$\mathcal{NLL}$ Models

John Nerbonne
nerbonnedfki.uni-sb.de

Deutsches Forschungszentrum für Künstliche Intelligenz
Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, Germany

Abstract
$\mathcal{NLL}$ is a semantic representation language which has been developed for use in natural language processing (NLP). It has found use in interface applications at Hewlett-Packard Labs—in natural language database query and a natural language interface to a PC operating system—and at the German Research Center for Artificial Intelligence (DFKI, Deutsches Forschungszentrum für Künstliche Intelligenz)—in a natural language interface to an agents system (appointment management) and in a speech understanding system (train schedule information). The language is implemented and has been described syntactically in some detail (in reports in the literature references). There exist interfaces and indeed interface tools both for interfacing to syntactic description systems and to application systems, a number of inference rules (all of a simplifying type), and documentation of these, too, may be found in the references section.

The present paper provides a model theory for the language, including a mapping to the theory of relations, boolean combinations, quantified expressions, etc. It also tackles less traditional aspects of the language—role-based predication, restricted parameters, complex determiners, variable-binding term-forming operators, predicate operators expressing comparative and superlative derivations, etc.—one at a time in order to state clearly the impact of including these in the language. The purpose of developing the model theory is the guarantee that this provides that the language is consistent, the security it offers for future experimentation, and the independent notion of validity it constitutes, against which inference rules may be justified.

The development of $\mathcal{NLL}$ is continuing at HP Labs and the DFKI and the focus continues to be its use in natural language understanding.

Keywords: semantics, natural language understanding, natural language processing, computational semantics, meaning representation.

$\mathcal{NLL}$ Models

Contents

$\neg\phi$	
1	0
0	1
u	u

`if`(ϕ,ψ)	1	0	u
1	1	0	u
0	1	1	u
u	u	u	u

`AND`$\{\phi,\psi\}$	1	0	u
1	1	0	u
0	0	0	u
u	u	u	u

`OR`$\{\phi,\psi\}$	1	0	u
1	1	1	u
0	1	0	u
u	u	u	u

`iff`$\{\phi,\psi\}$	1	0	u
1	1	0	u
0	0	1	u
u	u	u	u

`XOR`$\{\phi,\psi\}$	1	0	u
1	0	1	u
0	1	0	u
u	u	u	u

The n-ary connectives are defined only for the $n = 2$ case above, but the more general definitions are straightforward generalizations. Each of the complex formulas is defined iff all of its components are. Otherwise,

$$\begin{array}{lll} [\![\texttt{AND}\ \Phi]\!]_{\mathcal{M}} = 1 & \Longleftrightarrow & \forall\phi \in \Phi\ [\![\phi]\!]_{\mathcal{M}} = 1 \\ [\![\texttt{OR}\ \Phi]\!]_{\mathcal{M}} = 1 & \Longleftrightarrow & \exists\phi \in \Phi\ [\![\phi]\!]_{\mathcal{M}} = 1 \\ [\![\texttt{IFF}\ \Phi]\!]_{\mathcal{M}} = 1 & \Longleftrightarrow & \forall\phi,\phi' \in \Phi\ [\![\phi]\!]_{\mathcal{M}} = [\![\phi']\!]_{\mathcal{M}} \\ [\![\texttt{XOR}\ \Phi]\!]_{\mathcal{M}} = 1 & \Longleftrightarrow & \exists!\phi \in \Phi\ [\![\phi]\!]_{\mathcal{M}} = 1 \end{array}$$

4 Quantification and Variable-Binding

The $\mathcal{NLL}$ treatment of quantification is based on the theory of GENERALIZED QUANTIFIERS (cf. Westerståhl 1989 and references there). The notion of a generalized quantifier[5] is based on the relation between two sets—a restriction set R and a scope set S. $R \cap S$ is called the intersection set I. The theory thus assumes a binary view of quantification in contrast to the normal unary (first-order) view. The theory was developed at least in part because of the realization that many natural quantification relations, that expressed by *most* in *Most men smoke* could not be expressed using unary quantification. This is appreciated if one considers the failure of the unary rendering to be equivalent: *Most things are men smokers.* Cf. Westerståhl 1989, p.16 for details.

4.1 Preliminaries

Syntactically, a generalized quantifier is expressed by ⟨Quantifier⟩ which determines the restriction set (R). The ⟨Scope⟩ of a ⟨Quantified Wff⟩ de-

[5] We do not make any linguistic assumptions here about which natural language determiners ought to be translated as "quantificational"; in particular, we do not wish to make any claims here concerning possessives, vague quantifiers, and quantifiers expressing defaults (such as "usually"). Our concern is to provide a useful tool for computational semantics, not to defend particular hypotheses about NL meaning.

termines the scope set (S). The meaning of a ⟨Quantified Wff⟩ can be expressed as a relation involving R and the intersection set $I = R \cap S$.[6]

⟨Quantified Wff⟩ ::= (⟨Quantifier⟩ ⟨Scope⟩)
⟨Quantifier⟩ ::= ⟨Determiner⟩ ⟨Variable⟩ $^+$ ⟨Restrictor⟩
⟨Quantifier⟩ ::= ⟨Ternary Quantifier⟩
⟨Ternary Quantifier⟩ ::= ⟨Order Relation⟩ ⟨Variable⟩ $^+$
⟨Restrictor⟩ ⟨Pole Restrictor⟩
⟨Restrictor⟩ ::= ⟨Wff⟩
⟨Pole Restrictor⟩ ::= ⟨Wff⟩
⟨Scope⟩ ::= ⟨Wff⟩
⟨Determiner⟩ ::= ⟨Simple Determiner⟩ | ⟨Complex Determiner⟩
⟨Simple Determiner⟩ ::= exists | forall | the | most | several | max | min
⟨Complex Determiner⟩ ::= (⟨Maximally Specified Measure⟩)

Before launching in, notes on strategy and extent: even though the logical syntax here assumes that quantified formulas have two parts, quantifier and scope, and that the former in turn has three (or four), determiner, variable, and restrictor (and perhaps pole restrictor), the definitions below provide a semantics only for quantified formulas directly, without interpreting the quantifier. This strategy is technically simpler because variable binding must be attended to in order to treat scope, but the variable being bound is only "visible", as it were, within the quantifier. Cf. Dalrymple et al. 1991, 414ff for a discussion of alternatives here. Turning to the intended extent of the present treatment, note that the definitions above allow for POLYADIC QUANTIFICATION (cf. van Benthem 1989)—i.e., quantifiers which bind more than one variable. This is a syntactically simple extension, whose interpretation, moreover, is likewise a straightforward generalization of the single-variable case (cf. below). But this possibility has never undergone any real experimentation in $\mathcal{NLL}$, and there have never been inference rules written based on it; we still regard it as an "experimental extension". Accordingly, we provide definitions of the monadic case and indicate how it generalizes to polyadic ones. On the other hand we have employed more extensively a generalization of another sort—from 2-place relations on sets to 3-place relations. These are the "ternary" quantifiers `more` and `fewer` (and closely related ones) as these are used in *More men than women smoke*, as investigated by Keenan and Moss 1984.

4.2 Variable Binding

In order to formulate the semantics of variable binding, essential to quantification, we make frequent reference to variable assignment functions β, which map variables to elements of U, the domain of discourse, and occa-

[6] Below we introduce the number-theoretic characterization of the quantifiers which we employ instead of the set-theoretic ones.

sionally to $\beta\frac{x}{a}$, the assignment function like β except perhaps at x, where it has the value a:

$$\beta\frac{x}{a}(x') = \begin{cases} a & \text{if } x' = x \\ \beta(x') & \text{otherwise} \end{cases}$$

We proceed now by placing further conditions on the interpretation function I, and on the relation '$\models$'.

- for every n-place determiner name, D^n, $I(D^n)$ is an n-place relation on subsets of U, i.e.,

$$(\text{a subset of} \underbrace{2^U \times \ldots \times 2^U}_{n}).$$

Then, for every 2-place determiner D, variable x, and formulas ϕ and ψ, we first note the conditions under which the quantified proposition is defined—we regard as undefined ONLY quantified formula in which there is no way of satisfying the restrictor. N.B. this is recognition of a semantic source of ill-formedness.

$$[\![(Dx\,\phi\psi)]\!]_{(\mathfrak{A},\beta)} \text{ is defined} \iff \{a | [\![\phi]\!]_{(\mathfrak{A},\beta\frac{x}{a})} = 1\} \neq \emptyset$$

Given this, we may specify truth conditions.

$$[\![(Dx\,\phi\psi)]\!]_{(\mathfrak{A},\beta)} = 1 \iff \langle\{a | [\![\phi]\!]_{(\mathfrak{A},\beta\frac{x}{a})} = 1\}, \{a | [\![\psi]\!]_{(\mathfrak{A},\beta\frac{x}{a})} = 1\}\rangle \in [\![D]\!]_{(\mathfrak{A},\beta)}$$

(the generalization to 3-place determiners—ternary quantifiers—is the generalization from 2-place to 3-place relations on sets and is straightforward. Cf. below.) The generalization to polyadic quantifiers (cf. above) is the generalization from relations on sets to relations on relations. E.g., in the dyadic case we examine, for restrictor ϕ (and scope ψ):

$$\{< a, b > | [\![\phi]\!]_{(\mathfrak{A},\beta\frac{x}{a}\frac{y}{b})} = 1\}$$

to see whether this stands in the proper relation to the relation provided by $[\![\psi]\!]$. But as note above, we shall not provide complete definitions here.

Many generalized quantifiers **Q** may be defined as binary predicates:

$$\lambda(r, i)\mathbf{Q}'(r, i)$$

which are applied to the cardinality of the restriction set and the cardinality of the restriction set intersected with the scope set. I.e., for a quantified formula of the form `(Det ?x` $\phi\psi$`)`, let $R = \{a | [\![\phi]\!]_{\mathfrak{A},\beta\frac{?x}{a}} = 1\}$ (restriction set), analogously $S = \{a | [\![\psi]\!]_{\mathfrak{A},\beta\frac{?x}{a}} = 1\}$ (scope set), $r = |R|$ and $i = |R \cap S|$ (n.b., that this is the cardinality of the INTERSECTION set, and not simply the scope set). We have in mind here those quantifiers which obey the Axiom of Quantity (Westerståhl 1989, 66ff). This number-theoretic characterization of generalized quantifiers is adequate for finite sets. There is an equivalent formulation in terms of sets and relations among sets (for a comparison see Westerståhl 1989), which, moreover generalizes to infinite cases—but we shall not be concerned with this elaboration. We shall employ this characterization in specifying the semantics of some of the logical quantifiers, to which we now turn.

4.3 Logical Determiners

Although ALL determiners have semantics in keeping with the general definition above, still several are important enough to warrant more exact specification. $\mathcal{NLL}$ fixes the denotation of the following. We provide both the number-theoretic and the set-theoretic characterization for the first example, and only the number-theoretic for the others. (We use $[\![D]\!]'$ to refer to the number-theoretic characterization of the denotation.)

Existential Determiner: (exists ?x $\phi\psi$)

(number-theoretically) $[\![\texttt{exists}]\!]'$ is $\{< r, i > | i \geq 1\}$, i.e.,
$[\![(\texttt{exists ?x } \phi\ \psi)]\!]_{\mathfrak{A},\beta} = 1$ iff $|R \cap S| > 0$ (or $R \cap B \neq \emptyset$)
where $R = \{a | [\![\phi]\!]_{\mathfrak{A},\beta \frac{?x}{a}} = 1\}$ and $S = \{a | [\![\psi]\!]_{\mathfrak{A},\beta \frac{?x}{a}} = 1\}$

Universal Determiner: (forall ?x $\phi\psi$)

$[\![\texttt{forall}]\!]'$ is $\{< r, i > | r = i\}$

Negative Existential Determiner: (no ?x $\phi\psi$)

$[\![\texttt{no}]\!]'$ is $\{< r, i > | i = 0\}$

Negative Universal Determiner: (notall ?x $\phi\psi$)

$[\![\texttt{notall}]\!]'$ is $\{< r, i > | r > i\}$

4.4 Further Examples

Even if the following are not logical determiners, they are of some interest:

Most: (most ?x $\phi\psi$)

$[\![\texttt{most}]\!]'$ is $\{< r, i > | i > r/2\}$

which, however, runs into difficulties over infinite domains, where we need:

Most: (most ?x $\phi\psi$)

$[\![(\texttt{most ?x } \phi\ \psi)]\!]_{\mathfrak{A},\beta} = 1$ iff $|R \cap S| > |R \cap \overline{S}$
where R, S as above.

This is a bit more precise than some would prefer to be with the natural language *most*, but it is of interest as the source of early demonstrations that even relatively simple natural language meanings resist first-order characterization (Barwise and Cooper 1981). We are unable to represent `most` without reference to cardinality—this would have been possible for the logical quantifiers above (where we might have appealed to subset relations between restrictor and scope, etc.).

Several: (several ?x $\phi\psi$)

$[\![\texttt{several}]\!]'$ is $\{< r, i > | i \geq 2\}$

The: (the ?x $\phi\psi$)

$[\![\texttt{the}]\!]'$ is $\{< r, i > | r = i = 1\}$

Finally, we examine two determiners which have proven useful in comparative semantics. They have been applied in domains where there was a need for quantification over an ordered set of measures of a property of a finite set of individuals; thus they assume an antecedently specified ordering ($\leq$). We assume that measures (such as cardinality or weight) are taken from the natural or real numbers, whose finite subsets are completely ordered (so that maxima are always unique in these finite subsets). If we wish to employ infinite domains, then we should use least upper bounds. It is further noteworthy that these determiners have no number-theoretic characterization.

Max: ($\texttt{max}_{\leq}$?m $\phi\psi$)

$$[\![(\texttt{max}_{\leq}\texttt{?m}\ \phi\psi)]\!]_{\mathfrak{A},\beta} = 1 \iff [\![\psi]\!]_{\mathfrak{A},\beta\frac{?m}{max_{\leq}(R)}} = 1$$

where R is, as usual, $\{a | [\![\phi]\!]_{\mathfrak{A},\beta\frac{?m}{a}} = 1\}$ and max is a function that selects the greatest of a set of numbers, i.e.,

$$n = max(R) \iff n \in R \wedge \forall n' \in R(n' \leq n)$$

Intuitively, $\texttt{max}_{\leq}$ finds the greatest value (with respect to $\leq$) which satisfies the restrictor, and asserts it of the scope. $\texttt{min}_{\leq}$ is just parallel.

Min: ($\texttt{min}_{\leq}$?m $\phi\psi$)

$$[\![(\texttt{min}_{\leq}\ \texttt{?m}\ \phi\psi)]\!]_{\mathfrak{A},\beta} = 1 \iff [\![\psi]\!]_{\mathfrak{A},\beta\frac{?m}{min_{\leq}(R)}} = 1$$

where R is, as usual, $\{a | [\![\phi]\!]_{\mathfrak{A},\beta\frac{?m}{a}} = 1\}$ and min is a function that selects the least of a set of numbers, i.e.,

$$n = min(R) \iff n \in R \wedge \forall n' \in R(n' \geq n)$$

In employing these quantifiers we often drop the reference to the antecedently specified order relation, since this is normally clear given the context (an indeed given merely the entities involved).

4.5 Complex Determiners

Complex determination (*At most five,...*) is more properly treated as a part of the $\mathcal{NLL}$ extension for plurals and mass terms (cf. below)—but it has a straightforward (and first-order definable) interpretation here, as the availability of number theoretic definitions shows.

$[\![(> n)]\!]' = \{< r, i > \vert i > n\}$	more than n
$[\![(< n)]\!]' = \{< r, i > \vert i < n\}$	fewer than n
$[\![(= n)]\!]' = \{< r, i > \vert i = n\}$	(exactly) n
$[\![(<= n)]\!]' = \{< r, i > \vert i \leq n\}$	at most n
$[\![(>= n)]\!]' = \{< r, i > \vert i \geq n\}$	at least n

This may be generalized even further, as Nerbonne 1994 shows, but because such quantification interacts so crucially with plural semantics, we shall be content with the sketch here for present purposes. (This leaves some complex determiners undefined—but the extension to plurals provides for their definition.)

It is perhaps objectionable that the components of these complex determiners remain undefined in isolation—but Nerbonne 1994 provides non-syncategorematic treatments of these quantifiers, and several more complex ones.

4.6 Ternary Quantifiers

As we noted above there is good semantic (and not merely mathematical) motivation for examining 3-place relations among sets. Cf. Keenan and Moss 1984, 76ff for elaboration. $\mathcal{NLL}$ provides for five basic variants, and Nerbonne 1994 shows how each of these is subject to further (infinite) parametric extension. The five basic variants, together with an example of a sentence each might symbolize, are as follows (the final variant is linguistically odd for reasons we cannot deal with—it is included for symmetry):

$$\left\{\begin{array}{l}\text{More}\\ \text{Fewer}\\ \text{(Exactly) as many}\\ \text{At least as many}\\ \text{At most as many}\end{array}\right\}\ \text{boys}\ \left\{\begin{array}{c}\text{than}\\ \text{"}\\ \text{as}\\ \text{"}\\ \text{"}\end{array}\right\}\ \text{girls swim}$$

```
  >
  <
( =   ?x boy(inst:?x) girl(inst:?x) swim(inst:?x))
  >=
  <=
```

In analogy to the number-theoretic characterization above, we may define (for a ternary quantified formula of the form (D ?x $\phi\phi'\psi$):

$$\begin{array}{l} R1 = \{a | [\![\phi]\!]_{\mathfrak{A},\beta\frac{?x}{a}} = 1\} \\ r1 = |R1| \\ R2 = \{a | [\![\phi']\!]_{\mathfrak{A},\beta\frac{?x}{a}} = 1\} \\ r2 = |R2| \\ S = \{a | [\![\psi]\!]_{\mathfrak{A},\beta\frac{?x}{a}} = 1\} \\ i1 = |R1 \cap S| \\ i2 = |R2 \cap S| \end{array}$$

Then we may provide number-theoretic characterizations of the ternary determiner denotations, always specified as a relation between $i1$ and $i2$ as defined above. We have:

$[\![>]\!]' = \{< i1, i2 > | i1 > i2\}$
$[\![<]\!]' = \{< i1, i2 > | i1 < i2\}$
etc.

It appears that each of these quantifiers (except perhaps '=') allows the definition of the Rescher quantifier, which brings us beyond first-order (cf. Westerståhl 1989, 22)—a step we took independently in earlier allowing the quantifier directly ('`most`').

5 Substitution

We shall want a rule of substitution at any number of points in computational semantics: anaphora resolution, inference, perhaps even in semantics construction. The SUBSTITUTION LEMMA guarantees that the substitution of like-denoting terms preserves equivalences. Schematically:

Substitution Lemma For formulas ϕ, terms `t`, and variables `?x`:

$$[\![\phi[\texttt{?x} \mapsto \texttt{t}]]\!]_{\mathfrak{A},\beta} = [\![\phi]\!]_{\mathfrak{A},\beta\frac{\texttt{?x}}{[\![\texttt{t}]\!]_{\mathfrak{A}}}}$$

where $\phi[\texttt{?x} \mapsto a]$ is the formula where all free occurrences of `?x` are replaced by `t`.

Given an inductive definition of $\phi[\texttt{?x} \mapsto a]$, the substitution lemma is normally proved through an induction on the construction of ϕ. The only point in $\mathcal{NLL}$ at which this standard result could be in doubt would be at the level of atomic formula, since the syntax and semantics are somewhat novel.

But even here the substitution lemma must hold. The only applicable case obtains when `?x` occupies an argument position in the list of role-argument pairs, so that ϕ is of the form $\texttt{P(}\ldots\ \texttt{r}_1\texttt{:?x}\ \ldots\ \texttt{r}_n\texttt{:?x}\ \ldots\texttt{)}$ (where `?x` does not otherwise occur freely in ϕ), and the right side of the lemma equivalence is satisfied whenever there is an n-tuple in $[\![\texttt{P}]\!]_{\mathfrak{A}}$ whose i_{r_1}- to i_{r_n}-th projections are all $[\![\texttt{?x}]\!]_{\mathfrak{A},\beta\frac{\texttt{?x}}{[\![\texttt{t}]\!]_{\mathfrak{A}}}}$, i.e. $\beta\frac{\texttt{?x}}{[\![\texttt{t}]\!]_{\mathfrak{A}}}(\texttt{?x})$, which is of course just $[\![\texttt{t}]\!]_{\mathfrak{A}}$. Given a substitution $[\texttt{?x} \mapsto \texttt{t}]$, the formula on the left side of the lemma equivalence holds iff there is an n-tuple whose relevant projections are likewise all equal to $[\![\texttt{t}]\!]_{\mathfrak{A}}$, which is what we wish to show. □

We shall not repeat the rest of the standard proof (cf. Ebbinghaus et al. 1978, 65), since it transfers to $\mathcal{NLL}$ transparently.

Summary

This completes our treatment of the kernel of $\mathcal{NLL}$. The following sections treat extensions of various sorts: § 6 treats function terms, variable-binding term operators, and restricted parameters, and other complex terms; § 7 provides a lambda operator and some further derived predicate operators. § 8 treats topics connected with plural semantics.

6 $\mathcal{NLL}$ Terms

We introduced constants and functions without added comment above in § 2 above. In this section we discuss function terms, skolem functions, restricted parameters, variable-binding term-forming operators, and Term-Forming Propositional Operators. We introduce further term types—group terms and sigma terms—in the section on the treatment of plurals (§ 8), we reserve remarks on locations terms until the fundamental ideas of structured ontologies are introduced (with plurals, again in § 8.

6.1 Function Terms

⟨Function Term⟩ ::= ⟨Function Name⟩(⟨Term⟩, ..)
⟨Function Name⟩ ::= ⟨Identifier⟩

The treatment of function terms is absolutely standard.

- for f an n-place function name, $t_1, \ldots, t_n$ terms

$$[\![f(t_1, \ldots, t_n)]\!]_{\mathcal{M}} = [\![f]\!]_{\mathcal{M}}([\![t_1]\!]_{\mathcal{M}}, \ldots, [\![t_n]\!]_{\mathcal{M}})$$

  as long as $[\![t_1]\!]_{\mathcal{M}}, \ldots, [\![t_n]\!]_{\mathcal{M}}$ are each defined, and is undefined wherever one of them is.

We eschewed role-marking here even though it is used throughout atomic formulas because there is little use for anadic functional terms (cf. § 2). Once an argument is dropped from a function, we tend no longer to obtain a unique value, so that this variability—a raison d'être of role-based predication—is therefore of little use.[7]

6.2 Skolem Functions

We intend to add skolem functions (including 0-place functions, i.e. skolem constants) to $\mathcal{NLL}$ at some time in the future. These are functions model-theoretically—but ones which are constrained to yield values which satisfy the open sentences they are used in (in place of selected bound variables) Fitting 1990, 187-90. But the latter is a constraint in a theory formulated in a logical language, not a constraint on the language itself. So nothing needs to be added to the model theory.

6.3 Restricted Parameters

⟨Restricted Parameter⟩ ::= ([⟨Determiner⟩] ?⟨Identifier⟩ | ⟨Restriction⟩)
⟨Restriction⟩ ::= ⟨Wff⟩

[7]An interesting exploration might allow that missing arguments are supplied by default, in the way that functions in some role-based programming languages are. Cf. Steele 1984, 61. But given the projection rule, § 2, an addition of this sort would make $\mathcal{NLL}$ nonmonotonic.

We ignore the case in which determiners are specified in this section—that possibility is taken up in § 6.3.1.

Restricted parameters are introduced in Barwise 1987 and their linguistic utility is developed in Gawron and Peters 1990, who together with Westerståhl 1990 are at pains to interpret these as denoting a special class of objects, PARAMETERS—which we, however, have not admitted to the domain of discourse. One can provide very nearly parallel treatments (to Gawron-Peters), by allowing that restricted parameters simply as terms which indefinitely denote something satisfying a given restriction. In this case they have a simple semantics:

- for ?x an individual variable, ϕ a formula

$$[\![(\texttt{?x} \mid \phi)]\!]_{\mathfrak{A}_\beta} = \begin{cases} a & \text{if } [\![\phi]\!]_{\mathfrak{A}_\beta \frac{?x}{a}} = 1 \\ \text{undefined} & \text{otherwise} \end{cases}$$

  We note that the denotation function '$[\![\;]\!]$' is nondeterministic when applied to restricted parameters.

We note that Gawron and Peter's ABSORPTION PRINCIPLE, which would effectively forbid the occurrence of bound variables in restrictions (we provide an example formulated in a syntactic way for the sake of concreteness–Gawron and Peters formulation is semantic), is not enforced here. It is trivial to show that, e.g., the following the formula is interpreted, in violation of the absorption principle.

```
(forall ?x  man(inst:?x)
            love(source:?x theme:(?y | child(inst:?y of:?x)) ) )
```

This indicates that the absorption principle is either an independent notion or that it depends on the nature of parameters as opposed to variables.[8]

6.3.1 Quasi-Logical Forms

But restricted parameters are used in another way as well, viz. to provide a "quantifier-in-place" representation in $\mathcal{NLL}$, e.g., we may write:

```
walk(agt:(most ?x | man(inst:?x)))
```

This representation is a often convenient when decisions about quantifier scoping are to be postponed—but it is an uninterpreted "quasi-logical"

[8] One way to construe the absorption principle is to postulate (i) that restrictions on bound and unbound variables are fundamentally different; and (ii) that any variable binding—absorption—must respect whatever restrictions have accumulated on the variables being bound. In the example in the text, the restriction on ?y restricts equally ?x. Thus, once ?x is bound, its restrictions (which include restrictions on ?y) are bound-variable restrictions, and no longer unbound-variable-restrictions. Formally, this would seem to commit us to binding neither ?x nor ?y independently—i.e., we require polyadic variable-binding operators.

extension of $\mathcal{NLL}$ (cf. Alshawi and others July 1989 for its usefulness). If one wished, one could interpret *formulas* ϕ in which terms such as '`(most ?x | man(inst:?x))`' occur: we obtain these interpretations by examining the formulas in which all occurrences of the restricted parameter are replaced by the variable `?x` BOUND BY QUANTIFIERS OF THE FORM `most ?x man(inst:?x)`. Note that we require only the the variable be somehow properly bound—so that we obtain in general a set of formulas ϕ', whose disjunction is equivalent to ϕ.

6.4 Complex Terms

$\mathcal{NLL}$'s complex terms involve variable-binding term-forming operators; the syntax of complex terms is:

⟨Complex Term⟩ ::= (⟨VB-TF-Operation⟩ ⟨Variable⟩, ...
⟨Operand⟩ | ⟨Wff⟩)
⟨VB-TF-Operation⟩ ::= *sum | *product | *min | *max | *avg
⟨Operand⟩ ::= ⟨Term⟩

A variable-binding term former — ⟨`VB-TF-Operation`⟩ — is used to construct aggregations over the domain.

Simple Example:

```
operation  bound-vars  operand          restriction

 (*sum       ?z, ?y       ?y      |  sales(agt:?z jan$:?y))
```

Assuming that the 'sales' relation holds of agent ?z and jan$?y iff salesman ?z sells products worth ?y dollars in January, then the complex term refers to the total dollar sales for all salesmen in January. Suppose we model the relation 'sales' as a set of tuples, called "Sales", and jan$ is the accessor to the `jan$` field of a tuple (similarly for `agent`). Then this is equivalent to

$$\sum_{s \in \text{Sales}} \text{jan\$}(s) \mid \exists z \; \text{agent}(s) = z$$

We have assumed that `sales(agt:?z jan$:?y)` holds iff `?z`'s January sales total `?y`.

Complicated Example: The operand part of a ⟨Complex Term⟩ is not limited to a single variable; instead, any ⟨Term⟩ can be used here as the following example shows:

(*sum ?z,?m1,?m2,?m3 plus(?m1,?m2,?m3) |
sales(agt:?z jan$:?m1 feb$:?m2 mar$:?m3))

with the meaning

$$\sum_{s \in \text{Sales}} \text{jan\$}(s) + \text{feb\$}(s) + \text{mar\$}(s) \mid \exists z \; \text{agent}(s) = z$$

But note the the operand can always be reduced to a single variable if we add a conjunction to the restriction requiring that a (new variable) be equal to the operand. Thus the above could be reduced to:

```
(*sum ?z,?m1,?m2,?m3,?y              ?y |
                     AND { sales(agt:?z jan$:?m1 feb$:?m2 mar$:?m3)
                                  =(id1:?y id2:plus(?m1,?m2,?m3)) } )
```

We exploit this equivalence, providing a model theory only for the latter form in which the operand is simple. This avoids technical complications on variable bindings.

In order to introduce a general model theory of these terms it will be useful to introduce the notion of a RELATION EXPRESSED BY AN OPEN FORMULA. We present the definitions below first as if the bound-vars in the complex term were exactly the variables occurring freely in the restriction. This will make the presentation simpler, and there is a straightforward generalization for cases with some variables bound in the complex term and others elsewhere (or unbound).

- For ϕ a formula, and $\{?\mathbf{x}_1, \ldots, ?\mathbf{x}_n\}$ the variables occurring freely in ϕ (taken in order of their occurrence), the relation expressed by ϕ, R_ϕ
$$R_\phi = \{< a_1, \ldots, a_n > \mid [\![\phi]\!]_{\mathfrak{A}_\beta \frac{?\mathbf{x}_1}{a_1} \cdots \frac{?\mathbf{x}_n}{a_n}} = 1\}$$
We have suppressed an implicit dependency on the model $\mathcal{M}$ interpreting ϕ. This will always be clear in context.
- When we generalize to cases where only some variables $\{?\mathbf{x}_1, \ldots, ?\mathbf{x}_i\}$ are being bound, we examine the RELATION EXPRESSED BY ϕ OVER $?\mathbf{x}_1, \ldots, ?\mathbf{x}_i$. We assume that $?\mathbf{x}_1, \ldots, ?\mathbf{x}_i$ occur freely in ϕ, and define the relation over $?\mathbf{x}_1, \ldots, ?\mathbf{x}_i$ expressed by ϕ:
$$R_{\phi; ?\mathbf{x}_1, \ldots, ?\mathbf{x}_i} = \{< a_1, \ldots, a_i > \mid [\![\phi]\!]_{\mathfrak{A}_\beta \frac{?\mathbf{x}_1}{a_1} \cdots \frac{?\mathbf{x}_n}{a_i}} = 1\}$$

We need to be careful about one further point in the interpretation of these expressions, viz. that our collection of operands may contain duplicates—and these may not be ignored. Thus if I ask for the sum of sales and the figure $10. appears twice, both occurrences must be included separately in the sum. If there are no further figures, the sum should be $20, and not $10. We need therefore to use operators defined not over sets (as is customary), but rather over multisets.

- For R^n a n-place relation, i_j the j-th projection function, let $i_j^M(R^n)$ be the MULTISET of elements occurring in R^n in the j-th position. I.e., a occurs in $i_j^M(R^n)$ with multiplicity m iff there are m distinct tuples $< a_i >_{i \leq n}$ in R^n such that $a_j = a$.

Given this notion, we intuitively obtain the denotation of the complex term in two further steps: first we examine the multiset obtained by evaluating the operand with respect to the relation expressed by ϕ over the bound vars (effectively the (multiset) image of the relation under the operation expressed by the operand); then the operator is applied to this multiset. Our simplification (above) to operands consisting of simple variables allows us to simplify the first step: rather than examine the image of R_ϕ under a function supplied by the operand (where variable bindings must be administered), we can examine the simpler (multiset) PROJECTION of R_ϕ onto the operand variable. This leads us to the following definition:

- for τ a complex term of the form `(OP ?x`$_1, \ldots,$`x`$_n$`,`$?y$` ?y` | ϕ`)`, `OP` a VB-TF-Operation, ϕ a formula:

$$[\![\tau]\!]_{\mathfrak{A}_\beta} = [\![\texttt{OP}]\!]_{\mathfrak{A}}(i^M_{n+1}(R_{\phi;\texttt{?x}_1,\ldots,\texttt{?x}_n,\texttt{?y}})$$

Of course this formulation assumes that the definitions of sum, etc. based on multisets are provided—but this is quite straightforward.

6.5 Measure Terms

⟨Measure Term⟩ ::= ⟨Specified Measure⟩
| ⟨Unspecified Measure⟩
| ⟨Maximally Specified Measure⟩
⟨Specified Measure⟩ ::= { ⟨Specifier⟩ ⟨Unspecified Measure⟩ }
⟨Unspecified Measure⟩ ::= ⟨Variable⟩ | ⟨Simple Measure⟩ | ⟨Complex Measure⟩
⟨Simple Measure⟩ ::= ⟨Numeral⟩

$\mathcal{NLL}$ reserves a part of its domain for numbers, denoted by numerals—we allow that numerals refer in $\mathcal{NLL}$ (and thus require that the universe of discourse include $\mathbb{R}$, the reals). Thus `1, 1.5, 477.35,...` are all SIMPLE MEASURES and therefore $\mathcal{NLL}$ terms. They denote transparently; they satisfy (in pairs) the customary order relations (`>, <, =, >=, <=`), just as expected; and they appears as arguments to the arithmetical functions addition '`plus(n,m)`', subtraction '`minus(n,m)`', multiplication '`times(n,m)`', and division '`/(n,m)`'. Instead of '`=`' we normally write '`=m`' to denote equivalence between commensurate measures (for motivation, see below).

- for n a simple measure, $[\![n]\!]_{\mathfrak{A}} \in \mathbb{R}$

It will be useful to reserve `?n, ?n1, ..., ?m, ?m1, ...` for variables which are restricted to ranging over measures (simple or complex).

6.5.1 Complex Measure Terms

In addition to the simple measures, the numerals, $\mathcal{NLL}$ also allows reference to COMPLEX MEASURE PHRASES such as `[1.85m]`, in a sentence such as *Sam is 1.85 meters tall*:

```
tall(theme:s measure:[1.85 m])
```

We shall take these to denote individuals, intuitively points on scales (of meters, etc.). In this we follow *inter alia* Cresswell 1976 and Pinkal 1989. We furthermore take a very simple-minded view of these individuals—modeling them as pairs of numbers and otherwise undenoted atoms.

We need to be able to refer to dimensional scales and to points on them.

- for s a scale—`m` (meter), `kw` (kilowatt), etc.

  $$[\![s]\!]_{\mathfrak{A}} \in U$$

  i.e., this is simply an atom.
- for `[n s]` a complex measure

  $$[\![\texttt{[n s]}]\!]_{\mathfrak{A},\beta} = \langle [\![\texttt{n}]\!]_{\mathfrak{A},\beta}, [\![\texttt{s}]\!]_{\mathfrak{A}} \rangle$$

Given this construal of complex measure terms, we cannot express measure equivalences as identities. Thus statements such as the following are logically false:

```
=(id1:[1 km] id2:[1000 m])
```

since the ordered pairs denoted must be distinct (which is forced, since the second elements must be). The consequence here is that the equivalence relation among measures will be denoted using a distinct relational symbol, `=m`, at least when we are writing carefully. In place of the above, we shall write:

```
=m(theme:[1 km] pole:[1000 m])
```

which we can generalize as

```
(forall ?n1  number(inst:?n1)
             =m(theme:[?n1 km] pole:[times(1000,?n1) m])
```

where '`times`' denotes the functional constant of multiplication.

There is of course an alternative development of the model theory in which the identity relation might hold properly of measures such as `[1 km]` and `[1000 m]`. For this to be possible, we would need to postulate an abstract dimension of length, of which `km` and `m` were alternative individuations. In this construal complex measure terms would be taken to denote points within the abstract dimension, so that measures using alternate scales might genuinely denote the same abstract point. And in this case axioms such as the last equation would no longer need to be stated IN $\mathcal{NLL}$, instead they would be a part of the model theory. This would be an interesting alternative as far as $\mathcal{NLL}$ is concerned—we present the other version here because it is simpler, but not because it seems definitely superior to the other.

We allow measures to serve as the arguments of the arithmetic functions, with the expected definitions:

- for $m = \langle n, s \rangle, m' = \langle n', s \rangle$ measures ON THE SAME SCALE, n'' a simple measure (number)

$$\begin{aligned} m + m' &= \langle (n + n'), s \rangle \\ m - m' &= \langle (n - n'), s \rangle \\ m \times n'' &= \langle (n \times n''), s \rangle \\ m/n'' &= \langle (n/n''), s \rangle \end{aligned}$$

Note that we have not required that '`plus`' is SYNTACTICALLY restricted to measures which mention the same scale. Thus it may make perfect sense to refer to '`plus([6 ft],[2 in])`'—even though nothing will be derivable from this in the absence of axioms showing how to link the `ft` and `in` scales.

application Before concluding this section we examine an important application of measure terms—adjectival semantics, including comparatives—let us note a perhaps nonobvious consequence of $\mathcal{NLL}$'s anadic predication (cf. § 2.4.2 as well). Suppose we take the semantics of the adjectival phrase *1m tall* to be properly rendered as suggested above, i.e., `tall(theme:?x measure:[1 m])`, then it is natural to ask how this differs from that of adjectival phrases without degree specification, e.g., *tall* as in *Sam is tall.* Within $\mathcal{NLL}$ the following must hold:

$$\frac{\text{tall(theme:j meas:[1 m])}}{\text{tall(theme:j)}}$$

In fact, since $\mathcal{NLL}$ always validates projections, effectively everything measurable will satisfy the `theme`-projection predicate of the relation `tall` used above—even though it is certainly false, e.g., for `j` an adult human being, that *J is tall* if *J is 0.7m tall.*

In order to use $\mathcal{NLL}$ for gradable adjectival semantics, we have to do one of two things—either allow that the relations denoted by adjectives with and without specifiers are distinct (which is counterintuitive), or allow that there may be a nonmonotonic step of supplying a DEFAULT argument in cases where none is syntactically explicit. This indeed is the step that has been taken in using $\mathcal{NLL}$ for comparative semantics. In this case we represent the meaning of *tall* as `tall(theme:?x meas:(?m | >=(th:?m pole:[1.7 m])))`. Alternatively, we could attempt to use contextual information in order to fill the `measure` role more satisfactorily. But these are both techniques which are beyond $\mathcal{NLL}$ directly, and thus which require auxiliary support.

6.5.2 Specified Measures

SPECIFIED MEASURES allow us to underspecify measure terms:

⟨Specified Measure⟩ ::= { ⟨Specifier⟩ ⟨Unspecified Measure⟩ }

⟨Specifier⟩ ::= $< \mid \leq \mid = \mid \neq \mid > \mid \geq$

Example:

"Jones is at most 6 feet tall."
`tall(theme:Jones measure:{<= [6 foot]})`

Semantically, a specified measure should be just equivalent to a restricted parameter, as the following examples illustrate:

`{<= [4 l]}` ≡ `(?m | <=(theme:?m pole:[4 l]))`

`{> 4 }` ≡ `(?m | >(theme:?m pole:4))`

But specified measures do not introduce parameters to which further reference is possible, and they are more compact in expression. We therefore find them a useful defined syntax.

- for `R` a specifier, `m` a measure, `{R m }` is a specified measure, with the interpretation:

  ⟦`{R m }`⟧$_{\mathcal{M}}$ = ⟦`(?m′ | R(theme:?m′ pole:m)`⟧$_{\mathcal{M}}$

  where `?m′` is a variable not used in the expression under evaluation.

We therefore regard specified measures (and maximally specified measures, cf. § 6.5.3) as syntactically defined—and therefore in no need of special model theoretic development.

6.5.3 Maximally Specified Measures

MAXIMALLY SPECIFIED MEASURES differ from specified measures only in that they allow specifying a delta (e.g., "2 more than") or a factor (e.g., "twice as many as"). These have been employed primarily as the bases from which plural and mass quantifiers have been derived (cf. § 8.4), and their usefulness extends to adjectival comparison as well (cf. *twice as tall as x, 2 cm taller than x*.

Examples:

"5 more X [than Sam hired consultants]" `{> ?n delta:2}`
"4 liters less X [than Sam drank water]" `{< ?m delta:[4 l]}`
"twice as many as 3 kg" `{= [3 kg] *:2}`

⟨Maximally Specified Measure⟩ ::=
{ ⟨Specifier⟩ ⟨Unspecified Measure⟩
[`delta` {:⟨Unspecified Measure⟩ | :⟨Specified Measure⟩ }]
[* :{ ⟨Simple Measure⟩ | ⟨Variable⟩ }] }

Note that the arguments to `delta` and `*` cannot be maximally specified measures (no recursion).

- for `R` a specifier, `m`, `d` measures, `{R m1 delta:d}` is a maximally specified measure with the interpretation:

```
[[{ R m1 delta:d}]]_M = [[(?m' | AND{  R(th:?m' pole:m1)
                                       =m(th:d pole:abs-val(?m,m1))
                                   }  )]]_M
```

where `?m'` is a variable not used in the expression under evaluation.

The formulation '`AND{R(th:?m' pole:m1) =m(th:d pole:abs-val(?m,m1))}`' generalizes over the cases expressed by *more* on the one hand and *less* or *fewer* on the other. It is equivalent to '`R(th:?m' pole:plus(m1,d))`' for *more*, and '`R(th:?m' pole:minus(m1,d))`' for *less* or *fewer*. Cf. Nerbonne 1994 for further discussion.

- for `R` a specifier, `m` a measure, `n` a simple measure (number) `{R m1 *:n}` is maximally specified measure, with the interpretation:

$$[\![\{\texttt{ R m1 *:n}\}]\!]_{\mathcal{M}} =$$

$$\begin{cases} [\![\texttt{(?m' | R(th:?m' pole:times(n,m1)))}]\!]_{\mathcal{M}} & \text{for } \texttt{R} = \text{`}\texttt{=m}\text{'} \\ [\![\texttt{(?m' | =m(th:?m' pole:times(n,m1)))}]\!]_{\mathcal{M}} & \text{for } \texttt{R} = \text{`>'}, \texttt{n} > 1 \\ [\![\texttt{(?m' | =m(th:?m' pole:times(plus(1,n),m1)))}]\!]_{\mathcal{M}} & \text{for } \texttt{R} = \text{`>'}, \texttt{n} < 1 \\ [\![\texttt{(?m' | =m(th:?m' pole:times(/(1,n),m1)))}]\!]_{\mathcal{M}} & \text{for } \texttt{R} = \text{`<'}, \texttt{n} > 1 \\ [\![\texttt{(?m' | =m(th:?m' pole:times(minus(1,n),m1)))}]\!]_{\mathcal{M}} & \text{for } \texttt{R} = \text{`<'}, \texttt{n} < 1 \end{cases}$$

where `?m'` is a variable not used in the expression under evaluation. These last semantic definitions have been expressed through $\mathcal{NLL}$, rather than directly, in order to emphasize that they add only convenience to the language, no new expressive capacity. The effects of the definitions are summarized in the table below, from Nerbonne 1994, which provides motivation:

Type	**Factor**	**Example**	**Proportion**	**Formula**
=	> 1	three times as much	$x/y = 3$	$x/y = f$
	< 1	one-third as much	$x/y = 1/3$	$x/y = f$
>	> 1	three times more	$x/y = 3$	$x/y = f$
	< 1	one-third more	$x > y \wedge$ $\lvert x - y\rvert/x = 1/3$	$x/y = 1 + f$
<	> 1	three times less	$x/y = 1/3$	$x/y = 1/f$
	< 1	one-third less	$x < y \wedge$ $\lvert x - y\rvert/x = 1/3$	$x/y = 1 - f$

6.6 Term-Forming Propositional Operators

The PROPOSITIONAL SENSE and STATE-OF-AFFAIRS operators are purely experimental, but they are useful especially in applications where it is necessary to reason about the intentions and goals of interlocutors in conversation—the correct recognition of SPEECH ACTS depends on this. We even provide alternative syntaxes for those who might wish to experiment with both propositions and states of affairs, but we do not attempt to specify a model theory.

$\hat{}\phi$ is intended to denote the proposition *that* ϕ

<<ϕ>> is intended to denote the state of affairs *denoted by* ϕ

7 Derived Predicates

We treat derived predicates in this section, both λ-predicates and predicates derived from predicate operators. In giving free rein to the definition of DERIVED predicates, we of course allow relations to be denoted whose projections need not be prominent in natural language—which is not problematic, but needs to be noted (cf. also § 2.4.2).

7.1 Lambda Abstraction

In many approaches to natural language semantics construction λ, the predicate-forming abstraction operator, is ultimately responsible for binding every argument to its position in a relation, and the operator seems indispensable even in alternative semantics construction schemes (cf. Nerbonne 1992a for elaboration).

$\mathcal{NLL}$ therefore provides a λ-predicate. Because of the role-based nature of predication in $\mathcal{NLL}$, and because we wished λ-predicates to combine with sets of role-argument pairs in a fashion exactly parallel to that in which simple predicates combine, we need a role-based version of lambda. Since we know of such thing in the literature, this $\mathcal{NLL}$ extension probably should be regarded as experimental, but it seems straightforward.

$$\langle\lambda\text{-Predicate}\rangle ::= (\text{lambda } arg_1{:}\langle\text{Variable}\rangle, \ldots, arg_n{:}\langle\text{Variable}\rangle\ \langle\text{Wff}\rangle)$$

There is a familiar theorem (Barendregt 1984, 63) about λ-operators which looks quite different in a role-based setting, namely η-reduction, i.e., for all predicates P

$$\lambda x\, P(x) \equiv P$$

where x is not free in P. The problem arises because roles are required when one asserts a relation in role-based formalisms. It would natural to require that the role name used by the λ abstraction operator be the same as that used in the scope of the λ abstraction, so that one could write:

$$\lambda\texttt{r:x P(r:x)} \equiv \texttt{P}$$

And this would make the λ-predicate appear similar to its simpler counterpart: in particular, it would combine with the same sets of role-argument pairs. But this tack cannot generalize to more complicated examples where several argument positions are abstracted over, e.g.:

```
λr:?x P(r1:?x r2:?x)
λr:?x AND{P(r1:?x) Q(r2:?x r3:?x)}
```

We conclude from such examples that it is hopeless to attempt to identify roles in λ-predicates and their scopes. Indeed such examples suggest further that no finite set of roles will suffice (if one generalizes to simultaneous abstraction over several variables, as we do). We also accept the stronger point, and provide for an unlimited number of nonce roles **`arg1, arg2,...`**

A final qualifying remark: since $\mathcal{NLL}$ does not have an identity relation for predicates, the η-reduction theorem cannot be stated directly in $\mathcal{NLL}$, but we can guarantee that the expressions on either side of the equivalence receive the same denotation, modulo permutations. Cf. below.

We turn then to the definition and interpretation of λ-predicates:

$\langle\lambda$-Predicate$\rangle$ ::= (lambda arg_1:$\langle$Variable$\rangle$, ...,arg_n:$\langle$Variable$\rangle$ $\dot{\{}$**`dist`**$\dot{\}}$ $\langle$Wff$\rangle$)

where the dotted braces indicate that '**`dist`**' is optional; this is a distribution operator, whose interpretation will be provided for in the section on plurals, § 8. The variant without distribution is interpreted in the following manner:

- for all variables $?\mathtt{x}_1 \ldots ?\mathtt{x}_n$, and all formulas ϕ

  $$[\![\texttt{(lambda arg1:?x}_1 \ldots \texttt{argn:?x}_n\ \phi\texttt{)}]\!]_{\mathfrak{A},\beta} \subseteq \underbrace{U \times \ldots \times U}_{n}$$

  i.e., a relation on U^n (just as the denotation of any n-place atomic predicate), where

  $$\langle a_1, \ldots, a_n\rangle \in [\![\texttt{(lambda arg1:?x}_1 \ldots \texttt{argn:?x}_n\ \phi\texttt{)}]\!]_{\mathfrak{A},\beta} \leftrightarrow [\![\phi]\!]_{\mathfrak{A},\beta\frac{x_1}{a}\ldots\frac{x_n}{a_n}} = 1$$

  and each role $\mathbf{arg}_j$ is interpreted as j-th projection function.

α-REDUCTION in the λ-calculus is the rule guaranteeing equivalence under variable renaming. We account for α reduction (Barendregt 1984) in the usual manner: the variable assignment function is irrelevant when we examine the values of bound variables. This is obvious in the interpretation of λ specified above.

η-REDUCTION. The denotation specified for the λ-predicates guarantees that for ϕ an atomic formula of the form $\mathtt{P}^n\texttt{(r}_1\texttt{:?x}_1\texttt{, ..., r}_n\texttt{:?x}_n\texttt{)}$, $\mathtt{P}^n$ an n-place predicate, then for every $\mathfrak{A}, \beta$:

$$[\![\texttt{(lambda arg1:?x}_1 \ldots \texttt{argn:?x}_n\ \phi\texttt{)}]\!]_{\mathfrak{A},\beta} \in \prod [\![\mathtt{P}^n]\!]_{\mathfrak{A},\beta}$$

The denotation of the λ-predicate is a permutation of the denotation of the atomic predicate.

β-REDUCTION guarantees the validity of λ-application.

β-Reduction For all formulas ϕ, variables $?\mathtt{x}_1, \ldots, ?\mathtt{x}_n$, and terms $\mathtt{t}_1, \ldots, \mathtt{t}_n$

$$\texttt{IFF}\ \{\ \texttt{(lambda arg1:?x}_1 \ldots \texttt{argn:?x}_n\ \phi\texttt{)}\ \texttt{(arg1:t}_1 \ldots \texttt{argn:t}_n\texttt{)}$$
$$\phi\ \texttt{[?x}_1 \mapsto \texttt{t}_1\texttt{, ... ?x}_n \mapsto \texttt{t}_n\texttt{]}\ \}$$

where $\phi[x \mapsto a]$ is the formula where all free occurrences of x are replaced by a.

We sketch a proof of this. The first (λ-unreduced) formula is an atomic formula which is therefore satisfied iff

$$\exists \langle a_i \rangle_{0 \leq i \leq n} \in [\![\texttt{(lambda arg1:?x}_1 \ \ldots \ \texttt{argn:?x}_n \ \phi\texttt{)}]\!]_{\mathfrak{A},\beta}$$

such that for each $\texttt{t}_j$ in $\texttt{t}_1, \ldots, \texttt{t}_n$, the arg-j-th projection of $\langle a_i \rangle_{0 \leq i \leq n}$ is $[\![\texttt{t}_j]\!]$, i.e., iff

$$\langle [\![\texttt{t}_i]\!] \rangle_{0 \leq i \leq n} \in [\![\texttt{(lambda arg1:?x}_1 \ \ldots \ \texttt{argn:?x}_n \ \phi\texttt{)}]\!]_{\mathfrak{A},\beta}$$

And the λ-clause of the model definition (above) tells us this holds iff

$$[\![\phi]\!]_{\mathfrak{A},\beta \frac{\texttt{?x}_1}{[\![t_1]\!]_{\mathfrak{A}}} \cdots \frac{\texttt{?x}_n}{[\![t_n]\!]_{\mathfrak{A}}}} = 1$$

And it follows from the substitution lemma (§ 5) that this holds iff

$$[\![\phi \ \texttt{[?x}_1 \mapsto \texttt{t}_1\texttt{,} \ \ldots \ \texttt{?x}_n \mapsto \texttt{t}_n\texttt{]} \ ]\!]_{\mathfrak{A},\beta} = 1$$

which is just the second (λ-reduced) formula. The demonstration that the formulas are defined under the same circumstances is similar. □.

7.2 Questioners and Question-Wffs

QUESTIONERS and QUESTION-WFFS are constructs which are used to represent the content of WH-phrases (*Which competitor*) and WH-questions (*Which competitor won the race?*) respectively. There is no distinctive $\mathcal{NLL}$ construct used to represent ALTERNATIVE or YES-NO questions.

⟨Question Wff⟩ ::= (⟨Questioner⟩ ⟨Scope⟩)
⟨Questioner⟩ ::= ?lambda ⟨Variable⟩, … ⟨Restriction⟩
⟨Scope⟩ ::= ⟨Wff⟩
⟨Restriction⟩ ::= ⟨Wff⟩

A questioner should be understood as parallel to a quantifier: each consists of an operator, a variable and a wff restrictor. In the case of questioners, the operator must be the lambda operator. Questioners and quantifiers are further parallel in their supra-syntax—the environments in which they function. Quantifiers are found in quantified wffs and questioners in question-wffs. The reasons for this parallelism are primarily the striking similarity in the information which the two constructs convey, and the parallel syntactic environments which give rise to them.

Examples:

What did O'Brian sell?
`(?lambda ?x thing(inst:?x) sale(agent:O'Brian product:?x))`

How many copies of Advancelink did IG-Farben buy in December?

```
(?lambda ?n NatNum(inst:inst:?n)
        sale(product:Advancelink
             recipient:Ig-farben
             date:December))
```

Which woman manages what department?

```
(?lambda ?x,?y
        and{woman(inst:?x) department(inst:?y)}
        manage(agt:?x pat:?y))
```

The general position in theoretical natural language semantics is that a question ought to be analyzed as denoting the set of all true answers (Groenendijk and Stokhof 1984, but cf. Ginzburg 1992 for an interesting dissenting view). Thus, the meaning of (?lambda ?x $\phi(?x)$ $\psi(?x)$) should be the set of propositions that arise from substituting some denoting term n for `?x` in

$$\text{and}\{\phi(?x)\psi(?x)\}$$

The trouble with this view is just that $\mathcal{NLL}$ has no notion of proposition beyond truth-value, which is clearly much too coarse for the purposes here.

$\mathcal{NLL}$ therefore treats questions semantically as λ operators—in fact just as it treats λ-predicates. This has the advantage that the characterization of the relation between question and direct answer is clear: in case the answer denotes an individual term, it may be understood as indirectly expressing the proposition formed by applying the question to the answer. In case the answer denotes a quantifier (*Who left? —No one.*), it may be taken to indirectly express the proposition formed by applying the quantifier to the question.

Who left?	`(?lambda ?x person(inst:?x) left(theme:?x))`
No one	`no ?y person(inst:?y)`
	`(no ?y  person(inst:?y)`
	`      (?lambda ?x person(inst:?x) left(theme:?x)) (?y))`

Question Wffs have a special status in the following sense: it is convenient to treat these as parallel to quantified wffs—and in particular as a kind of formula—in order to exploit the parallelism noted above. At the same time they cannot be of the same semantic type as formulas according to all the theories of the semantics of questions (cf. above). They form a category by themselves with the semantics of λ-predicates and a syntax like quantified formulas.

- for all variables $?\mathtt{x}_1 \ldots ?\mathtt{x}_n$, and all formulas ϕ, ψ

  $$[\![(\mathtt{?lambda}\ ?\mathtt{x}_1\ \ldots\ ?\mathtt{x}_n\ \phi\ \psi)]\!]_{\mathfrak{A},\beta} \subseteq \underbrace{U \times \ldots \times U}_{n}$$

  i.e., a relation on U^n (just as the denotation of n-place λ-predicates), where

$$\langle a_1, \ldots, a_n \rangle \in [\![\texttt{(lambda ?x}_1 \ldots \texttt{?x}_n\ \phi\ \psi\texttt{)}]\!]_{\mathfrak{A},\beta} \leftrightarrow [\![\phi]\!]_{\mathfrak{A},\beta\frac{x_1}{a}\ldots\frac{x_n}{a_n}} = 1 \text{ and } [\![\psi]\!]_{\mathfrak{A},\beta\frac{x_1}{a}\ldots\frac{x_n}{a_n}} = 1$$

This is just a simpler version of the interpretation for λ-predicates. The simplification is possible because we need not interpret roles at this point (since we do not need to combine these expressions in the same manner as predicates).

There is a point in the $\mathcal{NLL}$ definition where question wffs serve as components for more complex language constructs—and where therefore their double status could be troublesome, and that is where they may serve as the propositions in the propositional-sense and state-of-affairs constructs (cf. § 6.6)—in order to represent the meanings of indirect questions. But since these constructs are semantically (still) undefined, we shall not attempt to pursue all of the ramifications here.

7.3 Complex Predicates with Operators

⟨Complex Predicate⟩ ::= ⟨Predicate Operator⟩ ⟨Simple Predicate⟩
⟨Predicate Operator⟩ ::= -er | -as | -less | -est | -least | -too alt -enough

The predicate operators **-er, -less, -est, -least, as, -too** and **-enough** may be applied to predicates which are GRADABLE—i.e., those which express a two-place relation between an individual and a measure (cf.§ 6.5 above). In what follows we shall assume that the individual and measure places of the relation are denoted by the `theme` and `measure` roles, respectively. In defining various COMPARISON operators, we should like to allow for the ready expression of such relations as:

Example:

Jones is 2 inches taller than 6 feet.
-er(tall)(theme:Jones pole:[6 feet] spec:[2 inch]) ≡
tall(theme:Jones meas:plus([6 feet],[2 inch]))

There is some dispute—where $\mathcal{NLL}$ is decidedly agnostic—about whether examples such as the one above should be read *Jones is* AT LEAST *two inches taller than 6 ft.* or *Jones is* EXACTLY *two inches taller than 6 ft.* (and similarly for *Jones is 6 ft tall.*) $\mathcal{NLL}$ allows the expression of either meaning, and is in that sense agnostic. The NL system in which it has been used has always assumed the latter meaning, because (among other reasons) it is difficult to derived the further specified meanings (*at most 2 in taller than 6 ft* from bases with an opposite bias. We mention this here because our examples assume the *exactly* meanings throughout.

7.3.1 -er—Positive Comparison

The meaning of the predicate operator **-er** is definable in $\mathcal{NLL}$:

```
(forall  ?x thing(inst:?x)
         (forall  ?m,?m' AND { measure(inst:?m)
                               measure(inst:?m') }
                  IFF { -er(P) (theme:?x pole:?m spec:?m')
                        P(theme:?x
                          meas:(?m''| =m(th:?m'' pole:plus(?m,?m')))) }
           ) )
```

We arrive at the following schematic truth conditions for derived comparatives.

$$\texttt{-er}(P)(\texttt{theme:}x\;\texttt{pole:}p\;\texttt{spec:}d) \equiv$$
$$\exists m \in M_P\; p <_{M_P} m \wedge : |m - p| = d \wedge P(\texttt{theme:}x\;\texttt{spec:}m)$$

where M_P is the set of measures participating in the relation P, and $<_{M_P}$ is the order relation defined on it.

We note in passing that the "**specifier**" position of the comparative relation may be occupied by a specified measure (§ 6.5), allowing us representations such as:

Example:

Jones is at most 2 inches taller than 6 feet.
`-er(tall)(theme:Jones pole:[6 feet] spec:<= [2 inch])`

7.3.2 -less—Negative Comparison

Similarly, the meaning of the predicate operator "-less" is given by:

```
(forall  ?x thing(inst:?x)
         (forall  ?m,?m′ AND { measure(inst:?m)
                               measure(inst:?m′) }
                  IFF { -less(P) (theme:?x pole:?m spec:?m′)
                        P(theme:?x
                           meas:(?m″| =m(th:?m″ pole:minus(?m,?m′)))) }
           ) )
```

whose schematic truth conditions are thus:

$$\texttt{-less}(P)(\texttt{theme:}x\,\texttt{pole:}p\,\texttt{spec:}d) \equiv$$
$$\exists m \in M_P\ m <_{M_P} p \wedge |m - p| = d \wedge P(\texttt{theme:}x\ \texttt{spec:}m)$$

where M_P and $<_{M_P}$ are as above.

The natural language predicates which the `-er(P)` and the `-less(P)` relations are designed to represent benefit from the anadic framework of $\mathcal{NLL}$ in that (i) they are often used without filling the specifier roles (*taller than Smith* rather than *c cm taller than Smith*); and (ii) when they are so used, they denote projections (thus the latter parenthesized example implies the former, for all c).

7.3.3 -est, -least—Superlative Derivation

The definition of `-est` anticipates the treatment of plurals to some extent. The basic meaning of the superlative '`-est(P)`' is postulated to be a relation between an individual x and a group of individuals Y which is defined iff x is in Y and which obtains iff x stands in the `-er(P)` relation to every other y in Y—this does not allow there may be more than one individual standing in a given `-est(P)`-relation to a given group Y.

```
(forall  ?x,?Y AND { thing(inst:?x)
                     thing(inst:?Y) }
         IF(  -est(P)(theme:?x wrt:?Y)
              (forall ?y′ AND { i-part(inst:?y′ in:?Y)
                                ~ =(id1:?y′ id2:?x) }
                   -er(P)(theme:?x pole:?y′)
                ) ) )
```

where the relation `i-part` is defined below. We employ the already defined comparative operator, but this amounts to the following schematic truth conditions for derived superlatives:

$$\texttt{-est}(P)(\texttt{theme:}x\ \texttt{wrt:}Y) \equiv$$
$$\exists m \in M_P\ P(\texttt{theme:}x\ \texttt{meas:}m)\ \wedge\ \forall y \in Y, y \neq x\ P(\texttt{theme:}y\ \texttt{pole:}(m'|m' < m))$$

The **-least** operator is defined in exactly parallel fashion, requiring that all individuals stand in the **-less**-P relation to any **theme** argument. Both in the case of **-est** and in that of **-least** it is also interesting to define a further superlative relation—holding between individuals and locations iff the basic superlative relation holds of between individual and all others in the location. This allows a fairly direct rendering of superlative predicates such as *tallest in Palo Alto.* We omit the definitions here.

7.3.4 -as—Equatives

For the sake of completeness we include an equative comparison operator, which, however, has never been used (as far as one can tell).

```
(forall  ?x thing(inst:?x)
         (forall  ?m measure(inst:?m)
                  IFF { -as(P) (theme:?x pole:?m)
                        P(theme:?x pole:?m) }
           ) )
```

This is a suspiciously uninteresting definition which was motivated mainly by the desire to be able to modify these constructs freely, in particular to make sense of constructions such as *at most as tall as Tom [is tall].*

We should emphasize here that, in allowing the language construct in $\mathcal{NLL}$ we wished to accommodate other potential definitions. The trivial (identity-mapping) semantics is intended to provide a concrete proposal for those who wish to use the language without experimenting in this area.

7.3.5 -too, enough (Experimental)

These were included only in order to support experimentation (cf. Flickinger and Nerbonne 1992). Each is treated as relation between individuals and properties denoted by open sentences. We examine the semantics of **-too** here; **-enough** is quite analogous. The intention is roughly that an individual stands in the **-too**-P relation to property P' just in case he stands in the P-relation to the measure m, and all individuals who stand in the P-relation to any $m' \geq m$ are such that they do not (or better cannot) satisfy the relation P'. Consider *Peter is too old for Mary to date.* We would view **old** as P is, **date(source:Mary theme:x)** as the open sentence defining the property P' of being an x such that Mary marries x, and regard the sentence as true just in case Peter is n years old and Mary does not (cannot) stand in the date relation to anyone $n' \geq n$ years old. I am waffling on the modality for the obvious reason: some nonextensional notion is needed, and $\mathcal{NLL}$ does not support it. The expressive devices are included for the sake of experimentation.

8 Lattice Structured Domains

This section draws on the now extensive literature on the logic of plurals and mass terms (cf. especially Link 1983 and Link 1987)[9] in order to include some of the more easily implementable ideas as an extension to $\mathcal{NLL}$. As useful as this literature has been, still it contains gaps—notably in the extension of plural logic to multiplace relations—which prompted some theoretical innovation here. This section assumes familiarity with Link 1983.

8.1 Plurals

To deal with plural reference and predication, we need to represent and reason about properties which hold of groups without holding of their individual members. Examples of such properties are *be a threesome*, *numerous*, *disperse*, and *meet*—in all of their uses, but also *write a paper*—in some of its uses (e.g., *write a paper together*). We represent groups by imposing algebraic structure on the domain of discourse, by common consensus the structure of an atomic boolean algebra less the bottom element (cf. Lønning 1989 for discussion and justification of the steps from a join-semilattice to a full atomic boolean algebra). The situation is similar for mass-term ontologies but the condition of atomicity is dropped. Given the isomorphism between powerset domains ordered by $\subseteq$ and boolean algebras with their partial orderings, it is also possible to use powersets to provide the required structure—but this is felt to introduce a distracting question as to the distinction between individuals and the singleton set containing them.

We therefore add to our model definition the conditon that the universe of discourse U be a boolean algebra less the bottom element. We now introduce some metasemantic terminology we shall employ in giving the model theory below. We use '$\sqsubseteq_i$' to denote the inclusion relation (partial order) on U, and '$\sqcup_i$' for the join operation which returns the least upper bound of pairs in U (under '$\sqsubseteq_i$'). '$\sqsubseteq_i$' is the relation that holds both between subgroups and groups, and also between individuals and groups containing them. The atoms in the plural lattice satisfy the predicate atom(x), and they correspond to individuals; the nonatomic elements correspond to groups of individuals. Cf. Figure 2. Since boolean algebras are complete, there exists a supremum for every arbitrary set A, i.e., $\bigvee A$; this is just the lub (under $\sqsubseteq_i$) of the atoms in A.

[9] The (relatively straightforward) generalization to mass reference will not be dealt with in detail because time prohibits examining it separately.

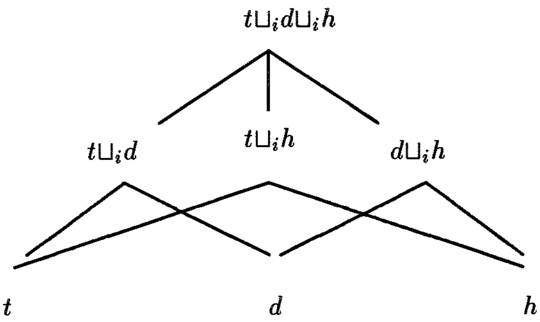

Figure 2: Sample of semilattice for plural reference (for 3-individuals). '$\sqcup_i$' is lattice join; 'a $\sqsubseteq_i$b' holds if one can travel from a up to b along $\sqcup_i$-lines. E.g. d $\sqsubseteq_i$d $\sqcup_i$t $\sqcup_i$h.

⟨Group Term⟩ ::= +{⟨Term⟩, ⟨Term⟩}
(sigma [⟨Determiner⟩] ⟨Variable⟩ | ⟨Restricting Wff⟩)

(The optional determiner serves the same purpose here it did with restricted parameters—it allows experimentation with quasi-logical form. In the case of measure determiners—cf. § 8.4.3—an interpreted as restriction would be straightforwardly justified. But we shall not attempt a general model-theoretical interpretation.) We stipulate that the meanings of **i-part**, **atom**, and **atomic-i-part** receive the obvious interpretation; we let '+{ }' denote a generalized $\sqcup_i$; and we use '`(sigma ?x | ` ϕ`)`' to designate the supremum of x satisfying ϕ). Thus we require that every model $\mathcal{M} = (\mathfrak{A}, \beta), \mathfrak{A} = (U, I)$ satisfy the following requirements:

- $[\![\texttt{i-part}]\!]_I = \{< a, b > | a \sqsubseteq_i b\}$
- $[\![\texttt{atom}]\!]_I = \{a | \text{atom}(a)\}$
- $[\![\texttt{atomic-i-part}]\!]_I = \{< a, b > | \text{atom}(a) \wedge a \sqsubseteq_i b\}$
- for all terms $t_1, \ldots, t_n$,

$$[\![\texttt{+\{ } t_1, \ldots, t_n\texttt{\}}]\!]_{\mathcal{M}} = [\![t_1]\!]_{\mathcal{M}} \sqcup_i (\ldots \sqcup_i [\![t_n]\!]_{\mathcal{M}})$$

- for all variables x, and all formulas ϕ

$$[\![\texttt{(sigma ?x |}\phi\texttt{)}]\!]_{\mathfrak{A}_\beta} = \vee\{a | [\![\phi]\!]_{\mathfrak{A}_{\beta\frac{x}{a}}} = 1\}$$

This model theory already justifies two important inference rules concerning plurals, which we provide instances of here:

```
% "Tom, Dick, and Harry ... one of them"
%
% atomic-i-part(th: ?x in: +{ T, D, H})
```

```
% ----------------------------
% OR{=(ID1: ?x ID2: H) =(ID1: ?x ID2: D)
%      =(ID1: ?x ID2: T)}
% "one of the men"
%
% atomic-i-part(th: ?x in: (sigma ?Z MAN(inst/i: ?Z)))
% --------------------------------------
% MAN(inst: ?x)
```

We shall have more to say about the ROLE MEREOLOGY '`/i`' below (§ 8.3), and the careful reader will have noted that the first rule's validity turns on the constants' (`T, D, H`) all denoting atomic individuals. The $\mathcal{NLL}$ definition does not require that individual constants be restricted to denoting atomic individuals, but indeed always been respected. If this restriction were relaxed, the rule would need to be constrained.

8.2 Distributivity Operators

Furthermore, it is useful to be able to use an analogue of Link's distributive predicate operator, ^{D}; As Link defines this, for any predicate P, $^{D}P(x) \leftrightarrow \forall x'(x'\text{atom-}\sqsubseteq_i x \rightarrow P(x'))$, i.e. ^{D}P is true of objects whenever P is true of their component atoms. The operator is useful in natural language semantics at least for the representation of the adverbial particle *each*, as in *They each spoke*, and for the representation of the ANTIQUANTOR *each* as it *They read a book each*. For reasons detailed further in § 8.3, we have implemented this as variation of λ-abstraction. In $\mathcal{NLL}$ we therefore have:

⟨λ-Predicate⟩ ::= (lambda arg_1:⟨Variable⟩, ...,arg_n:⟨Variable⟩ {**dist**} ⟨Wff⟩)

where the dotted braces indicate that '**dist**' is optional; the variation with distributive operator is interpreted here:

- for all variables $\texttt{?x}_1 \ldots \texttt{?x}_n$, and all formulas ϕ

  $$[\![\texttt{(lambda arg1:?x}_1 \ \ldots \ \texttt{argn:?x}_n \ \texttt{dist} \ \phi\texttt{)}]\!]_{\mathfrak{A}_\beta} \subseteq \underbrace{U \times \ldots \times U}_{n}$$

  i.e., a relation on U^n (just as the denotation of any n-place atomic predicate), where

  $$\begin{array}{cl} < a_1, \ldots, a_n > & \in [\![\texttt{(lambda arg1:?x}_1 \ \ldots \ \texttt{argn:?x}_n \ \phi\texttt{)}]\!]_{\mathfrak{A}_\beta} \\ \leftrightarrow & \forall a'_1 \sqsubseteq_i a_1 \ldots \forall a'_n \sqsubseteq_i a_n \ [\![\phi]\!]_{\mathfrak{A}_\beta \frac{x_1}{a'_1} \ldots \frac{x_n}{a'_n}} = 1 \end{array}$$

  and each role $\texttt{arg}_j$ is interpreted as j-th projection function.

Thus, for an entity to satisfy `lambda arg1:?x dist child(inst:?x)` each of its parts must likewise satisfy `child(inst:?x)`. The version with the distributive operator is satisfied by groups of children, i.e., entities all of whose subparts are children.

All of the above is fairly standard machinery in plural logics. On the other hand, we have not yet said anything about Link's plurality operators '*' and '+' which are definable within $\mathcal{NLL}$ extension provided thus far. For P a one-place predicate on atoms, for all ?x, we define *P, +P:

```
IFF  { *P(inst:?x)
         i-part(inst:?x in:(sigma ?y|P(inst:?y))) }
IFF  { +P(inst:?x)
         AND { *P(inst:?x) ~atom(inst:?x) } }
```

We shall therefore not provide separately for it as a primitive. We are somewhat unsure about how these operators should generalize to multiplace relations in any case, a topic we turn to now.

8.3 Distributive Relations

The language extension discussed in this section is clearly experimental—indeed, it appears to be novel. But it arises naturally in $\mathcal{NLL}$ (because of the importance of roles and anadic predication, which figured prominently above (§ 2)), the extension is worth some special discussion.

The innovation concerns distributivity and the closure properties of predications over plurals. As discussed above (§ 8.2) there are several means of expressing distributivity in $\mathcal{NLL}$, and there are various natural generalizations to the case of many-place relations. I should warn that I believe that the generalization from the treatment of distributivity in predicates to that in multiplace relations has not been examined linguistically. Link 1983 and subsequent authors allow for this in allowing a distributivity operator for one-place predicates, and a λ-abstraction operator (which creates one-place predicates from arbitrary formulas, including relational ones subject to further abstraction). The possibilities of these systems are illustrated in the following $\mathcal{NLL}$ formula, followed by its equivalent in the Link LP logic:

```
(lambda  arg2:?y
         dist (lambda  arg1:?x dist  chase(agt:?x theme:?y))
                                          (sigma ?u | boy(inst:?u)))
             (sigma ?v | girl(inst:?v))
```

$$^{D}\lambda\, y\, [^{D}\lambda\, x\, \mathrm{chase}(x,y)\, (\sigma\, u\, \mathrm{boy}(u))]\, (\sigma\, v\, \mathrm{girl}(v))$$

which, as may be verified, hold iff each boy and girl stand in the *chase* relation. Thus the expressive means provided in $\mathcal{NLL}$ are already as general as is normally provided for in plural logics. I.e., given the distributive operator, we can describe whatever distributivity is normally provided elsewehere.

There are two points at which these mechanisms seem inadequate to the task of modeling plural meanings in natural language, "distributivity" in group-based predications and so-called CUMULATIVE QUANTIFICATION. By the first I refer to the well-known observation that one sees a kind of "distributivity" in predications involving groups essentially (Scha and Stallard 1988). Thus *meet* is taken to denote a predicate which cannot be true of individuals, and thus cannot be distributive in the technical sense above (which would have the consequence that the predicate would then necessarily hold of individuals). But a sentence such as *Several groups met* is interpreted very naturally as meaning either that the groups met separately or that they convened together in a kind of plenary session. A kind of upward closure on predicates seems more appropriate to modeling this kind of phenomenon.

Scha 1981

noted that a kind of cumulative quantification seems to underlie the interpretation of *100 Dutch firms bought 5,000 American computers* in which there is a group of 100 Dutch firms and similarly a group of 5,000 American computers, and where each firm is involved in a purchase of one of the computers and each computer is likewise involved. Scha 1981 proposes a modeling in terms of a kind of polyadic quantifier, discussed further in van Benthem 1989.

It would be more in the spirit of $\mathcal{NLL}$ to seek a solution, not in a particular kind of quantification, but rather in the underlying theory of relations. This prompts one postulate and one generalization of the usual notion of distributivity. We postulate first that relations are closed under i-sums, i.e.,

Relations closed under $\sqcup_i$ For $a_1, \ldots, a_n, b_1, \ldots, b_n$, R an n-place relation, R is closed under $\sqcup_i$, i.e.

$$\langle a_1, \ldots, a_n\rangle \in R \wedge \langle b_1, \ldots, b_n\rangle \in R \Rightarrow \langle (a_1 \sqcup_i b_1), \ldots, (a_n \sqcup_i b_n)\rangle \in R$$

Note immediately that this accounts for the example above of *Several groups met.* But it also generalizes beyond simple predicates to cases involving relations. For example the following would be predicted to hold:

Dan read *Stuart Little* to Chris.
Dan read *Stuart Little* to Matt.
Dan read *Stuart Little* to Andrea.
———————————————————
Dan read *Stuart Little* to Chris, Matt and Andrea.

This could be modeled without the closure postulate, but only by proposing that a distributive operator is at play somewhere. But the closure postulate is not sufficient for explanations of how one can infer properties of atoms from properties of groups, only vice versa. This is the issue of distributivity, which is discussed above in connection with distributive operators (§ 8.2), and which is fundamental in the logic of plurals. This is the inference from the premise that a (potentially relational) predication holds of a group to one or more predications about the individuals in the group. It is generally interpreted by the $*P$ complex predicate mentioned in § 8.2 above.

We turn now to the treatment of distributive roles.

⟨Complex Role⟩ ::= ⟨Individual Role⟩ | ⟨Group Role⟩
⟨Individual Role⟩ ::= ⟨Simple Role⟩ /i
⟨Group Role⟩ ::= ⟨Simple Role⟩ /g

What we need is a generalization from one-place predicates to many-place predicates. The following achieves this in terms of the projection functions which proved so useful in allowing anadic predication:

Role Mereology The j-th projection of relation R is INDIVIDUAL-BASED iff for all n-tuples $\langle A_1, \ldots, A_j, \ldots, A_n\rangle \in R$, there are atoms $a_{j,1}, \ldots, a_{j,k}$ such that

$$\bigvee\{a_{j,1}, \ldots, a_{j,k}\} = A_j$$

and there exist $a_{i,k}$ i-parts of A_i, such that

$$\langle a_{i,k}\rangle_{1\leq i\leq n} \in R$$

and such that

$$\bigvee\{a_{i,k'} \mid 1 \leq k' \leq k\} = A_i$$

Note that the $a_{i,k}$ need be neither (i) proper i-parts of A_i (they may each be A_i), nor (ii) atomic nor (iii) distinct from one another.

To see how this condition can work, we unpack beginning with the last clause, which requires that

$$\begin{array}{llllllll} \langle a_{1,1}, & \ldots & a_{j-1,1}, & a_{j,1}, & a_{j+1,1}, & \ldots, & a_{n,1}\rangle & \in R \\ & & & \vdots & & & & \\ \langle a_{1,k} & \ldots & a_{j-1,k}, & a_{j,k}, & a_{j+1,k}, & \ldots, & a_{n,k}\rangle & \in R \\ \hline \langle A_1 & \ldots & A_{j-1}, & A_j, & A_{j+1}, & \ldots, & A_n\rangle & \in R \end{array}$$

That is, the components must form n-tuples of R horizontally and must sum ($\sqcup_i$) vertically to the original tuple.

Four aspects of this definition are worth special mention. First, given that relations are closed under $\sqsubseteq_i$, it is a generalization of Link's original +P and *P operators, so that it covers the distributivity inferences covered by them. This is of course just the case of 1-place R—whose single projection may be seen to be individual-based just in case, where any A satisfies R, its component atoms do as well. Second, the proposal will have distinct linguistic consequences from treatments via polyadic quantifiers, since e.g., we need not assume that the terms involved here have scope. But the detailed examination of these consequences will have to await a more linguistically oriented work. Third, the definition will have the effect of the "cumulative quantifiers", so that, e.g., if *The three men danced with the two women*, then there must be groups of three men and two women such that each of the men danced with one or more of the women and each of women danced with one or more of the men. In particular, it need not be the case that "all pairs" are found in the `dance-with` relation.

Fourth and finally, the definition is based on a property of particular projections on roles, so that it licenses a notation in which distributivity is noted on roles, to which we may now turn. We begin with an example of an $\mathcal{NLL}$ distributivity inference.

Example: The natural language predicate `secretary` is best modeled distributively, so that if Tom, Dick, and Harry are secretaries, then each of them is a secretary.

```
IFF {  secretary(inst/i:  +T,D,H)
       AND {  secretary(inst/i:T)
              secretary(inst/i:D)
              secretary(inst/i:H) } }
```

The notation '`rolename/i`' is introduced formally below. It is intended to designate an INDIVIDUAL-BASED role, i.e., one which, if played by a group, is played by all its members. This sort of role licenses distributive inferences like the one above. It is worth emphasizing that roles which are marked as "individual-based" may nonetheless be filled by a group-denoting arguments in virtue of their applicability to the individual members of that group. Hence marking an argument position "individual-based" may be regarded as tagging it for a "distributivity" inference.

Individual-based roles may be contrasted with GROUP-BASED roles, which may never be occupied by atomic individuals, and which are designated '`rolename/g`':

`couple(inst/g:?x)` *be a couple*

And some roles are neither group- nor individual-based. They indifferently allow either interpretation. E.g., one can write music individually or in collaboration.

We provide the formal definitions now. For all relation names `P`, all rolenames `r`,

- a rolename is marked `r/i` iff the projection associated with it is individually based.
- a rolename is marked `r/g` iff the projection associated with it includes no atomic individuals

Given our earlier definition of what is for an individual to occupy a role in a given situation, and given our wish to guarantee the projective inference property (§ 2.4.1), it is also important to note that the mereological role-markings "individual-based" and group-based are preserved in projections—they in no way depend on the constellation of roles in a predication. On the contrary, if a role is marked as "individual-based", then the projection it defines must hold distributively of any argument.[10]

There are of course roles which are neither individual- nor group-based, those which may contain both, e.g. the agent role of **carry**—both individuals and groups may participate in carrying.

We conclude this section with an example from the implemented distributive inference rule `DISTRIBUTE-PREDICATIONS-IN-ATOMS-EQUIV`.

```
%  "The three men hit the two dogs"
%
% HIT(agt: (sigma ({>= 3})
%            ?X | MAN(inst: ?X))
%      pat: (sigma ({>= 2})
%              ?Y | DOG(inst: ?Y)) loc: ?Z)
%
% we infer:
%
% (exists ?ISUM-4
%    AND{MAN(inst: ?ISUM-4)
%          size(th: ?ISUM-4 meas: ({>= 3}))}
%    (exists ?ISUM-3
%       AND{DOG(inst: ?ISUM-3)
%             size(th: ?ISUM-3 meas: ({>= 2}))}
%       AND{(for-all ?X i-part(th: ?X in: ?ISUM-4)
%              (exists ?Y i-part(th: ?Y in: ?ISUM-3)
%                 HIT(agt: ?X pat: ?Y loc: ?Z)))
%             (for-all ?Y i-part(th: ?Y in: ?ISUM-3)
%                (exists ?X i-part(th: ?X in: ?ISUM-4)
%                   HIT(agt: ?X pat: ?Y loc: ?Z)))}))
%
```

8.4 Quantification

Link 1987
discusses the need for genuinely quantifying over the plural entities, which he sees in sentences like the following:

Any two engineers could solve that problem.

$\mathcal{NLL}$ follows Link in viewing this sort of quantification as peripheral. One can formulate such meanings, but we have not worked on supporting inferences connected with it.

[10] But the reverse need not't hold, i.e. the fact that a given projection is distributive does NOT imply that it is individual-based (in the sense defined). In particular, given that a relation R holds of an n-tuple of groups $G_1, \ldots, G_n$, the mere fact that one of R's projections is distributive does not't guarantee that the atomic i-parts of a given group G_i stand in the R relation to i-parts of other groups $G_1, \ldots, G_{i-1}, G_{i+1}, \ldots, G_n$ etc. This is what is required for the cumulativity.

It is of course fine for there to be (determined) NP's which are interpreted not as generalized quantifiers, but rather as simple referring expressions, i.e., restricted parameters. Cf. Nerbonne et al. 1990 for a development along of plural logic with indefinite referring expressions. But this complication will not concern us below.

We turn now to plural (and other) quantification, which, although it can be reduced to quantification over individuals, interacts with assumptions about the plural domain in crucial ways. The general $\mathcal{NLL}$ tack here is to show how measure terms can function as determiners, and we shall sketch the developments need here. The section below is of necessity fairly dense, focusing on issues which arise in embedding this sort of quantification in $\mathcal{NLL}$, but the general issues are discussed at much greater length in Nerbonne 1994. I should also note that the background for the treatment below is furnished by Krifka 1991. See Nerbonne 1994 and Krifka 1991 for further references.

8.4.1 Plurals, Mass Terms and Measure Determiners

Plural and mass objects are MEASURABLE; for plural objects, cardinality is the salient measure, for mass objects, weight and volume are normally the more useful measures. Figure 3 illustrates the function of measuring: mapping a structured domain onto a set of MEASURES. The measures are of course expressed by $\mathcal{NLL}$ measure terms (cf. § 6.5). It is clear that measure mappings should respect (homomorphically) the plural/mass structure, e.g., that the measure of the sum of (nonoverlapping) objects should be the sum of the component measures. The relevant requirement (cf. Krantz et al. 1971) is that the measure be EXTENSIVE or ADDITIVE. We assume an ordered set of measures **MEAS**, and formulate the requirements as follows:

μ is a *measure* function $\stackrel{\text{def}}{=}$

$$\mu : E \mapsto \mathbf{MEAS} \qquad \text{e.g. } \mathbf{MEAS} = \mathrm{N}, R^{+} \cup 0$$

$$x \sqcap y = 0 \;\rightarrow\; \mu(x \sqcup y) = \mu(x) + \mu(y) \qquad \textbf{Additive}$$

$$x \sqsubseteq y \wedge \mu(x) \neq 0 \;\rightarrow\; \exists n > 0 \quad n \cdot \mu(x) \geq \mu(y) \qquad \textbf{Multiplicative (Archimedean)}$$

It is worth noting that these requirements cannot be placed on relations between individuals and measures in general—not even on those relations which in some sense "measure" individuals. There are any number of measures of individuals which do not obey these axioms, e.g., temperature or

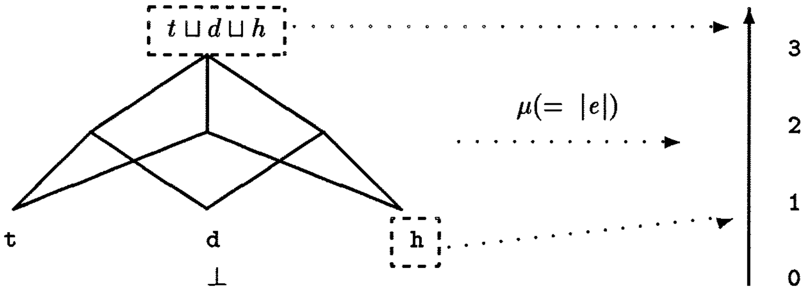

Figure 3: Measure functions μ maps elements of the domain, E, onto ordered measures (of various dimensions). Cardinality is a measure function over the plural lattice.

height. But relations which give rise to determiners appear to be exclusively extensive.

Cardinality is a measure function in the sense defined above. It maps the plural domain to N, and is additive and multiplicative in the required senses. The requirement that measures be additive and multiplicative was needed in particular to treat some of the complex measure phrases we examined above § 6.5.

It is useful to specify object-language expressions for the measure functions just introduced.

- **f-card, f-size-lb, f-size-kg, f-size-liter,**... are extensive measure functions with values are in $\mathbb{N}$ bzw. $\mathbb{R}$.
- **size** is a functional relation between objects and measures. I.e. for all objects ?x, measures ?m, ?m′

```
IF(AND{  size(th:?x meas:?m)
         size(th:?x meas:?m′)},   =m(th:?m pole:?m′))
```

- **card** denotes a functional relation between individuals and their cardinalities. For all ?x

```
card(th:?x meas:f-card(?x))
```

Our general strategy in this section will be to show how a theory of plural (and mass term) quantification arises from the theory of extensive measurement of plural (and mass term) measurement. This is largely independent of $\mathcal{NLL}$ particulars, which therefore play a mere facilitating role once the theoretical background is clear.

8.4.2 Deriving Determiners from Measures—Problems

In general, measures contribute to quantifiers by providing RESTRICTORS. The basic idea is simple: given a measure m we wish to derive a determiner D_m. This is accomplished by obtaining the inverse image of m under a

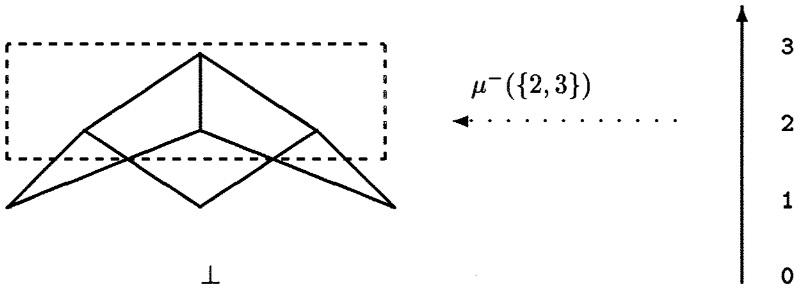

Figure 4: $\mu^-(\{2,3\})$, image of $\{2,3\}$ under the inverse measure function.

measure function μ, i.e. $\mu^-(m)$, which is then in turn available to restrict an (existential) determiner. Thus e.g.

(DET_m ?x $\phi\,\psi$) iff (exists ?x AND { in(th:?x loc:μ^-(m))
ϕ }
ψ)

This simple basic picture is complicated (i) by comparatives and other modifiers of measurement phrases, which lead us to consider not just single measures, but variously specified SETS of measures; and (ii) by the plural structure on the domain of discourse E, in particular the conditon on (distributive) predicates that they be closed under $\sqcup_i$. We take up these issues in turn.

Comparatives (*more than one* or *more than one ounce*) refer not to a single measure, but to specified sets of measures (those greater than one or those greater than one ounce). Allowing reference to sets of measures is straightforward, however. Figure 4 illustrates the obvious generalization from taking the inverse of a single measure to taking the image of a set of measures under the inverse measure function. The definitions below map SETS of measures onto determiners. For example, we can now generalize the definition above:
For $M \subset$ **MEAS** define binary DET_M:

(DET_M ?x $\phi\,\psi$) iff (exists ?x AND { in(th:?x loc:μ^-(M))
ϕ }
ψ)

The $\mathcal{NLL}$ category of MEASURE TERM includes specified measures (§ 6.5.2) and maximally specified measures (§ 6.5.3), which refer indefinitely to measures fulfilling a given description. Thus a general scheme allowing the derivation of a determiner from a measure term will be general enough for these purposes.

We turn then to the second problem, i.e., the interaction of plural struc-

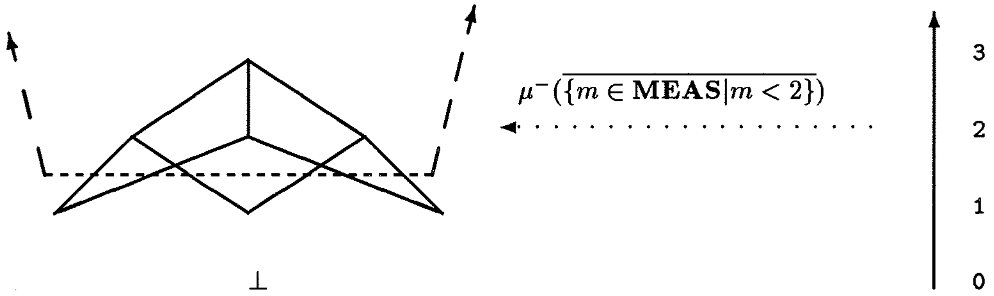

Figure 5: Inverse image of complement of $\{m \in \mathbf{MEAS}|m < 2\}$, downwardly closed measure set. Derived determiner "fewer than 2" lives on this image.

ture and comparative determiners. The rough construal of measure-based determiners (as providing restrictors on quantifiers), needs further generalization to deal with non-upwardly-closed measure specifications. This may be seen in examples such as the following:

(1) Fewer than 3 children sang.

Let's assume that $[\![sang]\!]$ is closed under $\sqcup_i$, so that if $x, y \in [\![sang]\!]$ then $x \sqcup_i y \in [\![sang]\!]$. This means e.g. that if the 2-sets in Figure 4 are in $[\![sang]\!]$, so is their join, the 3-set. But, by the simple derivation of measure determiners proposed above, sentence last (1) could be true, since there's a 2-set with the required properties! This is is clearly incorrect, and, moreover, it's the direct result of working in the plural structure. This structure must inform the derivation of determiners from measures and measure sets. In this case, we'd like the result that *fewer than n* holds of P, Q just in case there's no entity of size n or greater such that P and Q may be predicated of it.

8.4.3 Determiners Derived from Measure Sets

The general scheme for deriving determiner meanings from the specification of measure sets is as follows:

Measure Determiners Let **MEAS** be the range of a measure function, ordered by $\leq_{\mathbf{MEAS}}$, and let $M \subseteq \mathbf{MEAS}$. We obtain the determiner based on M, DET_M:

```
(DET_M ?x φ ψ) iff  (max≤ ?m  AND{  measure(inst:?m)
                                    (exists ?x  size(th:?z meas:?m)
                                                AND{φ ψ})
                                   }
                                in(th:?m loc:M) )
```

> Given this definition of the effects of the measure determiners depend only on their denoting a set of measures. We attend to this in the following.

I.e., among the measures in M is the maximal measure of objects satisfying the predicates ϕ and ψ. Note that this handles the "fewer than 3" as well as the "more than 3" cases. The "fewer than 3" case comes out right because the definition here requires that the maximal measure (max) satisfying the properties involved falls within the measure set.[11]
Nerbonne 1994

demonstrates logical and algebraic properties of these determiners and how they depend on the properties of the sets of measures M they're based on, especially whether M is upwardly (downwardly) closed, or convex. In particular, upwardly closed measure sets give rise to existential quantifiers and downwardly closed ones to negative existentials. The significance of the reductions is twofold. On the one hand, they are useful when it comes to adducing monotonicity properties, because existential and negative existential quantifiers are well-studied. More interestingly, from the point of view of design for meaning representation languages, the reductions show that the properties of complex determiners ("more than 3") arise from the (closure) properties of the measures sets they are derived from. Since these in turn are inherent in comparison, we have an opportunity to derive complex determiner meanings from the type of comparison involved.

8.4.4 Language Definitions

⟨Determiner⟩ ::= ⟨Simple Determiner⟩ | ⟨Complex Determiner⟩
⟨Complex Determiner⟩ ::= (⟨Maximally Specified Measure⟩)

In this section we wish to stipulate how measure terms may be uniquely associated with a set of measures in order to take advantage of the very general relationship developed above—between any set of measures and an associated determiner. The intuitive relation is quite straightforward: simple measures such as `5, [5 kg]`, and `[500 ml]` are associated with the singleton set whose only member is their own denotation (wrt $(\mathfrak{A}, \beta)$). In examining specified measures such as `{> 5}, {>= [5 kg]}`, and `{< [500 ml]}`, it is more convenient to examine the basic restricted parameter form, i.e., `(?x| >(th:?x pole:5))`, and `(?x| >(th:?x pole:[5 kg]))`.

Let us furthermore use the notation '$\Diamond D(t, \mathcal{M})$' to denote the set of possible denotations of a term wrt a model. (This is a bit similar to the notion of a relation expressed by an open formula we introduced above, in § 6.4, but we shall abstract away from variables in the term under evaluation) .

[11] For measure sets in the reals, maxima may not exist, so that we should prefer to use suprema (least upper bounds) rather than maxima. On the other hand, we'll never measure such suprema, so that this may be a nicety. The proof of reduction for the case of ↑-closed M (Nerbonne 1994) requires that maxima, not suprema be available.

- For all models $\mathcal{M} = (\mathfrak{A}, \beta)$, and t a term, the possible denotations of t with respect to $(\mathfrak{A}, \beta)$ is given by the formula below.

$$\Diamond D(t, \mathcal{M}) = \begin{cases} \{a | [\![\phi]\!]_{\mathfrak{A}\beta\frac{a}{?\mathtt{x}}} = 1\} & \text{if } t \text{ a restricted parameter w. form } (\mathtt{?x}|\,\phi) \\ \{[\![t]\!]_{\mathcal{M}}\} & \text{otherwise} \end{cases}$$

For constants c, this is simply the singleton $\{[\![\mathtt{c}]\!]_{\mathcal{M}}\}$, while for restricted parameters—the fundamental means of expressing specified measure phrases such as *more than five* '[>5] (= (?x| >(th:?x pole:5)))', the set of possible denotations is

$$\{a|\,[\![\mathtt{>(th:?x\ pole:5)}]\!]_{\mathfrak{A}\beta\frac{a}{?x}} = 1\}$$

which is of course just $\{6, 7, \ldots\}$. Note that the set of measures associated with every measure term is a subset of **MEAS**. We need this for the deployment of the '$\Diamond D$' sets as measure determiners.

We have thus stipulated the semantics of formulas which employ measure terms as determiners. The final step is simply the combination of the definition of the possible denotations of a measure term (immediately above) with the definition of measure determiners (in § 8.4.3 above).

It is clear that we are now employing measure terms polymorphically—both in order to denote measures, which we have construed as numbers or pairs consisting of numbers and scales, and also as determiners, roughly as existential or negative existential determiners with an additional assertion about the size of the maximal satisfying instance.

8.4.5 Simple Examples

The discussion above shows that determiner definitions follow once measure sets are provided. This is quite general; the definitions are available not only for measure sets provided by comparative phrases, but for measure sets quite generally. It is now time to provide some examples, both for the sake of further clarification, and in order to illustrate how the closure properties of measure sets can be put to use. In each case, we assume information about closure properties in order to provide reasonable determiner definitions.

We consider first upwardly closed measure sets M, e.g., those associated with the measure terms *more than two, more than two liters*, and *at least two liters*. Given the treatment above, these will hold of ?x, ϕ, ψ iff the greatest β(?x) has a measure in M. If we examine relations on ϕ and ψ which are more inclusive, then the largest element must be $\geq \beta$(?x in measure, thus also in M. Thus the derived determiner must be upwardly monotonic in both the left and right positions (cf. van Benthem 1983 for discussion of the monotonicity properties of determiners). By a similar argument we can show that downwardly closed measure sets give rise to (left and right) downwardly monotonic determiners (cf. Nerbonne 1994 for details).

Upwardly closed measure sets Natural language examples include: *More than 2 children sang, More than 2 liters of water spilled, At least 2 children sang,* etc.

```
(({> [2l]}) ?x water(inst:?x) spill(th:?x))
```

Downwardly-closed measure sets (that are not also upwardly-closed) Natural language examples include: *Fewer than 2 children sang, Less than 2 liters of water spilled, At most 2 children sang, Not more than seven children sang,* etc.[12]

```
(({<= 2}) ?x child(inst/i:?x) sang(th/i:?x))
```

8.5 Location Terms

$\mathcal{NLL}$ implements a version of the theory of locative reference detailed in Creary et al. 1989, according to which locative expressions such as *south of Page Mill Rd.* denote regions, which may stand in relations to nonregional individuals (frequently playing a `location` role), and which are organized in a lattice structure, in which simple juxtaposition normally denotes the lattice meet operation.

We thus require that a subset of the universe of discourse be $\mathcal{R}$, the set of regions, and that these have the structure of a meet-semilattice. Let '$\sqcap_{\mathcal{R}}$' denote the lattice meet operation, and '$\sqsubseteq_{\mathcal{R}}$' the subsumption relation.

⟨Location Term⟩ ::= reg-X{⟨Term⟩, ...}

Location expressions are functional terms denoting regions. These are of two general sorts, simple—locative function terms—and intersective—location terms. The simple ones consist of a locative function applied to an appropriate argument, while the intersective ones consist of a regional intersection functor `reg-X` applied to a set of regions. As an example, consider the following location term, which might serve as the translation (in a given context) of the iterated locatives *on the Ohio in Kentucky near Illinois*:

```
reg-X{f-on(the-ohio),f-in(kentucky),f-near(illinois)}
```

This is a location term whose components are locative function terms. `f-near` should denote (e.g.) a function that maps Illinois onto a region beginning at its borders and extending out a short distance.

The functor of an intersective location term denotes the regional intersection function, which maps r_1, r_2, $\ldots$, r_n onto their intersection r. The commutativity and associativity of the lattice meet operation justify specifying its arguments via sets. The order-indifference of set specification accounts for the permutability of locative component arguments

[12] Cf. Nerbonne 1994 for discussion of apparent counterexamples to the downward monotonicity of these determiners.

(Creary et al. 1989). We will also make use of the familiar lattice theorem: $r_1 \sqcap_{\mathcal{R}} r_2 \sqsubseteq_{\mathcal{R}} r_1$, according to which location terms must denote subregions of their component arguments.

This is codified in two further requirements on $\mathcal{NLL}$ models:

- for all locative functions, `f-loc`,

$$[\![\texttt{f-loc}]\!] \mapsto \mathcal{R}$$

- $[\![\texttt{reg-X}]\!] = \sqcap_{\mathcal{R}}$

8.5.1 Located Predications

This is a fact about (most) situations being located in space: if an event or state occurs or obtains within a region r, then it occurs or obtains within any region r' containing r, i.e. there is an upward monotonicity for the location arguments of relations. This accounts for the correctness of locative-simplifying inferences, such as:

	Al works in NY on 5th Ave		Al works in NY on 5th Ave
∴	Al works in NY	∴	Al works on 5th Ave

On the other hand, there are relations between individuals and regions in which the region argument is not upwardly monotonic—e.g., the relation of being the tallest individual in a given region. In $\mathcal{NLL}$ we reserve the **`location`** role (**`loc`**) for the designation of location argument positions which ARE upwardly monotonic.

- for all relations R, the argument position denoted by the **`location`** role is upwardly monotonic: for all n tuples $\langle a_1 \ldots a_l \ldots a_n \rangle, 1 \leq l \leq n$ in R, where a_l is the value of the locative projection, $i_{\texttt{loc}}$, then for all $a'_l \ni a_l \sqsubseteq_{\mathcal{R}} a'_l \quad \langle a_1 \ldots a'_l \ldots a_n \rangle \in R$

This concludes our presentation of $\mathcal{NLL}$ models.

9 Conclusions and Prospects

Although $\mathcal{NLL}$ contains no profound expressive novelty, still it contains a great enough range of logical devices normally studied in isolation for its model theory to be nontrivial. The point of laying it out here was to provide a firm foundation for work in inference and in interfacing $\mathcal{NLL}$ both to natural language grammars and to applications, knowledge representation components, and speech act representation.

Although we see need for improvement in the model-theoretic treatment of questions, in the introduction of sorts, and especially in allowing some nonextensionality—each of which would probably imply a need for modifications in the model theory, nonetheless further development of the expressive capabilities of $\mathcal{NLL}$ is NOT a focus of current work.

Instead, our current work (1992) focuses on making $\mathcal{NLL}$ easier to use in concrete NLP applications.

- provision of a high-level language for the specification of interfaces and inference (Joachim Laubsch).
- development of a concrete theory of temporal reference (along Quinian lines, i.e., without introducing propositional operators). This could prompt minor modifications to the model theory. (joint work with Walter Kasper).
- investigation of issues in sortal disambiguation, especially the extent to which an integration with semantic representation is desirable (Nerbonne 1992b)
- provision of interface tools for use in connection with feature-based grammars (Diagne and Nerbonne 1992).

These foci arise from a conviction that good software should not only be theoretically well-understood, but that it must meet the demands of practice as well. We hope that $\mathcal{NLL}$ can contribute both to a better understanding and to better application of computational semantics.[13]

10 Acknowledgements

Joachim Laubsch and I originally implemented and employed $\mathcal{NLL}$ in the Hewlett-Packard Labs HP-NL project (Nerbonne and Proudian 1987). We are extremely grateful to Derek Proudian, who collaborated extensively in early phases (especially, but by no means exclusively in software design) and Masayo Iida, who developed mappings from HPSG phrase structures into $\mathcal{NLL}$. Dan Flickinger, Brett Kessler and Diana Roberts stoically helped us through various problems in early development and integration phases.

The semantic ideas in $\mathcal{NLL}$ owe a good deal to conversations with Lew Creary, Mark Gawron, Bill Ladusaw and Carl Pollard. For specific remarks about the presentation here I thank Jürgen Allgayer, Kader Diagne, Judith Engelkamp, Gregor Erbach, Walter Kasper, Karsten Konrad, Brigitte Krenn, Hans-Ulrich Krieger, Holger Maier, Karel Oliva, Carola Reddig, Werner Saurer and Espen Vestre.

This work was supported by research grants ITW 9002 0 and ITV 9102 from the German Bundesministerium für Forschung und Technologie to the DFKI DISCO and ASL projects led by Hans Uszkoreit (DISCO) and Hans Uszkoreit and me (ASL).

References

Alshawi, H., et al. July 1989. Research Programme in Natural Language Processing. Final report. SRI Cambridge Research Centre: Alvey Project No.

[13] We would like to make $\mathcal{NLL}$ available as public domain software during 1992 or sometime in early 1993, and we have written assurance of cooperation from the owner of (most of) its current implementation, the Hewlett-Packard Corporation. If you would like to be informed of its availability, please write nerbonnedfki.uni-sb.de, and I will include you on a mailing list for announcements.

ALV/PRJ/IKBS/105.

Bach, Emmon. 1986. The Algebra of Events. *Linguistics and Philosophy* 9(1):5–16.

Barendregt, Henrik P. 1984. *The Lambda Calculus: Its Syntax and Semantics.* Amsterdam: Elsevier.

Barwise, Jon. 1987. Noun Phrases, Generalized Quantifiers, and Anaphora. In *Generalized Quantifiers and Natural Language*, ed. Peter Gärdenfors. 1–29. Dordrecht: Reidel.

Barwise, Jon, and Robin Cooper. 1981. Generalized Quantifiers and Natural Language. *Linguistics and Philosophy* 4(2):159–219.

Barwise, Jon, and John Perry. 1983. *Situations and Attitudes.* Cambridge: MIT Press.

Creary, Lewis G., J. Mark Gawron, and John Nerbonne. 1989. Reference to Locations. In *Proc. of the 29th Annual Meeting of the Association for Computational Linguistics*, 42–50.

Creary, Lewis G., and Carl J. Pollard. 1985. A Computational Semantics for Natural Language. In *Proc. of the 25th Annual Meeting of the Association for Computational Linguistics*, 172–179.

Cresswell, Max J. 1976. The Semantics of Degree. In *Montague Grammar*, ed. Barbara H. Partee. 261–292. New York: Academic Press.

Dalrymple, Mary, Stuart M. Shieber, and Fernando C.N.Periera. 1991. Ellipsis and Higher-Order Unification. *Linguistics and Philosophy* 14(4):399–452.

Date, C. J. 1986. *An Introduction to Database Systems.* Addison-Wesley. 4 edition.

Devlin, Keith. 1991. *Logic and Information.* Oxford: Oxford University Press.

Diagne, Abdel Kader, and John Nerbonne. 1992. Flexible Semantics Communication in Integrated Speech/Language Architectures. In *KONVENS 92*, ed. Günter Göerz, 348–352. Berlin. Springer.

Dowty, David. 1991. Thematic Proto-Roles and Argument Selection. *Language* 67(3).

Ebbinghaus, Heinz-Dieter, Jörg Flum, and Wolfgang Thomas. 1978. *Einführung in die Mathematische Logik.* Darmstadt: Wissenschaftliche Buchgesellschaft.

Fillmore, Charles. 1985. Pragmatically Controlled Zero Anaphora. In *Berkeley Linguistic Society.*

Fitting, Melvin. 1990. *First Order Logic and Automated Theorem Proving.* New York: Springer Verlag.

Flickinger, Daniel, and John Nerbonne. 1992. Inheritance and Complementation: A Case Study of *Easy* Adjectives and Related Nouns. *Computational Linguistics* 19(3):269–309.

Gawron, Jean Mark, and Stanley Peters. 1990. *Anaphora and Quantification in Situation Semantics.* Stanford University: CSLI Lecture Notes.

Ginzburg, Jonathan. 1992. *Questions, Queries, and Facts: A Semantics and Pragmatics for Interrogatives.* Doctoral dissertation, Stanford University, Palo Alto, March.

Groenendijk, Jeroen, and Martin Stokhof. 1984. *Studies on the Semantics of Questions and the Pragmatics of Answers.* Doctoral dissertation, University of Amsterdam, Amsterdam.

Johnson, Mark. 1988. *Attribute Value Logic and the Theory of Grammar.* Stanford: Center for the Study of Language and Information.

Keenan, Edward L., and Lawrence Moss. 1984. Generalized Quantifiers and the Expressive Power of Natural Language. In *Generalized Quantifiers*, ed. Johan van Benthem and Alice ter Meulen. 73–124. Dordrecht: Foris.

Krantz, David H., R. Duncan Luce, Patrick Suppes, and Amos Tversky. 1971. *Foundations of Measurement.* New York: Academic Press.

Krifka, Manfred. 1991. Nominal Reference, Temporal Constitution and Quantification in Event Semantics. In *Semantics and Contextual Expressions*, ed. Renate Bartsch, Johan van Benthem, and Peter van Emde Boas. Dordrecht: Foris.

Laubsch, Joachim. 1992. The Semantics Application Interface. In *Applied Natural Language Processing*, ed. Hans Haugeneder. ??address: ??publisher.

Laubsch, Joachim. December 1989. Logical Form Simplification. STL report. Hewlett-Packard.

Laubsch, Joachim, and John Nerbonne. 1991. An Overview of $\mathcal{NLL}$. Technical report. Palo Alto: Hewlett-Packard Laboratories, July.

Link, Godehard. 1983. The Logical Analysis of Plurals and Mass Terms: A Lattice-Theoretical Approach. In *Meaning, Use, and the Interpretation of Language*, ed. Rainer Bäuerle, Urs Egli, and Arnim von Stechow, 302–323. Berlin. de Gruyter.

Link, Godehard. 1987. Generalized Quantifiers and Plurals. In *Generalized Quantifiers and Natural Language*, ed. Peter Gärdenfors, 151–180. Dordrecht. Reidel.

Lønning, Jan Tore. 1989. Some Aspects of the Logic of Plural Noun Phrases. Technical Report 11. Department of Mathematics, University of Oslo.

Nebel, Bernhard, and Gert Smolka. 1991. Representation and Reasoning with Attributive Descriptions. In *Sorts and Types in Artificial Intelligence*, ed. K.-H. Bläsius, U. Hedstück, and C.-R. Rollinger. 112–139. Berlin: Springer Verlag.

Nerbonne, John. 1992a. Constraint-Based Semantics. In *Proc. of the 8th Amsterdam Colloquium*, ed. Paul Dekker and Martin Stokhof, 425–444. Institute for Logic, Language and Computation. also DFKI RR-92-18.

Nerbonne, John. 1992b. Natural Language Disambiguation and Taxonomic Reasoning. In *DFKI Workshop on Taxonomic Reasoning (DFKI D-92-08)*, ed. Jochen Heinsohn and Bernhard Hollunder, 40–47. Saarbrücken. DFKI.

Nerbonne, John. 1994. A Semantics for Nominal Comparatives. In *Proc. of the 9th Amsterdam Colloquium*, ed. Paul Dekker and Martin Stokhof, 487–506. Institute for Logic, Language and Computation.

Nerbonne, John, Masayo Iida, and William Ladusaw. 1990. Semantics of Common Noun Phrase Anaphora. In *Proc. of the 9th West Coast Conference on Formal Linguistics*, ed. Aaron Halpern, 379–394. Stanford. CSLI.

Nerbonne, John, and Joachim Laubsch. 1990. Logics for Meaning Representation. unpub. documentation, Hewlett-Packard Laboratories, Palo Alto, Feb.

Nerbonne, John, and Derek Proudian. 1987. The HP-NL System. Technical report. Hewlett-Packard Labs.

Pinkal, Manfred. 1989. Die Semantik von Satzkomparativen. *Zeitschrift für Sprachwissenschaft* 8(2):206–256.

Scha, Remko. 1981. Distributive, Collective and Cumulative Quantification. In *Truth, Interpretation and Information*, ed. Jeroen Groenendijk, Theo Janssen, and Martin Stokhof. 131–158. Dordrecht: Foris.

Scha, Remko, and David Stallard. 1988. Multi-Level Plurals and Distributivity. In *Proc. of the 26th ACL Meeting*, 17–24. ACL.

Schubert, Lenhart K., and Francis Jeffrey Pelletier. 1982. From English to Logic: Context-Free Computation of Conventional Logic Translations. *Journal of the Association for Computational Linguistics* 165–176.

Sondheimer, Norman K., Ralph M. Weischedel, and Robert J. Bobrow. 1984. Semantic Interpretation Using KL-ONE. In *Proc. of COLING84*, 101–107. ACL.

Steele, Guy. 1984. *Common Lisp: The Language.* Burlington, Massachusetts: Digital.

Turner, Raymond. 1984. *Logics for Artificial Intelligence.* West Sussex: Ellis Horwood.

van Benthem, Johan. 1983. Determiners and Logic. *Linguistics and Philosophy* 6:447–478.

van Benthem, Johan. 1989. Polyadic Quantifiers. *Linguistics and Philosophy* 12(4):437–464.

Westerståhl, Dag. 1989. Quantifiers in Formal and Natural Languages. In *Handbook of Philosophical Logic, Vol.IV*, ed. Dov Gabbay and Franz Guenthner. 1–132. Dordrecht: Reidel.

Westerståhl, Dag. 1990. Parametric Types and Propositions in First-Order Situation Theory. In *Proc. of the First Conference on Situation Theory and Its Applications*, ed. Robin Cooper, Kunikai Mukai, and John Perry, 193–230. Stanford University. CSLI Lecture Notes.