\mathit{any}\ x\ \mathcal{A}(x))$, and by replacing all open occurrences of indefinite terms $(\mathit{some}\ y\ (q_1,\ldots,q_n)\ \mathcal{C}(y))$ with $(\mathit{some}\ y\ (x, q_1,\ldots,q_n)\ \mathcal{C}(y))$.

[6] Although we prefer using a paraconsistent logic in our KRR system (Shapiro 2000c), we will present this logic as a classical logic to avoid confusing two independent issues.

($\mathbf{anyI_2}$) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(a)$ and a is a term that does not occur in Γ, and x is a variable that does not occur open in $\mathcal{B}(a)$, nor does a occur within $\mathcal{B}(a)$ inside the scope of $\lfloor_x \ldots \rfloor$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(any\ x)$, where $\mathcal{B}(any\ x)$ is derived from $\mathcal{B}(a)$, by replacing all occurrences of a with $(any\ x)$, and by replacing all open occurrences of indefinite terms $(some\ y\ (q_1, \ldots, q_n)\ \mathcal{C}(y))$ with $(some\ y\ (x, q_1, \ldots, q_n)\ \mathcal{C}(y))$.

(anyE$_1$**)** If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}(a)$ and $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(any\ x\ \mathcal{A}(x))$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(a)$, for any term a, where $\mathcal{A}(a)$ and $\mathcal{B}(a)$ are derived from $\mathcal{A}(x)$ and $\mathcal{B}(x)$, respectively, by replacing every open occurrence of $(any\ x\ \mathcal{A}(x))$ by a, and every open occurrence of $(some\ y\ (q1, \ldots, q_i, x, q_{i+1}, \ldots, q_n)\ \mathcal{C}(y))$ by $(some\ y\ (q1, \ldots, q_i, q_{i+1}, \ldots q_n)\ \mathcal{C}(y))$.

($\mathbf{anyE_2}$) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(any\ x)$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(a)$, for any term a, where $\mathcal{B}(a)$ is derived from $\mathcal{B}(any\ x)$, by replacing every open occurrence of $(any\ x)$ by a, and every open occurrence of $(some\ y\ (q1, \ldots, q_i, x, q_{i+1}, \ldots, q_n)\ \mathcal{C}(y))$ by $(some\ y\ (q1, \ldots, q_i, q_{i+1}, \ldots q_n)\ \mathcal{C}(y))$.

($\mathbf{someI_1}$) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}(a) \wedge \mathcal{B}(a)$, for any term a, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(some\ x\ ()\ \mathcal{A}(x))$.

($\mathbf{someI_2}$) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(a)$, for any term a, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(some\ x\ ())$.

($\mathbf{someE_1}$) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(some\ x\ ()\ \mathcal{A}(x))$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{A}(a) \wedge \mathcal{B}(a)$, for any ground term a that does not occur in Γ nor in $\mathcal{B}(some\ x\ ()\ \mathcal{A}(x))$.

($\mathbf{someE_2}$) If $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(some\ x\ ())$, then $\Gamma \vdash_{\mathcal{L}_A} \mathcal{B}(a)$, for any ground term a that does not occur in Γ nor in $\mathcal{B}(some\ x\ ())$.

Examples Table 5 shows a proof that *Some child of every woman all of whose sons are doctors is busy* follows from *Some child of every person all of whose sons are professionals is busy*, *Every woman is a person*, and *Every doctor is a professional*.[7] Notice, in particular, the use of $(any\ z\ sonOf(z, a))$ as a term in lines 4–6. In a comparable $\mathcal{L}_S$ derivation, this one use of $\mathbf{anyE_1}$ would require two uses of $\forall\mathbf{E}$, two uses of $\Rightarrow \mathbf{E}$, one use of $\Rightarrow \mathbf{I}$, and one use of $\forall\mathbf{I}$.

There is a 24-step proof[8] of $\vdash_{\mathcal{L}_A} \neg\lfloor_x \neg A(any\ x\ B(x))\rfloor \Leftrightarrow A(some\ x\ ()\ B(x))$, but space limits preclude presenting it here.

Soundness and Completeness of $\mathcal{L}_A$

Since we are taking the meaning of any wff $\mathcal{A}$ of $\mathcal{L}_\mathcal{A}$ to be the meaning in $\mathcal{L}_\mathcal{S}$ of $tr_{\mathcal{AS}}(\mathcal{A})$,

$$\mathcal{A} \models_{\mathcal{L}_A} \mathcal{B} \text{ iff } tr_{\mathcal{AS}}(\mathcal{A}) \models_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B})$$

Therefore,

Lemma 1 *$\mathcal{L}_A$ is sound if, for every sentence $\mathcal{A}$ and $\mathcal{B}$ of $\mathcal{L}_\mathcal{A}$, if $\mathcal{A} \vdash_{\mathcal{L}_A} \mathcal{B}$ then $tr_{\mathcal{AS}}(\mathcal{A}) \models_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B})$*

[7] This is based on an example in (Woods 1991).

[8] This is in response to a question from an anonymous reviewer of this paper.

and, since $\mathcal{L}_S$ is sound,

Lemma 2 *$\mathcal{L}_A$ is sound if, for every sentence $\mathcal{A}$ and $\mathcal{B}$ of $\mathcal{L}_\mathcal{A}$, if $\mathcal{A} \vdash_{\mathcal{L}_A} \mathcal{B}$ then $tr_{\mathcal{AS}}(\mathcal{A}) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B})$*

Theorem 2 *$\mathcal{L}_A$ is sound.*

Proof: Using Lemma 2, we only need to show that, for each rule of inference, if the translations of the antecedent derivation(s) into $\mathcal{L}_S$ can be done in $\mathcal{L}_S$, then so also can the the translations of the consequent derivation(s) into $\mathcal{L}_S$. These proofs are shown in Appendix A.

Theorem 3 *$\mathcal{L}_A$ is complete.*

Proof: It can be shown that, for every rule of inference of $\mathcal{L}_S$, the translation of the conclusion into $\mathcal{L}_A$ follows from the translations of the antecedents into $\mathcal{L}_A$ using the rules of inference of $\mathcal{L}_A$ from the section, "Proof Theory of $\mathcal{L}_A$". Therefore, since $\mathcal{L}_S$ is complete, so is $\mathcal{L}_A$.

Subsumption Reasoning in $\mathcal{L}_A$

Subsumption Reasoning in $\mathcal{L}_A$ depends on derived rules of inference, several of which are presented here. The proofs of aaSubsumption and iiSubsumption are shown in Appendix B.

(aaSubsumption) $\mathcal{A}(any\ x\ \mathcal{B}(x)), \mathcal{B}(any\ y\ \mathcal{C}(y)) \vdash_{\mathcal{L}_A} \mathcal{A}(any\ y\ \mathcal{C}(y))$
For example, from *Every mammal is hairy* and *Every elephant is a mammal* to *Every elephant is hairy.*

(iiSubsumption) $\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y)) \vdash_{\mathcal{L}_A} \mathcal{A}(some\ x\ \phi\ \mathcal{C}(x))$
For example, from *Some albino elephant is valuable* and *Every albino elephant is an elephant* to *Some elephant is valuable.*

(aiSubsumption) $\mathcal{A}(any\ x\ \mathcal{B}(x)), \mathcal{C}(some\ y\ \phi\ \mathcal{B}(y)) \vdash_{\mathcal{L}_A} \mathcal{A}(some\ y\ \phi\ \mathcal{C}(y))$
For example, from *Every mammal is hairy* and *Some mammal is a pet* to *Some pet is hairy.*

aaSubsumption and aiSubsumption are the traditional syllogisms called Barbara and Darii, respectively (Lejewski 1967).

MRS and $\mathcal{L}_A$

An MRS structure (Copestake *et al.* 1999) is a triple, $\langle \mathcal{T}, \mathcal{L}, \mathcal{C} \rangle$, where $\mathcal{L}$ is a bag of reified elementary predications (EPs), each labeled by a "handle", $\mathcal{T}$ is the handle of the topmost EP, and $\mathcal{C}$ is a bag of handle constraints, which, in a scope-resolved MRS structure will all be handle equalities. For example a scope-resolved MRS structure for the sentence *every nephew of some fierce aunt runs*, in which *every nephew* outscopes *some fierce aunt* is (Copestake *et al.* 1999, p. 10) $\langle h1, \{h2 : \text{every}(x, h3, h4), h5 : \text{nephew}(x, y), h6 : \text{some}(y, h7, h8), h9 : \text{fierce}(y), h9 : \text{aunt}(y), h10 : \text{run}(x)\}, \{h1 = h2, h3 = h5, h4 = h6, h7 = h9, h8 = h10\}\rangle$. The following changes to scope-resolved MRS structures would make them notational variants of the SNePS 3 implementation of $\mathcal{L}_A$: 1) give each EP its own handle, but allow a handle to be equated to a set of handles in $\mathcal{C}$; make the top handle the central EP, instead of

1.	$\Gamma, \mathcal{A} \vdash_{\mathcal{L}_\mathcal{A}} Woman(a)$	$\mathbf{axiom}; \wedge\mathbf{E}$
2.	$\Gamma, \mathcal{A} \vdash_{\mathcal{L}_\mathcal{A}} Person(any\ x\ Woman(x))$	$\mathbf{axiom}$
3.	$\Gamma, \mathcal{A} \vdash_{\mathcal{L}_\mathcal{A}} Person(a)$	$\mathbf{anyE_1}, \mathbf{1}, \mathbf{2}$
4.	$\Gamma, \mathcal{A} \vdash_{\mathcal{L}_\mathcal{A}} Doctor(any\ z\ sonOf(z, a))$	$\mathbf{axiom}; \wedge\mathbf{E}$
5.	$\Gamma, \mathcal{A} \vdash_{\mathcal{L}_\mathcal{A}} Professional(any\ x\ Doctor(x))$	$\mathbf{axiom}$
6.	$\Gamma, \mathcal{A} \vdash_{\mathcal{L}_\mathcal{A}} Professional(any\ z\ sonOf(z, a))$	$\mathbf{anyE_1}, \mathbf{4}, \mathbf{5}$
7.	$\Gamma, \mathcal{A} \vdash_{\mathcal{L}_\mathcal{A}} Person(a) \wedge Professional(any\ z\ sonOf(z, a))$	$\wedge\mathbf{I}, \mathbf{3}, \mathbf{6}$
8.	$\Gamma, \mathcal{A} \vdash_{\mathcal{L}_\mathcal{A}} Busy(some\ x\ (any\ y\ Person(y) \wedge Professional(any\ z\ sonOf(z, y)))\ child\_of(x, y))$	$\mathbf{axiom}$
9.	$\Gamma, \mathcal{A} \vdash_{\mathcal{L}_\mathcal{A}} Busy(some\ x\ ()\ child\_of(x, a))$	$\mathbf{anyE_1}, \mathbf{7}, \mathbf{8}$
10.	$\Gamma \vdash_{\mathcal{L}_\mathcal{A}} Busy(some\ x\ (any\ y\ Woman(y)\ Doctor(any\ z\ sonOf(z, y)))\ child\_of(x, y))$	$\mathbf{anyI}, \mathbf{9}$

Table 5: Example proof in $\mathcal{L}_A$ of $\Gamma \vdash Busy(some\ x\ (any\ y\ Woman(y)\ Doctor(any\ z\ sonOf(z, y)))\ child\_of(x, y))$, where Γ stands for $Busy(some\ x\ (any\ y\ Person(y) \wedge Professional(any\ z\ sonOf(z, y)))\ child\_of(x, y))$, $Person(any\ x\ Woman(x)), Professional(any\ x\ Doctor(x))$ and $\mathcal{A}$ stands for $Woman(a) \wedge Doctor(any\ z\ sonOf(z, a))$.

a quantifier EP; eliminate the scope argument from quantifier EPs; add a set of supporting variables as an argument in a some EP; add a closure EP. After these changes the above MRS structure would be $\langle h8, \{h1: \text{every}(x, h2), h3: \text{nephew}(x, y), h4 : \text{some}(y, \{x\}, h5), h6 : \text{fierce}(y), h7 : \text{aunt}(y), h8: \text{run}(x)\}, \{h2 = h3, h5 = \{h6, h7\}\}\rangle$.

Current Implementation Status

An implementation of $\mathcal{L}_A$ as the logic of (the not yet released) SNePS 3 is currently under way, and partially completed.

A discussion of SNePS 3 may be found in (Shapiro 2000a). However, the actually implemented representation of $\mathcal{L}_A$ sentences differs in several respects from the representation discussed in that paper.

Acknowledgments

The author greatly appreciates discussions with Jean-Pierre Koenig, David Pierce, William Rapaport, and other members of the University at Buffalo's SNePS Research Group, comments on a previous draft by Kit Fine, and the comments of the anonymous reviewers of this and previous versions of this paper. This work was supported in part by the U.S. Army Communications and Electronics Command (CECOM) Intelligence and Information Warfare Directorate (I2WD), Ft. Monmouth, NJ, through Contract #DAAB-07-01-D-G001 with Booze·Allen & Hamilton.

Appendix A: Soundness Proofs

For each rule of inference of $\mathcal{L}_A$, we need to show that if the translations of the antecedent derivation(s) into $\mathcal{L}_S$ can be done in $\mathcal{L}_S$, then so also can the the translations of the consequent derivation(s) into $\mathcal{L}_S$.

Proofs of the following rules of inference are trivial, since they are the same as the corresponding rules of inference of $\mathcal{L}_S$: $\mathbf{axiom}, \mathbf{hyp}, \mathbf{cut}, \neg\mathbf{I}, \neg\mathbf{E}, \wedge\mathbf{E}, \vee\mathbf{E}, \Rightarrow \mathbf{E}, \Leftarrow \mathbf{I}, \Leftrightarrow \mathbf{E}$.

The proofs of **closureI** and **closureE** are trivial since, by rule 2 of the translation from $\mathcal{L}_A$ to $\mathcal{L}_S$, $tr_{\mathcal{AS}}(\lfloor_x \mathcal{A} \rfloor) = tr_{\mathcal{AS}}(\mathcal{A})$,

The following rules of inference of $\mathcal{L}_A$ have restrictions on the wffs. In the cases where these restrictions are satisfied, the $\mathcal{L}_A$ are the same as the corresponding $\mathcal{L}_S$ rules of inference: $\wedge\mathbf{I}, \vee\mathbf{I}, \Rightarrow \mathbf{I}$.

This leaves, as relatively non-trivial, the introduction and elimination rules for quantified terms. For $\mathcal{L}_S$ proofs, I will use the proof theory given in (Shapiro 2000b).

$\mathbf{anyI_1}$

Assume, without loss of generality, that no open bound quantified term in $\mathcal{A}(a)$ uses the same variable as any open bound quantified term in $\mathcal{B}(a)$.

$tr_{\mathcal{AS}}(\Gamma), tr_{\mathcal{AS}}(\mathcal{A}(a)) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(a))$	**assumption**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{A}(a)) \Rightarrow tr_{\mathcal{AS}}(\mathcal{B}(a))$	$\Rightarrow \mathbf{I}$
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{A}(a) \Rightarrow \mathcal{B}(a))$	$\mathbf{tr}_{\mathcal{AS}}$**rule 4**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} \forall x\, tr_{\mathcal{AS}}(\mathcal{A}(x) \Rightarrow \mathcal{B}(x))$	$\forall\mathbf{I}$
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(any\ x\ \mathcal{A}(x)))$	$\mathbf{tr}_{\mathcal{AS}}$**rule 9**

$\mathbf{anyI_2}$

$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(a))$	**assumption**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} \forall x\, tr_{\mathcal{AS}}(\mathcal{B}(x))$	$\forall\mathbf{I}$
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(any\ x))$	$\mathbf{tr}_{\mathcal{AS}}$**rule 8**

$\mathbf{anyE_1}$

Let $(any\ \overline{u}\ \overline{R_1(u)})$ be the supporting variables of indefinite terms $(some\ \overline{w}\ (\overline{u})\ \overline{R_2(w)})$ and $(some\ \overline{v}\ (\overline{u}))$ that occur in both $\mathcal{A}(a)$ and $\mathcal{B}(a)$, and let $(any\ \overline{y})$ and $(any\ \overline{z}\ \overline{R_3(z)})$ be additional arbitrary terms that occur in both $\mathcal{A}(a)$ and $\mathcal{B}(a)$.

To save space, let
$\Delta = tr_{\mathcal{AS}}(\Gamma), tr_{\mathcal{AS}}(\overline{R_1(c1)}), tr_{\mathcal{AS}}(\overline{R_3(c3)})$.

$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{A}(a))$	**assumption**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(any\ x\ \mathcal{A}(x)))$	**assumption**
$\Delta \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\overline{R_1(c1)})$	**axiom**
$\Delta \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\overline{R_3(c3)})$	**axiom**
$\Delta \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{A}(a))$	**Hyp**
$\Delta \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(any\ x\ \mathcal{A}(x)))$	**Hyp**
$\Delta \vdash_{\mathcal{L}_S} \forall\overline{u}\exists\overline{v}\exists\overline{w}\forall\overline{y}\forall\overline{z}[tr_{\mathcal{AS}}(\overline{R_3(z)}$ $\Rightarrow (\overline{R_2(w)} \wedge (\overline{R_1(u)} \Rightarrow \mathcal{A}'(a))))]$	$\mathbf{tr}_{\mathcal{AS}}$**rules 5–9**

$\Delta \vdash_{\mathcal{L}_S} \forall \overline{u} \exists \overline{v} \exists \overline{w} \forall \overline{y} \forall \overline{z} [tr_{\mathcal{AS}}(\overline{R_3(z)}$
$\quad \Rightarrow (\overline{R_2(w)} \wedge (\overline{R_1(u)} \Rightarrow \mathcal{B}'(any\ x\ \mathcal{A}'(x)))))]$ **tr$_{\mathcal{AS}}$rules 5–9**

$\Delta \vdash_{\mathcal{L}_S} \forall \overline{u} \exists \overline{v} \exists \overline{w} \forall \overline{y} \forall \overline{z} [tr_{\mathcal{AS}}(\overline{R_3(z)})$
$\quad \Rightarrow (tr_{\mathcal{AS}}(\overline{R_2(w)}) \wedge (tr_{\mathcal{AS}}(\overline{R_1(u)})$
$\quad \Rightarrow tr_{\mathcal{AS}}(\mathcal{A}'(a)))))]$ **tr$_{\mathcal{AS}}$rule 4**

$\Delta \vdash_{\mathcal{L}_S} \forall \overline{u} \exists \overline{v} \exists \overline{w} \forall \overline{y} \forall \overline{z} [tr_{\mathcal{AS}}(\overline{R_3(z)})$
$\quad \Rightarrow (tr_{\mathcal{AS}}(\overline{R_2(w)}) \wedge (tr_{\mathcal{AS}}(\overline{R_1(u)})$
$\quad \Rightarrow tr_{\mathcal{AS}}(\mathcal{B}'(any\ x\ \mathcal{A}'(x))))))]$ **tr$_{\mathcal{AS}}$rule 4**

$\Delta \vdash_{\mathcal{L}_S} [tr_{\mathcal{AS}}(\overline{R_3(c3)}) \Rightarrow (tr_{\mathcal{AS}}(\overline{R_2(c2)})$
$\quad \wedge (tr_{\mathcal{AS}}(\overline{R_1(c1)}) \Rightarrow tr_{\mathcal{AS}}(\mathcal{A}'(a)))))]$ $\forall$**E**, $\exists$**E**

$\Delta \vdash_{\mathcal{L}_S} [tr_{\mathcal{AS}}(\overline{R_3(c3)})$
$\quad \Rightarrow (tr_{\mathcal{AS}}(\overline{R_2(c2)}) \wedge (tr_{\mathcal{AS}}(\overline{R_1(c1)})$
$\quad \Rightarrow tr_{\mathcal{AS}}(\mathcal{B}'(any\ x\ \mathcal{A}'(x))))))]$ $\forall$**E**, $\exists$**E**

$\Delta \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{A}'(a))$ $\Rightarrow$**E**, $\wedge$**E**, $\Rightarrow$**E**

$\Delta \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}'(any\ x\ \mathcal{A}'(x)))$ $\Rightarrow$**E**, $\wedge$**E**, $\Rightarrow$**E**

$\Delta \vdash_{\mathcal{L}_S} \forall x\ tr_{\mathcal{AS}}(\mathcal{A}'(x) \Rightarrow \mathcal{B}'(x))$ **tr$_{\mathcal{AS}}$rule 9**

$\Delta \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{A}'(a) \Rightarrow \mathcal{B}'(a))$ $\forall$**E**

$\Delta \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{A}'(a)) \Rightarrow tr_{\mathcal{AS}}(\mathcal{B}'(a))$ **tr$_{\mathcal{AS}}$rule 4**

$\Delta \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}'(a))$ $\Rightarrow$**E**

$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} \forall \overline{u} \exists \overline{v} \exists \overline{w} \forall \overline{y} \forall \overline{z} [tr_{\mathcal{AS}}(\overline{R_3(z)}$
$\quad \Rightarrow (\overline{R_2(w)} \wedge (\overline{R_1(u)} \Rightarrow \mathcal{B}'(a))))]$ $\Rightarrow$**I**, $\wedge$**I**, $\forall$**I**, $\exists$**I**

$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(a))$ **tr$_{\mathcal{AS}}$rules 5–9**

anyE$_2$

$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(any\ x))$ **assumption**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} \forall x\ tr_{\mathcal{AS}}(\mathcal{B}(x))$ **tr$_{\mathcal{AS}}$rule 8**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(a))$ $\forall$**E**

someI$_1$

$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{A}(a) \wedge \mathcal{B}(a))$ **assumption**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} \exists x\ tr_{\mathcal{AS}}(\mathcal{A}(x) \wedge \mathcal{B}(x))$ $\exists$ **I**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(some\ x\ ()\ \mathcal{A}(x)))$ **tr$_{\mathcal{AS}}$rule 7**

someI$_2$

$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(a))$ **assumption**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} \exists x\ tr_{\mathcal{AS}}(\mathcal{B}(x))$ $\exists$ **I**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(some\ x\ ()))$ **tr$_{\mathcal{AS}}$rule 6**

someE$_1$

$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(some\ x\ ()\ \mathcal{A}(x)))$ **assumption**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} \exists x\ tr_{\mathcal{AS}}(\mathcal{A}(x) \wedge \mathcal{B}(x))$ **tr$_{\mathcal{AS}}$rule 7**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{A}(a) \wedge \mathcal{B}(a))$ $\exists$ **E**

someE$_2$

$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(some\ x\ ()))$ **assumption**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} \exists x\ tr_{\mathcal{AS}}(\mathcal{B}(x))$ **tr$_{\mathcal{AS}}$rule 6**
$tr_{\mathcal{AS}}(\Gamma) \vdash_{\mathcal{L}_S} tr_{\mathcal{AS}}(\mathcal{B}(a))$ $\exists$ **E**

Appendix B: Proofs of Subsumption Rules

aaSubsumption

$\mathcal{A}(any\ x\ \mathcal{B}(x)), \mathcal{B}(any\ y\ \mathcal{C}(y)) \vdash_{\mathcal{L}_A} \mathcal{B}(any\ y\ \mathcal{C}(y))$ **axiom**
$\mathcal{A}(any\ x\ \mathcal{B}(x)), \mathcal{B}(any\ y\ \mathcal{C}(y)) \vdash_{\mathcal{L}_A} \mathcal{A}(any\ x\ \mathcal{B}(x))$ **axiom**
$\mathcal{A}(any\ x\ \mathcal{B}(x)), \mathcal{B}(any\ y\ \mathcal{C}(y)) \vdash_{\mathcal{L}_A} \mathcal{A}(any\ y\ \mathcal{C}(y))$ **anyE$_1$**

iiSubsumption

Let $\phi = (any\ z_1 P_1(z_1), \ldots, any\ z_n P_n(z_n))$, $a_1, \ldots, a_n, b$ be ground terms that do not occur in $\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y))$, and $\overline{P(a)} = P_1(a_1), \ldots P_n(a_n)$, and where $\mathcal{A}'$ and $\mathcal{B}'$ are derived from $\mathcal{A}$ and $\mathcal{B}$, respectively, by replacing every open occurrence of $(any\ z_i\ P_i(z_i))$ by a_i, and every open occurrence of $(some\ x\ (z_1, \ldots, z_{i-1}, z_i, z_{i+1}, \ldots, z_n)\ Q(z_i))$ by $(some\ x\ (z1, \ldots, z_{i-1}, z_{i+1}, \ldots z_n)\ Q(a_i))$.

$\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y)), \overline{P(a)} \vdash_{\mathcal{L}_A} \overline{P(a)}$ **axiom**

$\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y)), \overline{P(a)}$
$\quad \vdash_{\mathcal{L}_A} \mathcal{A}(some\ x\ \phi\ \mathcal{B}(x))$ **axiom**

$\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y)), \overline{P(a)}$
$\quad \vdash_{\mathcal{L}_A} \mathcal{A}'(some\ x\ ()\ \mathcal{B}'(x))$ **anyE$_1$**

$\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y)), \overline{P(a)} \vdash_{\mathcal{L}_A} \mathcal{B}'(b) \wedge \mathcal{A}'(b)$ **someE$_1$**

$\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y)), \overline{P(a)} \vdash_{\mathcal{L}_A} \mathcal{B}'(b)$ $\wedge$**E**

$\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y)), \overline{P(a)}$
$\quad \vdash_{\mathcal{L}_A} \mathcal{C}(any\ y\ \mathcal{B}'(y))$ **axiom**

$\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y)), \overline{P(a)} \vdash_{\mathcal{L}_A} \mathcal{C}(b)$ **anyE$_1$**

$\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y)), \overline{P(a)} \vdash_{\mathcal{L}_A} \mathcal{A}'(b)$ $\wedge$**E**

$\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y)), \overline{P(a)} \vdash_{\mathcal{L}_A} \mathcal{C}(b) \wedge \mathcal{A}'(b)$ $\wedge$**I**

$\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y)), \overline{P(a)}$
$\quad \vdash_{\mathcal{L}_A} \mathcal{A}'(some\ x\ ()\ \mathcal{C}(x))$ **someI$_1$**

$\mathcal{A}(some\ x\ \phi\ \mathcal{B}(x)), \mathcal{C}(any\ y\ \mathcal{B}(y)) \vdash_{\mathcal{L}_A} \mathcal{A}(some\ x\ \phi\ \mathcal{C}(x))$ **anyI$_1$**

References

Ali, S. S., and Shapiro, S. C. 1993. Natural language processing using a propositional semantic network with structured variables. *Minds and Machines* 3(4):421–451.

Ali, S. S. 1993. A structured representation for noun phrases and anaphora. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, 197–202. Hillsdale, NJ: Lawrence Erlbaum.

Ali, S. S. 1994. *A "Natural Logic" for Natural Language Processing and Knowledge Representation*. Ph.D. dissertation, Technical Report 94-01, Department of Computer Science, SUNY at Buffalo, Buffalo, NY.

Allen, J. 1995. *Natural Language Understanding*. Redwood City, CA: Benjamin/Cummings, second edition.

Barwise, J., and Cooper, R. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 4(2):159–219.

Brachman, R. J., and Levesque, H. J., eds. 1985. *Readings in Knowledge Representation*. San Mateo, CA: Morgan Kaufmann.

Brachman, R. J., and Levesque, H. J. 2004. *Knowledge Representation and Reasoning*. San Francisco, CA: Morgan Kaufmann.

Copestake, A.; Flickinger, D.; Sag, I. A.; and Pollard., C. 1999. Minimal recursion semantics: an introduction. Draft of September 1999. CSLI, Stanford University, Stanford, CA.

Davis, E. 1990. *Representations of Commonsense Knowledge*. San Mateo, CA: Morgan Kaufmann.

Fahlman, S. 1979a. *NETL*. Cambridge, MA: MIT Press.

Fahlman, S. 1979b. *NETL: A System for Representing and Using Real-World Knowledge*. Cambridge, MA: MIT Press.

Fine, K. 1983. A defence of arbitrary objects. *Proceedings of the Aristotelian Society* Supp. Vol. 58:55–77.

Fine, K. 1985a. Natural deduction and arbitrary objects. *Journal of Philosophical Logic*.

Fine, K. 1985b. *Reasoning with Arbitrary Objects*. New York: Blackwell.

Geach, P. T. 1962. *Reference and Generality*. Ithaca, NY: Cornell University Press.

Jurafsky, D., and Martin, J. H. 2000. *Speech and Language Processing*. Upper Saddle River, NJ: Prentice Hall.

Kay, M. 1973. The MIND system. In Rustin, R., ed., *Natural Language Processing*. New York: Algorithmics Press. 155–188.

Kulas, J.; Fetzer, J. H.; and Rankin, T. L., eds. 1988. *Philosophy, Language, and Artificial Intelligence*. Studies in Cognitive Systems. Dordrecht: Kluwer.

Lehmann, F., ed. 1992. *Semantic Networks in Artificial Intelligence*. Oxford: Pergamon Press.

Lejewski, C. 1967. Ancient logic. In Edwards, P., ed., *The Encyclopedia of Philosophy*, volume Four. New York: Macmillan Publishing Co. 516.

McCarthy, J. 1979. First order theories of individual concepts and propositions. In Hayes, J. E.; Michie, D.; and Mikulich, L. I., eds., *Machine Intelligence 9*. Chichester, England: Ellis Horwood Limited. 129–147. Reprinted in (Brachman & Levesque 1985, pp. 524–533).

McCawley, J. D. 1981. *Everything that Linguists have Always Wanted to Know about Logic* *but were ashamed to ask*. Chicago, IL: The University of Chicago Press.

Orilia, F., and Rapaport, W. J., eds. 1998. *Thought, Language, and Ontology: Essays in Memory of Hector-Neri Castañeda*. Dordrecht: Kluwer Academic Publishers.

Rapaport, W. J.; Shapiro, S. C.; and Wiebe, J. M. 1997. Quasi-indexicals and knowledge reports. *Cognitive Science* 21(1):63–107. Reprinted in (Orilia & Rapaport 1998, pp. 235–294).

Schubert, L. K.; Goebel, R. G.; and Cercone, N. J. 1979. The structure and organization of a semantic net for comprehension and inference. In Findler, N. V., ed., *Associative Networks: The Representation and Use of Knowledge by Computers*. New York: Academic Press. 121–175.

Schubert, L. K. 1976. Extending the expressive power of semantic networks. *Artificial Intelligence* 7(2):163–198.

Sekar, R.; Ramakrishnan, I. V.; and Voronkov, A. 2001. Term indexing. In Robinson, A., and Voronkov, A., eds., *Handbook of Automated Reasoning II*. Cambridge, MA: MIT Press. 1853–1964.

Shapiro, S. C., and Rapaport, W. J. 1991. Models and minds: Knowledge representation for natural-language competence. In Cummins, R., and Pollock, J., eds., *Philosophy and AI: Essays at the Interface*. Cambridge, MA: MIT Press. 215–259.

Shapiro, S. C., and Rapaport, W. J. 1992. The SNePS family. *Computers & Mathematics with Applications* 23(2–5):243–275.

Shapiro, S. C., and The SNePS Implementation Group. 2002. *SNePS 2.6 User's Manual*. Department of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo, NY.

Shapiro, S. C. 1979. The SNePS semantic network processing system. In Findler, N. V., ed., *Associative Networks: The Representation and Use of Knowledge by Computers*. New York: Academic Press. 179–203.

Shapiro, S. C. 1980. Review of (Fahlman 1979a). *American Journal of Computational Linguistics* 6(3–4):183–186.

Shapiro, S. C. 1993. Belief spaces as sets of propositions. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)* 5(2&3):225–235.

Shapiro, S. C. 2000a. An introduction to SNePS 3. In Ganter, B., and Mineau, G. W., eds., *Conceptual Structures: Logical, Linguistic, and Computational Issues*, volume 1867 of *Lecture Notes in Artificial Intelligence*. Berlin: Springer-Verlag. 510–524.

Shapiro, S. C. 2000b. Propositional, first-order and higher-order logics: Basic definitions, rules of inference, and examples. In Iwańska, Ł. M., and Shapiro, S. C., eds., *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*. Menlo Park, CA: AAAI Press/The MIT Press. 379–395.

Shapiro, S. C. 2000c. SNePS: A logic for natural language understanding and commonsense reasoning. In Iwańska, Ł., and Shapiro, S. C., eds., *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*. Menlo Park, CA: AAAI Press/The MIT Press. 175–195.

Thomason, R. H. 1974. Introduction. In Thomason, R. H., ed., *Formal Philosophy: Selected Papers of Richard Montague*. New Haven, CT: Yale University Press. 1–69.

Woods, W. A., and Schmolze, J. G. 1992. The KL-ONE family. *Computers & Mathematics with Applications* 23(2–5):133–177.

Woods, W. A. 1991. Understanding subsumption and taxonomy. In Sowa, J., ed., *Principles of Semantic Networks*. Los Altos, CA: Morgan Kaufmann. 45–94.