Quantifiers as Terms and Lattice-Based Semantics

Hiroyuki Uchida and Nicholas L. Cassimatis

March 20, 2014

Abstract

When a first-order logic (FOL) is used to represent the semantics of natural language, natural language quantifiers are dealt with in terms of explicitly quantified FOL formulas. This semantic account has several problems. It fails to provide a uniform semantics for all the noun phrases that appear in an argument position of a verb in the surface syntax. With this analysis, it is harder to compositionally derive the sentential meanings from the lexical semantics. Also, how human reasoning is connected to other abilities, such as perceptual information processing, is less clear in this theory, since the knowledge representations involving quantifiers and other operators have structures quite different from the data-structures that other kinds of information processing (e.g., visual/spatial computation) uses. Based on such considerations, this paper provides a uniform semantic analysis for both referential and quantificational NPs (QNPs) as arguments of predicates. To provide adequate denotations for QNPs as arguments, we extend Godehard Link's join semi-lattice structures for plural NPs and turn them to complete lattice structures equipped with both join $\oplus$ and meet $\otimes$. We also posit positive and negative complete lattices as duals, capturing the fact that entailment relations between propositions are reversed when quantified natural language sentences are negated. By postulating singular semantic terms and their model theoretic denotations for quantificational NPs, we can remove explicit quantifiers from our first-order semantic representations, running the quantificational inferences only in the background and as required in the interpretation process. This allows us to provide a uniform semantic analysis of natural language NPs and improve the grounding of logical language expressions in their denotational interpretation structures.

1 Introduction

Using a first-order logic (FOL) language, we can represent the semantics of *Meg rocks* as '$Rock'(meg')$,' which is true if and only if the individual denoted by the argument term meg' is a member of the set denoted by the predicate $Rock'$.[1] The semantics of quantificational NPs such as *every boy* cannot be easily represented in this way, since the standard FOL semantic model does not have an individual object that the expression *every boy* can denote. Thus, the first order form '$\forall x(Boy'(x) \rightarrow Rock'(x))$' is used instead, where the explicit quantifier '$\forall x$' binds the variable x whose interpretation ranges over the members of the domain of individuals in the model. However, this semantic analysis has several problems. First, it fails to assign a uniform semantic type to all the NPs and fails to maintain the uniform predicate-argument relation between the VP and the subject NP in their semantics. Secondly, with this analysis, it is harder to compositionally derive the sentential meanings from the lexical semantics. Finally, and somewhat related to the previous two problems, FOL is not grounded well in its own denotational model of the real world. As we discussed further in Uchida, Cassimatis, and Scally (2012), this is especially problematic since the standard interpretation models of FOL characterize the perceptual representations of the real world reasonably well, that is, the FOL interpretation models include only concrete objects and the properties and relations attributed to those concrete objects. As we also discussed there, integrating the logical language expressions that characterize human reasoning and the perceptual representations that perceptual information processing uses is important for several reasons. It helps explain how human reasoning has evolved from the perceptual abilities shared by human ancestors. It is also useful in a hybrid computational architecture for Human-Level AI, which uses different computational methods and data-structures depending on the tasks, cf. Cassimatis, Bignoli, Bugajska, Dugas, Kurup, Murugesan, and Bello (2010). If the structures of the 'logical' language is similar to the data structures that other computations use, then it becomes easier to spontaneously switch between different computational methods, one of which will use the logic-like knowledge representations while the others use 'non-logical' data-structures such as (vector spaces embedding) two-dimensional graph structures, cf. Riesen and Bunke

[1] We attach the prime mark '$'$' to the constant terms and predicates in FOL. On the one hand, this convention helps distinguish constant expressions from variables in the formal language. This will make it clear that the logical language that we propose instead of FOL does NOT include any variables. Also, attaching '$'$' to the (non-variable) expressions of the symbolic logical language will help distinguish symbolic logic expressions from their interpretations, which we will explain using some meta-language notations that would otherwise look like symbolic logic expressions, as we see in Section 2 onwards.

(2012), and topological spaces, cf. Duan (2003), for visual/spatial computation. We briefly discuss these three problems of FOL here.

Look at (1), in which each English sentence is paired with a FOL formula that represents the semantics of that sentence.[2]

(1) a. [$_{\text{NP}}$ Meg] rocks.
FOL: $Rock'(meg')$

b. [$_{\text{NP}}$ Every girl] rocks.
FOL: $\forall x(Girl'(x) \rightarrow Rock'(x))$

c. [$_{\text{NP}}$ Some girl] rocks.
FOL: $\exists x(Girl'(x) \wedge Rock'(x))$

The three expressions that are labeled as NP in (1) occupy the same surface position in the English sentences (i.e., each of them is the subject of the verb *rocks*). In contrast, this verb-subject relation is interpreted in terms of a simple predicate argument relation only in (1-a). In both (1-b) and (1-c), the semantics of NP spread all over the FOL formulas. Even before we consider a precise interpretation algorithm, it is easy to see that the interpretation of the English sentences into the FOL formulas in (1-a)∼(1-c) is more complex than the interpretation of the same sentences into the logical formulas that we propose below, i.e., $Rock'(meg')$, $Rock'(uboy')$ and $Rock'(eboy')$, in which the above-mentioned verb-subject relation is interpreted as the simple predicate-argument relation for all the English sentences.[3]

Next, using FOL to represent the semantics of natural language expressions makes it difficult to compositionally construct the sentential semantics starting with the lexical semantics. To see this point, try to provide FOL expressions that can represent the lexical semantics of *every* and *some* respectively, assuming that the sentential meanings are as

[2] We are not concerned about the precise interpretation of English sentences here, and thus their semantics in terms of FOL ignore complexities such as tense and aspect.

[3] In a Chomskyan syntactic framework, May (1977) covertly moves each quantified NP such as *every girl* and *some girl* into a co-indexed higher-syntactic position at an abstract level of natural language syntactic structures (which is called 'LF'), such as [$_{\text{TP}}$ [$_{\text{NP}}$ Every girl]$_1$ [$_{\text{XP}}$ t$_1$ rocks]]. Some might then argue that abstract syntactic structures such as LF can be interpreted into FOL in a simpler manner. However, LF structures still deviate from the structures of FOL (which is why many Chomskyan semanticists use some higher-order logical language, rather than FOL, to represent natural language semantics). Besides, interpreting English sentences into their semantics by way of abstract syntactic structures that deviate from the surface English structures only makes the overall natural language interpretation process more complex. The same applies to the well-known Montagovian interpretation of natural language sentences, which includes the so-called (semantic) type-lifting operation on noun phrases. See Montague (1973) and Montague (1970) for details.

the FOL formulas in (1-b) and (1-c) suggest and the lexical semantics are maintained in those sentential interpretations. Again, the semantics of *every* and *some* spread all over the corresponding FOL formulas and it is impossible to provide the lexical semantics of these natural language determiners in terms of some well-formed sub-expressions of the FOL formulas. Actually, as Dowty, Wall, and Peters (1981) discussed in details and as we briefly explain below, interpreting FOL expressions into their standard set-theoretic interpretation models in a compositional manner is not easy either. This is because the intended interpretation of variables such as x in (1) can be different locally (where x is a free-variable) and globally (where x can be bound by a quantifier such as $\forall x$ and $\exists x$). Jacobson (1999) discusses both empirical and model theoretic problems with the use of variables in the natural language semantic representations, motivating the use of a higher-order logic instead of FOL to represent the natural language semantics. This 'model theoretic' problem is related to the third problem of using FOL to represent the natural language semantics, which we discuss next.

As we indicated above, some of the crucial expressions of FOL are not grounded well in their standard interpretation models. For comparison, we first consider the FOL expressions that are grounded well in their denotational models. In the FOL interpretation models (cf. van Dalen, 2004), each constant term, such as meg' in (1), is mapped to a member of the domain of individuals.[4] Similarly, the interpretation function in each semantic model maps each unary predicate expression in FOL, such as $Rock'$ in (1), to a set of individuals (i.e., it maps each unary predicate to a member of the power set of the domain of individuals). For example, the interpretation of $Rock'$ is the set of individuals who rock in the given interpretation model.[5] The denotation of a two-place predicate expression, such as $Like'$, is a set of ordered pair of individuals taken from the domain set, e.g., $\{\langle Meg, Jack\rangle, \langle Sid, Nancy\rangle, \langle Courtney, Curt\rangle, ...\}$, in which each ordered-pair '$\langle a, b\rangle$' represents the 'liker-likee' pair in the given first-order interpretation model. This line of interpretation can be extended to n-ary predicate that requires n arguments, as

[4] For convenience, we let English names such as Meg and Jack represent such model theoretic individual objects. Do not confuse these with the constant terms in the logical language, such as meg' and $jack'$. Also, unlike 'Meg' and 'Jack' used as English names, each of which can be shared by many different individuals in the real world, 'Meg' and 'Jack' as meta-notations of the model theoretic objects correspond to exactly one object each by assumption.

[5] The standard FOL interpretation models can distinguish the meanings of two predicate expressions only to the degree that the sets they denote contain different individuals. When the natural language semantics require finer granularity than that, natural language semanticists often extend FOL into some sort of Modal Logic, in which each unary predicate denotes a function from the set of possible worlds to the power set of the domain of individuals. However, such additional complexities are irrelevant to the current discussion.

in $Pred^n(a_1, ..., a_n)$, where $Pred^n$ denotes a set of ordered n-tuples of individuals taken from the domain of individuals. This of course allows us to provide the interpretation of FOL formulas such as $Rock'(meg')$ in a compositional and precise manner (that is, this formula is true if and only if meg' denotes a member of the set that $Rock'$ denotes in the given interpretation model) and similar for the interpretation of the formulas in the form of $Pred^n(a_1, ..., a_n)$, when $a_1, ..., a_n$' are all constant terms (and hence denote particular members of the domain of individuals).

Now, when the logical language includes variables and quantifiers, its interpretation does not go this straightforwardly, and most importantly in the current discussion, the grounding of FOL in their denotational structures is significantly compromised. First, we look at the interpretation of variables more closely. When the variable is not bound by a quantifier, say, in $Rock'(x)$, the interpretation of the variable depends on the given variable assignment function. For example, if the given assignment function, say, g_1, maps x to Meg, then x denotes Meg, whereas when a different assignment function, say, g_2, is chosen, which maps x to Jack, then x denotes Jack, even if the interpretation model itself stays the same. For comparison, the interpretations of meg' and $Rock'$ do not change within one model.[6]

Now, when the variable is bound by a quantifier, say, in $\forall x Rock'(x)$, then the interpretation of x no longer depends on a particular interpretation function such as g_1 and g_2. Roughly, $\forall x Rock(x)$ is true if and only if for all the logically possible value assignment to x with the given domain of individuals, the assigned value (i.e., a member of the domain of individuals) is a member of the set denoted by $Rock'$.[7] Again, this kind of interpretation makes the semantic interpretation 'non-compositional', since the local interpretation of x is not maintained globally. But more importantly in the current discussion, variable assignment functions are NOT part of the interpretation models. Instead, it is an external tool that allows one to *somehow* interpret FOL formulas with variable terms within the first-order set theoretic interpretation models. As Jacobson (1999) suggested, variable assignment functions do not mesh well with at least the spirit of model theoretic semantics of logical language expressions, which is a clear down-side of FOL as a logical language, independent of whether FOL can adequately characterize natural language semantics or

[6]In fact, most logicians assume that the denotations of constant terms such as meg' stay the same in all the interpretation models. This is partly because the constant terms in a logical language act as the unique identifiers of all the individuals that can ever be included in an interpretation model. However, this convention is not particularly important in the current discussion.

[7]It is logically possible to map each x to ANY member of the domain of individuals, which allows us to capture the semantics of *Everybody rocks* in terms of $\forall x Rock'(x)$.

not.

As we indicated above, the imperfect grounding of FOL in its own interpretation models is not only a formal problem. The FOL interpretation models can characterize perceptual representations of the real world reasonably well, since they include only concrete objects and concrete properties and relations on those objects (i.e., properties and relations that can be characterized as sets of (ordered n-tuples of) those concrete objects).[8] Thus, if we ignore elements such as quantifiers and variables in FOL, which are not grounded well in the FOL interpretation models, using FOL to characterize natural language semantics (and also human reasoning) can actually help ground human reasoning in perceptual representations. As it is, however, FOL does include quantifiers and variables, which we try to get rid of in our novel formal language.

Before we sketch the logical language we propose, we briefly explain why we do not adopt a higher-order logic for representing natural language semantics. Using higher-order logic expressions such as '$(every_{(et)((et)t)}\ boy_{et})smoke_{et}$' together with Barwise and Cooper's (1981) Generalized Quantifier analysis as their interpretations can solve the first two of the three problems with FOL that we explained above. However, such an analysis involves some form of generalized type raising in the syntactic derivation that pairs phonological strings with higher-order logic expressions and this may make the grammar undecidable.[9] Also, first-order logics are generally preferred to higher-order logics in terms of the efficiency of inferences. Finally, the intended interpretation models for higher-order logic, which include complex semantic structures such as (characteristic functions of) sets of sets of individuals and functions from sets of individuals to sets of sets of individuals, deviate too much from perceptual models of the real world, which would make it hard to ground the

[8]Note that most computational models of human perception involves objects categorized into some sort of classes. The expressive power of the first-order interpretation model is not more expressive than such computational data structures.

[9]Mark Steedman (p.c.) points out that higher-order representations can be used only as a 'glue' to construct a first-order representation by way of the syntactic calculus, as we can see with the lexical assignment of '$\lambda P_{et}.\lambda Q_{et}.\forall x_e(Px \to Qx)$' to the determiner *every*. We are actually happy with this use of type theory. Note that higher-order 'types' such as '(et)t' can be seen as a propositional formula in a linear logic, as becomes clearer if we represent it as '$e \to (e \to t)$,' where we can see e and t as atomic propositional letters. Similarly, we can see forward and backward slashes '$\backslash, /$' in Lambek Calculus as (directional) implication arrows '$\leftarrow, \rightarrow$' in a variant of propositional linear logic. However, note also that the Generalized Quantifier analysis essentially interprets the semantics of quantificational determiners such as *every* and *some* as denoting relations between two sets of individuals and there is more to the GQ analysis of QNPs than the use of types as glues. Thus, we think that the formal problems that we alluded to in the main text is more serious with the GQ theory than with a mere use of type theories in order to construct first order formulas. Also, there is an additional complication with regard to interpreting QNPs in non-subject positions by way of the GQ semantics (Heim and Kratzer, 1998; Hendriks, 1987; Uchida, 2008). But we leave further discussion for another paper.

natural language semantics in perceptual representations. Use of such an expressive logical language also makes it difficult to switch from the 'logical' computation using the higher-order logic and other computational methods in a hybrid computational architecture for Human-Level AI that we discussed above.

Based on such considerations, we propose a first order language that includes individual terms for quantificational NPs as well as for referential NPs, as in (2).

(2) a. $Smoke'(uboy')$ for *Every boy smokes.*
b. $Smoke'(eboy')$ for *Some/a boy smokes.*

Since $uboy'$ and $eboy'$ are (constant) individual terms, we might want a new kind of individuals in our model theoretic objects that these newly added individual terms can denote.[10] To demonstrate that such model theoretic objects are indeed mathematically well-defined, we introduce a particular lattice structure in which the individual denotation for a universal term such as $uboy'$ is the minimal element (i.e., the greatest lower bound of the lattice) and the denotation for an existential individual term such as $aboy'$ is the maximal element (i.e., the least upper bound of the lattice). The partial order for such a lattice is provided in such a way that the logical entailment relations between sentences that include quantificational and referential NPs are maintained. We then show how such lattice structures can capture the interaction between natural language quantifiers and negation by way of duality.

The idea of using lattice structures for analyzing the NP semantics is from Link (1983) which used semi-lattice structures for the semantics of plural and mass NP expressions. However, as we discuss further in Section 2, Link (1983) did not use complete lattice structures for individuals and therefore did not use the additional inferential power associated with complete lattice. In contrast, since we assume that lattice structures for individuals are complete, which have both the join $\oplus$ and the meet $\otimes$, we can also use the least upper bound (lup) and the greatest lower bound (glb) of the lattice.[11] As we explain in Section 2, the lub of the lattice for a partially-ordered set of individuals is the 'existential' individual in that set while the glb of the lattice for the same set of individuals is the 'universal' individual in that set. As I also discuss there, the universal and the existential individuals

[10] Actually, we later show that $uboy'$ and $aboy'$ can be interpreted in the standard FOL interpretation models that we explained above without such novel model theoretic objects, as long as we ignore the semantics of plural quantifiers such as *most* and the 'collective' interpretation of plural NPs.

[11] For those who are not familiar with order theory in math, we provide brief explanation of the basic lattice theoretical notions with examples in the following sections.

are dualized when we construct a complete lattice that is the dual of the above lattice. This paper focuses on novel algebraic objects for 'singular' quantifiers, e.g., *every boy* and *some/a girl*. However, in Section 2.2, we also suggest model theoretic objects for *most boys*, which can be extended to other plural quantifiers in future research.

As we indicated in footnote 10, after introducing the interpretation models including the denotations for *uboy′* and *eboy′*, we also show that these 'quantificational' terms in our formal language can actually be interpreted in the standard FOL interpretation models without including the lattice structures that we mentioned above. Lattice structures then are only used to meta-theoretically explain how we can interpret these quantificational terms in the standard FOL interpretation models in an empirically adequate manner. Since the semantic analysis of plural quantifiers such as *most* and the collective interpretation of plural individual terms can use the full-expressive power of the above-mentioned lattice structures included in the interpretation model, some might wonder what is the point of showing that we can remove the lattice structures in our treatment of natural language semantics, ignoring such an important subpart of the NP semantics. However, including lattice structures for plural individuals makes the interpretation model structures much more complex than the standard FOL interpretation models, which some might want to avoid even if the additional expressive power is useful for dealing with the semantics of some natural language expressions. Also, it is actually possible to deal with even the above mentioned interpretation of the plural quantifiers/individuals without including the suggested lattice structures in the interpretation models (although that would require us to either shift to at least second-order logic interpretation models, or alternatively, shift to the interpretation models for sorted-first-order language). Discussing all of these complications is beyond the scope of this paper, but we briefly discuss these possible future research topics after we explain the lattice structures for model theoretic individuals.

Section 2.1 introduces the lattice structures in terms of which we can interpret the singular quantifier terms in the first-order interpretation models. As we mentioned above, we first include the lattice structures in the interpretation models for our logical language that does not include quantificational operators or variables, but later we suggest that singular quantifier terms (as well as plural terms with the distributive interpretation) can be interpreted in the standard first-order interpretation model, while using the lattice structures only meta-theoretically, that is, in order to show how quantifier terms are interpreted in the way we explain. Section 2.2 shows how we can deal with basic scope ambiguity data in this logical language. We also discuss what additional tools will be required to deal with more complex quantifier data. Section 3 provides the definition of a quantifier-free first-order language that exemplifies our proposal in Section 2 and Section 4 shows the interpretation of that language in both the interpretation models with and without the above-mentioned lattice structures. Section 5 provides a concluding remark.

2 Lattice structures for NP Semantics

For the reasons that we explained in section 1, we remove explicit quantifiers from our first-order logic language that represents the semantics of natural language. Instead, our logical language includes terms for quantificational NPs, which can fill the argument slots of predicate expressions just like any other terms do. For example, the terms *uboy′* for *every boy* and *eboy′* for *some/a boy* can fill the argument slot of $Noisy'$, as in $Noisy'(uboy')$ (for *Every boy is noisy*) and $Noisy'(eboy')$ (for *Some/a boy is noisy*). Since these are constant terms, we need the corresponding model theoretic objects that characterize their intended interpretations.[12] Subsection 2.1 provides model structures that include model theoretic denotations for both (singular) quantificational and referential NPs. 2.2 briefly shows how our quantifier-free first-order language with novel model structures deals with basic natural language quantifier data.

2.1 Model structures with complete NP lattices

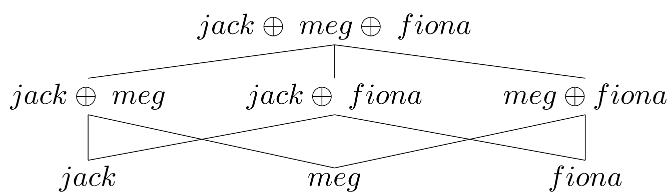
Dealing with the semantics of NPs in natural language, Link (1983) introduced the notion of plural individuals. For example, if the natural language expression *Jack* denotes the model theoretic object $jack$ and *Meg* denotes meg, then the denotation of the plural NP *Jack and Meg* is $jack \oplus meg$, which is called a 'plural individual', with $jack$ and meg being its 'atomic' components. When we interpret the above plural NP in natural language sentences, the interpretation can be 'distributive', for example, in the salient interpretation of *Jack and Meg passed the driving test*, which means that Jack passed his

[12] Any non-variable expressions in a logical language should ideally be matched up with some sort of mathematically well-identified intended interpretations, such as some sort of algebraic objects/relations, etc., just for the purpose of understanding precisely what those expressions mean. Whether we include those algebraic objects/relations into the interpretation models with regard to which the truth-values of the logical formulas are evaluated is a separate issue. For comparison, although every categorial formula and structures that are used in Type Logical Grammar (cf. Moortgat, 1997; Uchida, 2008) should ideally be paired with some algebraic intended interpretations, not all of those algebraic interpretations need to be included in the theoretical representations of English expressions and their semantics that the grammar theory connect in terms of the categorial inference.

driving test and Meg passed her driving test, most naturally at a different time. In terms of the Link's plural individual, $jack \oplus meg$, we can say that the property of passing the exam is distributed to each atomic component of this plural individual, i.e., $jack$ and meg. The interpretation of the above NP can also be 'collective', e.g., in the interpretation of *Jack and Meg met up at the Penn station*, in which the property of 'meeting up at the Penn station' is collectively attributed to the plural individual '$jack \oplus meg$.'[13] In this paper, we mostly ignore the collective interpretation of plural NPs, as well as the interpretation of mass and uncountable terms such as *water* and *audience*.

When there are three atomic individuals, say, $jack$, meg and $fiona$, then Link's plural individuals produce the 'join' semi-lattice structure in (3).

(3)



The lattice structure in (3) represents a particular partial order between the individuals in its vertices, where the partial order '$\leq$' means that for all the individuals a, b: $a \leq b$ if and only if a and b have the same atomic component(s) or b has more components than a.[14] In terms of the graph structure in (3), $a \leq b$ iff either $a = b$ or if b can be reached from a by continuously going up one or more edges. With this partial order, the least upper bound (lub) of this partially ordered set of individuals in (3) is $jack \oplus meg \oplus fiona$.[15] Link's structure above uses only one connective, $\oplus$, and is not complete. Also, he discusses only semilattice structures whose 'atomic' components (i.e., the leaf nodes at the bottom in (3)) are fully specified. Because of these, although the semi-lattice structure in (3) does have the least upper bound (lub), that is, $jack \oplus meg \oplus fiona$, we cannot make use of the maximal inferential power associated with lattice structures. More specifically, the lub

[13]Note that a single person cannot have this property, since one person cannot meet on his/her own. Linguists would say that the predicate 'meet' is obligatorily collective.

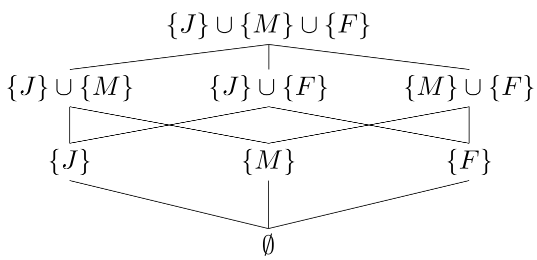
[14]This interpretation works only for Link's semi-lattice structures. Our interpretation is based on a certain kind of logical entailment relation between the individuals in the lattice. See below.

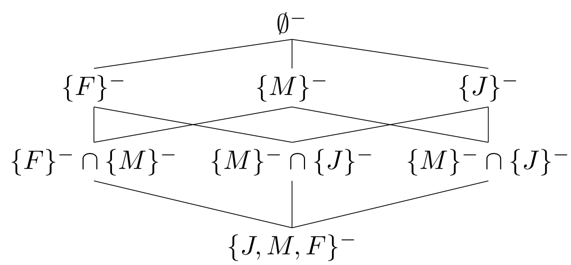
[15]Suppose that the universal domain set U is partially ordered. Then, the lub of a subset $D \subseteq U$, if there is any, is a member u of U such that $\forall d \in D : d \leq u$ and $\forall v \in U$, if $d \leq v$ for all $d \in D$, then $u \leq v$. A join semi-lattice is a partially ordered set that contains a lub for any of its finite subsets. Note that a join-semilattice can be an infinite set.

'$jack \oplus meg \oplus fiona$' can only act as the denotation of the plural NP, *Jack, Meg and Fiona* and we cannot use the 'dual' of the lattice structure in (3).

To illustrate the additional inferential power of complete lattice, we first discuss a well-known complete lattice structure, that is, the power set of a set of objects. Consider the power set of the set '$\mathbb{D} = \{J, M, F\}$' in (4-a), which is a complte lattice, like its dual in (4-b).

(4) a. $\wp(\mathbb{D})$: b.





In (4), the partial-order '$\leq$' is the subset relation '$\subseteq$,' i.e., $\emptyset \subseteq \{J\} \subseteq \{J, M\} \subseteq \{J, M, F\}$. Note that $\{J\} \cup \{M\} = \{J, M\}$ and $\{J\} \cup \{M\} \cup \{F\} = \{J, M, F\}$, where '$\cup$' represents the set union. As a matter of notation, for each complete lattice $\mathbb{L}$, we let $\bigvee \mathbb{L}$ represent the least upper bound (lub) of $\mathbb{L}$ and let $\bigwedge \mathbb{L}$ represent the greatest lower bound (glb) of $\mathbb{L}$. For (4a), $\bigvee \wp(\mathbb{D}) = \{J, M, F\}$ and $\bigwedge \wp(\mathbb{D}) = \emptyset$.

The dual of (4a) is (4b), which orders the complements of the sets in (4a). We use the set-complement notation, A^- ($= \mathbb{D} \backslash A$, i.e., the complement of the set A with $\mathbb{D}$ being the universal domain set), decorating each vertex of the lattice in (4b), in order to graphically represent the duality, but notice that denotationally, (4a) and (4b) are the same lattice (e.g., $\emptyset^- = \{J, M, F\} = \{J\} \cup \{M\} \cup \{F\}$, $\{F\}^- = \{J, M\} = \{J\} \cup \{M\}$, etc). For those readers who are familiar with classical propositional logic (CPL), the denotational correspondence between (4a) and (4b) is for the same reason why contraposition (i.e., $\forall \phi, \psi \in CPL$: $\phi \rightarrow \psi \Leftrightarrow \neg\psi \rightarrow \neg\phi$) is valid.

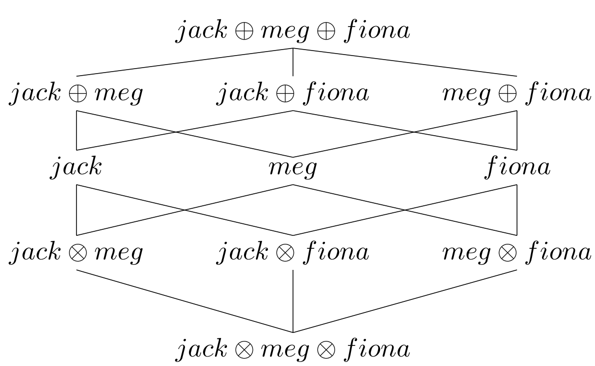
Note that when the empty-set $\emptyset$ is removed from the 'positive' lattice in (4a), the structure is the same as Link's join-semilattice in (3). This is not a co-incidence since the 'join' connective $\oplus$ in Link's join semilatice is meant to be interpreted in essentially the same way as the set union. The difference is only that Link's $jack \oplus meg$ is an object and hence can act as the intended interpretation of a plural term (which can then saturate an argument-slot of a predicate expression, as in $Pred'(term')$), whereas the corresponding

$\{J, M\}$ in (4a) is a set and cannot be the denotation of a term in the logical language, unless we turn the logical language into a higher-order one or its equivalent.

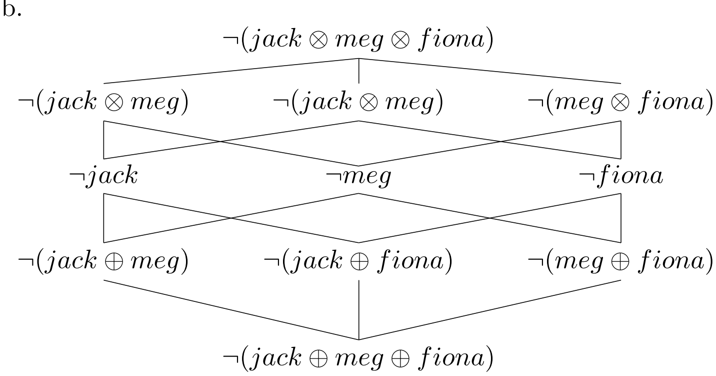
Now, can we turn Link's semi-lattice into a complete one by simply replacing the set in each vertex of the lattice in (4a) with the corresponding plural individual? Actually, even if we pull that off by adding to Link's structures a new connective that corresponds to the set intersection $\cap$, such a complete lattice does not serve our purpose. Among several reasons, the most fatal one is that the lub in (4a) is the empty set. Whatever '(plural) object' that corresponds to the empty set would not serve as the intended interpretation of the universal quantifier term.

We start the explanation of our lattice structures for NP semantics with a simple example. Suppose that the domain of individuals is $\mathbb{D} = \{jack, meg, fiona\}$. Then our complete lattice structure is (5-a) with (5-b) being its dual.

(5) a.



b.



Consider the 'positive' complete lattice in (5-a). The atomic individuals, $jack$, meg, $fiona$ are like the individuals in the stnadard FOL interpretaiton models. The join '$\oplus$' means something different from what it means in Link's plural individuals. It actually is closer to the original interpretation of 'join', whose interpretation can be captured in terms of 'OR' in some sense. That is, for any individuals $a, b:\ a \oplus b$ means 'a or b.'[16] More specifically, for any set of individuals in the interpretation model, $a \oplus b$ is in that set if and only if a or b is in that set.[17] Since 'or' here is inclusive, it means that $a \oplus b$ is in that set if and only if at least one of a and b is in that set. For example, $jack \oplus meg$ is in the set of kids iff at least one of $jack$, meg is in the set of kids, and $jack \oplus meg \oplus fiona$ is in the set of kids iff at least one of $jack$, meg, $fiona$ is in that set.

Next, $a \otimes b$ means 'a AND b.' More specifically, for any set of individuals A and any pair of individuals a and b (simple or complex): $a \otimes b$ is in A if and only if both a is in A and b is in A. For example, $jack \otimes meg$ are in the set of kids iff both $jack$ and meg are in that set.

Remember the graphic notation is such that $a \leq b$ if and only if either $a = b$ or b can be reached from b by going up one or more edges in the graphic lattice structure. Thus, in (5-a), $(jack \otimes meg) \leq jack \leq (jack \oplus meg)$. How does this partial order follow from the interpretations of $\oplus$ and $\otimes$ that we we explained above? Actually, it is easy to see that this partial order does follow. For any set of individuals, say, A, suppose that $jack \otimes meg$ is a member of A. Then, according to the interpretation of $\otimes$ as above, both $jack$ and meg are in A. It follows that $jack$ is in A. In other words, if $jack \otimes meg$ is in A, then $jack$ is in A. This entailment relation corresponds to the partial order, $(jack \otimes meg) \leq jack$. Entailment does not hold in the opposite direction. That is, suppose that $jack$ is in A. From there it does NOT follow that $(jack \otimes meg)$ is in A since meg might not be in A. That is, the partial order is $(jack \otimes meg) \leq jack$, not the other way around. It is easy to see that this partial order holds everywhere in the lower half of the lattice in (5-a).

Similarly, suppose that $jack$ is in A. Then, it follows that $(jack \oplus meg)$ is in A (since $(a \oplus b)$ is in A iff at least one of a, b is in A). In contrast, suppose that $(jack \oplus meg)$ is in A. From there, it does NOT follow that $jack$ is in A (since it might be only meg who is in A). Thus, $jack \leq (jack \oplus meg)$. It is easy to see that this partial order holds everywhere in the upper half of the lattice and hence, in combination with the above, the suggested partial order is valid in the entire lattice in (5-a).

Now, for any set of individuals: if the greatest lower bound (glb) of the lattice in (5-a)

[16] We omit the parentheses in $(a \oplus b)$ unless we think that it is misleading.

[17] Recall that $a \oplus b$ is a model theoretic object and NOT a term in the logical language (which carries the prime mark '$'$' in our notation). As we indicated above, we later try removing these 'complex individuals' from the interpretation models and interpret the corresponding term $a' \oplus b'$ in our logical language in terms of the interpretations of its atomic component terms, i.e., a' and b'.

(i.e., $(jack \otimes meg \otimes fiona)$) is in that set, then it follows that every individual in that lattice is in that set. Similarly, suppose that at least one of the three atomic individuals, $jack$, meg, $fiona$, is in any set of individuals. Then it follows that the least upper bound (lub) of the lattice in (5-a) (i.e., $(jack \oplus meg \oplus fiona)$) is in that set. As we explain later, this fact is used when we let each universal quantifier term denote the glb of the corresponding lattice and each existential quantifier term denote the lub of the corresponding lattice.

Turning to the dual lattice in (5-b), for any set of objects: the negation of each atomic individual, say, $\neg jack$, represents a state of affairs in which the positive individual $jack$ is NOT in that set. In other words, for each set of individuals A and for each individual a: $\neg a \in A$ if and only if $a \notin A$. Note that each vertex in (5-b) is occupied by the negation of the individual in the corresponding vertex in (5-a). Note also that the partial order $\leq$ does correspond to the bottom-to-top order across the edges in (5-b). For exmple, for any set of individuals A: $\neg(jack \otimes meg)$ is in A if and only if $(jack \otimes meg)$ is not in A, which means that at least one of $jack$, meg is not in A.[18] Now if $\neg jack$ is in A, it means that $jack$ is not in A, and then the above condition is satisfied. It means that if $\neg jack$ is in A, it follows that $\neg(jack \otimes meg)$ is in A. That is, $\neg jack \leq \neg(jack \otimes meg)$. Similarly, for any set A: $\neg(jack \oplus meg)$ is in A iff $(jack \oplus meg)$ is not in A, which means that it is not the case that at least one of $jack$, meg is in A, which in turn means that neither $jack$ nor meg is in A. Now if this situation holds, then it follows that $jack$ is not in A, and hence $\neg jack$ is in A. Thus, $\neg(jack \oplus meg) \leq \neg jack$.

Notice that $\oplus$ formulas are placed in the top half in (5-a) but in the bottom half in (5-b) and vice versa for $\otimes$. This is because the entailment relations that we explained above are reversed between (5-a) and (5-b) (i.e., the set membership of a lower individual entails the set membership of a higher individual via continous edges). And this is of course what duality means.

We can finally explain the denotations of $ukid'$ (for *every kid*) and $ekid'$ (for *some/a kid).* First, suppose that the set of kids in some interpretation model is $[\![kid']\!]^M = \{jack, meg, fiona\}$ (i.e., Jack, Meg and Fiona are the only kids in this model).[19] Then, we can construct the lattice in (5-a) above. Next, suppose that the glb of this lattice, i.e., $(jack \otimes meg \otimes fiona)$ is a member of another set, say, the set of noisy individuals, in this model.[20] Now, remember from the above explanation that $(jack \otimes meg \otimes fiona)$ is a

[18]Note that a De Morgan's law does apply here, that is, $\neg(jack \otimes meg)$ if and only if $(\neg jack \oplus \neg meg)$, which means that either $jack$ is not in A or meg is not in A (or neither).

[19]For each logical expression α: $[\![\alpha]\!]^M$ means the denotation of α in the interpretation model M. We sometimes omit the model name M, when it is not important in any way.

[20]Actually, this 'another set' can be the same set of kids, but for the sake of presentation, we consider

member of a set if and only if all the component individuals are members of that set. More specifically, $(jack \otimes meg \otimes fiona)$ is a member of the noisy individuals if and only if $jack$, meg, $fiona$ are all members of the set of noisy individuals. Notice that this is exactly what the proposition 'Every kid is noisy' (or $Noisy'(ukid')$ in our language) will require in this interpretation model (i.e., in order for the proposition to be true in this interpretation model). In other words, in this particular interpretation model, we can assume that the term $uboy'$ denotes the glb of the ('positive') lattice structure for the set of boys, i.e., $(jack \otimes meg \otimes fiona)$ in (5-a).

In the above interpretation model, the set of kids is a particular finite set, but this does not have to be the case. Even if the set of atomic kids is (countably) infinite in some interpretation model (to include all the kids from distant past to eternity, for example), as long as the set of atomic kids does not exceed countability/enumerability, the lattice for that set has both a unique glb and a unique lub.[21] That is, if the atomic members of the set of kids are $a_1, ..., a_n$ (where $n \leq \infty$), then, the glb of the positive lattice for this is $(a_1 \otimes ... \otimes a_n)$, which we can take as the denotation of $ukid'$ in this model.

Getting back to the model that has (5-a) for the set of kids, suppose that its lub, which is $(jack \oplus meg \oplus fiona)$, is a member of another set, say, the set of cheeky individuals. Then, given the above interpretation of this complex object, what it means is that at least one of $jack$, meg, $fiona$ is in the set of individuals. Note that this is exactly what the proposition, 'Some/a kid is cheeky' (or $Cheeky'(ekid')$ in our language) requires to hold in this interpretation model. Thus, in a model in which the set of kids denotes $\{jack, meg, fiona\}$, we can take the lub of the lattice for this set, that is, $(jack \oplus meg \oplus fiona)$, as the denotation of $ekid'$.

Again, we do not need to identify the entire membership of the set of kids. If the set of atomic kids is countably infinite, such as $\{a_1, ..., a_n\}$, where $n \leq \infty$, then the denotation of $ekid'$ would be still the lub of the lattice for that countably infinite set, i.e., $(a_1 \oplus ... \oplus a_n)$.

Next, still assuming that the set of atomic kids is $\{jack, meg, fiona\}$ in the given model, consider the 'negative' lattice in (5-b). Suppose that its glb, that is, $\neg(jack \oplus meg \oplus fiona)$, is in another set, say, the set of grown-ups, in that model. With the interpretations of '$\oplus$' and '$\neg$' that we explained above, this means that none of $jack$, meg, $fiona$ is a member of the set of grown-ups in that model. Note that this is precisely what the proposition 'No kid is a grown-up' (or $GrownUp(\neg ekid)$ in our language) requires to hold in order to be

a different set.

[21]Technically, every finite or countably infinite complete lattice has both a unique glb and a unique lub. Cf. Blyth (2005).

true in this interpretation model. Thus, we can let the 'negative' quantifier term $\neg ekid$ denote the glb of the negative lattice (5-b) in this interpretation model. Again, the set of kids does not have to be finite or specific, in order to get this result, since every countably infinite (and every finite) complete lattice has a unique glb.

Similarly, to say that $\neg(jack \otimes meg \otimes fiona)$ is a member of the set of tall individuals in an interpretation model is the same as saying that at least one of $jack$, meg, $fiona$ is not tall in that interpretation model, and thus we can capture the interpretation of 'Some/a kid is not tall' (or $Tall(\neg ukid')$ in our language) in this interpretation model and we can let $\neg ukid'$ denote the lub of the above negative lattice, i.e., $\neg(jack \otimes meg \otimes fiona)$.

In our account above, the denotations of $ukid'$ and $ekid'$ depend on which atomic individuals the set of kids contain in that interpretation model[22] Some might then wonder if the situation would then be the same as using variables in the logical language. Actually, it is not the same. Remember that the interpretation of variables is independent of the interpretation models (and it has to be that way in FOL). For each logically possible interpretation model, there are as many variable assignment functions that can map each variable to each member of the domain of individuals. In contrast, in our case, if we fix the interpretation model, the denotation of each quantifier term in the form of either $uPred'$ or $ePred'$ (or $\neg ePred'$ or $\neg uPred'$) is fixed into exactly one element, that is, the lub or the glb of the positive (negative) complete lattice for the set denoted by the predicate expression $Pred'$.

Next, some might wonder if the denotation of the universal term $ukid'$ for *every kid* and the denotation of $(jack' \otimes meg' \otimes fiona')$ for *Jack, Meg and Fiona* are the same when the set of boys is $\{jack', meg', fiona'\}$.[23] This is correct,[24] but notice that the denotation of the term $(jack' \otimes meg' \otimes fiona')$ stays the same in any other interpretation model in which the membership of the set of kids is different, say, $\{jack, meg, fiona, bob, emily\}$.

[22]Recall that the denotations of $ukid'$ and $ekid'$ always exist in any first-order interpretation model, since every set included in a first-order interpretation model is either finite or countably infinite. When the set of boys is empty, then the glb and the lub of the lattice for the set are the same, i.e., the 'non-existing' object, $\perp$, which represents the absolute lack of existence and hence is not a member of any set. This means that $Noisy'(uboy')$ is FALSE when the set of boys is empty. This is different from the truth value of the FOL formula, $\forall x(Boy(x) \rightarrow Noisy(x))$ in the same context, since this FOL formula is TRUE when the set of boys is empty. We are not sure which is empirically better, but since humans do not have a decisive intuition about whether the proposition 'Every boy is noisy' is true or false when there is no boy, this difference on its own does not give any empirical advantage to either theory.

[23]Recall that $ukid'$ and $(jack' \otimes meg' \otimes fiona')$ are terms of our logical language, not individuals in the interpretation models. Here we overload information with the symbol $\otimes$ by using it both in the meta-notation for complex model theoretic individuals and in the terms of the logical language. Since the atomic terms in the logical language always carry $'$, this should not be misleading.

[24]Recall that we are ignoring the collective interpretation of *Jack, Meg and Fiona* in this paper.

The denotation of $ukid'$ on the other hand is different in such an alternative model. It would be the glb of the lattice for the set of kids in that model, that is, $(jack \otimes meg \otimes fiona \otimes bob \otimes emily)$ in the alternative model above. Also, in application, we do not need to identify the membership of the set of kids at the start of the inference.[25] For example, we can use a meta-notation such as $\forall kid$, to talk about the glb of the set of kids in any interpretation models. What $\forall kid$ actually is of course depends on the membership of the set of kids in each interpretation model.[26] Also, note that it is not a problem that two expressions in the logical language denote the same thing in the given interpretation model (e.g., two predicate expressions in FOL can denote the same set of individuals in some interpretation model), as long as their denotations are not the same in **every logically possible** interpretation model, which again is never the case between $ukid'$ and the complex term $(jack' \otimes meg' \otimes fiona')$.

We have argued above that the denotations of $ukid'$ and $(jack' \otimes meg' \otimes fiona')$ in our logical language are not the same since the former denotes the least upper bound (lub) of the positive complete lattice for the set of kids, whereas the latter denotes the 'meet' of the denotations of $jack'$, meg' and $fiona'$, i.e., the meet of the three atomic individuals $jack$, meg and $fiona = (jack \otimes meg \otimes fiona)$.[27] The denotations of the above two terms may coincide in some interpretation models, but if we consider their denotations in all the possible interpretation models, their interpretations are not the same.

That being said, there is still some similarity in the denotation of $ukid'$ and the denotation of $(jack' \otimes meg' \otimes fiona')$. The same applies to the denotation of $ekid'$ and the denotation of $(jack' \oplus meg' \otimes fiona')$. That is, the former two are similar in the sense that the lub that $ukid'$ denotes is formed in terms of the meet, $\otimes$, which is also in the denotation of $(jack' \otimes meg \otimes fiona')$. In the same way, the glb that $ekid'$ denotes is formed in terms of the join, $\oplus$, which is also used in the denotation of $(jack' \oplus meg' \oplus fiona')$. Now, there is some natural language data support for this account. Consider (6)$\sim$(7).

(6) a. Dare-mo-ga urusai.
Who-and-nom noisy.
"Everybody is noisy."

[25] We use a computational model in which new object terms are continually introduced into the knowledge representation. See Cassimatis, Murugesan, and Bignoli (2009b) for details.

[26] The same applies to the interpretation of the notation $\exists kid$ in different interpretation models, which is the lub of the lattice for the set of kids in the given interpretation model.

[27] Again, we overload the information with $\otimes$ here, using it both in the logical language and in (the meta-notation of) its denotation. The term $(jack' \otimes meg' \otimes fiona')$ in our logical language denotes the plural individual $(jack \otimes meg \otimes fiona)$ in our interpretation model.

b. Jack-mo Meg-mo Fiona-mo urusai.
Jack-and Meg-and Fiona-and noisy.
"Jack and Meg and Fiona are noisy."

(7) a. Dare-ka-ga namaikida.
Who-or-nom cheeky.
"Someone is cheeky."

b. Jack-ka Meg-ka Fiona-ga namaikida.
Jack-or Meg-or Fiona-nom cheeky.
"Jack or Meg or Fiona is cheeky."

In the Japanese sentence in (6-a), the universal-quantifier subject *dare-mo-ga* (i.e., *everyone*) can be decomposed into the morphemes *dare* (literally, '*who*'), *mo*, (which is a kind of *and*[28]), and *ga*, which is the 'nominative' case-marker that suggests that this NP is in the syntactic subject position. Now, note that the second morpheme, that is, '*mo*' is used in (6-b) as well, where it conjoins *Jack*, *Meg* and *Fiona* in such a way that the complex subject *Jack-mo Meg-mo Fiona-mo* can be translated into *Jack, Meg and Fiona* in English. As we indicated above, the suffix *mo* is not the word to which the English *and* is usually translated. It is most likely that the English sentence in (6-b) is translated to *Jack-to Meg-to Fiona-ga urusai.* However, since (6-a) has basically the same truth condition as this Japanese sentence and the suffix *mo* in (6-b) does play the role of conjoining *Jack*, *Meg* and *Fiona*,[29] we can treat *mo* as some sort of conjunction connective. Based on that, we can characterize this connective *mo* in terms of $\otimes$ (recall that $\otimes$ means 'AND' in some sense). Then the subject of (6-a) will be '*dare*-$\otimes$' and the subject in (6-b) will be '$jack \otimes meg \otimes fiona$'. Now, the question is what 'dare' (or 'who') means. Following a traditional account in Japanese linguistics (Kuroda, 1965), Kratzer (2005) assumes that *dare* is an 'indeterminate pronoun'. The possible denotations of pronouns of course range over the members of the domain of individuals and what 'indeterminate' means above is that the exact interpretation of the pronoun in question varies depending on where it appears in the syntactic structure. Thus, we can say that the interpretation of *dare* ranges over (atomic) individuals and the indeterminateness of *dare* together with the meet '$\otimes$' leads to the exhaustive coverage of all the atomic individuals in the domain in question, '$a_1, \ldots, a_n$,' as in the glb, '$(a_i \otimes ... \otimes a_n)$.' That is, if the domain of individuals is $\{jack, meg, fiona\}$, then the subject *dare-mo*, which we analyze in terms of '*pro*-$\otimes$' (where

[28]As we explain below, this is not the normal *and*.

[29]The suffix *mo* has additional pragmatic connotation similar to the connotation that the English word *too* has. But such non-truth conditional meaning is not crucial in the current discussion.

pro ranges over the atomic individuals), is cashed out as $(jack \otimes meg \otimes fiona)$. But this is exactly how we analyzed the semantics of *everybody* in the lattice-based semantics above. Again, when the domain is $\{jack, meg, fiona\}$, then $ubody'$ for *dare-mo* is co-denotational with $(jack' \otimes meg' \otimes fiona')$ for *Jack, Meg and Fiona* (in its distributive interpretation) in this account.

The data support is clearer in (7), in which the Japanese existential quantifier *Dare-ka* (literally, 'who-or') and the plural referential NP, *Jack-ka Meg-ka Fiona* (literally, 'Jack-or Meg-or Fiona') shares the disjunction connective, 'ka' (i.e., 'or'), which we can characterize in terms of the join connective '$\oplus$.' Again, if we assume that the semantics of 'dare' (i.e., 'who') ranges over all the individuals in the domain in question while the suffix '*ka* (i.e., 'or') denotes the join, '$\oplus$', then the entire NP, '*dare-ka*' will denote the lub of the domain of individuals. When the domain is $\{jack, meg, fiona\}$, then the denotation of *dare-ka* will be $(jack \oplus meg \oplus fiona)$, the same as the denotation of *Jack-ka Meg-ka Fiona* in (7-b). Again, this is exactly how we analyzed the semantics of *somebody* and *Jack, Meg or Fiona* in the lattice-based semantics in this section. Thus, we could say that the semantics of Japanese sentences as in (6)$\sim$(7) provides some data support for our analysis of the semantics of argument NPs.

As we indicated a few times in this section, we can exclude all the non-atomic individuals from our interpretation models. More specifically, in order to define the model theoretic truth-condition for each formula in the form of $Pred'(ukid')$ and $Pred'(ekid')$, we do not really need to include any complex individuals as in (5-a). That is, for any set A, whether any complex individual in the form of $(a \oplus b)$ or $(a \otimes b)$ is in A can be computed solely depending on the set memberships of its atomic-component individuals. Thus, we can include complex terms in the form of $(a'_1 \oplus ... \oplus a'_n)$ and $(a'_1 \otimes ... \otimes a'_n)$ only in our logical language and then interpret those complex terms in terms of the denotations of their atomic component terms a_1,...,a_n in the interpretation model in question. Similarly, the interpretations of 'negative' individuals such as $\neg jack$ and $\neg meg$ can be functionally computed from the interpretations of their atomic components, say, *jack* and *meg*. Thus, we can eliminate $\neg jack$ from the interpretation model and then interpret the term $\neg jack'$ in the logical language, say, in $Kid'(\neg jack')$, based on whether *jack* (which is the denotation of $jack'$) is in the set of kids in the interpretation model in question (e.g., in the above model, this formula is false since *jack* is indeed in the set of kids in that model). Actually, we do not even need these 'negative' terms in our logical language, since $Kid(\neg jack')$ is equivalent to $\neg Kid'(jack')$ with regard to the standard FOL interpretation models. Thus, all we need are I) for each predicate expression $Pred'_i$ in the logical language: $upred'$ and $epred'$, and II) for all the atomic terms $a_1, ..., a_n$ ($n \leq \infty$): the complex terms in the form of $(a_1 \oplus ... \oplus a_n)$ and $(a_1 \otimes ... \otimes a_n)$. As we see in Section 3, II) will generate all the terms such as $(jack' \otimes meg' \otimes fiona')$ and $(jack' \oplus meg' \oplus fiona')$[30].

2.2 Basic quantifier data analysis

In this subsection, we show our treatment of basic natural language quantifier data.

First, we sketch how we deal with reflexives and pronouns bound by quantifiers, since reflexive/pronominal binding is often taken to provide an extra motivation for using explicit quantifiers in the semantic representations. Consider (8).

(8) a. Every boy likes himself.
b. $\forall x(Boy'(x) \rightarrow Like'(x, x))$
c. Every boy$_1$ rides his$_1$ bike.
d. $\forall(Boy'(x) \rightarrow (Ride'(x, theBikeOfx)))$

FOL with quantifiers can simply put an additional variable in the argument position for the reflexive or the pronoun and have it bound by the explicit quantifier, as it is done in (8-b) for (8-a) and in (8-d) for (8-c).[31]

However, we can deal with reflexive binding and bound pronouns in our quantifier-free language without a major problem.

For reflexive binding as in *Every girl likes herself*, we follow the analyses in Szabolcsi (1992) and in Carpenter (1997) and generate a predicate term '$Self\text{-}like'$' for a complex expression *like herself*, which denotes the set of individuals that like themselves.[32] This predicate term can then take our singular term '*uboy*' for *every boy* as an argument, as shown in (9).[33]

[30] Some adequate 'structural' properties would be presupposed here , such as $(jack' \otimes meg') = (meg' \otimes jack')$ (i.e., the complex terms are commutative and associative with regard to $\otimes$ and $\oplus$. Also, many of the complex terms that can be formally generated will not be used in application, but that often happens with the use of a productive logical system.

[31] We ignore the detailed semantics of *his bike*, which is commonly analyzed as a definite description. The definiteness effect is not important for the current discussion.

[32] We use English-like expressions in our logical language only for the convenience of helping the reader's intuitive understanding of the logical language. If we used a typed-lambda expression, then this particular VP functor would be represented as '$\lambda x.like'xx$' with its usual model theoretic interpretation.

[33] Instead of this analysis, we could add an index to each occurrence of a quantifier term and represent the semantics in (9-a) as '$Like'(uboy_1, uboy_1)$,' which would be interpreted in the same way as '$Self\text{-}like'(uboy')$' is interpreted in the current analysis. The semantics of *Every boy likes every boy* would then be

(9) a. $Self\text{-}like'(uboy')$ for *Every boy likes himself.*

b. $Self\text{-}like'(uboy') \Rightarrow Self\text{-}like'(boy_i) \Rightarrow Self\text{-}like'(eboy')$
where boy_i can be any atomic term that denotes an atomic member of the set of boys in the given interpretation model, e.g., $jack'$, bob', etc.

Remember the intended interpretation of $uboy'$ from the previous section, which is the glb of the complete lattice for the set of boys. When this object is a member of the individuals who like themselves, then it follows that any atomic individual who is a member of the set of boys, say, $jack$, also likes himself in this model. This in turn entails that the lub of the set of boys, i.e., the denotation of $eboy'$, is a member of the set of self-liking individuals in this interpretation model, as the entailment relation in (9-b) suggests.

Our treatment of pronominal binding is basically the same. For example, consider (10)

(10) a. Every boy_1 said that he_1 likes logic.

b. A boy_1 said that he_1 likes logic.

c. $Pred_j(uboy') \rightarrow Pred_j(boy_i) \rightarrow Pred_j(eboy')$
where '$[\![Pred_j]\!]$' is the set of individuals that like logic, and boy_i is as in (9-b) above.

Dealing with the pronominal binding in (10-a), we first generate a predicate expression, '$Pred_j$,' which denotes $\{x \in \mathbb{D} \mid\ x \text{ likes logic }\}$ and then attribute this predicate to the term '$uboy'$' for *every boy*. Here, we have ignored the semantics of *say (that)* in (10-a). For (10-b), we attribute the same predicate to the term '$eboy'$' for *a boy*. The entailment relation between (10-a) and (10-b) is then captured in terms of the entailment relation that we explained at (5) in Section 2.1, as shown in (10-c).

We do not show how we generate the set-denoting term '$Pred_j$' for the complex English expression *say that he₁ likes logic* in this paper (and again, we have ignored the semantics of *say (that)* in the above analysis), but in the Cognitive Substrate theory (cf. Cassimatis, Murugesan, and Bugajska, 2008), we can create a new set-denoting term on the fly throughout the inference.[34]

represented as '$Like'(uboy_1, uboy_2)$,' which would be interpreted in the same way that '$Like'(uboy', uboy')$' is interpreted in the current analysis. Though this alternative analysis has some merits, such as a more straightforward relation between the phonological string and the semantic representation especially for pronominal binding as we discuss below, distinguishing each occurrence of a singular term adds extra complexity to the theory. We leave this choice as an open question.

[34]In this AI framework, we also continually introduce new object denoting terms as inference proceeds. Because of a particular Artificial Intelligence (AI) algorithm sketched in Cassimatis, Murugesan, and

We do not discuss empirical merits and demerits of our treatment of reflexive and pronominal binding in comparison to alternative accounts based on FOL in this paper. However, we point out that it is not clear if the semantics of the reflexive pronoun in (8-a) can be correctly captured in terms of a variable x as in (8-b). In FOL, when a variable such as x appears outside the scope of an explicit quantifier, it is 'free' and is interpreted as the individual that is assigned by a variable assignment function. For example, the formula $Run'(x)$ can be interpreted as "Bob runs" when the given assignment function maps x to the individual Bob. However, a reflexive pronoun can never be interpreted in this way[35] and this begs a question with regard to whether the lexical semantics of a reflexive can be correctly captured in terms of a free variable.[36] Also, the characteristic locality constraint on reflexive binding (which does NOT affect pronominal binding) needs to be explained somehow, which is at least not clear from the semantics of a reflexive as a variable. Finally, as we indicated before and as Jacobson (1999) and Jacobson (2008) explained in details, if we assume that the lexical semantics of a pronoun is simply a free variable, say, x, as used in a FOL language, then the local interpretation of a bound pronoun (which is whatever individual that a variable assignment function maps it into) and the global interpretation of the bound pronoun (after the quantifier that binds that pronoun is added) are different, going against at least the spirit of the semantic compositionality from the lexical level.

In our account, both for reflexive binding and pronominal binding, we first derive a predicate for a possibly complex natural language expression. Thus, different locality constraints on reflexive binding and pronominal binding can be assigned when we derive such predicate expressions for the two kinds of bound expressions. Also, as a result of this analysis, a free variable such as x is never inserted into the position for the reflexive or the bound pronoun at any stage of the syntactic derivation and thus, the compositionality problem as indicated above does not arise in our account. For an analysis of the different localities of reflexive binding and pronominal binding along this line and also for a compositional semantic analysis of reflexive and pronominal binding, see Uchida (2008).

Next, we show how we deal with scope ambiguity in natural language data. First,

---

Bignoli (2009a), this continual introduction of new terms does not lead to a fatal scaling problem. We think that the ability to continually introduce new objects into inference is an important characteristic of human reasoning. See Cassimatis, Murugesan, and Bignoli (2009b) for relevant discussions.

[35] Unless it is a logophor, which we assume is different from a proper reflexive.

[36] Of course, using a first-order language with explicit quantifiers to represent the semantics of natural language sentences does not force the researcher to assume that the lexical semantics of a reflexive or a pronoun is a (at least locally) free variable. However, if we do not insert a free variable in the argument position that a reflexive or a bound pronoun occupies, it is less clear whether the first-order formulas as in (8-b) and (8-d) have additional merits in comparison to our analysis.

remember the intended interpretations of '$(a \otimes b)$' and '$(a \oplus b)$' from section 2. For any set A: $(a \otimes b)$ means that both a and b are members of A. $(a \oplus b)$ means that at least one of a, b is in A. Thus, when $(a \otimes b)$ is in A, it follows that both a and b are in A, which in turn entails that $(a \oplus b)$ is in A. Now, although we have discussed such entailment relation only in terms of sets of objects in the previous section, this entailment relation actually holds for any set in the first-order set-based interpretation models, that is, for any set of n-tuples of objects, as long as we put the terms such as $(a \otimes b)$ and $(a \oplus b)$ in the same i-th position in the n-tuples where all the other positions in the n-tuples stay the same through the replacement in the i-th position. That is, (11) follows from the intended lattice-theoretic interpretation of $\otimes$ and $\oplus$.

(11) a. For any set X of k-tuples of individuals $(2 \leq k)$, and for any member of X, such as $\langle u_1, ..., u_k \rangle$ and for any complex objects $(a_1 \otimes ... \otimes a_n)$ made out of n atomic objects $a_1, ..., a_n$ $(n \leq \infty)$:
If the k-tuple $\langle u_1, ..., u_k \rangle$ with $(a_1 \otimes ... \otimes a_n)$ being u_i $(1 \leq i \leq k)$ is in X, then the same k-tuple except for u_i being replaced by any of the atoms a_j $(1 \leq j \leq n)$ is also in X.

b. For any set X of k-tuples of individuals $(2 \leq k)$, and for any member of X, such as $\langle u_1, ..., u_k \rangle$ and for any complex objects $(a_1 \otimes ... \otimes a_n)$ made out of n atomic objects $a_1, ..., a_n$ $(n \leq \infty)$:
If the k-tuple, $\langle u_1, ..., u_k \rangle$ with a_j $(1 \leq j \leq n)$ being u_i $(1 \leq i \leq k)$ is in X, then the same k-tuple except for u_i being replaced by $(a_1 \oplus ... \oplus a_n)$ is in X.

Although the precise theorems look complex, their contents are actually simple. Consider the three-place predicate $Introduce'$, which denotes a set of triples, '$\langle a, b, c \rangle$,' such that a introduces b to c. That is, $[\![Introduce']\!] = \{\langle a, b, c \rangle \mid a \text{ introduces } b \text{ to } c\}$, where a, b, c are atomic or complex model theoretic individuals that we have discussed. Now suppose that $\langle (jack \otimes meg), fiona, kieth \rangle$ is a member of this set, that is, $(jack \otimes meg)$ introduces $fiona$ to $kieth$ (i.e., both $jack$ and meg introduce $fiona$ to $kieth$). Then what (11-a) means is that both $\langle jack, fiona, kieth \rangle$ and $\langle meg, jack, kieth \rangle$ are members of this set, that is, $jack$ introduces $fiona$ to $kieth$ and meg introduces $fiona$ to $kieth$. (11-a) is the most general description of such entailment relations. Similarly, about the same set of triples that $Introduce'$ denotes (in some model), suppose that $\langle jack, fiona, kieth \rangle$ is a member of this set, that is, $jack$ introduces $fiona$ to $kieth$. Then, what (11-b) means is that it follows from there that $\langle (jack \oplus meg), fiona, kieth \rangle$ is also a member of this set, that is, at least one of $jack, meg$ introduces $fiona$ to $kieth$.

It should be clear from the above explanation why (11) follow from the interpretation of the complete lattice at (5) in the previous section. For example, for any set of triples $\langle a, b, c\rangle$, we can fix any two of a, b, c, say, b and c, to particular individuals, say, $fiona$ and $kieth$ and form a subset of the original set of triples. That is, the set of triples such that the second member of the triple is $fiona$ and the third member is $kieth$. This set will look like $\{\langle a_1, fiona, kieth\rangle, ..., \langle a_k, fiona, kieth\rangle\}$, where $a_1, ..., a_k$ range over all the individuals who introduce $fiona$ to $kieth$. Now, it is clear that this subset of triples is isomorphic to the set of individuals who introduce $fiona$ to $kieth$, that is, $\{d \mid \langle d, fiona, kieth\rangle \in [\![Introduce']\!]\}$ (where d ranges over (simple/complex) individuals while $[\![Introduce']\!]$ is as above). Thus, given the intended interpretation of $\otimes$ (i.e., for ANY set A of individuals, if $(jack \otimes meg)$ is in A, it follows that $jack$ is in A and meg is in A), it must be the case that both $\langle jack, fiona, kieth\rangle$ and $\langle meg, fiona, kieth\rangle$ are members of $[\![Introduce']\!]$.

Similarly, if $jack$ is a member of the set of individuals who introduce $fiona$ to $kieth$, then given the intended interpretation of $\oplus$, it must be the case that $(jack \oplus meg)$ is a member of that set. Thus, the corresponding set of triples $[\![Introduce']\!]$ must contain $\langle (jack \oplus meg), fiona, kieth\rangle$ as a member.

Given (11), which again follows from the intended interpretations of $\otimes$ and $\oplus$, it should be clear that whenever $\langle ..., (a_1 \otimes ... \otimes a_n), ...\rangle$ is in a set of k-tuples, then it follows that $\langle ..., \langle (a_1 \oplus ... \oplus a_n), ...\rangle$ is also a member of that set (where '...' in $\langle ..., X, ...\rangle$ stay the same and the position X can be anywhere in the k-tuple).

Finally, since $ukid'$ (for *every kid*) denotes the glb of the set of kids, that is, $(a_1 \otimes ... \otimes a_n)$ with all a_i $(1 \leq i \leq n)$ being atomic kids, and since $ekid'$ (for *some/a kid*) denotes the lub of the same set, that is, $(a_1 \oplus ... \oplus a_n)$ with the same a_i $(1 \leq i \leq n)$, with (11) above, it follows that whenever $\langle ..., [\![ukid']\!], ...\rangle$ is a member of a set of k-tuples, $\langle ..., [\![ekid']\!], ...\rangle$ is also a member of the same set of k-tuples.[37]

Things get a bit complex here. (12) follows from (11).

(12) a. For every binary relation R and for any two members $x, y \in \mathbb{QL}$, if $x \leq y$, then '$R(.., x) \subseteq R(.., y)$.'

b. For every binary relation R and for any two members $x, y \in \mathbb{QL}$, if $x \leq y$, then '$R(x, ..) \subseteq R(y, ..)$.'

[37] Recall that $[\![ukid']\!]^{M_i} = (a_1 \otimes ... \otimes a_n)$ and $[\![ekid']\!]^{M_i} = (a_1 \oplus ... \oplus a_n)$ when the set of atomic kids in M_i is $\{a_1, ..., a_n\}$.

$\mathbb{QL}$ represents a complete lattice such as the one we explained at (5) in section 2.1. Thus, x, y ranges over atomic/complex individuals, such as $(jack \otimes meg)$, $jack$ and $(jack \oplus meg)$. Recall that $\forall a_1, ..., a_n \in \mathbb{D} : (a_1 \otimes ... \otimes a_n) \leq a_i \leq (a_1 \oplus ... \oplus a_n)$, where $2 \leq n \leq \infty$ and $1 \leq i \leq n$ (i.e., a_i represents any one of the atomic objects $a_1 \sim a_n$). For example, $x \leq y$ in (12) can be cashed out as $(jack \otimes meg) \leq meg$, $meg \leq (jack \oplus meg)$, etc.

Now, (12-a) means that if the individual y is higher than the individual x in the partial order that we explained at (5), then the set that we get by taking x as the second argument of the relation R is a subset of the set that we get by taking y as the second argument of the relation R. We used '..' to indicate the 'unsaturated' position of the relation.[38]

As an example, (12-a) means that the set of individuals that love every girl is a subset of the set of individuals that love at least one girl. Note that each member of the latter set does not need to love the same girl; the set just requires each member x to love at least one girl individual. To see that (12-a) does follow from (11), first, note that each two place relation R can be seen as the corresponding set of ordered pairs of individuals, i.e., $\{\langle a, b\rangle \mid R(a, b)\}$. Now, according to (11), for any individual, say, Bob, 'Bob loves every girl' entails 'Bob loves at least one girl', since (11) says that whenever $\langle bob, [\![ugirl']\!]\rangle$ is a member of $[\![Love]\!]$ ($= \{\langle a, b\rangle \mid a \text{ loves } b\}$), $\langle bob, [\![egirl']\!]\rangle$ is also a member of $[\![Love]\!]$. That is, according to (11), $\{a \mid \langle a, [\![ugirl']\!]\rangle \in [\![Love']\!]\} \subseteq \{a \mid \langle a, [\![egirl']\!]\rangle \in [\![Love']\!]\}$.[39] It is easy to see that this holds true for any two-place relation. That is, according to (11), for any set X of ordered pairs of individuals, $\{a \mid \langle a, [\![ugirl']\!]\rangle \in X\} \subseteq \{a \mid \langle a, [\![egirl']\!]\rangle \in X\}$. Finally, according to (11), '$[\![ugirl']\!]$' and '$[\![egirl']\!]$' above can be any two members of a complete lattice such as the lattice in (5) that enter into the partial order $\leq$. That is, according to (11), $\forall x, y \in \mathbb{QL}$ such that $x \leq y$ and for all set X of ordered pairs of individuals, $\{a \mid \langle a, x\rangle \in X\} \subseteq \{a \mid \langle a, y\rangle \in X\}$, which is the same as (12-a). Thus, (12-a) follows from (11).

In (12-b), we leave the second individual that the relation R relates unspecified, as '..' indicates. As an example, (12-b) means that the set of individuals that every boy loves is a subset of the set of individuals that at least one boy loves. The proof from (11) to (12-b) is basically the same as the proof from (11) to (12-a).

Actually, (12) holds in terms of n-ary relation when we put $x \leq y$ in one of the n-positions such as $R_n(..., x, ...) \subseteq R_n(..., y, ...)$, but (12) is more convenient for explaining

[38]Recall that the AI architecture that contains the proposed logical language can create a new set-denoting term on the fly during the inference, which will make it possible to allocate a new predicate term to the set of individuals that we generate from a two-place relation in the above manner.

[39]Again, recall that $[\![ugirl']\!] = a_1 \otimes ... \otimes a_n$ and $[\![egirl']\!] = a_1 \oplus ... \oplus a_n$, when $a_1, ..., a_n$ are atomic girls.

out treatment of scope ambiguity.[40]

Given (12), consider the sentence *Every boy loves a girl.* Interpreting this sentence, we may use (12-a) first and generate the set of individuals that love at least one girl, which we represent as '$L(.., \exists girl)$.'[41] Again, this is a particular set of individuals (which is a proper super-set of the set represented as '$L(.., \forall \text{girl})$,' for example). Now, let X represent $L(.., \exists girl)$. Then, since X is a set of individuals, '$\forall boy$' can be a member of X. Note that given (5) and given the order '$\forall boy \leq \exists boy$,' it follows that '$\forall boy \in X \Rightarrow \exists boy \in X$.' If '$\forall boy \in X$' is as above, then this corresponds to the narrow scope interpretation of the indefinite with *Every boy loves a girl*, that is, each boy loves a possibly different girl.

Instead of (12-a), we can use (12-b) and generate the set '$L(\forall boy, ..)$' first (which again, is a subset of another set '$L(\exists boy, ..)$'). Again, let Y represent $L(\forall boy, ..)$. Then, since Y is a set of individuals, '$\exists girl$' can be a member of this set, as in $\exists girl \in Y$. If it is, this corresponds to the wide scope interpretation of the indefinite with *Every boy loves a girl*, that is, every boy loves the same girl. Note again that '$\forall girl \in Y \Rightarrow \exists girl \in Y$' in our account (at least in cases when there exists at least one atomic girl individual in the model).

At the level of logical language expressions, these two interpretation model structures can be specified in terms of the dependency predicate $DependOn'$ as in (13).

(13) *A girl likes every boy* (scope: '$a > every$' or '$every > a$').

a. Indefinite narrow scope: $Like'(egirl', uboy') \wedge DependOn'(egirl', uboy')$

b. Indefinite wide scope: $Like'(egirl', uboy') \wedge DependOn'(uboy', egirl')$

(13-a) corresponds to the model structure with '$\forall boy \in L(\exists girl, ..)$,' where $L = [\![Like']\!]$. (13-b) corresponds to the model structure with '$\exists girl \in L(.., \forall boy)$.' Note that (13-a) is true in both the structures since the indefinite wide scope interpretation logically entails the indefinite narrow scope interpretation.

All the scope relations that FOL can represent are linear, as in (14).

(14) $\exists x \forall y \exists z \forall u \exists v \phi$

An existential quantifier such as $\exists x$ takes narrow scope with regard to all the universal

[40] Besides, such a general theorem would be an equivalent restatement of (11) in terms of n-place relations (instead of sets of n-tuples).

[41] Recall that the meta-notation '$\exists girl$' represents the lub of the lattice for the set of girls. That is, when $a_1, ..., a_n$ are atomic girls, $\exists girl = (a_1 \oplus ... \oplus a_n)$.

quantifiers to the left and takes wide scope with regard to all the universal quantifiers to the right according to the linear, left-to-right order as in (14) (N.B. there is no scope interaction between two existential quantifiers in FOL; neither is there any between two universal quantifiers). Our language can express any of these relations as in (13), and hence it is clear that all the scope relations expressible in FOL are expressible in our language. On the other hand, given (13), notice that not including the formula $DependOn(egirl', uboy')$ in the formula does **not necessarily** mean that $egirl'$ takes wide scope over $uboy'$. In order to encode the existential wide scope, the formula $DependOn'(uboy', egirl')$ needs to be included. What this means is that our language gives us a possibility to state that the scope of $egirl'$ and the scope of $uboy'$ are independent from one another, giving us the same sort of expressive power that a monadic second-order language has, as we see below.[42] On the other hand, the way of interpreting multiple quantifiers that we explained at (12) above restricts us to the linear-scope interpretations that are expressible in standard FOL. Thus, although the interpretation procedures we explained at (12) is not the only logically possible way to interpret multiple quantifiers, at the moment, we stick to it and hence we can ignore non-linear scope dependency readings in this paper.

Alternatively to the formulation at (13), we could leave the logical representation underspecified as '$Like'(egirl', uboy')$' and make this uniform logical formula can be interpreted by way of either of the two model structures that we sketched above. This would amount to saying that the formula '$Like'(egirl', uboy')$' is ambiguous between the two interpretations. This alternative in a sense would be more restrictive since, as we indicated above, using a predicate such as '$DependOn'$' that relates singular quantifier terms gives our language a similar potential expressive power as Dependence Logic (cf. Väänänen, 2007) and IF logic (Hintikka and Sandu, 1997). However, maintaining truth conditional ambiguity at the level of the logical language representation is not unproblematic especially because a main point of our proposal is to provide a matching model structure to our novel first-order language without explicit quantifiers. Thus, at the moment, we stick to the analysis as in (13).

Our data analysis in this section indicates that our computational mechanism can spontaneously compute a set such as 'the set of individuals that like every girl', or an even more complex set such as 'the set of odd numbers that is larger than the largest prime', as long as the interpretation model is fixed. That is, in each interpretation model with a

[42]Note that this is not possible with FOL since all the quantifiers are linearized left-to-right in FOL and there are only two scope relations between $\forall x$ and $\exists y$, that is, either $\forall x \exists y \phi$ (existential narrow scope) or $\exists y \forall x \phi$ (existential wide scope), in FOL.

specific finite domain of individuals $\mathbb{D}$, both of the above examples simply denote some subset of $\mathbb{D}$. For example, if $\mathbb{D}$ contains only single digit natural numbers, $\{n : 1 \leq n \leq 9\}$, then the set of odd numbers that is larger than the largest prime is $\{9\}$ in this particular model. It is easy to see that we can compute such a set for each domain that contains only a finite set of natural numbers. The same applies to sets of non-numerical objects such as 'the set of individuals that like every girl.' That is, when the domain of individuals is fixed and both the set of girls and the set of ordered pairs of individuals that the predicate $Like'$ denotes are fixed, then it is easy to calculate the set of individuals that like every girl in that particular interpretation model.

Now, at this point, there are two possible options with regard to the further specifics of our logical language. One possibility is to spontaneously generate a new predicate symbol that corresponds to the set of individuals that like every girl, for example, which will lead to the indefinite wide scope reading of the sentence *A boy likes every girl*. The merit of such a formal logical language is that at each stage of computation, there is stronger correspondence between the logical language expressions and their denotations in the model theoretic semantic structures. On the other hand, its demerit is that creating new predicate symbols on the fly in this manner compromises the bottom-up compositionality from the lexical semantics. An alternative option is to stick to the expressions as in (13) at the level of the logical language expressions. With this option, we do not generate any constituent predicate expression that corresponds to the set of individuals that every girl likes for (13-a) and similarly, our language will not have any constituent predicate expression for the set of individuals that like every boy for (13-b). Instead, these sets are constructed in the given interpretation model only for evaluating the truth-conditions of the logical formulas in (13-a) and (13-b) in a systematic manner. An example of a formal logical language that we present in Section 3 is compatible with either of these two options and we leave further specification of our logical language in this regard for future research.

This section has explained complete lattice structures with which we can interpret our first-order language in which singular QNPs such as *every boy* and *a girl* are treated as arguments of predicates, rather than as explicit quantifiers. We have also shown how we account for basic quantifier data with our quantifier-free language and model structures. We have left our analysis of plural QNPs such as *all (the) boys*, *most boys* and *few boys* for another paper. The treatment of *all (the) boys* will be especially interesting since the scope ambiguity possibility can be influenced by whether we have *every boy* or *all boys* in the sentence.[43] Given our treatment of singular QNPs, our first-order language would treat these plural QNPs as argument terms of predicates as well, as in $Noisy'(mboys')$ for *Most boys smoke*. Roughly, when the set of kids is $\{jack, meg, fiona\}$, then $Noisy'(mkids')$ will be true if and only if $(jack \otimes meg) \oplus (jack \otimes fiona) \oplus (meg \otimes fiona)$ is a member of $[\![Noisy']\!]$ (i.e., the set of noisy individuals).[44] However, in this account, as the cardinality of the set of atomic kids increases, the complexity of the denotation of $mkids'$ will massively increase (basically, when $|[\![Kid']\!]| = n$, there are $C(n, n/2 + 1)$ subcomponents connected with $\oplus$ in the complex object that $mkids'$ denotes). We think that at least this level of complexity is involved in any other account of *most kids* in an alternative logical language, but substantiating this speculation will require more work. In contrast, in the case of *all boys*, it would be relatively straightforward since it is a matter of how to add the collective reading possibilities as in *All the kids met up at the station*, to our treatment of *every kid*. We leave further treatment of these plural quantifiers for future research.

In section 3, we present a logical language that instantiates the idea that we have sketched so far. In section 4, we interpret that language in terms of both the interpretation models including the lattice structures as above and the standard FOL interpretation models (i.e., excluding the lattice structures above).

3 PL^-

This section provides the definition of a first-order language, PL^-, which instantiates our proposal in the previous sections. In this paper, we ignore extra complications such as tense, aspect and events.

First, Sig is the signature of this language, that is, the set of the atomic symbols that appear in our logical language, which is the union of the component sets as in (15)

(15) $Sig := Ind \mid Cat \mid Pred \mid F \mid \{\bot\} \mid \{\neg, \wedge, \vee, \rightarrow\} \mid \{(,)\}$

(16) shows what the component sets in (15) contain.

(16) a. Ind: a set of atomic individual constants.
b. Cat: a set of category-object constants.

[43] Thanks to Mark Steedman by reminding me of this data observation.
[44] The idea here is that either $(jack \otimes meg)$ OR $(jack \otimes fiona)$ OR $(meg \otimes fiona)$ constitutes the majority of kids. Note that including $(jack \otimes meg \otimes fiona)$ as the fourth alternative in this complex object is redundant, since any one of the other three alternatives 'entails' this fourth possibility.

c. $Pred$: a set of n-ary predicate symbols

d. F: a set of n-ary function symbols.

e. $\{\bot\}$: Falsum

f. $\{\neg, \wedge, \vee, \rightarrow\}$

g. $\{(,),\text{-}\}$

(17) shows some examples of (16) using intuitive notations.

(17) a. $Ind = \{jack', meg', fiona', ...\}$

b. $Cat = \{kid', boy', girl', logician'...\}$

c. $R = \{IsA', Run', Like', Introduce' \ldots\}$

d. $F = \{u', e', I', f_i'\}$ $\quad (i \in \mathbb{N}.)$

We use English expressions in the notations for helping the reader to grasp our logical language in terms of its application in natural language semantics but this notational choice is arbitrary. Note also that the elements of Sig are atomic and do not have any internal structures.

Our individual terms are sorted at least in two sorts, the proper individual sort which characterizes the semantics of *Meg*, *that girl* and *every girl* etc., and the 'category-object' sort which characterizes the semantics of nominal restrictions in natural language. Ind in (16-a) and (17-a) contains the atomic elements for the former sort whereas Cat in (16-b) and (17-b) contains the atomic elements for the latter. That is, Ind is the set of constant terms that denote 'atomic individuals' such as $jack$ and meg in Section 2.1. This set is the same as the set of individual constant terms in standard FOL. In contrast, Cat deviates from standard FOL; it is the set of constant terms that we use to characterize the semantics of nominal restrictions in natural language, such as *boy* and *girl* in English. As we indicated above, Cat only contains the atomic 'category-object-denoting' expressions which we expand to the set K, which contains both the atomic and complex category-object-denoting expressions, as we show later at (19). K inherits the sort for Cat.

$Pred$ in (16-c) and (17-c) is the set of predicate expressions, which denote either a set of individuals or a set of ordered n-tuples of individuals, as in standard FOL. We have an arity function that maps each symbol in $Pred$ to the number of argument terms that it requires in order to form a propositional formula, but we omit this detail in this paper. The expressions in $Pred$ are sort sensitive though most predicate expressions that we use in this paper require all of their argument terms to belong to the proper individual sort, such as $jack'$ and meg'. As we see below, 'IsA'' is exceptional in this regard. Since we treat the semantics of nominal restriction expressions in natural language with category-object terms in the set K, $Pred$ is mostly the set of expressions for verb phrases.

F in (16-d) and (17-d) is the set of 'functor symbols.' Functor symbols also require arguments of particular sorts, as we show below.

Next, we define the expressions of PL^{-} using the atomic elements in Sig in (15). Firstly, as we indicated above, we define 'K' in (18), which is a complete set of category-object-denoting terms.

(18) K is the minimal set that contains (18-a)∼(18-c).

a. $\forall y \in Cat : y \in K$.

b. $\forall p, q \in K$: p-$q \in K$
E.g., *nice′-boy′*, *mean′-male′-doctor′*, etc.

c. $\forall p \in K$: $f_i(p) \in K$
E.g., $alleged'(criminal')$, $fake'(diamond')$, etc.

The axiom in (18-b) defines the set of category-object terms which characterize the semantics of nominal restrictions that contain intersective adjectives and relative clause modifiers, such as *nice boy* and *boy who is nice*.

The axiom in (18-c) generates terms which characterize the semantics of nominal restrictions that contain non-intersective adjectives, such as *alleged criminal* and *fake diamond*. We analyze the semantics of the modifiers such as *alleged* and *fake* in terms of functors f_i, which are members of F in (16-d) above. Obviously, at (18-c), we assume that each functor f_i requires the category-object sort.

The axioms in (18) might be too generous in that they generate many expressions that are not useful in linguistic application. To avoid this profusion of expressions, we could use different sets of object terms for different kinds of expressions that appear in the nominal restrictions of NPs, such as adjectives and adverbs that modify such adjectives. However, introducing too much complexity into the logical language can lead to the incompleteness of the logical inference system. Also, we can filter out many of the formally possible logical expressions at the level of application by way of restrictive lexical assignments to natural language expressions as well as an adequately restrictive natural language syntactic theory that pairs phonological expressions with their semantics (which we represent by using PL^{-} as meta-language).

Whether we assign such additional restrictions logically or linguistically, it is clear that the rules in (18) can generate an infinite set of category-object terms from a finite number of atomic category-object terms in Cat. However, as we indicated in footnote 34, with our AI algorithm, this property of our logical language does not lead to a fatal scaling problem in computational application. But we do not go into the details of computational implementation in this paper. Again, see Cassimatis, Murugesan, and Bignoli (2009a) in this regard.

Next, we show how we generate the set of expressions $TERM$. $TERM$ is the minimal set that contains (19-a)∼(19-d).

(19) $TERM$:

a. $\forall x \in Ind : x \in TERM$.

b. $\forall x_1, ..., x_n \in Ind : (x_1 \oplus ... \oplus x_n) \in TERM$ and $(x_1 \otimes ... \otimes x_n) \in TERM$ (where $2 \leq n \leq \infty$).

c. $\forall p \in K$: $u'(p), e'(p) \in TERM$
E.g., $u'(boy'), e'(nice'\text{-}boy')$ etc.

d. $\forall t \in TERM : \neg t \in TERM$.
E.g., $\neg jack', \neg(jack \otimes meg), \neg e'(girl'), \neg u'(alleged'(criminal')), etc.$

$TERM$ is the set of argument expressions, i.e., 'terms', in our logical language. In linguistic application, $TERM$ contains the expressions that we use to represent the semantics of argument NPs such as *Bob*, *that boy* and *every boy* in natural language. First of all, all the atomic individual terms in Ind are members of $TERM$. Moreover, as we have proposed in the previous sections, our language has argument terms for quantificational NPs as well, which we explain next.

As we explained for (16-d)∼(17-d) above, F contains the set of functor symbols with which we can compose 'complex members' of $TERM$, as shown in (19-c). In this paper, what is important is 'u'' for creating universal quantifier individuals such as '$u'(boy')$' for *every boy* and 'e'' for creating existential quantifier individuals such as '$e'(girl')$' for *a girl*. Here we add additional parentheses in these complex terms to make their internal structures clearer. In the notations that we used in the previous sections, $u'(boy')$ is $uboy'$ and $e'(girl')$ is $egirl'$. We explain how the logical inference works with these quantifier terms at (23) below.

F contains some other functor expressions such as I', which denotes a function from atomic individuals to atomic individuals. This function is basically the same as the '$< e, e >$' function expression that Jacobson (1999) uses to account for the semantics of pronouns. The basic idea is that *he* in *Bob said that he will run* can be analyzed in terms of $I'(bob') = bob'$, where the antecedent NP *Bob* encodes bob', which can then be taken in as the argument of the functor I'. The binding of this pronoun by a quantifier in *Every student said that he will attend* can be analyzed in a similar manner. We omit further details.

F in (17-d) also contains f_i', which range over the functor terms with which we characterize 'non-intersective adjectives' in natural language, such as the functor $alleged'$ for *alleged* in English.

As we have indicated before, we do not really need the 'negative' terms in (19-d) above, since (19-a)∼(19-c) together with the propositions in the form of '$\phi \to \perp$' can cover all the necessary natural language semantic data. But for the purpose of showing the correspondence between the set of terms in PL^- and the pair of positive and negative lattice structures that we explained in 2.1, we include (19-d) in PC^-. We also assume that superfluous terms such as $\neg\neg\neg jack'$ is filtered out in practical application via logical equivalence, i.e., this one is equivalent to $\neg jack'$.

The set of formulas $FORM$ is the minimal set that contains (20-a)∼(20-c).

(20) $FORM$:

a. $\forall R^n \in Pred, \forall t_1, ..., t_n \in TERM,\ R^n(t_1, ..., t_n) \in FORM$

b. $\perp \in FORM$

c. $\forall \phi, \psi \in FORM$: $(\phi \star \psi) \in FORM$
where $\star \in \{\wedge, \vee, \to, \leftrightarrow\}$.

The way to compose atomic formulas out of individual terms and predicate expressions and the way to compose formulas out of atomic formulas are both the same as with standard FOL, except for the lack of explicit quantifiers in our language. In (20-a), we have added n as superscript, which indicates how many argument terms the predicate expression requires. Note that n here is NOT the exponent (R^n in (20-a) is an unstructured meta-variable in any case). Some examples of the 'atomic formula expressions' that we can generate with (20-a) are $Run'(bob')$, $Like'(e'(boy'), u'(girl'))$ and $DependOn'(e'(boy'), u'(girl'))$. '$\perp$' in (20-b) is also an atomic formula, which means Falsum. We can then generate complex formulas such as $Run'(bob') \wedge Like'(e'(boy'), u'(girl'))$ and $Run'(bob') \to \perp$ with (20-a)∼(20-c).

Finally, we explain how we generate 'quantifier argument terms' in our language. First, 'IsA'' denotes a relation between the denotations of terms (i.e., members of the set $TERM$)

and the denotations of 'nominal restriction terms' (i.e., members of K), as shown in (21).

(21) A sorted relation $[\![IsA']\!] \subset [\![TERM]\!] \times [\![K]\!]$:
E.g., $IsA'(bob', boy')$, $IsA'(meg', girl')$, etc.

In terms of the interpretation models that we explained in Section 2.1, '$IsA'(bob', boy')$' means that Bob is a boy in the given model M (i.e., $bob \in [\![Boy']\!]^M$) and '$IsA(meg', girl')$' means that Meg is a girl in the given model M (i.e., $meg \in [\![Girl']\!]^M$). With this relation expression'IsA',' we can provide restrictions to the category-object terms that we have defined in (18-b), as shown in (22).

(22) $\forall a, b \in K$, $\forall x \in TERM$: $IsA'(x, a\text{-}b)$ iff $IsA'(x, a) \wedge IsA'(x, b)$.

For example, $IsA'(meg', english'\text{-}girl')$ iff $IsA'(meg', english') \wedge IsA'(meg', girl')$.[45]

We now provide the essential inference rules for singular quantifier individuals such as '$u'(boy')$' and '$e'(girl')$' in (23).

(23) a. $\forall t \in Ind, \forall p \in K$: $IsA'(t, p) \vdash \phi[u'(p)] \to \phi[t]$.
b. $\forall t \in Ind, \forall p \in K$: $IsA'(t, p) \vdash \phi[t] \to \phi[e'(p)]$.

The notation $\phi[u'(p)]$ means that the term $u'(p)$ occupies a distinguished position inside the formula ϕ. In the right-hand side of $\to$ in (23-a), the term t replaces $u'(p)$ in exactly the same position inside ϕ. For example, suppose '$IsA'(bob', boy')$' holds. Suppose also that '$Smoke'(u'(boy'))$' holds. Then with (23-a), '$Smoke'(bob')$' must also hold. In the same way, given (23-b), if both '$IsA(meg', girl')$' and '$Run'(meg')$' hold, then '$Run'(e'(girl'))$' must also hold.

Corresponding to the dual lattice that we saw in (5b), the placement of the 'negative quantifier terms' is the opposite in comparison to the placement of the quantifier terms in (23), as shown in (24).

(24) a. $\forall t \in Ind, \forall p \in K$: $IsA'(t, p) \vdash \phi[\neg e'(p)] \to \neg\phi[t]$.
b. $\forall t \in Ind, \forall p \in K$: $IsA'(t, p) \vdash \neg\phi[t] \to \phi[\neg u'(p)]$.

[45] In the computational implementation of inferences using PL^-, we will run the 'categorial' inferences involved in (22) outside the logical inference. That is, for each complex category-object term in the form of a-b, we simply provide an atomic new term and this newly introduced term is defined to be a sub-category of the existing category a and b in a background inference system which we call a 'specialist.' This AI architecture improves the efficiency of the overall infrences. See Cassimatis, Bignoli, Bugajska, Dugas, Kurup, Murugesan, and Bello (2010).

For example, suppose that '$IsA'(meg', girl')$' and '$Run'(\neg e'(girl'))$' both hold.[46] Then, given (24-a), '$\neg Run(meg')$' must hold. Similarly for (24-b).

As we have indicated before, we do not really need expressions in the form of '$\neg e'(p)$' and '$\neg u'(p)$', but in order to show the soundness and completeness of $TREM$ with regard to the dual lattice structures that we explained at (5), these 'negative quantifier terms' are convenient. (24) then maintains the soundness and completeness of the entire logical inference system with regard to their interpretation models when we include these negative quantifier terms in the logical language.[47]

As we have indicated before, PL^- does not use any variable expressions. Since we do not use explicit quantifiers in our language, we do not need variables bound by such quantifiers. Since we characterize the semantics of pronouns with the functor I' in (16-d)$\sim$(17-d), as we sketched above, we do not need free variables for pronouns either.

This section has shown a possible logical language that incorporates the proposal that we have made in this paper. There are still several loose ends in the language, such as an exact specification of the sorted inferences and precisely how we construct complex category-object terms. We will also need to provide a formal comparison between the complexity of this language and the complexity of an alternative language that serves a similar purpose, such as second or higher-order predicate calculus languages and other sorted first-order languages. The next section shows the semantics of PL^-.

4 Semantics

As we pointed out a few times, PL^- in Section 3 can be interpreted either in the standard FOL interpretation model or in the FOL interpretation model plus the complete lattices for each set of objects.[48] Strictly speaking, PL^- does not maintain the standard FOL semantics completely, since we have posited two kinds of object denoting terms, i.e., I) Ind, which is the standard set of terms denoting usual individuals, such as *bob* and *meg* and II) K, which is a set of terms denoting 'category objects.' Intuitively, each category object is the identifier (ID) of each set of (usual) individuals. For example, for the set

[46] Remember that '$Run'(\neg e'(girl'))$' is for *Not a girl runs*, interpreted as "No girl runs."

[47] We probably should have shown the entailment relations as in (24) in terms of $\neg p$ with p being ANY term, not only with $\neg e'(p)$ and $\neg u'(p)$, but in this paper, it is more important to show this with regard to 'negative' quantifier terms, dealing with the semantics of negative quantifiers such as *no kids* in natural language.

[48] As we discussed in Section 2.2, at the moment, we only consider linear-scope dependencies among multiple quantifiers, which are expressible in standard FOL.

of boys in each interpretation model, there exists its unique category object. Category-denoting terms in PL^{-} then denote such category objects. We assume that when the set of boys and the set of lads, for example, contain exactly the same individuals in an interpretation model, then there is only one set there and therefore, there is only one category object as the unique identifier of that set and both boy' and lad' denote that category object in that model.

We first show how to interpret the main expressions of PL^{-} in the standard FOL interpretation models plus the above-mentioned category objects and a function associated with category objects. We then show how to do the same in the models enriched with the complete lattices. We ignore the interpretations of the function expressions other than u' and e'.

(25) Basic model M:
$M := \langle \mathbb{D}_i, \mathbb{D}_k, SN, I \rangle$

In (25), $\mathbb{D}_i$ is the domain of individuals in M, which include all the individuals in the model. I is the interpretation function in M that maps each non-variable expression of the logical language to its denotation in M, as we explain below. SN is the 'set-naming' function as we explained above, thus, each member of $\wp(\mathbb{D}_i)$ (i.e., the power-set of $\mathbb{D}_i$) is mapped to its own unique category object by SN. For example, SN maps each of $\{meg, bob\}$ and $\{meg, bob, fiona\}$ to its own unique identifier in this manner. Note that the function SN is one-to-one and surjective. That is, for each member s of $\wp(\mathbb{D}_i)$: SN maps s to exactly one member of $\mathbb{D}_k$ and for all the member d_k of $\mathbb{D}_k$, there is exactly one member s of $\wp(\mathbb{D}_i)$ such that $SN(s) = d_k$.

Although sets of objects here are extensional (i.e., any two sets that contain the same individuals are regarded as one and the same set), for convenience, I use the intuitive category name such as *boy* to refer to the category object for the set of boys in the given interpretation model. For example, we may let BOY represent the set of boys in the given interpretation model. Suppose that $BOY = \{bob, john, tom\}$. Then we write $SN(BOY) = boy$, when *boy* is the category object for the set $\{bob, john, tom\}$. Again, when some other predicate term, say, lad' in PL^{-} denotes this set of individuals in the model, then $I(lad') = boy$, since *boy* here represents the category object that is uniquely allocated to $\{bob, john, tom\}$ in this interpretation model.

Since PL^{-} does not have variables, we do not need assignment functions.

Given (25), the interpretations of the key expressions in PL^{-} are as in (26).

(26) a. $\forall d \in Ind : [\![d]\!]^M = I(d) \in \mathbb{D}_i$.
b. $\forall p \in K : [\![p]\!]^M = I(p) \in \mathbb{D}_k$.
c. $\forall R^n \in Pred : [\![R^n]\!]^M = I(R^n) \subseteq \underbrace{\mathbb{D}_i \times ... \times \mathbb{D}_i}_{n}$.
d. $[\![IsA']\!]^M = I(IsA') \subseteq \mathbb{D}_i \times \mathbb{D}_k$.

In words, each individual-denoting term d is interpreted as a member of $\mathbb{D}_i$ and each 'category-object-denoting' term p is interpreted as a member of $\mathbb{D}_k$. Recall that each set of individual objects, that is, each member of $\wp(\mathbb{D}_i)$ is mapped to its own unique category object, i.e., a member of $\mathbb{D}_k$ as its identifier by the above-mentioned 'set-naming function' SN. Each n-ary predicate R^n in $Pred$ is interpreted as a subset of the Cartesian product $\mathbb{D}_i^n$ $(= \{\langle a_1, ..., a_n\rangle \mid a_1, ..., a_n \in \mathbb{D}\})$.[49] That is, R^n denotes a set of ordered n-tuples each of whose component atoms is taken from $\mathbb{D}_i$. As we have explained in Section 3, IsA' denotes a set of ordered pairs $\langle a, p\rangle$ such that $a \in \mathbb{D}_i$ and $p \in \mathbb{D}_k$. Given the above-mentioned 'set-naming' function SN, whenever an individual, say, *bob* is a member of a set of individuals, say, the set of boys, in the given model M_i, then $[\![IsA']\!]^{M_i}$ includes the corresponding ordered pair, say, $\langle bob, boy\rangle$ as member (where *boy* is the identifier of the set of boys in M_i).

The interpretations of PL^- formulas are as in (27). For all n-ary predicate R^n $(\in Pred)$ ('$1 \leq\ n \leq \infty$' except for (27-d)$\sim$(27-e), where it is '$2 \leq n \leq \infty$'.):

(27) a. $\forall t_1, ..., t_n \in Ind : [\![R^n(t_1, ..., t_n)]\!]^M = True$ iff $\langle [\![t_1]\!]^M, ..., [\![t_n]\!]^M\rangle \in [\![R^n]\!]^M$
b. $\forall p \in K : [\![R^n(..., u'(p), ...)]\!]^M = True$ iff $\forall d \in \mathbb{D}_i$: if $\langle d, [\![p]\!]^M\rangle \in [\![IsA']\!]$, then $[\![R^n(..., a, ...)]\!]^M = True$ with $[\![a]\!]^M = d$.
c. $\forall p \in K : [\![R^n(..., e'(p), ...)]\!]^M = True$ iff $\exists d \in \mathbb{D}_i$ such that $\langle d, [\![p]\!]^M\rangle \in [\![IsA']\!]^M$ AND $[\![R^n(..., a, ...)]\!]^M = True$ with $[\![a]\!]^M = d$.
d. $\forall a_1, ..., a_n \in Ind$ $(2 \leq n \leq \infty)$: $[\![R^n(..., (a_1 \otimes ... \otimes a_n), ...)]\!]^M = True$ iff $[\![R^n(..., a_i, ...)]\!]^M = True$ for all i such that $1 \leq i \leq n$.
e. $\forall a_1, ..., a_n \in Ind$ $(2 \leq n \leq \infty)$: $[\![R^n(..., (a_1 \otimes ... \otimes a_n), ...)]\!]^M = True$ iff $[\![R^n(..., a_i, ...)]\!]^M = True$ for at least one i such that $1 \leq i \leq n$.

The interpretation of formulas in the form of $IsA'(a, p)$ with a being a complex term is similar to (27-d)$\sim$(27-e) and so we omit them here. We also omit the interpretation

[49]The notation here is a bit messy. Recall that n in R^n indicates the arity of the relation and hence is NOT the exponent, whereas n in $\mathbb{D}_i^n$ here is the exponent, since $\mathbb{D}_i^n$ is the Cartesian product $\underbrace{\mathbb{D}_i \times ... \times \mathbb{D}_i}_{n}$.

of negative terms, such as $\neg u'(kid')$, which we have explained informally before. (27) shows that it is possible to interpret PL^- in the standard FOL interpretation models enriched with the category objects and the 'set-naming' function SN, which is good from the viewpoint of keeping the interpretation models for PL^- as simple as the standard FOL models. However, there is one thing that is not really satisfactory with this way of interpreting PL^-. Notice that in (27-b), the universal quantifier term $u'(p)$ is interpreted in essentially the same manner in which the universal quantifier $\forall x$ is interpreted in the corresponding formula, $\forall x(P(x) \rightarrow R^n(..., x, ...))$, where P denotes the set that SN maps to p in (27-b). Basically, with this model theoretic semantic rule, $u'(p)$ is interpreted in the same manner that 'universal parametric term' is interpreted in the Herbrand interpretation of quantifiers (cf. Fitting, 1996). Since the Herbrand interpretation models for quantifiers allow us to interpret FOL quantifiers in terms of concrete parametric argument terms that replace quantifiers, this is not necessarily a bad thing. However, the lack of correspondence between the structure of the logical language and the structure of their denotations is not really resolved if we interpret our language as in (27). Similar criticism can be made for (27-c)$\sim$(27-e), in which we interpret the existential quantifier term $e'(p)$ and the complex terms such as $(a_1 \otimes ... \otimes a_n)$ and $(a_1 \oplus ...a_n)$, and also for the interpretation of 'negative' terms, which we omit.

We can resolve this problem by adding a set of complex individuals, which will provide the denotations for all the quantifier terms and complex/plural terms in PL^-. That is,

(28) Lattice-enriched model M:
$M := \langle \mathbb{D}_i, \mathbb{D}_k, \mathbb{D}_c, SN, I \rangle$

As well as everything we explained at (25) above, this model has $\mathbb{D}_c$, the set of complex objects. $\mathbb{D}_c$ is defined in (29).

(29) a. All members of $\mathbb{D}_i$ are also members of $\mathbb{D}_c$.
b. $\forall a_1, ..., a_n \in \mathbb{D}_i(\ 1 < n \leq \infty) : (a_1 \otimes ... \otimes a_n) \in \mathbb{D}_c$.
c. $\forall a_1, ..., a_n \in \mathbb{D}_i(\ 1 < n \leq \infty) : (a_1 \oplus ... \oplus a_n) \in \mathbb{D}_c$.
d. $\forall x \in \mathbb{D}_c : \neg x \in \mathbb{D}_c$.
e. $\mathbb{D}_c$ is the minimal set derived via (29-a)$\sim$(29-d).

The reader can check that when $\mathbb{D}_i = \{jack, meg, fiona\}$, (29) generates all the individuals in the lattices in (5). The interpretations of $\otimes$ and $\oplus$ are as we explained in Section 2.1, that is, 'AND' and 'OR' in a specific sense.

Actually, (29) generates superfluous individuals such as $(jack \otimes jack \otimes jack)$ and $(meg \oplus jack)$, but these are assumed to be the same objects as $jack$ and $(jack \oplus meg)$ in terms of monotonicity and commutativity. We put some, but not all of the structural rules in (30). Lack of internal parentheses indicate that the terms are fully associative. The other structural properties of complex terms should be obvious from the intended interpretations of $\oplus$ and $\otimes$ that we explained in Section 2.1.

(30) a. Contraction invariance I: $\forall x \in \mathbb{D}_c : (...x \otimes x...) = (...x...)$
b. Permutation invariance I: $\forall x, y \in \mathbb{D}_c : (...x \otimes y...) = (...y \otimes x...)$
c. Contraction invariance II: $\forall x \in \mathbb{D}_c : (...x \oplus x...) = (...x...)$
d. Permutation invariance II: $\forall x, y \in \mathbb{D}_c : (...x \oplus y...) = (...y \otimes x...)$

In fact, we showed a streamed-lined generation rule at (29). That is, (29) does not generate superfluous complex terms such as $(\neg jack \otimes \neg meg)$, which is equivalent to $\neg(jack \oplus meg)$ which (29) can generate. We could provide a more productive definition of the set of complex individuals, such as $\forall x, y \in \mathbb{D}_c : (x \otimes y) \in \mathbb{D}_c$ instead of (29-b), but such a productive rule would produce a larger number of superfluous complex objects which we then would need to identify in terms of structural association, permutation, contraction, De Morgan laws etc. At the moment, (29) is enough, given the intended interpretations of $\otimes$, $\oplus$ and $\neg$, as we explained in section 2.1.

Some might wonder if we need to include a pair of complete lattices for each subset of $\mathbb{D}_i$, since the denotation of $u'(boy')$, for example, is the glb of the positive lattice for the set of boys in the given interpretation model. However, note that for any subset of the domain set $\mathbb{D}_i$, the glb of the 'positive' lattice for that subset is a member of $\mathbb{D}_c$ in (30). Actually, for any subset of $mathbbmD_i$, both the positive and negative complete lattice structures are subsets of $\mathbb{D}_i$. Thus, at the basic level, we can only include $\mathbb{D}_i$, which include all the complex individuals that we want as the denotations of all the terms in PL^-, and then depending on which category-denoting term p is included in $u'(p), e'(p), \neg u(p), \neg e(p)$, etc., we can choose an appropriate subset of $\mathbb{D}_c$ and construct the adequate pair of lattices for the set of individuals in question.

Now, we briefly show how the inclusion of the set of complex individuals $\mathbb{D}_c$ improves the interpretation of PL^- in terms of the above problem. First, obviously, we can now provide the model theoretic denotations for quantifier terms and complex terms, as in (31).

(31) a. $\forall p \in K : [\![u'(p)]\!]^M = (a_1 \otimes ... \otimes a_n)$, where the set of individuals that SN

maps to $[\![p]\!]^M$ is $\{a_1, ..., a_n\}$.

b. $\forall p \in K:\ [\![e'(p)]\!]^M = (a_1 \oplus ... \oplus a_n)$, where the set of individuals that SN maps to $[\![p]\!]^M$ is $\{a_1, ..., a_n\}$.

c. $\forall p \in K:\ [\![\neg u'(p)]\!]^M = \neg(a_1 \otimes ... \otimes a_n)$, where the set of individuals that SN maps to $[\![p]\!]^M$ is $\{a_1, ..., a_n\}$.

d. $\forall p \in K:\ [\![\neg e'(p)]\!]^M = \neg(a_1 \oplus ... \oplus a_n)$, where the set of individuals that SN maps to $[\![p]\!]^M$ is $\{a_1, ..., a_n\}$.

e. $\forall t_1, ..., t_n \in Ind\ (1 < n \leq \infty) : [\![(t_1 \otimes ... \otimes t_n)]\!]^M = ([\![t_1]\!]^M \otimes ... \otimes [\![t_n]\!]^M)$.

f. $\forall t_1, ..., t_n \in Ind\ (1 < n \leq \infty) : [\![(t_1 \oplus ... \oplus t_n)]\!]^M = ([\![t_1]\!]^M \oplus ... \oplus [\![t_n]\!]^M)$

g. $\forall t_1, ..., t_n \in Ind\ (1 \leq n \leq \infty) : [\![\neg(t_1 \otimes ... \otimes t_n)]\!]^M = \neg([\![t_1]\!]^M \otimes ... \otimes [\![t_n]\!]^M)$.

h. $\forall t_1, ..., t_n \in Ind\ (1 \leq n \leq \infty) : [\![\neg(t_1 \oplus ... \oplus t_n)]\!]^M = \neg([\![t_1]\!]^M \oplus ... \oplus [\![t_n]\!]^M)$

The interpretation rule at (31-a) means that the denotation of the universal quantifier term $u'(p)$ is the glb of the positive complete lattice for the set $\{a_1, ..., a_n\}$, which the set-naming function SN maps to p and hence, this is the appropriate nominal restriction set for the universal quantifier term, such as the set of boys for $u'(boy')$. (31-b) is basically the same, mapping $e'(p)$ to the lub of the positive complete lattice for the set for p.

The interpretation rules at (31-c)$\sim$(31-d) show the denotations of negative quantifier terms. As we indicated above, we do not really need these terms in PL^-, but we show their interpretations for the sake of completeness. Note that the denotation of $\neg u'(p)$ is the lub of the 'negative' lattice for the set of individuals for p and that the denotation of $\neg e'(p)$ is the glb of the 'negative' lattice for the set of individuals for p (see (5)), reflecting the duality of the positive and the negative lattices for the set for p.

The interpretation of complex individual terms in (31-e) and (31-f) should be obvious. Recall that we overload the information with $\otimes$ and $\oplus$ by using these both as the connectives in complex terms in PL^- and in the meta-notations for the complex lattice theoretic denotational objects. Similarly, we overload the information with '$\neg$' in (31-g) and in (31-h), where we show the interpretation of negative non-quantifier terms. The reason why we put '$(1 \leq n \leq \infty)$' there, rather than '$(1 < n \leq \infty)$' in (31-e)$\sim$(31-f) is that (31-g)$\sim$(31-h) are meant to interpret the terms such as $\neg jack'$ and $\neg meg'$, as well as more complex terms such as $\neg(jack' \otimes meg' \otimes fiona')$ and $\neg(meg \oplus fiona)$.

Now, given (31), the interpretation of the PL^- formulas at (27) is significantly simplified, as in (32).

(32) a. $\forall R^n \in Pred : [\![R^n]\!]^M = I(R^n) \subseteq \underbrace{\mathbb{D}_c \times ... \times \mathbb{D}_c}_{n}$.

b. $\forall R^n \in Pred \;\&\; \forall x_1, ..., x_n \in TERM\ (1 \leq n \leq \infty) :\ [\![R^n(x_1, ..., x_n)]\!]^M = True$ iff $\langle [\![x_1]\!]^M, ..., [\![x_n]\!]^M \rangle \in [\![R^n]\!]^M$.

For example, (31) means that the formula $Introduce'(uboy', bob', meg')$ is True in the model M iff the ordered triple of the denotations of *uboy′*, *bob′*, *meg′* in M is a member of the set denoted by $Introduce'$ in M. Actually, we need a bit more than (32) when the formula contains multiple quantifier/negative terms. Basically, we need to generate a 'complex' set denoting predicate expression as we explained at (12) in section 2.2. Again, we omit the details of this process in this paper.

This section has showed the interpretation of PL^- in model theoretic semantics, both with regard to the interpretation models that are basically the same as the standard FOL interpretation models and with regard to the FOL interpretation models enriched with complete lattice structures that we discussed in section 2.1. In terms of the complexity of the semantic models, the former approach will be preferable, but in terms of the completeness of the interpretation and also in terms of the improved correspondence between the logical language structures and the denotation model structures, the latter approach is preferable. Thus, a crucial future research possibility will be how to minimize the number of complex individuals included into the denotation models while still completely covering the interpretations of the quantifier-terms, etc. in PL^-.

5 Conclusion

We have proposed a novel logical language in which the semantics of quantificational NPs (QNPs) in natural language are represented as arguments of predicates, rather than as quantifiers that bind variables in argument positions as in standard first-order logic. In order to provide adequate denotations for the QNP argument terms in this logical language, we have proposed particular interpretation model structures with complete lattice structures. If we use this logical language to represent the natural language semantics, we can treat both referential and quantificational NPs in the argument positions of verbs in the surface syntax as arguments of the predicates that those verbs denote in the semantics. This makes the translation between the phonological strings and their semantics via the natural language syntax simpler. The proposed logical language also makes it easier to switch between the logical inference using this language and other kinds of computation with alternative data-structures, such as graphs and topological spaces.

After explaining the basic idea of the formal language and its interpretation models including the lattice structures, we have shown how to deal with basic scope ambiguity data in English. We have also defined a particular logical language that instantiates the above idea and then shown that we can interpret this language both in the interpretation models that are essentially the same as the standard FOL semantic models and in the FOL models enriched with the complete lattice structures that we mentioned above. How to minimize the complexity of the latter interpretation models is left for future research, as well as the precise treatment of 'plural' quantifiers such as *most kids* and *all the kids*.

Acknowledgement

The authors would like to thank Nicholas Allott, Emmon Bach, Selmer Bringsjord, Mark Steedman, Naveen Sundar, Robert Truswell and the audience at SPE2010 (Paris) and at ESPP2010 (Bochum-Essen). This work was supported in part by grants from the Office of Naval Research (N000140910094), the Air Force Office of Scientific Research (FA9550-10-1-0389), and MURI award (N000140911029).

References

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

Blyth, T. S. (2005). *Lattices and Ordered Algebraic Structures*. Berlin: Springer-Verlag.

Carpenter, B. (1997). *Type-Logical Semantics*. Cambridge, Mass: MIT Press.

Cassimatis, N., A. Murugesan, and P. Bignoli (2009a, December). Relational probabilistic inference over unknwon objects. Revision submitted to Journal of Artificial General Intelligence.

Cassimatis, N. L., P. G. Bignoli, M. D. Bugajska, S. Dugas, U. Kurup, A. Murugesan, and P. Bello (2010). Adaptive algorithmic hybrids. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 903–914.

Cassimatis, N. L., A. Murugesan, and P. G. Bignoli (2009b). Inference with relational theories over infinite domains. In *Proceedings of the FLAIRS-09 Conference*, pp. 21–26.

Cassimatis, N. L., A. Murugesan, and M. D. Bugajska (2008). A cognitive substrate for natural language understanding. In *AGI*, pp. 99–106.

Dowty, D. R., R. E. Wall, and S. Peters (1981). *Introduction to Montague Semantics*. Dordrecht: D. Reidel Publishing Company.

Duan, Y. (2003). *Topology Adaptive Deformable Models for Visual Computing*. Ph. D. thesis, State University of New York, Stony Brook, NY.

Fitting, M. (1996). *First-Order Logic and Automated Theorem Proving*. Berlin: Springer-Verlag.

Heim, I. and A. Kratzer (1998). *Semantics in Gernerative Grammar*. Oxford: Blackwell.

Hendriks, H. (1987). Type change in semantics: the scope of quantification and coordination. In E. Klein and J. van Benthem (Eds.), *Categories, Polymorphism and Unification.*, Chapter 6, pp. 96–119. Centre for Cognitive Science, University of Edinburgh. Institute for Language, Logic and Information. University of Amsterdam.

Hintikka, J. and G. Sandu (1997). Game-theoretic semantics. In J. van Benthem and A. ter Meulen (Eds.), *Handbook of Logic and Language, vol. 1*, pp. 361–410. Boston, MA: MIT Press.

Jacobson, P. (1999). Towards a variable-free semantics. *Linguistics and Philosophy 22*, 117–185.

Jacobson, P. (2008). Direct compositionality and variable-free semantics: the case of antecedent contained deletion. In K. Johnson (Ed.), *Topics in Ellipsis*, Chapter 3, pp. 30–68. Cambridge University Press.

Kratzer, A. (2005). Indefinites and the operators they depend on: From japanese to salish. In G. N. Carlson and F. J. Pelletier (Eds.), *Reference and Quantification: The Partee Effect*, pp. 113–142. Stanford: CSLI Publications.

Kuroda, S.-Y. (1965). *Generative Grammatical Studies in the Japanese Language*. Ph. D. thesis, Massachusetts Institute of Technology, Cambridge, Mass.

Link, G. (1983). The logical analysis of plurals and mass terms. In R. Bäuerle, C. Schwarze, and A. von Stechow (Eds.), *Meaning, Use and the Interpretation of Language*, pp. 303–323. Berlin: Walter de Gruyter & Co.

May, R. (1977). *The grammar of quantification*. Ph. D. thesis, MIT, Cambridge, Mass.

Montague, R. (1970). English as a formal language. In *Formal philosophy: selected papers of Richard Montague*, pp. 108–221. New Haven: Yale University Press.

Montague, R. (1973). The proper treatment of quantificaiton in ordinary english. In J. Hintikka, J. Moravcsik, and P. Suppes (Eds.), *Approaches to Natural Language: Proceedings of the 1970 Standford Workshop of Grammar and Semantics*, pp. 221–242. Dordrecht: Reidel.

Moortgat, M. (1997). Categorial type logics. In J. van Benthem and A. ter Meulen (Eds.), *Handbook of logic and language*, Chapter 2, pp. 93–178. Cambridge, Mass: MIT Press.

Riesen, K. and H. Bunke (2012). Classification and clustering of vector space embedded graphs. In C.-H. Chen (Ed.), *Emerging Topics in Computer Vision and its Applications*. New Jersey: World Scientific.

Szabolcsi, A. (1992). Combinatory grammar and projection from the lexicon. In I. A. Sag and A. Szabolcsi (Eds.), *Lexical matters*, Number 24 in CSLI Lecture Notes, Chapter 9, pp. 241–268. CSLI Publications.

Uchida, H. (2008). *Scope Alternation and Type Logical Grammar*. Ph. D. thesis, University College London, London.

Uchida, H., N. L. Cassimatis, and J. R. Scally (2012). Perceptual simulations can be as expressive as first-order logic. *Cognitive Processing 13*(4), 361–369.

Väänänen, J. (2007). *Dependence Logic*. Cambridge: Cambridge University Press.

van Dalen, D. (2004). *Logic and Structure, 4th edition*. Berlin: Springer.