Toward a Logical Description of Double-R Grammar

Jerry T. Ball

Last Updated on 16 July 2015

Introduction

This paper presents a logical description of Double-R Grammar – a computational Cognitive Grammar of English, implemented as a large-scale cognitive model in the ACT-R Cognitive Architecture (Anderson, 2007). The intent is to describe Double-R Grammar in a way that abstracts away from the underlying computational implementation in ACT-R, where possible. It is hoped that this description will facilitate understanding of the basic representation and processing commitments. We will occasionally need to make reference to the computational implementation, where a declarative logical representation has not yet been worked out. Full details of Double-R Grammar are available as Ball et al. (2014).

In this paper, the pronoun we is used throughout. In part, this reflects a personal proclivity of the author. In part, it reflects the collaborative nature of the broader research program of which Double-R Grammar is a part. That broader research program is focused on the development of language and task enabled synthetic teammates for use in team training simulations (cf. Ball et al., 2010). Key collaborators include Chris Myers, Mary Freiman, Stu Rodgers, Michelle Caisse, Nancy Cooke, Steve Shope and Andrea Heiber. Many others have also contributed. Pursuit of the broader research program would not have been possible without the help of these individuals and the extensive support of the Warfighter Readiness Research Division and higher level organizations of the Air Force Research Laboratory. Support from the Office of Naval Research is also acknowledged. Computational implementation of Double-R Grammar in its present form would not have been possible without the ACT-R cognitive architecture which is based on 40+ years of computational and psychological research. Thanks to Mark Burstein for adding a multiple-inheritance capability to ACT-R.

The focus of Double-R Grammar is on the grammatical encoding of referential and relational meaning. Relational meaning is encoded by relational lexical items and their associated arguments, classic predicate-argument structure. Referential meaning is encoded by the referring expressions which function as the arguments of predicates, but also by the situation that the predicate-logic expression, as a whole, describes. In Double-R Grammar there are two core types of referring expression; object referring expression and situation referring expression. These are not the only types of referring expression – others include location referring expression and oblique referring expression – but they are the focus of this paper. There are also subtypes of these referring expressions – e.g. wh-question situation referring expression, existential there situation referring expression, time referring expression, manner referring expression – which will not be discussed.
Object referring expressions are more usually called noun phrases (NP) or nominals in other grammatical treatments. Situation referring expressions are more usually called clauses or sentences (S). Since the focus of Double-R Grammar is on the grammatical encoding of referential and relational meaning, the use of the term referring expression is appropriate, despite the fact that all representations in Double-R Grammar are grammatical. This aligns with the Cognitive Linguistic roots of Double-R Grammar and the assumption that grammatical representations are semantically motivated and not purely syntactic. It also aligns with the following position from Lyons (1977, p. 445): “Looked at from a semantic point of view, nominals are referring expressions”.

In Double-R Grammar, meaning is captured in the grammatical representations in combination with their mapping into a situation model – the mental stage for building non-linguistic representations of described situations, along with a secondary mapping from the situation model into the mental universe of accumulated experience of an individual. The referring expressions in the grammatical representations refer to objects and situations in the situation model. These objects and situations may in turn, be linked to objects and situations in the mental universe of experience.

The grammatical representations are more expressive than typical first order predicate logic representations. The situation model representations are more expressive than set theory which is broadly used to provide the model that is associated with a model-theoretic predicate logic. Our basic approach is to assume the validity of the grammatical representations and to make adjustments to the logical notation and the situation model, where needed, to support the description of the grammatical representations in the logical notation, and to support the interpretation in the situation model.

Grammatical representations in Double-R Grammar align with elements of a number of different logical approaches. We borrow from Situation Semantics (Barwise & Perry, 1983) in giving situation referring expressions first class status, and viewing situation model representations as partial, rather than complete worlds. We borrow from Davidson (1967) with respect to the use of event variables which provide a handle for events, effectively giving them first class status. We extend the idea of an event variable, with the addition of handle variables, to support the representation of grammatical structure. These handle variables are reminiscent of the handle variables used in Minimal Recursion Semantics (Copesteak et al., 2005), although they serve the purpose of representing grammatical structure, rather than being used to (under)represent quantifier scope. We leave quantifier scope underspecified, as occurs in Minimal Recursion Semantics and in Allen (1995), two logically based computational treatments. We borrow ideas incorporated into neo-Davidsonian representations (Parsons, 1990), which introduce semantic roles (e.g. agent, patient) for the semantic categorization of the arguments (and adjuncts) of predicates. However, we replace semantic roles with semantically motivated grammatical functions (e.g. subject, object) – Parsons (1990) did just the opposite, starting with grammatical functions and replacing them with semantic roles. We adapt ideas from Montague Grammar (Montague, 1973), using λ abstraction to represent referring expressions. We borrow from Discourse Representation Theory (DRT) (Kamp, 1981), eliminating the direct representation of existential and universal quantification, and treating indefinite as well as
definite descriptions as referring expressions. We also dynamically construct the situation model which provides the referents of the referring expressions, as is done in DRT and other dynamic logics. We add grammatical features to representations to support co-reference resolution, something that is seldom done in logical approaches, but which is necessary for a computational implementation. The encoding of grammatical structure, grammatical functions and grammatical features is crucial for co-reference resolution, as we know from linguistic studies (cf. Chomsky, 1981). We even encode the part of speech in our semantically motivated grammatical representations, since we view parts of speech as abstract semantic categories, in accordance with Cognitive Grammar (Langacker, 1987, 1991) and traditional grammar. The result of all these borrowings and adaptations is fully articulated grammatical representations which squeeze out as much semantically motivated grammatical information from the linguistic input as is possible. We call these representations \textit{logico-grammatical}.

In this paper, we do not focus on truth conditions, quantifier scope or logical inference, three key topics in logic. From a logical perspective, truth conditions can be formulated in terms of an object language mapping into a meta-language, as in Tarski’s classic example:

\begin{quote}
\textit{snow is white} if and only if \textit{snow is white} (Tarski, 1944)
\end{quote}

In this example, the conditions under which the object language expression \textit{snow is white} is true, are provided by the meta-language statement \textit{snow is white}. From a computational cognitive linguistic perspective, we view the meta-language statement as a promissory note to create a non-linguistic mental representation of the truth conditions expressed by the meta-language statement. In Double-R Grammar, the non-linguistic mental representation of truth conditions is the primary function of the situation model. The actual truth or falsity of an assertion may be derived from the situation model representation. Is the described situation, as represented in the situation model, compatible with the mental universe of experience of an individual? Currently, situation model representations in Double-R Grammar are ill-defined and impoverished. We know that they are more expressive than set theory. We know that “individuals” come in many different types: e.g. objects, situations, locations, times, instances, kinds, classes, collections, even empty collections (Ball, 2012). To use Hobbs’ (1985) term, situation model representations are \textit{ontologically promiscuous}. We know that situation model representations are non-linguistic. However, we do not know how to adequately represent individual concepts, let alone propositions or images. We do not view situation model representations as a “language of thought”, although the representations can have structure. Like our view of Tarski’s formulation of truth conditions in a meta-language, Double-R Grammar provides a promissory note to provide adequate situation model representations once a theory and computational implementation have been more fully worked out.

We represent quantifiers in situ, leaving quantifier scope underspecified, as is common in computational treatments. However, it may be that quantifier scope can be resolved in the translation into situation model representations.

We have not yet formulated any mechanisms of logical inference beyond default inheritance, although the deductive mechanisms of Episodic Logic (Schubert, 2000) and
the abductive mechanisms of Hobbs, Stickel & Martin (1993) may be compatible with Double-R Grammar.

To date, the developmental focus of Double-R Grammar has been on grammatical expressiveness and achieving broad grammatical and lexical coverage. Is Double-R Grammar capable of representing the major grammatical constructions of English? Can it also represent many of the minor grammatical constructions? Does the mental lexicon approach the size of the human mental lexicon? We believe the answer to the first question is “yes” – at least in terms of the creation of grammatical representations, and we are working to make the answer to the second question “yes”, as well. In terms of the mental lexicon, Double-R Grammar includes 58,000 lexical entries, which approximates some estimates of the size of the human mental lexicon. However, it lacks much of the fine-grained knowledge of the meaning of words that human possess, being largely restricted to encoding grammatical information about words (e.g. animacy, gender, number and person for nouns; tense, aspect, and modality for verbs).

Having achieved sufficiently broad coverage of the major grammatical constructions of English, we are now in a position to begin to focus on the mapping of grammatical representations into the situation model, and the linking to the mental universe, although there are difficult theoretical issues that will need to be overcome. We are also in a position to consider the finer shades of meaning that are needed to model ambiguities like modifier scope (e.g. prepositional phrase attachment, sentence vs. predicate modification), quantifier scope, scope of conjunction and disjunction, co-reference of definite descriptions (as opposed to pronouns which we already model using grammatical features), lexical ambiguity, metaphor, metonymy, synonymy, antonymy, etc.

In providing a computational implementation, Double-R Grammar is explicitly formal. It is possible to submit a linguistic expression or discourse to Double-R Grammar and see the computational results. In providing a logical description, this paper expresses the formalism in a way that makes it accessible without having to execute the program. However, ultimately, the program and its execution provide the actual formalism. In building the computational implementation, many representational and processing choices were initially made that had to be altered to improve consistency, and to achieve better overall performance. Despite these improvements, many inconsistencies remain and additional improvements are foreseen. The computational implementation makes it possible to formally explore and investigate complexities of language that would otherwise be out of reach. This is the real power of a computational implementation. It is also one of the reasons that automated machine learning techniques, which change the basic nature of exploration, have been largely avoided.

We begin with a consideration of the logical representation of object referring expressions from the perspective of predicate logic, extended with generalized quantifiers, the equality operator and \( \lambda \) abstraction.
The Logical Representation of Object Referring Expressions in Predicate Logic and Extensions

In Double-R Grammar, an object referring expression is a type of expression that refers to an object (or reified object) in the situation model. In traditional grammar, object referring expressions are more usually called noun phrases (NPs), but this terminology causes problems since many noun phrases are lacking a noun head. The more recent trend to call such expressions determiner phrases (DPs) headed by a determiner is even more problematic, since object referring expressions are more likely to be missing a determiner than a noun. The treatment of the determiner as the head is called the functional head hypothesis (Abney, 1987). The functional head hypothesis is broadly accepted in Chomskyan grammar. However, as I like to say, “the functional head hypothesis is wrong-headed”. Or as McCawley laments “...all sorts of things that to me are obvious modifiers now get represented as heads of things they aren’t heads of” (Cheng & Sybesma, 1998). Double-R Grammar is based on the assumption that both the specifier, which McCawley didn’t recognize as a grammatical function distinct from modifier, and the head make important grammatical contributions to encompassing expressions (Ball, 2007). The Double-R Grammar treatment aligns with Cann’s (1999) treatment of the specifier as a secondary head, although the term secondary head is not used. In Double-R Grammar, the head is the semantically most significant element of a referring expression. In the case of an object referring expression, the head determines the type of object that is referred to. The specifier functions to indicate the definiteness of the referring expression. If the specifier is definite, a previously introduced or otherwise salient object is referred to. If the specifier is indefinite, a new object is referred to.

In predicate logic, there are two kinds of things that come close to the notion of an object referring expression: logical constants and definite descriptions. Logical constants correspond to names that refer to a unique individual within a model, and typically do so rigidly across worlds in possible worlds semantics. Definite descriptions are logical expressions that describe a unique individual within a model, but do not do so rigidly across worlds.

Quantified expressions and indefinite descriptions, on the other hand, quantify over a set of individuals identified by a predicate, but do not refer to any particular individual – i.e. they are not referring expressions in this sense. For example, in the predicate logic expression

$$\exists x [\text{man}(x)]$$

the predicate man denotes the set of individuals in the model that satisfy the predicate and the variable x is existentially quantified over those individuals. As long as there is at least one individual in the model that satisfies the predicate, this logical expression is true, given the model.

Although this logical expression can be put forward as an approximate logical representation for the indefinite object referring expression

a man
the logical representation fails to capture much of the meaning of this object referring
equation. In particular, it fails to capture the intent of this expression to refer to some
individual that is not currently in the model, but should be introduced.

Another problem with this representation is that it fails to capture the context dependent
nature of quantifiers. Since Aristotle, quantifiers have been treated as binary relations
between properties. For example, the sentence

\[
\textit{all men are mortal}
\]

is commonly represented as

\[
\forall x[\text{man}(x) \rightarrow \text{mortal}(x)]
\]

where the universal quantifier \( \forall \) relates the predicate \text{man} to the predicate \text{mortal} via the
variable \( x \). The predicates \text{man} and \text{mortal}, which denote properties, are within the scope of
the universal quantifier which quantifies over \( x \). Note that there is no single element in this
representation that corresponds to the object referring expression \textit{all men} – i.e. the
bracketing indicates that \([\text{man}(x) \rightarrow \text{mortal}(x)]\) is an element, but not \( \forall x[\text{man}(x)] \). A basic
problem with this representation is that it doesn’t correspond closely to the grammatical
structure of the linguistic expression. Another problem is that the representation differs
significantly from the representation of

\[
\textit{John is mortal}
\]

which is commonly given as

\[
\text{mortal}(J)
\]

where \( J \) is a logical constant that refers to the person named \textit{John}.

There is also no simple predicate logic representation for a basic definite description like

\[
\textit{the man}
\]

since neither the universal nor the existential quantifier captures the idea of reference to a
unique individual.

Montague developed a system of intensional logic with typed \( \lambda \) calculus, called \textbf{Montague
Grammar}, that addresses some of these concerns (cf. Montague, 1973). In Montague
Grammar as described by Partee (2014), quantifiers (e.g. \textit{all}, \textit{some}) and the indefinite
determiner \textit{a} are treated as binary relations of two properties with existential (\( a, some \)) or
universal (\textit{all}) quantification over a single variable; the definite determiner \textit{the} is treated as
a binary relation between two properties, with two variables – one existentially quantified
and one universally quantified – that are equated (thereby uniquely identifying an
individual); and, proper nouns are treated as properties of a logical constant. For example,
the quantified description \textit{all men} is represented as:

\[
\textit{all men}: \lambda P[\forall x(\text{man}(x) \rightarrow P(x))]
\]
where $\lambda P$ is a $\lambda$ variable that will be replaced by the (main) predicate of the sentence. Abstracting away from the noun *men* leads to the $\lambda$ abstracted representation of the quantifier *all*:

$$all: \lambda Q[\lambda P[\forall x(Q(x) \rightarrow P(x))]]$$

where the abstracted noun, represented by $Q$, is also treated as a property. Given specific properties for $Q$ and $P$, the $\lambda$ variables are eliminated via a process called $\beta$ reduction:

$$all \text{ men are mortal: } \forall x[\text{man}(x) \rightarrow \text{mortal}(x)]$$

where

$$\lambda Q[Q(x)](\text{man}(x)) \leftrightarrow \text{man}(x)$$
$$\lambda P[P(x)](\text{mortal}(x)) \leftrightarrow \text{mortal}(x)$$

The $\lambda$ abstracted representation of quantifiers like *all* is commonly called a **generalized quantifier**. The $\lambda$ abstracted representation is generalized to determiners, proper nouns and pronouns, as well as quantifiers. The generalized quantifier representations for the determiners *a* and *the*, and for the proper noun *John* are shown below:

$$a: \lambda Q[\lambda P[\exists x(Q(x) \land P(x))]]$$
$$\text{the}: \lambda Q[\lambda P[\exists x(Q(x) \land \forall y[Q(y) \rightarrow y = x]) \land P(x)]]$$
$$John: \lambda P[P(J)]$$

For the sentences

$$a \text{ man is mortal}$$
$$\text{the man is mortal}$$
$$John \text{ is mortal}$$

following $\beta$ reduction, we have

$$a \text{ man is mortal: } \exists x[\text{man}(x) \land \text{mortal}(x)]$$
$$\text{the man is mortal: } \exists x[\text{man}(x) \land \forall y[\text{man}(y) \rightarrow y = x] \land \text{mortal}(x)]$$
$$John \text{ is mortal: mortal}(J)$$

The representation for *the man is mortal* is problematic since the object referring expression *the man* does not pick out a unique individual, unless there is only one man in the model.

Barwise and Cooper (1981) modify Montague’s treatment of generalized quantifiers in arguing that noun phrases (NPs) are generalized quantifiers, whereas *all, some, a* and *the* are determiners. It is the combination of a determiner with a common noun (i.e. a set expression) to form an NP that results in a generalized quantifier.
In the case of proper nouns and pronouns which can occur without a determiner, it is still the NP, of which they are the only element, that functions as a generalized quantifier, not
the proper noun or pronoun.

The treatment of all noun phrases as generalized quantifiers is often cited as a major
strength of Montague Grammar (cf. Partee, 1986), since it facilitates the alignment of
syntactic rules with semantic rules, adhering to Bach’s (1989) rule-by-rule hypothesis, and
thereby supporting the compositional generation of logical representations from linguistic
inputs. This treatment works well as long as the noun phrase is functioning as the subject,
but is problematic when the noun phrase is functioning as a non-subject argument or as a
predicate nominal. Partee (1986) extends Montague’s treatment by allowing noun phrases
to have three distinct semantic types. She calls these semantic types referential, predicative
and quantificational (in Montague grammar which is based on type theory, referential = e,
predicative = <e, t> and quantificational = <<e, t>, t>, but these explicit types from type
theory are avoided in this paper). Each noun phrase is associated with a basic semantic
type – depending on the head of the noun phrase – that can be type-shifted to another
semantic type, depending on the context of use. For example, noun phrases headed by
proper nouns and common nouns are normally referential, but noun phrases headed by
common nouns can be used as predicate nominals, suggesting a derivative predicative use,
and noun phrases headed by proper nouns can be conjoined with quantificational
expressions, suggesting a derivative quantificational use:

*John* is laughing – referential use of *John*
*John is a man* – predicative use of *a man*
*Could John or any doctor come to the emergency room?*
 – quantificational use of *John* and *any doctor*

Type shifting avoids the need to have multiple lexical entries for proper nouns and
common nouns to handle the different uses, but it violates the rule-by-rule hypothesis (i.e.
more than one semantic type is associated with a syntactic type), and it complicates the
compositional generation of logical representations.

In Double-R Grammar, the basic semantic type of all object referring expressions is
referential. Object referring expressions are allowed to function as predicate nominals, in
which case they have a predicative function, but they remain referential (i.e. they are co-
referential with the subject). Even the most recalcitrant quantifying expressions (e.g.
expressions with *every* like *every man*) are assumed to be referential. To support their
referential treatment, expressions like *every man* are allowed to refer to collections in the
situation model. However, there is also a strong sense in which object referring expressions
with *every* quantify over the individuals members of the collection (e.g. *each and every*
man is leaving). From a cognitive perspective, it is feasible to represent the individual members of a small collection in the situation model, but it does not seem feasible to individually represent a large number of members, which would exceed the capacity of working memory. For small collections with three or four members, the collection as a whole can be represented, along with the individual members. For larger collections, some mechanism for abstracting over most of the members is needed (e.g. a collection within a collection).

In arguing that expressions with every are necessarily quantifying (i.e. non-referring), the following contrast is often provided:

\[
\begin{align*}
\text{John is here. He arrived early.} \\
\text{Everyone is here. *He arrived early.}
\end{align*}
\]

The ungrammaticality of he arrived early in the second example is attributed to the fact that everyone is a quantifying expression, and the pronoun he cannot be co-referential with it. However, as argued above, expressions with every refer to collections. Co-reference to a collection typically requires use of a plural pronoun:

\[
\begin{align*}
\text{John and Mary are here. They arrived early.} \\
\text{Everyone is here. They arrived early.}
\end{align*}
\]

On this analysis, everyone is an object referring expression. It just refers to a collection, making use of the pronoun he inappropriate for co-reference (at least in English). To reference the individual elements of the collection, an explicitly distributional quantifier like each is required:

\[
\begin{align*}
\text{Everyone is here. Each one arrived early.}
\end{align*}
\]

Negative expressions like no man are also assumed to present challenges for their treatment as referring expressions. For such expressions, we assume the existence of empty collections. It is important to note that collections are typically associated with a type. The expression no man refers to an empty collection of type man. In this respect, collections are like kinds. However, collections normally denote contextually relevant subsets of a kind, not the entire kind.

The treatment of all object referring expressions as referential differs from Montague's treatment of them all as quantificational. Proper nouns and pronouns are paradigmatic referring expressions. Treating them as quantifying expressions in their most common uses reflects a desire to treat all nominal expressions uniformly. Partee (1986) modifies Montague's treatment to allow proper nouns and pronouns to be referential in their basic uses, but allows them to be type shifted to a quantificational use when the context requires (or at least suggests) it. Double-R Grammar allows object referring expressions to have alternative grammatical functions (e.g. predicate nominal, predicate modifier) without requiring a type shift. This is achieved via the use of constructions which support the alternative functions, and via context accommodation. For example, in

\[
\text{John is a doctor}
\]
the object referring expression *a doctor* functions as a predicate nominal and in

*John left yesterday*

the temporal object referring expression *yesterday* functions as a predicate modifier (i.e. *yesterday* is categorized as a nominal in Double-R Grammar, and projects a temporal object referring expression). These are both examples of context accommodation.

It should be noted that the trend in compositional semantics is toward the treatment of more and more nominal expressions as referring expressions (cf. Pelletier, 2010). This is reflected in the treatment of indefinite descriptions as referring expressions by Kamp (1981) and Heim (1983). It is also reflected in the treatment of bare plural nominal expressions as functioning like proper nouns that refer to kinds by Carlson (1977) (although the association of bare plural nominal expressions with proper nouns is problematic since proper nouns and bare plural nouns exhibit different grammatical behaviors and are semantically distinct). The range of non-referring nominal expressions has been reduced as the ontology of individuals in the model (e.g. kinds, collections, stuff, properties) has been expanded. It should, however, also be noted that most compositional semantic approaches continue to assume a subset of non-referential quantifying expressions. Chierchia and McConnell-Ginet (2000) present a referential approach that is largely compatible with Double-R Grammar as a straw man that is inadequate to handle recalcitrant quantifying expressions. We believe that the expansion of the ontology of individuals and the introduction of structured meaning representations, combined with the distinction between the situation model and the mental universe provides a suitable basis for meaning representation along the lines advocated in Double-R Grammar, but have not yet tried to address the challenges for a truth-conditional treatment of quantifying expressions that a purely referential treatment gives rise to.

**Overview of the Logical Representation of Object Referring Expressions in Double-R Grammar**

In Double-R Grammar, the representation of object referring expressions is divided into two parts: 1) lexical representations for each of the parts of speech of the words in the object referring expression, and 2) a construction that represents the object referring expression itself, and constructions for any subexpressions (e.g. object-head construction). This treatment aligns with that of Barwise and Cooper (1981), with object referring expressions corresponding roughly to their use of the term generalized quantifier.

Splitting the representation of object referring expressions into parts of speech and constructions has several advantages. First, it allows us to keep lexical representations simple and consistent. The lexical representations for nouns, proper nouns and pronouns are all very similar, as are the lexical representations for determiners and quantifiers (where quantifier is a part of speech). Second, it allows us to represent the relations between the lexical representations in a consistent manner. Third, it allows us to capture the full grammatical structure of object referring expressions. Fourth, it supports the generation of logical representations that are closely aligned with grammatical representations – in fact, they are the same. Finally, it allows us to minimize ambiguity in
the mental lexicon. There is a single lexical entry for each word in a given part of speech, ignoring homonyms. For example, there is no need for distinct entries in the mental lexicon to capture the attributive vs. predicative function of adjectives, or for separate entries for the function of a noun as a modifier vs. the head of an object referring expression.

In Double-R Grammar, an object referring expression (ore) construction is represented using a λ expression that indicates a promissory note to identify or introduce an object into the situation model that is referred to by the object referring expression. This use of λ variables differs from Montague’s use, but aligns roughly with the use of variables, called discourse referents, in DRT. In addition to the λ variable for the object referring expression construction itself, special η handle variables (η is the Greek equivalent to the Latin letter h) are needed to represent the grammatical functions that make up the object referring expression construction. Limiting the discussion in this overview to the specifier and head grammatical functions, we can represent an object referring expression construction as:

$$\lambda o \eta s \eta h . \text{ore}(o) \land \text{spec}(o, s) \land \text{head}(o, h)$$

where

- o is a λ variable of type object referring expression
- s and h are η handle variables for the grammatical functions (i.e. specifier and head) encoded by the referring expression
- ore (object referring expression) is an expression type
- spec (specifier) and head are grammatical functions
- \(\land\) = logical conjunction

An instantiated object referring expression construction will include a specifier and head, or just a head. We call the instantiated object referring expression construction an object referring expression. This representation of object referring expressions is reminiscent of neo-Davidsonian logical representations in which the λ variable functions like an event variable – although object referring expressions are not events, and grammatical functions replace semantic roles. The η handle variables are needed to allow reference to the parts of speech or subexpressions that are instantiated into the construction, from other parts of the representation.

Object referring expression constructions are projected from parts of speech or subexpressions (e.g. object head). Determiners project object referring expression constructions in which they are instantiated as the specifier. Proper nouns and pronouns project object referring expression constructions in which they are instantiated as the head. Common nouns may or may not project an object referring expression construction, depending on whether or not they are preceded by a determiner or quantifier which has already projected one. In either case, the common noun will be instantiated as the head. The representation of nouns as the heads of object referring expressions aligns with the traditional noun phrase (NP) analysis of nominal expressions, and is incompatible with the more recent determiner phrase (DP) analysis. In Double-R Grammar, the motivation for this treatment is primarily semantic – the noun is the main semantic element of an object referring expression. It provides the type specification for the object referring expression,
determining what kind of object is being referred to. Determiners encode a peripheral grammatical function in indicating the definiteness of object referring expressions.

From a grammatical perspective, the facts that determiners are ungrammatical without a noun head, as in

*\textit{the} is laughing

and they are not allowed to precede proper nouns in simple expressions like

*\textit{the John} is laughing

(in English), provide strong evidence against treating them as the head. Determiners functioning as specifiers often project object referring expressions, since they occur before noun heads and they are a strong indicator of an object referring expression, but they are not the heads of the expressions they project.

Once the head is instantiated into the projected object referring expression construction, an instantiated object referring expression that is capable of referring to some object in the situation model results. Instantiation of the specifier and head must occur according to the grammar. A snippet of the grammar is shown below:

\begin{verbatim}
o re --> spec head
   spec --> det | quantifier | demon-pron | poss-obj-spec ( | = or)
   head --> noun | prop-noun | pers-pron | quantifier | demon-pron
\end{verbatim}

Note that quantifiers (e.g. \textit{all, some}) and demonstrative pronouns (demon-pron) (e.g. \textit{this, that}) can function as either specifiers (spec) or heads. On the other hand, determiners (det) (e.g. \textit{a, the}) only function as specifiers, and nouns, proper nouns (prop-noun) and personal pronouns (pers-pron) only function as heads (in this snippet which ignores modifiers). Possessive object specifier (poss-obj-spec) is a construction containing a possessive nominal (e.g. \textit{John's}) or possessive pronoun (e.g. \textit{his}) that functions as a specifier.

The existence of object referring expression constructions makes it possible to have fewer, simpler and more consistent representations of parts of speech, that are semantically motivated. In this regard, they perform a function similar to Partee’s (1986) type-shifting operation, at least with respect to her referential and quantificational uses (predicate nominal constructions support the predicational use). The representations for the determiners \textit{a} and \textit{the}, with the addition of the grammatical feature definiteness (def), are shown below:

\begin{verbatim}
\textit{a}: \eta d \cdot \text{det}(d) \land \text{def}(d, \text{indef})
\textit{the}: \eta d \cdot \text{det}(d) \land \text{def}(d, \text{def})
\end{verbatim}

where

- d is a \eta handle variable
- det (determiner) is a part of speech
def (definiteness) is a grammatical feature, def(d, Def), such that Def ∈ {def (definite), indef (indefinite), univ (universal)}

In Double-R Grammar, the part of speech category determiner (det) is viewed as an abstract semantic category, as are all parts of speech. The representation of a differs from the only in terms of the grammatical feature definiteness (def) (ignoring number), with a being indefinite (indef) and the being definite (def). Grammatical features like definiteness are viewed as grammatically relevant, semantically motivated features. Since there can be more than one grammatical feature, the feature type (e.g. def) is represented as a predicate, and the value of the feature (e.g. indef) is represented as an argument.

If we instantiate the determiner a into the object referring expression construction as the specifier, we get:

\[ ore+s=a: \lambda o \eta s \eta h . ore(o) \land spec(o, s) \land det(s) \land head(o, h) \land def(o, indef) \]

As part of the instantiation process, the definiteness feature of a is projected to the object referring expression. This representation is viewed as a grammatically, and hence, semantically motivated logical representation. It is licensed by the grammar.

This representation corresponds roughly to the following logical representation for a:

\[ a: \lambda P_h[\exists x(P_h(x))] \]

where \( P_h \) corresponds to the head grammatical function which has not been instantiated and the specifier function is not explicitly represented. The logical representation is close to the representation of generalized quantifiers (i.e. NPs) proposed in Barwise and Cooper (1981), although Barwise and Cooper do not use \( \lambda \) abstraction. To get closer to Partee’s (1986, 2014) description of Montague’s generalized quantifier representation, we need to instantiate this object referring expression as the subject of a situation referring expression construction. The representation of a situation referring expression (sre) construction, ignoring irrelevant detail, is shown below:

\[ sre: \lambda sit \lambda subj \eta p . sre(sit) \land subj(sit, subj) \land head(sit, p) \]

where

sit and subj are \( \lambda \) variables for referring expressions, with subj functioning as the subj (subject)

sre (situation referring expression) is an expression type

p is a \( \eta \) handle variable for the (predicate) head grammatical function

subj (subject) and (predicate) head are grammatical functions

We use the term head instead of predicate for consistency with the representation of object referring expressions and in alignment with X-Bar Theory (Chomsky, 1970).

Instantiating the object referring expression as the subject gives:

\[ sre+ore=subj+s=a: \lambda sit \lambda subj \eta p \eta h . sre(sit) \land subj(sit, subj) \land ore(subj) \land spec(subj, s) \land det(s) \land head(subj, h) \land def(subj, indef) \land head(sit, p) \]
This representation corresponds roughly to the following generalized quantifier representation for $a$:

$$a : \lambda Q h_1 [\lambda P h_2 [\exists x (Q h_1(x) \land P h_2(x))]]$$

where $Q h_1$ corresponds to the head of the object referring expression functioning as subject of the situation referring expression, and $P h_2$ corresponds to the (predicate) head of the situation referring expression. This representation is close to that of Partee (1986, 2014).

Although there is a rough correspondence between these last two representations for $a$, such a representation would not be constructed in Double-R Grammar, because the processing of the determiner $a$ would not, by itself, project a situation referring expression, although it would project an object referring expression. The reason for not adopting the Montague like representation is that object referring expressions are capable of referring without participating in a sentence. Consider

*Look! A man!*

where *a man* refers without being the subject of a sentence. In Double-R Grammar, additional grammatical evidence is needed before a situation referring expression construction is projected – typically the occurrence of a verb following the subject.

To represent the indefinite object referring expression

*a man*

we need only instantiate the noun *man* as the head of the object referring expression construction projected by the determiner $a$. The lexical representation for *man* is shown below:

$$man : \eta n \cdot man(n) \land noun(n) \land def(n, \text{Def})$$

where

- $man$ is a type specification
- $noun$ is a part of speech
- $def$ (definiteness) is a grammatical feature such that
  - $\text{Def} \in \{\text{def}, \text{indef}\}$

Note that the definiteness feature of *man* is the variable $\text{Def}$. This is because singular count nouns do not encode for definiteness, as is suggested by the ungrammaticality of expressions like

*man is coming*

It is the lack of a definiteness feature that makes this example ungrammatical.

Instantiation of the noun *man* as the head of the object referring expression construction results in:
\[
\text{ore+s=a+h= man: } \lambda o \eta_s \eta_h . \text{ ore(o) } \land \text{ spec(o, s) } \land \text{ det(s) } \land \text{ head(o, h) } \land \text{ man(h) } \land \\
\text{ noun(h) } \land \text{ def(o, indef) }
\]

When \textit{man} is instantiated as the head, the definiteness feature of the determiner \textit{a} prevails and the instantiated object referring expression is indefinite.

Although this representation treats \textit{man} as a predicate, this is not quite right. The noun \textit{man} actually encodes a presupposed \textit{type specification}, not a predicate. A type specification may be viewed as a set of properties. The noun \textit{man} denotes the set of properties that are associated with the type (e.g. male, human, bipedal, hirsute), although this set of properties is only implicit. Unlike adjectives which typically denote a specific property (e.g. red, sad), nouns typically denote a type that may have many associated properties.

One way of logically representing this distinction between a presupposed type specification and a predicate for the sentence

\textit{a man sneezed}

is as follows:

\[\exists x: \text{man}(x)[\text{sneezed}(x)]\]

where the presupposed type specification occurs to the left of the square brackets and the predicate is within the brackets. In this notation, the presupposed type specification is called a \textit{restriction}, and the predicate within the brackets is called the \textit{body}.

Although we will retain the above treatment of \textit{man} as a predicate in this paper as \textit{syntactic sugar} highlighted in \textcolor{red}{red} to support comparison with standard predicate logic, a more perspicuous representation would explicitly indicate that it is a type:

\text{type(h, man)}

In a recent adaptation of Montague Grammar, called \textit{Modern Type Theory} (Luo, 2012), common nouns are treated as types and not predicates. This allows common nouns to be organized into a semantic type hierarchy and the arguments of predicates to be type specific. It also maintains a distinction between nouns, on the one hand, and verbs and adjectives, on the other hand. In Cognitive Grammar (cf. Langacker, 1991), nouns, which denote types of object, are distinguished from verbs and adjectives, which denote types of relations. Since verbs and adjectives denote types of relations, they combine with one or more arguments. Nouns do not do this, although their common treatment as predicates in predicate logic suggests that they do. Even semantically "relational" nouns like \textit{father} are ungrammatical with a bare argument without the preposition \textit{of}:

\* \text{the father John}
\text{the father of John}

The relational nature of the noun \textit{father} is implicit. Grammatically, it denotes a non-relational object type that functions as the head of an object referring expression.
In distinguishing nouns from verbs and adjectives, Modern Type Theory improves on predicate logic, but fails to recognize that verbs and adjectives are also types (i.e. relational types) that participate in a type hierarchy.

The representation of the definite object referring expression

*the man*

in Double-R Grammar, parallels that of the indefinite object referring expression:

\[ \text{ore} + s = \text{the} + h = \text{man} : \lambda o \eta s \eta h . \ \text{ore}(o) \land \text{spec}(o, s) \land \text{det}(s) \land \text{head}(o, h) \land \text{man}(h) \land \text{noun}(h) \land \text{def}(o, \text{def}) \]

The only representational difference is the definiteness feature. However, there is an important grammatical (and semantic) difference. Definite object referring expressions indicate the intent to refer to an existing and unique object in the situation model that satisfies the description. Indefinite object referring expressions indicate the intent to introduce a new object into the situation model that satisfies the description. This important difference is captured by the interpretation function that maps the object referring expression into the situation model – the promissory note mentioned above.

In Double-R Grammar, a proper noun typically functions as the head of an object referring expression. This allows the lexical representation for a proper noun to be kept simple. Consider the proper noun *John*, which is represented as:

\[ \text{John} : \eta \text{pn} . \ \text{prop-noun}(\text{pn}) \land \text{def}(\text{pn}, \text{def}) \]

ignoring grammatical features other than definiteness. When a proper noun is processed, it typically projects an object referring expression construction in which it is instantiated as the head. If we instantiate the representation of the proper noun *John* as the head of the projected object referring expression construction, we have:

\[ \text{ore} + h = \text{John} : \lambda o \eta s \eta h . \ \text{ore}(o) \land \text{spec}(o, s) \land \text{head}(o, h) \land \text{prop-noun}(h) \land \text{def}(o, \text{def}) \]

In this example, the specifier function is left uninstantiated since there is no determiner (i.e. the variable *s* does not correspond to any word or expression). Unlike the noun *man* and like the determiner *a*, there is no type specification in this representation. However, it does seem necessary to encode the name *John* somewhere in the representation. In Discourse Representation Theory (DRT) (Kamp, 1981), this is achieved using the equality operator:

\[ x \overset{\text{-------------------}}{=} x = \text{John} \]

where *x* is a discourse referent (i.e. a logical variable) that is set equal to *John*, and where *John* is a logical constant.
In Double-R Grammar, the name is indicated using an index:

\[ \text{John} : \eta \text{pn} . \; \text{prop-noun}(\text{pn}) \land \text{index}(\text{pn}, \text{John}) \land \text{def}(\text{pn}, \text{def}) \]

The index is part of the lexical entry. Indexes, which generalize over morphological variants of a word, are discussed further in the part of speech section.

A common logical treatment of pronouns is to use the existential quantifier combined with the equality operator to indicate co-reference. Given the expression

\[ \text{John is a man. He runs.} \]

a logical representation like

\[ \exists x[\text{man}(x) \land \text{run}(x) \land J = x] \]

is common.

In Double-R Grammar, we represent personal pronouns simply as:

\[ \text{he} : \eta \text{pn} . \; \text{pers-pron}(\text{pn}) \land \text{index}(\text{pn}, \text{he}) \land \text{def}(\text{pn}, \text{def}) \]

ignoring grammatical features other than definiteness and the index. In this representation, personal pronoun (pers-pron) is a semantically motivated part of speech. If we instantiate the representation of the personal pronoun he as the head of an object referring expression construction, we have:

\[ \text{ore} + \text{he} : \lambda o \; \eta s \; \eta h . \; \text{ore}(o) \land \text{spec}(o, s) \land \text{head}(o, h) \land \text{pers-pron}(h) \land \text{index}(h, \text{he}) \land \text{def}(o, \text{def}) \]

To indicate the co-reference between John and he, ignoring irrelevant detail, we equate the distinct \( \lambda \) variables \( o_1 \) and \( o_2 \):

\[ \lambda o_1 \; \eta h_1 . \; \text{ore}(o_1) \land \text{head}(o_1, h_1) \land \text{prop-noun}(h_1) \land \text{index}(h_1, \text{John}) \land \text{def}(o_1, \text{def}) \land \\
\lambda o_2 \; \eta h_2 . \; \text{ore}(o_2) \land \text{head}(o_2, h_2) \land \text{pers-pron}(h_2) \land \text{index}(h_2, \text{he}) \land \text{def}(o_2, \text{def}) \land \\
o_1 = o_2 \]

Although grammatical features other than definiteness and the index are not shown in this example, they are crucial for determining co-reference. Indeed, grammatical features were added to Double-R Grammar representations specifically to support co-reference resolution.

If we reduce and simplify the representation of object referring expressions in Double-R Grammar to the bare essentials, the expression

\[ \text{John} \]

can be represented roughly as

\[ \lambda x . \; \text{ore}(x) \land \text{head}(x, \text{John}) \]
In this representation, a \( \lambda \) variable \( x \) is introduced as a handle for the object referring expression, and \textit{John} is assigned the grammatical function head of \( x \). The proper noun \textit{John} is represented as the head of an object referring expression, just like the common noun \textit{book} in

\textit{the book}

which is represented as

\[
\lambda x . \text{ore}(x) \land \text{head}(x, \text{book})
\]

These \( \lambda \) expressions do not get reduced in Double-R Grammar. The reason they do not get reduced is that the \( \lambda \) variable may be referenced elsewhere, and reduction would eliminate the variable. For example, consider

\textit{John read the book. He liked it.}

In this short discourse, the pronoun \textit{he} refers back to \textit{John} and the pronoun \textit{it} refers back to \textit{the book}. (For now, we ignore the importance of grammatical features for determining co-reference.) The \( \lambda \) variables make it possible to bind these pronouns to their antecedents, establishing co-reference:

\[
\lambda b . \text{ore}(b) \land \text{head}(b, \text{book}) \land \lambda j . \text{ore}(j) \land \text{head}(j, \text{John}) \land \text{read}(j, b) \land \lambda h . \text{ore}(h) \land \\
\text{head}(h, \text{he}) \land h=j \land \lambda i . \text{ore}(i) \land \text{head}(i, \text{it}) \land i=b \land \text{liked}(h, i)
\]

The head grammatical function is essential to these representations, although it is not obvious in these simple examples. To see why the head function is critical, consider the expressions

\textit{The early Wittgenstein}  
\textit{A Reagan republican}

English grammar allows proper nouns to function as heads of complex object referring expressions with separate modifiers and specifiers, and as modifiers of heads, as well. These expressions can be represented in Double-R Grammar as:

\[
\lambda w . \text{ore}(w) \land \text{head}(w, \text{Wittgenstein}) \land \text{mod}(w, \text{early}) \\
\lambda r . \text{ore}(r) \land \text{head}(r, \text{republican}) \land \text{mod}(r, \text{Reagan})
\]

What we do not want, is to have separate part of speech entries in the mental lexicon (or part of speech conversions via type-shifting) for each function of a proper noun. We do not want to treat \textit{Wittgenstein} as a common noun in \textit{the early Wittgenstein}, or \textit{Reagan} as an adjective in \textit{a Reagan republican}. Doing so would break the connection between the parts of speech of words and meaning. The representation of grammatical functions in grammatical constructions that are distinct from lexical entries, allows us to represent the different functions of words without complicating their representation in the mental lexicon.

In predicate logic with extensions, the following representations are suggested:
\[ \lambda P[\exists x[\text{early-Wittgenstein}(x) \land \forall y[\text{early-Wittgenstein}(y) \rightarrow x=y] \land P(x)]] \]
\[ \exists x[\text{Reagan-republican}(x)] \]

These suggested representations treat \textit{early-Wittgenstein} and \textit{Reagan-republican} as composite, unanalyzed predicates. It is unclear how to do otherwise without introducing something equivalent to the grammatical functions head and modifier.

In this overview section, we have provided simplified (and incomplete) representations of object referring expressions that facilitate comparison to predicate logic and various extensions. Fully articulated representations, which add additional grammatical functions and features, are discussed below.

**The Logical Representation of Situation Referring Expressions in Predicate Logic and Extensions**

Predicate logic is better designed for the representation of situation referring expressions than object referring expressions. The typical predicate logic representation of a sentence is a relatively flat representation consisting of one or more predicates connected via the logical operators \(\land\) (logical conjunction), \(\lor\) (logical disjunction), \(\rightarrow\) (material implication), \(\leftrightarrow\) (logical equivalence), and \(\neg\) (logical negation), and perhaps a few others, with each predicate having one or more arguments which are existentially or universally quantified.

Given standard predicate logic, a sentence like

\textit{all men are mortal}

can be represented as

\[ \forall x[\text{man}(x) \rightarrow \text{mortal}(x)] \]

We have already discussed weaknesses of this logical representation as a representation for the object referring expression \textit{all men}. As a representation of the situation referring expression \textit{all men are mortal}, there are also problems. The auxiliary verb \textit{are} is represented via material implication when there is a universal quantifier, but the morphological variant \textit{is} is represented as a logical conjunction when there is an existential quantifier, as in the representation of

\textit{some man is laughing}

as

\[ \exists x[\text{man}(x) \land \text{laugh}(x)] \]

If we use the restriction notation, we can represent this as

\[ \exists x:\text{man}(x)[\text{laugh}(x)] \]

where the restriction \textit{man}(x) is implicitly connected to the body \textit{laugh(x)} by a conjunction. However, this does not work for \textit{all men are mortal}

\[ \forall x:\text{man}(x)[\text{mortal}(x)] \]
if the implicit relationship between the restriction \( \text{man}(x) \) and the body \( \text{mortal}(x) \) is material implication and not conjunction. To address this problem, we can implicitly treat the relationship as conjunction when the existential quantifier is used and material implication when the universal quantifier is used (cf. Gamut, 1991).

With unary predicates like \( \text{man}, \text{laugh} \) and \( \text{mortal} \), and use of the restriction notation, the representations have two parts that correspond roughly to the subject (i.e. some \( \text{man} \), all \( \text{men} \)) and predicate (i.e. is laughing, is mortal). Logical treatments prior to the introduction of predicate logic often assumed a division of the sentence into a subject and a predicate.

If we consider a binary predicate like \( \text{kicked} \) in

\[
\text{John kicked a ball}
\]

the following representation is suggested:

\[
\exists x: \text{ball}(x)[\text{kick}(\text{John}, x)]
\]

In this representation, the restriction notation is used for the object referring expression \( \text{a ball} \) which is the object of \( \text{kicked} \), and the subject \( \text{John} \) is represented by the logical constant \( J \). The subject \( \text{John} \) and the object \( \text{a ball} \) function as arguments of the predicate \( \text{kicked} \). There is no representation of the division into a subject \( \text{John} \) and a predicate \( \text{kicked a ball} \).

The bi-partite division of a sentence is assumed in many syntactic approaches which posit a top-level syntactic rule like:

\[
S \rightarrow \text{NP} \text{ VP}
\]

Of course the predicate or VP \( \text{kicked a ball} \) is not a word and has internal structure which is typically represented by a syntactic rule like

\[
\text{VP} \rightarrow \text{V} \text{ NP}
\]

where \( \text{kicked} \) is a V (verb) and \( \text{a ball} \) is an NP (noun phrase). Finally, the structure of the noun phrase \( \text{a ball} \) is also represented by a rule like

\[
\text{NP} \rightarrow \text{D} \text{ N}
\]

where \( a \) is a determiner (D) and \( \text{ball} \) is a noun (N). Most of this grammatical structure is flattened out or ignored in the predicate logic representation.

It is possible to approximate the grammatical notion of a VP or predicate, in predicate logic, with the use of \( \lambda \) abstraction. For example, the logical expression

\[
\lambda \text{subj}[\exists o: \text{ball}(o)[\text{kick}(\text{subj}, o)]]
\]

is suggested as an approximate representation for \( \text{kick a ball} \). In this representation, the \( \lambda \) variable \( \text{subj} \) can be instantiated by the subject \( \text{John} \) via \( \beta \) reduction giving
\[ \lambda \text{subj}[\exists o: \text{ball}(o)[\text{kick}(\text{subj}, o)]](J) \leftrightarrow \exists o: \text{ball}(o)[\text{kick}(J, o)] \]

Of course, following \( \beta \) reduction, the division into a subject and predicate is lost. An alternative representation which retains the division is suggested below:

\[ \lambda \text{subj}[\exists o: \text{ball}(o)[\text{kick}(\text{subj}, o)]] \land \text{subj} = J \]

In this representation, instead of reducing the \( \lambda \) variable \( \text{subj} \), it is set equal to \( J \).

In a linguistically oriented logical theory, Sag (1976) argues for \( \lambda \) abstracted representations of VPs to support verb phrase deletion. In a sentence like

*John kicked a ball and Bill did \( \emptyset \) too*

there is an implicit VP *kick a ball* in the elliptical expression *Bill did \( \emptyset \) too*. To provide an adequate logical representation for this sentence, some representation of the elided VP is needed. Based on Sag’s analysis, the following representation is suggested:

\[
<\text{John}, \lambda \text{subj}_1[\exists o_1: \text{ball}(o_1)[\text{kick}(\text{subj}_1, o_1)]]> \land
<\text{Bill}, \lambda \text{subj}_2[\exists o_2: \text{ball}(o_2)[\text{kick}(\text{subj}_2, o_2)]]>
\]

In this logical representation, the \( \lambda \) abstracted representation for the VP or predicate is repeated and the subject that will be substituted for the \( \lambda \) abstracted variable is listed before the predicate to reflect the subject-predicate surface order. In Sag’s transformational grammar based treatment, the repeated VP is deleted on the way to surface syntax, but remains in place in the logical form.

If we substitute a handle variable, \( \eta p \), for the repeated predicate we have

\[
<\text{John}, \lambda \text{subj}_1[\eta p^{**}[\exists o: \text{ball}(o)[\text{kick}(\text{subj}, o)]]]> \land
<\text{Bill}, \lambda \text{subj}_2[p]>\]

where \( p^{**} \) functions like the \( e^{**} \) operator of Episodic Logic (Schubert, 1999) in allowing reference to the complex formula \[\exists o: \text{ball}(o)[\text{kick}(\text{subj}, o)]\]. Given this representation, some mechanism other than operator scope is needed, so that the second use of \( p \) is correctly bound to the first use – perhaps a DRT-like approach in this case. (Episodic Logic and DRT are discussed below.) This representation obviates the need for a deletion transformation, although it leaves unexplained the question of how the tense winds up being expressed explicitly by *did*, and not implicitly by *kicked* in the elided VP. The representation aligns with Double-R Grammar which makes a commitment to predicate level representations to handle elliptical expressions. The \( \eta \) handle indicates that the predicate is not a referring expression. Predicates, in elliptical expressions, typically lack tense, as in this example, where the tense of the elliptical expression is carried by the explicit auxiliary verb *did*. The representation of tense is not shown in this example, but is represented as a grammatical feature that is projected from the auxiliary (or main) verb to the situation referring expression, in Double-R Grammar.

If we substitute a handle variable, \( \eta p^{**} \), for the repeated predicate we have

\[
<\text{John}, \lambda \text{subj}_1[\eta p^{**}[\exists o: \text{ball}(o)[\text{kick}(\text{subj}, o)]]]> \land
<\text{Bill}, \lambda \text{subj}_2[p]>\]

where \( p^{**} \) functions like the \( e^{**} \) operator of Episodic Logic (Schubert, 1999) in allowing reference to the complex formula \[\exists o: \text{ball}(o)[\text{kick}(\text{subj}, o)]\]. Given this representation, some mechanism other than operator scope is needed, so that the second use of \( p \) is correctly bound to the first use – perhaps a DRT-like approach in this case. (Episodic Logic and DRT are discussed below.) This representation obviates the need for a deletion transformation, although it leaves unexplained the question of how the tense winds up being expressed explicitly by *did*, and not implicitly by *kicked* in the elided VP. The representation aligns with Double-R Grammar which makes a commitment to predicate level representations to handle elliptical expressions. The \( \eta \) handle indicates that the predicate is not a referring expression. Predicates, in elliptical expressions, typically lack tense, as in this example, where the tense of the elliptical expression is carried by the explicit auxiliary verb *did*. The representation of tense is not shown in this example, but is represented as a grammatical feature that is projected from the auxiliary (or main) verb to the situation referring expression, in Double-R Grammar.

In predicate logic, the denotation of a logical sentence is a truth value. An expression like

\[ \exists x: \text{ball}(x)[\text{kick}(J, x)] \]
is true if there is an individual in the model, John, that is the first element of the set of ordered pairs denoted by the relation kick, with a second element that is an individual in the set of individuals denoted by the property ball, and false otherwise.

In predicate logic, a relation like kick denotes a set of ordered pairs of individuals in the model (i.e. those individuals involved in kicking events), and a property like ball denotes a set of individuals in the model (i.e. those individuals with the property ball). Relations and properties are essentially reduced to the individuals participating in relations and having properties. This treatment of relations as ordered tuples of individuals is foreign to Double-R Grammar.

Davidson (1967) extended predicate logic with the introduction of event variables. Although a logical sentence still denotes a truth value, the event variable makes it possible to reference the event itself within a logical representation. With the addition of an event variable, the representation of John kicked the ball becomes

\[ \exists e \exists x : \text{ball}(x) [\text{kick}(e, J, x)] \]

With an event variable available, it becomes possible to extend the logical representation to encode tense information. For example, we can indicate the past tense of this sentence as

\[ \exists e \exists x : \text{ball}(x) [\text{kick}(e, J, x) \land \text{past}(e)] \]

We can also encode the grammatical function of the arguments, giving:

\[ \exists e \exists x : \text{ball}(x) [\text{kick}(e, J, x) \land \text{past}(e) \land \text{subj}(e, J) \land \text{obj}(e, x)] \]

where subj(e, J) identifies John as the subject of the event and obj(e, x) identifies a ball as the object of the event. If we replace grammatical functions with semantic roles as Parsons (1990) did, we get

\[ \exists e \exists x : \text{ball}(x) [\text{kick}(e, J, x) \land \text{past}(e) \land \text{agent}(e, J) \land \text{patient}(e, x)] \]

where agent(e, J) identifies John as the agent of the event and patient(e, x) identifies a ball as the patient (i.e. affected object) of the event.

One reason for preferring semantic roles to grammatical functions is that the relationship between sentences like

- John kicked the ball
- The ball was kicked by John

...can be captured using semantic roles. In both these sentences, John is the agent, even though the ball is the subject of the second sentence, and John is part of a modifier phrase by John. Without semantic roles, it is necessary to represent the voice of these sentences so that the logical subject (i.e. John in both sentences) can be distinguished from the surface subject. Of course, even if semantic roles are used, consideration of voice is required to determine the appropriate role during creation of the logical representation. The primary advantage of this approach is that a single logical representation is created for both of these
sentences. The logical representation abstracts away from differences in voice. However, sometimes these differences are important, as in the classic phrase

*Mistakes were made*

in which the agent is intentionally left unexpressed. The logical representation

\[ \exists e \exists x \exists y : \text{mistake}(y) \land \text{make}(e, x, y) \land \text{agent}(e, x) \land \text{patient}(e, y) \land \text{past}(e) \]

abstracts away from this intention. (The plural grammatical feature of *mistakes* is not represented.)

Representations in which semantic roles reference event variables are called neo-Davidsonian, since Davidson did not use semantic roles in his representations. The use of semantic roles has the disadvantage of being grammatical ill-defined. Whereas grammatical functions like subject and object are reasonably well-defined and identifiable (in English), semantic roles like agent and patient often are not. For example, what is the semantic role of the subject in

*John is tall*

Agent does not seem appropriate. Theme is often suggested, but this role appears to be a catchall – no better than subject.

A recent extension to neo-Davidsonian representations involves the use of predicate specific semantic roles (cf. Sayeed & Demberg, 2012). Replacing agent with kicker, and patient with kickee, we have

\[ \exists e \exists x : \text{ball}(x) \land \text{kick}(e, J, x) \land \text{past}(e) \land \text{kicker}(e, J) \land \text{kickee}(e, x) \]

These fine grained semantic roles have the advantage of being easier to identify than abstract semantic roles, since they are lexically specified. If they are organized into a semantic role hierarchy (or better semi-lattice), then the abstract semantic roles can be inferred from the fine grained roles.

If we add back in the grammatical functions and the active voice grammatical feature (representing active voice as a predicate), we have

\[ \exists e \exists x : \text{ball}(x) \land \text{kick}(e, J, x) \land \text{past}(e) \land \text{subj}(e, J) \land \text{obj}(e, x) \land \text{active}(e) \land \text{kicker}(e, J) \land \text{kickee}(e, x) \]

The grammatical functions subject and object provide high level, semantically motivated grammatical information about the arguments, whereas the predicate specific semantic roles provide low level semantic information. For some purposes (e.g. logical inference), we may want to ignore parts of this representation, but for other purposes, the structure is needed.

Besides semantic roles, grammatical functions, and tense, the Davidsonian event variable makes it possible to represent event level modification. For the sentence
John kicked the ball hard

A representation in which the adverb hard modifies the event, is possible:

$$\exists e \exists x : \text{ball}(x) \land \text{kick}(e, J, x) \land \text{hard}(e) \land \text{past}(e) \land \text{subj}(e, J) \land \text{obj}(e, x) \land \text{active}(e) \land \text{kicker}(e, J) \land \text{kickee}(e, x)$$

Unfortunately, the event variable does not support a distinction between sentence modification and predicate modification. Fodor (1972) provides the examples:

John spoke clearly
Clearly, John spoke

In the first example, clearly modifies the predicate spoke. In the second example, clearly modifies the sentence John spoke. A Davidsonian event variable cannot handle both. Parsons (1972) proposed to handle sentence modification by treating sentence level adverbs as operators, leaving the event variable to represent predicate modification:

$$\exists e [\text{speak}(e, J) \land \text{clearly}(e)]$$
$$\exists e [\text{clearly} [\text{speak}(e, J)]]$$

In the latter representation, clearly is a logical operator, not a predicate.

Although this approach retains a first order status for sentence modification, since operators, unlike predicates, are allowed to range over sentences, it results in the introduction of a logical operator for each sentence adverbial (i.e. adverb or prepositional phrase). Such a proliferation of logical operators is akin to the treatment of predicate modification, without an event variable, by creating a composite predicate:

$$\text{speak-clearly}(J)$$

In this case, the proliferation is in terms of composite predicates, rather than operators (and composite operators), but the negative implications of these proliferations for compositional semantics are similar. In particular, it is difficult to see how the model can be pre-populated with static sets corresponding to these dynamically created properties (e.g. the set of speaking-clearly events).

Besides being insufficient for modeling the distinction between sentence and predicate modification, Davidsonian event variables are also inadequate to model nested modification. Consider the expression

John spoke very clearly

in which the adverb very modifies the adverb clearly. First order predicate logic with event variables provides no mechanism for representing this kind of modification. For each level of modification, we need something equivalent to an event variable – what is called a handle variable in Double-R Grammar. Adding in a handle variable for clearly, the following representation is suggested:

$$\exists e [\text{speak}(e, J) \land \exists c [\text{clearly}(e, c) \land \text{very}(c)]]$$
However, if we want to allow for the possibility of further modification, we should add a handle variable for *very* as well:

\[ \exists e \left[ \text{speak}(e, J) \land \exists c \left[ \text{clearly}(e, c) \land \exists v \left[ \text{very}(c, v) \right] \right] \right] \]

Such representations encode enough structure to support the representation of as many levels of predicate modification as occur. The handle variables make it possible to represent the scope of the modifiers to whatever depth is needed.

If we add in grammatical functions for each handle and the constant \( J \), and a fine-grained semantic role for the subject, we have:

\[ \exists e \left[ \text{speak}(e, J) \land \text{subj}(e, J) \land \text{active}(e) \land \exists c \left[ \text{mod}(e, c) \land \text{clearly}(c) \land \exists v \left[ \text{mod}(c, v) \land \text{very}(v) \right] \right] \right] \]

The only obvious way to flatten this representation without losing information about the scope of modifiers is to resort to the non-compositional creation of complex predicates:

\[ \exists e \left[ \text{speak-very-clearly}(e, J) \land \text{subj}(e, J) \land \text{active}(e) \land \exists c \left[ \text{mod}(e, c) \land \text{clearly}(c) \land \exists v \left[ \text{mod}(c, v) \land \text{very}(v) \right] \right] \right] \]

**A compositional treatment of modification requires the representation of sufficient grammatical structure to represent the scope of modification.**

First order predicate logic and its extensions result in relatively flat representations in which the arguments of predicates are always atomic individuals. The only things that are allowed to range over non-atomic formulas are logical operators.

**Modal logics** extend predicate logic by adding modal operators for *possibility* and *necessity*, among others. Consider the expression

*John might kick a ball*

which can be represented as

\[ \Box \left[ \exists e \exists x : \text{ball}(x) \left[ \text{kick}(e, J, x) \land \text{pres}(e) \land \text{subj}(e, J) \land \text{obj}(e, x) \land \text{active}(e) \land \text{kicker}(e, J) \land \text{kickee}(e, x) \right] \right] \]

where \( \Box \) is the possibility operator (i.e. *it is possible that...*).

**Temporal logics** extend predicate logic by adding *temporal* operators and/or variables for time. Assuming a temporal operator \( \text{pres} \) (present time), we can represent

*John kicks a ball*

as

\[ \text{pres} \left[ \exists e \exists x : \text{ball}(x) \left[ \left[ \text{kick}(e, J, x) \land \text{subj}(e, J) \land \text{obj}(e, x) \land \text{active}(e) \land \text{kicker}(e, J) \land \text{kickee}(e, x) \right] \right] \right] \]
where the operator \texttt{pres} (present time) has \([\text{kick} (e, J, x) \land \text{subj} (e, J) \land \text{obj} (e, x) \land \text{active} (e) \land \text{kicker} (e, J) \land \text{kickee} (e, x)]\) within its scope.

If we combine modality and time, the following representation is suggested:

\[
\diamond \left[ \exists e \exists x : \text{ball} (x) [\text{kick} (e, J, x) \land \text{pres} (e) \land \text{subj} (e, J) \land \text{obj} (e, x) \land \text{active} (e) \land \text{kicker} (e, J) \land \text{kickee} (e, x)] \right]
\]

where the present time operator is nested within the possibility operator. We can translate this representation back into English as

\textit{It is possible that at the present moment John kicks a ball}

In Double-R Grammar, modality and tense (i.e. grammatically encoded time) are represented as grammatical features of situation referring expressions. Grammatical features function like logical operators in having sentential scope. However, there is no nesting of grammatical features – they are unordered and conjoined – more like conjoined predicates than logical operators. Modality and tense are projected to the situation referring expression from the lexical items in the expression which encode them. The lexical items, themselves, are represented in situ.

The Davidsonian event variable makes it possible to avoid the nesting of operators, so long as the event variable is the right level (and right variable) for representing tense and modality. For example, the possibility and present tense operators can be represented as

\[
\exists e [ \exists x : \text{ball} (x) [\text{kick} (e, J, x) \land \text{possible} (e) \land \text{pres} (e) \land \text{subj} (e, J) \land \text{obj} (e, x) \land \text{active} (e) \land \text{kicker} (e, J) \land \text{kickee} (e, x)]]
\]

 retains a relatively flat representation that aligns with Double-R Grammar's treatment of modality and tense as (event=situation level) grammatical features. In this representation, modality and tense are represented as conjoined predicates, not sentential operators. However, their only argument is the event variable, giving them scope over the event.

Blackburn & Bos (2003) provide an alternative treatment of the possibility operator, inspired by intensional logic, that relies on the introduction of a variable to represent a possible world, rather than using the event variable. Using a possible world variable gives the following representation:

\[
\exists w : \text{world} (w) [\exists e : \text{event} (e) [\exists x : \text{ball} (x) [\text{kick} (w, e, J, x) \land \text{possible} (e) \land \text{pres} (e) \land \text{subj} (e, J) \land \text{obj} (e, x) \land \text{active} (e) \land \text{kicker} (e, J) \land \text{kickee} (e, x)]]]
\]

In this representation, \textit{John might kick a ball} is translated as meaning that there is some world in which there is an event of \textit{John kicking a ball at the present moment}. This representation introduces a special world predicate in addition to the possible world variable. For consistency, we have introduced a special event predicate as well, and treated both as restrictions.
As an alternative to the use of a temporal operator, or a temporal predicate which references the event variable, to represent time, a distinct time variable can be used. If we add a time variable $t$ in addition to an event variable, we can represent *John kicks a ball* as

$$\exists e \exists t (\exists x \text{ball}(x) \land \text{kick}(e, t, J, x) \land t = \text{now} \land \text{subj}(e, J) \land \text{obj}(e, x) \land \text{active}(e) \land \text{kicker}(e, J) \land \text{kickee}(e, x))$$

where *now* is a special temporal constant that represents the present moment. It is common in such approaches to include $>$ (later than) and $<$ (earlier than) as operators in addition to (or in place of) $=$ (at the same time as). As is the case with other variables, addition of the time variable supports reference from other parts of the logical representation.

It is a bit of an overstatement to describe predicate logic representations as being flat. We have already seen that the universal and existential quantifiers have scope over sentences, as does the negation operator. These three operators are unary operators. The other logical operators are all binary – they take scope over two sentences. When there is more than one operator in a logical representation, the scope of the logical operators must be represented, either implicitly via an operator preference hierarchy or explicitly via bracketing. For example, when there are two quantifiers in a representation, the relative scope of the operators must be determined. Consider the sentence

*Every man loves a woman*

which can be represented in standard predicate logic as

$$\forall x (\text{man}(x) \rightarrow \exists y (\text{woman}(y) \land \text{love}(x, y)))$$

In this representation, for every man, there is a woman (perhaps different), that he loves. This is represented by giving the universal quantifier scope over the existential quantifier (indicated by the bracketing). However, this is not the only possibility, since this sentence is ambiguous. There is an alternative interpretation in which there is a single woman that all men love. This can be represented as

$$\exists y (\text{woman}(y) \land \forall x (\text{man}(x) \rightarrow \text{love}(x, y)))$$

where the existential quantifier has scope over the universal quantifier.

It is external to the logic which interpretation is actually intended. If one is building a system that translates linguistic expressions into logical representations, one must either choose between the alternatives, or allow for the possibility of both. To do either of these, we must step outside of logic proper and provide a computational mechanism or mechanisms to do so. One frequently cited computational approach to quantifier scope resolution is based on a storage mechanism called Cooper storage (Cooper, 1983). The basic idea is to keep the quantifiers in a memory store until sufficient context is available to resolve their relative scope. The Cooper storage mechanism has the advantage of not changing the logical representations. It also makes it possible to deal with the fact that the
location of quantifiers in linguistic expressions differs significantly from their location in predicate logic representations.

Another approach, implemented in Minimal Recursion Semantics (Copestake et al., 2005), adds handle variables to the logical representations. These handle variables support the underspecification of quantifier scope until sufficient context is available to resolve the variables, perhaps in a separate computational process. This latter approach has the disadvantage of complicating the logical representations – a distinct handle variable must be added for each quantifier and for each argument of the quantifier, except the quantified variable, itself. For the example above, the following MRS representation is suggested:

\[ h_1: \text{every}(x, h_3, h_4), h_3: \text{man}(x), h_4: a(y, h_5, h_6), h_5: \text{woman}(y), h_6: \text{love}(x, y) \]

In this representation, the generalized quantifier \text{every} is represented as the three argument predicate \text{every}(x, h_3, h_4) (roughly equivalent to \( \lambda x [ \lambda P(\forall x (Q(x) \rightarrow P(x))) ] \)), and has scope over the generalized quantifier \text{a}, represented as \text{a}(y, h_5, h_6) (roughly equivalent to \( \lambda y [ \lambda P(\exists y (Q(y) \land P(y))) ] \)). Expressions like \text{every}(x, h_3, h_4) and \text{man}(x) are called \text{Elementary Predications} (EPs). Each EP has a handle, allowing the EP to be referenced elsewhere in the representation. There is also an implicit n-ary conjunction of all the EPs (e.g. \( \land (\text{EP}_1, \text{EP}_2, \text{EP}_3, \ldots, \text{EP}_n) \)) which has the effect of flattening the representation.

Once handle variables are added to logical representations, it is possible to generalize the representations so that quantifier scope is underspecified. The following representation does this:

\[ h_1: \text{every}(x, h_3, h_8), h_3: \text{man}(x), h_4: a(y, h_7, h_9), h_5: \text{woman}(y), h_6: \text{love}(x, y) \]

Note that in this representation, the third arguments of \text{every}, \text{h}_8, and \text{a}, \text{h}_9, do not reference any EP. If we set \text{h}_8 = \text{h}_4 and \text{h}_9 = \text{h}_6, we get the reading in which \text{every} has scope over \text{a}. On the other hand, if we set \text{h}_8 = \text{h}_6 and \text{h}_9 = \text{h}_1, we get the alternative reading in which \text{a} has scope over \text{every}.

The MRS representations have the advantage of not requiring quantifier scope disambiguation when it is not possible or necessary to do so. MRS representations are also very flat – the only structure above the EPs is the n-ary conjunction, although it is possible to recover the grammatical structure from the handle variables.

In Double-R Grammar, which is not focused on resolving quantifier scope, handle variables are used to encode grammatical structure in an otherwise flat representation. It may be possible to use these same variables to address the challenges of quantifier scope.

Discourse Representation Theory (DRT) aligns with MRS in presenting relatively flat logical representations. In DRT representations, called \text{Discourse Representation Structures} (DRS), there are \text{discourse referents, conditions} and \text{logical operators}, grouped into levels, with discourse referents corresponding to implicitly quantified variables and conditions within a level implicitly conjoined. A DRS for the sample sentence is shown below using a text based followed by a graphical box notation:
In these representations, there are two discourse referents, $x$ and $y$, three conditions, and the material implication logical operator organized into levels. According to Kamp (1981), DRT is a notational variant of predicate logic. It is possible to algorithmically generate the predicate logic equivalent of any DRS. The algorithm that does this relies on the context of the variables to determine which quantifiers are appropriate. In this example, variables before the material implication are universally quantified and variables after the material implication are existentially quantified. Conditions within a box are conjoined. This results in the following translation:

$$\forall x [\text{man}(x) \rightarrow \exists y (\text{woman}(y) \land \text{love}(x, y))]$$

which is one of the two readings for every man loves a woman. A suggested DRS for the alternative reading follows:
In this representation, the discourse referent, \( \nu \), in the outer box, is set equal to \( y \) in the inner right box. The introduction of \( \nu \) in the outer box is needed to get the quantifier scope correct. This DRS corresponds to the following predicate logic representation:

\[
\exists \nu [\forall x [\text{man}(x) \rightarrow \exists y [\text{woman}(y) \land \text{love}(x, y) \land y = \nu]]]
\]

From a processing perspective, the need to introduce \( \nu \) in the outer box cannot be determined until the expression \textit{a woman} is processed, unless a discourse referent is automatically associated with each box.

In Double-R Grammar, quantifiers are represented in situ. Quantifier scope is not determined by logico-grammatical representations. Quantifier scope resolution is presumed to occur in the mapping of referring expressions into the situation model. For this to work, the situation model must support structured representations – i.e. situation model representations capable of distinguishing quantifier scope. It is likely that the interpretation function that maps into the situation model will choose a single interpretation. For the example, the function is likely to give \textit{every man} wide scope. If the intention were for \textit{a woman} to have wider scope, the following expression is suggested:

\textit{every man loves the woman}

where use of the definite determiner \textit{the} indicates reference to a unique individual (in the model) that every man loves. It is clear from this expression that generic reference to an arbitrary woman is not intended. Although a wide scope interpretation of \textit{a woman} is logically possible, it is not practically achievable, since humans are likely to prefer to give wide scope to the leftmost quantifier. What \textit{a woman} is likely to do is introduce a single referent into the situation model that is not ground in the mental universe.

DRT was designed to support the representation of multi-sentential discourse – especially discourse involving pronouns. Given the expression

\textit{John likes Mary. She is nice.}

The following representation is suggested:
In this representation, three discourse referents are introduced. The third referent, $v$, is introduced by the pronoun *she*. The condition *female*($v$) encodes the gender feature of *she*. Other features are ignored. The referent, $v$, is set equal to $y$ to indicate co-reference. There is nothing in this representation to indicate why *she* is made co-referential with *Mary* and not *John*. In Double-R Grammar, the lexical entries for proper nouns like *John* and *Mary* have grammatical features just like pronouns. It is the female gender feature of both *Mary* and *she* that supports co-reference.

Episodic Logic (Schubert, 1999) introduces two additional operators $e^*$ and $e^{**}$ that support reference to complex situations consisting of multiple events (i.e. episodes). The $e^*$ operator follows Situation Semantics in characterizing a partial complex situation that may be described by multiple sentences, whereas the $e^{**}$ operator follows Davidson's event variable in characterizing a complete situation that may be described by multiple sentences. The $e^*$ and $e^{**}$ operators are essentially discourse level handles. For the multi-sentential discourse above, the following Episodic Logic representation is suggested:

$$e^{**}[\exists y \exists x (x = John \land y = Mary \land \text{like}(x, y) \land \text{nice}(y))]$$

Note that the $e^{**}$ operator has both $\text{like}(x, y)$ and $\text{nice}(y)$ within its scope. Separate Davidsonian event variables would be needed to represent this discourse:

$$\exists e_1 \exists e_2 \exists x \exists y (x = John \land y = Mary \land \text{like}(e_1, x, y) \land \text{nice}(e_2, y))$$

Multi-sentential event variables have not yet been added to Double-R Grammar, but are likely to be needed to better model discourse.

Situation Semantics (Barwise & Perry, 1983) has had an important influence on the development of Double-R Grammar. Barwise and Perry (1983) reject Frege's assertion that the reference or denotation (i.e. bedeutung in German) of a statement is a truth value in extensional contexts, replacing truth value with described situation. The interpretation of a linguistic expression describing a situation is the described situation, and the linguistic expression is said to reference the situation. Whether or not the described situation is true or false in the actual world can only be evaluated after interpretation of the linguistic expression. It is this perspective of Situation Semantics that motivates the use of the term situation referring expression in Double-R Grammar.

To overcome severe problems with the treatment of the denotation of a statement as a truth value, Frege introduced a secondary notion of sense. In many contexts, especially intensional contexts, the meaning of a statement is taken to be the sense, not the reference. For example, in

*John believes dogs are mammals*

the contribution of *dogs are mammals* to the meaning of the overall expression is assumed to be the sense and not the reference. If this were not so, then in

*John believes cats are mammals*
dogs are mammals and cats are mammals would both contribute true to the overall meaning of the expression. Further, in the expression

John believes bats are birds

bats are birds would contribute false to the overall meaning, even though John may actually believe this to be true. Treating the sense of an expression that describes a situation as its contribution to meaning comes close to affording situations first class status. Barwise and Perry (1983) take this logical next step.

Barwise and Perry (1983) also make an important distinction between efficient and non-efficient statements. Efficient statements refer to situations that are time and space dependent – i.e. context dependent. Non-efficient statements are statements that are universally true, irrespective of time and space. Whereas much of the focus in predicate logic has been on determining the validity of inference over non-efficient statements, ordinary language is primarily concerned with expressing efficient, or context dependent, statements. The situation model in Double-R Grammar is a stage for the construction of situations expressed by context dependent statements.

Kratzer (2014) extends situation semantics with the use of λ abstraction to represent situations, and proposes an integration with Davidsonian event variables. For the discourse,

John likes Mary. She is nice.

the following representation is suggested:

$$\lambda s[\exists e[e \leq p s \land \text{likes}(e)(s) \land \exists x[\text{nice}(x)(s) \land \text{female}(x)(s) \land x=M(s)]]]$$

In this representation, the Davidsonian event e (i.e. a liking event) is a subpart ($\leq p$) of the larger situation s which also includes the properties of Mary being nice and female (from she). In Kratzer’s notation, arguments of the predicate are individually parenthesized. Incorporating them into the predicates gives the more familiar:

$$\lambda s[\exists e[e \leq p s \land \text{likes}(e, J, M)] \land \exists x[\text{nice}(s, x) \land \text{female}(s, x) \land = (s, x, M)]]$$

To get this representation, it was necessary to use the Polish prenix notation for the equality operator (i.e. $=(s, x, M)$), since there are three arguments.

**Overview of the Logical Representation of Situation Referring Expressions in Double-R Grammar**

The key feature of Double-R Grammar based representations of situation referring expressions is that they are grammatical – i.e. they represent the grammatical structure of the linguistic input. In providing a logical description, we do not abstract away from grammatical structure. As a result, Double-R Grammar based logical representations diverge from predicate logic representations where predicate logic representations diverge from grammatical structure. For example, it is broadly accepted that English sentences are grammatically divided into a subject and a predicate. Double-R Grammar representations
of situation referring expressions respect this grammatical divide. Quantifiers are typically parts of object referring expressions, not parts of situation referring expressions. Double-R Grammar represents quantifiers in situ. Quantifiers are not treated as operators that take situation referring expressions within their scope, as in predicate logic. In fact, there are no equivalents to the universal and existential quantifiers in Double-R Grammar, although Double-R Grammar does have a wide range of different quantifiers. Quantifiers are identified on the basis of meaning, unlike many syntactic treatments which do not recognize quantifiers as a distinct part of speech. Modality is primarily expressed by modal auxiliaries (e.g. could, should, must). Like quantifiers, modal auxiliaries are represented in situ. However, the modality that is lexically expressed by a modal auxiliary is projected to the situation referring expression as a grammatical feature, affording modality an operator like status. Grammatical features are the closest equivalent to a unary operator in Double-R Grammar. Besides modality, they include tense, aspect, polarity and voice. Tense, aspect and polarity are represented as grammatical features of the lexical items that express them, but they get projected to situation referring expressions as situation level grammatical features. Semantically motivated grammatical functions like subject and object are explicitly represented. The voice grammatical feature is projected to situation referring expressions from the relevant lexical items (i.e. auxiliary and main verb) to support a distinction between the logical subject and the surface subject. In this manner, Double-R Grammar respects grammatical structure while supporting the determination of logical argument structure. Handle variables are introduced so that elements of a grammatical representation can be referenced by other elements of the grammatical representation. This is especially important for representing elliptical expressions, sentential and predicate modification and for co-reference resolution.

It is clear from linguistic studies (cf. Chomsky, 1981), that grammatical structure is crucial for determining meaning. How could it be otherwise? Logical representations which abstract away from grammatical structure do so at their peril. Simplified logical representations may make determination of truth, determination of scope, or logical inference more tractable, but to the extent that they abstract away from grammatically encoded meaning, they make full determination of meaning more difficult – perhaps impossible.

In attempting to motivate the autonomy of syntax hypothesis, generative grammarians have posited numerous syntactic structures which they claim to be semantically unmotivated. Cognitive Linguistics rejects both the autonomy of syntax hypothesis and the assumption that many of these syntactic structures are semantically unmotivated (cf. Lakoff, 1987). Double-R Grammar aligns with Cognitive Linguistics in this regard. If grammatical structure is semantically motivated, that structure needs to be encoded in meaning representations.

Many logical approaches (cf. Montague, 1973; Steedman, 2000) assume a rule-by-rule alignment for the simultaneous determination (or processing) of syntactic structure and generation of logical representation. These approaches are able to deal with misalignments, but to the extent that the resulting logical representations abstract away from the syntactic structure, they lose information. For some purposes, the logical
representations may be sufficient. But for many purposes, they will not be adequate. As a simple example, if the male gender feature of John is not represented, then determination of co-reference with the pronoun he is problematic in

_Mary likes John. He is nice._

Of course, this feature is lexically determined, but the co-reference of _him_ vs. _himself_ is grammatically determined:

_John likes him
John likes himself_

The personal pronoun _him_ functioning as object cannot be co-referential with the subject _John_, whereas the reflexive pronoun _himself_ must be. Either this information must be represented in the logical representation, or co-reference determination must consider syntactic structure as well as logical structure. In Double-R Grammar, they are one and the same.

Situation Semantics provides the logical back drop for the representation of situation referring expressions in Double-R Grammar. However, where Barwise and Perry (1983) use the term _reference_ with caution, in Double-R Grammar, it is embraced. Reference to situations is viewed as essentially equivalent to reference to objects, except that situations are not atomic in the way that objects are (at least in the naïve physics that most humans assume for objects). Situations are composed of relations and their arguments. Unlike model-theoretic predicate logic, situations are not reduced to ordered tuples of the individuals participating in the relation, or sets of individuals with a given property. Situations retain their structure in the situation model – they are the non-linguistic cognitive correlates of situation referring expressions. Situation referring expressions, and language more generally, provides a mirror for cognition, suggesting structured cognitive representations that go well beyond the representational capabilities of set theory. From the perspective of Double-R Grammar, it is incoherent to suggest that the predicate _run_ means the set of individuals who run, or that _kick_ means the set of ordered pairs of individuals such that the first individual is the kicker and the second individual is the kickee (assuming the arguments are somehow associated with these semantic roles). The set theoretic definition of predicates in terms of the individuals that function as arguments of the predicate, fails to capture much of the meaning. If we look at properties like _old_, it may seem feasible to define them in set terms. The property _old_ denotes the set of individuals who are old. Likewise for words like _man_ which denote a class. Here we make a distinction between _denotation_ and _reference_ (cf. Lyons, 1977). Nouns denotes classes (of individuals) and adjectives denote properties (of individuals). By comparison, object referring expressions typically refer to individual objects, and situation referring expressions typically refer to individual situations. Words like _old_ and _man_ are seldom used in isolation. Typically, they are used in expressions like _an old man_ or _the man is old_, in which they refer to some specific individual object or situation. Even when used in isolation, they are typically accompanied by some kind of pointing gesture that indicates reference to a specific individual. Ordinary language is all about pointing out objects and saying something about them. In the expression
the old man is laughing

the old man points out a salient object, and the predicate is laughing provides new information about that object – referencing a new situation (or extending an existing situation) in the process. There is no a priori set of laughing individuals in a model against which the truth value of this expression is evaluated. This expression is an instruction to the hearer to update the information about an individual who has already been introduced into the conversation or is otherwise salient. Once the information is added to the situation model, it can be evaluated against the mental universe:

no he’s not, he’s crying

Of course, there may be no access to the described situation, in which case, the information will simply be assumed to be true:

if you say so

Language is often, perhaps typically, used to describe situations at an unobservable distance – often a distance in time. If one is interested in modeling ordinary language, the primary task is to identify or introduce objects and situations into a situation model based on the linguistic input. It may be possible to verify the truth of some of the assertions, but it may well not be. To determine truth, an adequate situation model representation of the truth conditions is presupposed, but is not sufficient.

Since the typical declarative sentence is asserting some new piece of information, situation referring expressions are usually indefinite. They add new information to the situation model. It is, of course, possible to refer to an existing situation. However, reference to an existing situation is usually achieved by reifying the situation and referring to it with an object referring expression. For example, in

John kicked the ball. That was unexpected.

the demonstrative pronoun that reifies and refers back to the situation referred to by the situation referring expression John kicked the ball. In a logical system in which it is not possible to refer to a situation, this function of demonstrative pronouns is unexplained.

In embracing use of the term reference to describe the function of referring expressions, a distinction is made between reference to objects and situations in the constructed situation model and reference to objects and situations in the mental universe. An object referring expression may refer to an object in the situation model which has no correlate in the mental universe of experience of an individual. This is effectively an attributive use of the object referring expression (cf. Donnellan, 1966). Many of the purported non-referential uses of object referring expressions (cf. Partee, 1972) can be modeled in this manner. For example, in

John wants to catch a fish and eat it (Partee, 1972)

the object referring expression a fish introduces a referent into the situation model, even though this referent does not exist in the mental universe. This makes it possible to refer
back to the referent with the pronoun it. As Lyons (1977) notes, co-reference is problematic if the antecedent is not itself a referring expression. In ordinary language, referents are typically introduced into situation models via use of indefinite object referring expressions and referred back to via use of definite object referring expressions. Pronouns are just one way of expressing definite object referring expressions. Definite descriptions are an alternative way:

*John wants to catch a fish. I hope the fish is big enough to keep.*

In this example, the definite description *the fish* refers back to the referent introduced into the situation model by *a fish*.

Double-R Grammar differs from predicate logic in treating the subject as an *external argument* of the verb, and other arguments (e.g. object, indirect object) as internal to the verb. This adds a level of structure that is typical of grammatical treatments of English (i.e. verb phrase or VP), but not predicate logic representations. It represents the division of a sentence into a *subject* and a *predicate* – where predicate includes the verb, any non-subject arguments, and predicate modifiers. Early versions of Double-R Grammar (Ball, 1992) followed predicate logic in having flat predicate-argument representations. However, grammatical evidence for the division of a sentence into a subject and a predicate, in English, ultimately held sway, and it was introduced into Double-R Grammar.

As an example of grammatical evidence, the extra level of grammatical structure is important for representing elliptical expressions in dialog:

*John likes Mary. Doesn’t he?*

*John kicked the ball after I told him to ∅.*

There is an implicit (untensed) predicate *like Mary* in the question *Doesn’t he?* There is also an implicit (untensed) predicate *kick the ball* in the subordinate clause *I told him to*. Such examples create challenges for predicate logic based representations of dialog which have no level of structure corresponding to the (untensed) predicate. The addition of $\lambda$ abstraction to predicate logic offers a way of representing predicates. For example, the logical expression

$$\lambda s[\exists o[kick(s,o) \land ball(o)]]$$

is suggested as an approximate representation for *kick the ball*. In this representation, the $\lambda$ variable $s$ can be instantiated by the subject *John* via $\beta$ reduction giving

$$\lambda s[\exists o[kick(s,o) \land ball(o)]](J) \leftrightarrow \exists o[kick(J,o) \land ball(o)]$$

Alternatively, the $\lambda$ variable can be retained:

$$\lambda s[\exists o[kick(s,o) \land ball(o)]] \land J = s$$

Double-R Grammar aligns with this latter alternative, except that a $\eta$ handle variable, which is not reduced, is provided for the predicate. The $\eta$ handle variable indicates that the predicate is not a referring expression – i.e. the predicate is an incomplete situation
description in need of a subject and tense in order to refer. This handle variable gives the predicate first class status, allowing it to be referenced by other parts of the representation. First class status is needed to support predicate modification. The handle variable is like a Davidsonian event variable, except that the subject is represented at the level of the situation referring expression, which has its own handle variable, and not at the level of the predicate. Unlike Davidsonian events, Double-R Grammar has handle variables to support the representation of both predicate and situation referring expression level modification (cf. Thomason & Stalnaker, 1973; Fodor, 1972), and even nested modification.

To provide an example of the representation and processing of situation referring expressions in this overview, we will use the following sentence from the previous section:

_Every man loves a woman_

The processing of the object referring expression _every man_ results in the following representation:

\[
\text{ore} + s = \text{every} + h = \text{man} \quad \lambda o \ \eta s \ \eta h \ . \ \text{ore}(o) \ \wedge \ \text{spec}(o, s) \ \wedge \ \text{quantifier}(s) \ \wedge \ \text{index}(s, \ \text{every}) \ \wedge \ \text{head}(o, h) \ \wedge \ \text{man}(h) \ \wedge \ \text{noun}(h) \ \wedge \ \text{index}(h, \ \text{man}) \ \wedge \ \text{def}(o, \ \text{indef})
\]

Note that _every_ is a quantifier that functions as the specifier of the object referring expression (ore), and _man_ is a noun that functions as the head.

The processing of _every man_ does not result in the projection of a situation referring expression. Object referring expressions are often used to point out objects without participating in an assertion. Since the object referring expression corresponding to _every man_ does not project a higher level construction in which it can be instantiated, it must be retained in memory until such a construction is available. Double-R Grammar provides a grammatical function specific _subject buffer_ to store the uninstantiated subject (and to support later co-reference).

The processing of _loves_ results in its identification as a transitive verb (trans) via the _verb-frame_ grammatical feature:

\[
\text{loves} : \eta x \ . \ \text{love}(x) \ \wedge \ \text{verb}(x) \ \wedge \ \text{index}(x, \ \text{love}) \ \wedge \ \text{verb-frame}(x, \ \text{trans}) \ \wedge \ \text{tense}(x, \ \text{pres}) \ \wedge \ \text{voice}(x, \ \text{act})
\]

In this lexical representation, we ignore grammatical features other than _verb-frame, tense, voice_ and _index_. The verb _loves_ projects a predicate transitive verb (pred-trans-verb) construction in which it is integrated as the head:

\[
\text{pred-trans-verb} : \ \eta p \ \eta h \ \lambda \text{obj} \ . \ \text{pred-trans-verb}(p) \ \wedge \ \text{head}(p, h) \ \wedge \ \text{love}(h) \ \wedge \ \text{verb}(h) \ \wedge \ \text{index}(h, \ \text{love}) \ \wedge \ \text{obj}(p, \ \text{obj}) \ \wedge \ \text{tense}(p, \ \text{pres}) \ \wedge \ \text{voice}(p, \ \text{act})
\]

Note that _\eta p_ is the handle for the pred-trans-verb construction itself, and _\eta h_ is the handle for the head. The grammatical features tense and voice get projected to the pred-trans-verb construction from the verb _loves_. Note that there is no subject grammatical function in the pred-trans-verb construction. The subject is an _external argument_ of the predicate. It is instantiated at the level of the situation referring expression. Also note that the object
handle λ.obj is yet to be instantiated. The object handle is represented as a λ variable since it is a referring expression.

The pred-trans-verb construction projects a situation referring expression in which it is instantiated as the (predicate) head, and the object referring expression in the subject buffer is simultaneously instantiated as the subject:

\[ \lambda \text{sit} \lambda \text{subj} \eta s \eta h_1 \eta r \eta h_2 \lambda \text{obj}. \text{sre}(\text{sit}) \land \text{subj}(\text{sit}, \text{subj}) \land \text{ore}(\text{subj}) \land \text{spec}(\text{subj}, s) \land \text{quantifier}(s) \land \text{index}(s, \text{every}) \land \text{head}(\text{subj}, h_1) \land \text{man}(h_1) \land \text{noun}(h_1) \land \text{index}(h_1, \text{man}) \land \text{def}(\text{subj}, \text{def}) \land \text{head}(\text{sit}, p) \land \text{pred-trans-verb}(p) \land \text{head}(p, h_2) \land \text{loves}(h_2) \land \text{verb}(h_2) \land \text{index}(h_2, \text{love}) \land \text{obj}(p, \text{obj}) \land \text{tense}(\text{sit}, \text{past}) \land \text{voice}(\text{sit}, \text{act}) \]

Note that the handle variable η supports instantiation of the pred-trans-verb construction as the (predicate) head of the situation referring expression construction. Note also that the grammatical features tense and voice get projected from the pred-trans-verb construction to the situation referring expression construction.

Finally, the processing of the object referring expression \textit{a woman} leads to its instantiation as the object of the pred-trans-verb construction:

\[ \lambda \text{sit} \lambda \text{subj} \eta s_1 \eta h_1 \eta r \eta h_2 \eta s_3 \eta h_3 \lambda \text{obj}. \text{sre}(\text{sit}) \land \text{subj}(\text{sit}, \text{subj}) \land \text{ore}(\text{subj}) \land \text{spec}(\text{subj}, s_1) \land \text{quantifier}(s_1) \land \text{index}(s_1, \text{every}) \land \text{head}(\text{subj}, h_1) \land \text{man}(h_1) \land \text{noun}(h_1) \land \text{index}(h_1, \text{man}) \land \text{def}(\text{subj}, \text{def}) \land \text{head}(\text{sit}, p) \land \text{pred-trans-verb}(p) \land \text{head}(p, h_2) \land \text{love}(h_2) \land \text{verb}(h_2) \land \text{index}(h_2, \text{love}) \land \text{obj}(p, \text{obj}) \land \text{ore}(\text{obj}) \land \text{spec}(\text{obj}, s_3) \land \text{det}(s_3) \land \text{index}(s_3, a) \land \text{head}(\text{obj}, h_3) \land \text{woman}(h_3) \land \text{noun}(h_3) \land \text{index}(h_3, \text{woman}) \land \text{def}(\text{obj}, \text{indef}) \land \text{tense}(\text{sit}, \text{past}) \land \text{voice}(\text{sit}, \text{act}) \]

For the object referring expression \textit{a woman} to be instantiated as the object, the pred-trans-verb construction must be available in storage. A grammatical function specific predicate buffer is provided for this purpose.

In the Double-R Grammar representation below, the parts of the representation are highlighted that correspond most closely to the following predicate logic representation with situation and event variables:

\[ \lambda s \exists e[e \leq p, s \land \forall x[\text{man}(s, x) \rightarrow \exists y[\text{woman}(s, y) \land \text{love}(e, x, y)]]] \]

\[ \lambda \text{sit} \lambda \text{subj} \eta s_1 \eta h_1 \eta r \eta h_2 \eta s_3 \eta h_3 \lambda \text{obj}. \text{sre}(\text{sit}) \land \text{subj}(\text{sit}, \text{subj}) \land \text{ore}(\text{subj}) \land \text{spec}(\text{subj}, s_1) \land \text{quantifier}(s_1) \land \text{index}(s_1, \text{every}) \land \text{head}(\text{subj}, h_1) \land \text{man}(h_1) \land \text{noun}(h_1) \land \text{index}(h_1, \text{man}) \land \text{def}(\text{subj}, \text{def}) \land \text{head}(\text{sit}, p) \land \text{pred-trans-verb}(p) \land \text{head}(p, h_2) \land \text{love}(h_2) \land \text{verb}(h_2) \land \text{index}(h_2, \text{love}) \land \text{obj}(p, \text{obj}) \land \text{ore}(\text{obj}) \land \text{spec}(\text{obj}, s_3) \land \text{det}(s_3) \land \text{index}(s_3, a) \land \text{head}(\text{obj}, h_3) \land \text{woman}(h_3) \land \text{noun}(h_3) \land \text{index}(h_3, \text{woman}) \land \text{def}(\text{obj}, \text{indef}) \land \text{tense}(\text{sit}, \text{past}) \land \text{voice}(\text{sit}, \text{act}) \]

Note the following variable correspondences between the Double-R Grammar and predicate logic representations:

- \( \lambda \text{sit} \sim \lambda s \)
Of these, the correspondence between the predicate handle and predicate head variables of Double-R Grammar ($\eta p + \eta h_2$) and the event variable of the predicate logic representation ($\exists e$) is the weakest.

If we remove the extra detail, we have:

$$\lambda \text{sit} \lambda \text{subj} \eta h_1 \eta p \eta h_2 \eta h_3 \lambda \text{obj} \cdot \text{sre(sit)} \land \text{subj(sit, subj)} \land \text{ore(subj)} \land \text{index}(s_1, \text{every}) \land \text{head(subj, h}_1) \land \text{man(h}_1) \land \text{head(sit, p)} \land \text{pred-trans-verb(p)} \land \text{head(p, h}_2) \land \text{love(h}_2) \land \text{obj(p, obj)} \land \text{ore(obj)} \land \text{index}(s_3, a) \land \text{head(obj, h}_3) \land \text{woman(h}_3)$$

In this representation: 1) $\lambda$ variables provide handles for referring expressions, 2) $\eta$ variables provide handles to support reference to parts of the grammatical representation that are not referring expressions, 3) there is a top level division into a subject and a predicate (i.e. head of situation referring expression), 4) the subject is represented as an external argument of the predicate, 5) the predicate is headed by a transitive verb, and 5) the object of the transitive verb is represented as an internal argument of the predicate.

On the downside, the relative scope of the quantifiers $\text{every}$ and $a$ is not represented in Double-R Grammar. A possible way to indicate this scope is as follows:

$$\eta s_3 \subseteq \text{scope } \eta s_1$$

The ramifications of using such a mechanism have not yet been explored.

**The Logical Representation of Parts of Speech in Double-R Grammar**

In Double-R Grammar, parts of speech are viewed as abstract semantic categories and included in logico-grammatical representations. By themselves, lexical items categorized into parts of speech are not referring expressions, although they can project constructions which, when instantiated, function as referring expressions. In this section, we provide a brief overview of the parts of speech. Full details are available at Ball & Freiman (2014).

Parts of speech are organized into a multiple-inheritance lattice (technically semi-lattice is more accurate, since they have a single most general type, but not a single most specific type). The term **tangled hierarchy** is also used informally. The part of speech lattice is shown below:
This part of speech lattice can be represented logically with meaning postulates like:

\[
\forall x \left[ \text{proto-noun}(x) \rightarrow \text{part-of-speech}(x) \right] \\
\forall x \left[ \text{noun}(x) \rightarrow \text{proto-noun}(x) \right] \\
\forall x \left[ \text{nominal}(x) \rightarrow \text{proto-noun}(x) \right] \\
\forall x \left[ \text{determiner}(x) \rightarrow \text{part-of-speech}(x) \right] \\
\forall x \left[ \text{quantifier}(x) \rightarrow \text{nominal}(x) \right] \\
\forall x \left[ \text{quantifier}(x) \rightarrow \text{determiner}(x) \right]
\]

However, parts of speech in Double-R Grammar have internal structure (not shown in the diagram) that must also be represented. For example, determiners encode the definiteness and number grammatical features. A logical description of the grammatical features associated with each part of speech is yet to be developed, although representations like

\[
\forall x \left[ \text{determiner}(x) \rightarrow \text{def}(x, \text{Def}) \land \text{number}(x, \text{Num}) \right] \text{ such that} \\
\text{Def} \in \{\text{def, indef, univ}\} \\
\text{Num} \in \{\text{sing, plur}\}
\]

may be sufficient for the purposes of this paper.

Parts of speech that inherit from more than one parent (e.g. quantifier) are called composite. Composite parts of speech capture the behavior of words that typically exhibit multiple grammatical functions, without any particular grammatical function being central.
We limit the discussion in this paper to the following parts of speech:

- Determiner
- Noun (aka Common Noun)
- Proper Noun
- Pronoun
- Nominal
- Quantifier
- Adjective
- Adverb
- Verb
- Auxiliary Verb

**Determiner**

Determiners function as specifiers in object referring expressions and primarily indicate the definiteness of object referring expressions. Definiteness determines whether the object referring expression refers to an existing object in the situation model, to a salient object in the mental universe, or to an object that should be introduced into the situation model. There are few true determiners in Double-R Grammar – just the words *the*, *a*, *an*, and *no* (essentially the articles in traditional grammar plus *no*). Object referring expressions with determiners functioning as specifier are ungrammatical without a head noun:

- *the* is going
  - the man is going
- *no* is here
  - no one is here

The requirement for a head noun grammatically distinguishes determiners from quantifiers and demonstrative pronouns.

The logical representation of determiners (det) encodes the part of speech and two grammatical features: definiteness (def) and number (num). Double-R Grammar also encodes the word and an index which generalizes over morphological variants. The logical representation for all determiners is shown below:

\[
\text{det}: \exists x . \text{det}(x) \land \text{index}(x, \text{Index}) \land \text{word}(x, \text{Word}) \land \text{def}(x, \text{Def}) \land \text{num}(x, \text{Num})
\]

where

- \text{index} = \text{generalization over morphological variants (i.e. lexeme)}
  - with value \text{Index} (e.g. *a*, *the*)
- \text{word} = \text{a particular morphological variant with value \text{Word} (e.g. *an*, *the*)}
- \text{def (definite)} = \text{grammatical feature (GFeat)}
  - \text{Def} \in \{\text{def}, \text{indef}, \text{univ}\}
- \text{num (number)} = \text{GFeat}
  - \text{Num} \in \{\text{sing}, \text{plur}\}
The underlined element is the default value for the grammatical feature. By default, determiners are def (definite) and sing (singular). These default values can be overridden by individual lexical entries.

The logical representation for the determiner *the* is shown below:

\[
\text{the: } \eta x . \text{det}(x) \land \text{index}(x, \text{the}) \land \text{word}(x, \text{the}) \land \text{def}(x, \text{def}) \land \text{num}(x, \text{Num})
\]

This representation introduces no new logical notation, although it does replace the variable Def with the value def for the definiteness (def) feature – indicating that *the* is definite. On the other hand, the value of the number (num) feature is the variable Num and not the default value sing. This is because *the* occurs with both singular and plural nouns:

*the* man  
*the* men

The number feature of an object referring expression is determined by the head and not the determiner functioning as specifier, when the determiner is *the*.

The logical representations for the determiners *a* and *an* are shown next:

\[
\text{a: } \eta x . \text{det}(x) \land \text{index}(x, \text{a}) \land \text{word}(x, \text{a}) \land \text{def}(x, \text{indef}) \land \text{num}(x, \text{sing})
\]
\[
\text{an: } \eta x . \text{det}(x) \land \text{index}(x, \text{a}) \land \text{word}(x, \text{an}) \land \text{def}(x, \text{indef}) \land \text{num}(x, \text{sing})
\]

The index indicates that the words *a* and *an* are morphological variants of the same lexeme. Unlike the determiner *the*, the determiners *a* and *an* only occur with singular nouns, with one notable exception:

*a* man  
*an* otter  
*an* men  
*a few* men (exception)

As a result of this grammatical behavior, the determiners *a* and *an* are treated as singular. The determiners *a* and *an* are also indefinite. For the exception *a few men*, either *a few* is represented as a plural multi-word determiner, or some mechanism for reconciling the singular number feature of *a* with the plural number feature of *few* and *men*, is needed. Both are possible in Double-R Grammar and which mechanism is used is likely to vary from individual to individual.

**Noun**

Nouns (aka common nouns) typically function as the heads of object referring expressions. Singular count nouns are grammatically distinguished from mass nouns, plural count nouns, proper nouns, pronouns and nominals in that object referring expressions in which they function as the head require separate specification:

*man is here* – the noun *man* cannot function alone as an object referring expression
the man is here
rice is healthy – the mass noun rice can function alone
men are here – the plural noun men can function alone
john is here – the proper noun john can function alone
he is here – the pronoun he can function alone
somebody is here – the nominal somebody can function alone

The lexical representation for nouns includes an indication of the specific object subtype that each noun encodes – called a type specification, the noun part of speech, and an unordered sequence of grammatical features that are associated with nouns.

The logical representation for all nouns is shown below:

```
noun: \( \eta x \ 
\eta T \cdot T(x) \land \text{noun}(x) \land \text{index}(x, \text{Index}) \land \text{word}(x, \text{Word}) \land \text{def}(x, \text{Def}) \land \text{num}(x, \text{sing}) \land \text{anim}(x, \text{Anim}) \land \text{gen}(x, \text{Gen}) \land \text{pers}(x, 3rd) \land \text{count}(x, \text{count}) \)
```

where

- \( x \) and \( T \) are \( \eta \) handle variables
- \( T = \) subtype of object (i.e. type specification)
- \( \text{def, num, anim, gen, pers and count are grammatical features such that} \)
  - \( \text{Def} \in \{\text{def, indef}\} \)
  - \( \text{Num} \in \{\text{sing, plur}\} \)
  - \( \text{Anim} \in \{\text{inanimate, animate, human}\} \)
  - \( \text{Gen} \in \{\text{male, female, none}\} \)
  - \( \text{Pers} \in \{1st, 2nd, 3rd\} \)
  - \( \text{Count} \in \{\text{count, mass}\} \)

The logical representation for the noun man is shown next:

```
man: \( \eta x \ 
\text{man}(x) \land \text{noun}(x) \land \text{index}(x, \text{man}) \land \text{word}(x, \text{man}) \land \text{def}(x, \text{Def}) \land \text{num}(x, \text{sing}) \land \text{anim}(x, \text{human}) \land \text{gen}(x, \text{male}) \land \text{pers}(x, 3rd) \land \text{count}(x, \text{count}) \)
```

In this representation, \( \eta T \cdot T(x) \) is reduced to \( \text{man}(x) \) by substituting man for \( T \) on the right hand side of the period following \( \eta T \) and then eliminating \( \eta T \) (equivalent to \( \beta \) reduction).

Note that man is a singular count noun. Singular count nouns do not encode the definiteness feature (i.e. the value of this features is the variable Def), indicating that they are neither definite nor indefinite. This is because singular count nouns do not typically function as object referring expressions without a separate indication of definiteness:

*man is sad*

The logical representation for the plural noun men is shown next:

```
men: \( \eta x \ 
\text{man}(x) \land \text{noun}(h) \land \text{index}(x, \text{man}) \land \text{word}(x, \text{men}) \land \text{def}(x, \text{indef}) \land \text{num}(x, \text{plur}) \land \text{anim}(x, \text{human}) \land \text{gen}(x, \text{male}) \land \text{pers}(x, 3rd) \land \text{count}(x, \text{count}) \)
```
Note that $\eta T \cdot T(x)$ reduces to man(x) and not men(x). It is actually the index value that controls this reduction. In fact, given the index value, the type specification can actually be eliminated, but will be left in for convenience as syntactic sugar and for comparison to predicate logic:

$$\text{index}(x, \text{man}) \leftrightarrow \text{man}(x) \leftrightarrow \text{type}(x, \text{man})$$

Plural nouns like *men* are always indefinite and count. They are indefinite because they can function as object referring expressions without a separate indication of definiteness, and the resulting object referring expression is indefinite:

*men are coming*

They are count, because mass nouns, the other alternative for the count feature, are always singular.

The logical representation for the mass noun *rice* is shown below:

$$\text{rice} . \eta x . \text{rice}(x) \land \text{noun}(x) \land \text{index}(x, \text{rice}) \land \text{word}(x, \text{rice}) \land \text{def}(x, \text{indef}) \land \text{num}(x, \text{sing}) \land \text{pers}(x, 3\text{rd}) \land \text{count}(x, \text{mass})$$

Mass nouns behave grammatically like plural nouns in their ability to function as indefinite object referring expressions without a separate indication of definiteness:

*rice is good for you*

Mass nouns are always singular and indefinite.

**Proper Noun**

Proper nouns name individuals. They typically function as the heads of object referring expressions without separate specification:

*John is here*

The lexical representation of proper nouns aligns with that of common nouns, except that there is no type specification. However, as noted above, the type specification is syntactic sugar that duplicates the information provided by the index. In the case of proper nouns, this syntactic sugar is undesirable.

The logical representation for all proper nouns is shown below:

$$\text{prop-noun: } \eta x . \text{prop-noun}(x) \land \text{index}(x, \text{Index}) \land \text{word}(x, \text{Word}) \land \text{def}(x, \text{Def}) \land \text{num}(x, \text{sing}) \land \text{anim}(x, \text{Anim}) \land \text{gen}(x, \text{Gen}) \land \text{pers}(x, 3\text{rd}) \land \text{count}(x, \text{count})$$

where

- $x$ is an $\eta$ handle variable
- $\text{def, num, anim, gen, pers and count}$ are grammatical features such that

$$\text{Def} \in \{\text{def, indef}\}$$

$$\text{Num} \in \{\text{sing, plur}\}$$
Anim ∈ {inanimate, animate, human}
Gen ∈ {male, female, none}
Pers ∈ {1st, 2nd, 3rd}
Count ∈ {count, mass}

Note that the default values for the grammatical features of proper nouns differ from common nouns. In particular, proper nouns are definite and human by default.

The logical representation for the proper noun John is shown next:

\[ \text{John} : \eta x \cdot \text{prop-noun}(x) \land \text{index}(x, \text{John}) \land \text{word}(x, \text{John}) \land \text{def}(x, \text{def}) \land \text{num}(x, \text{sing}) \land \text{anim}(x, \text{human}) \land \text{gen}(x, \text{male}) \land \text{pers}(x, 3rd) \land \text{count}(x, \text{count}) \]

The proper noun John overrides the default gender feature none, setting it to male. We do not assume a neuter gender feature for English, although the value none can be projected to an encompassing object referring expression. For the purpose of co-reference resolution, none is not compatible with male or female, unlike the variable Gen which can be unified with either male or female.

Grammatically, proper nouns typically project object referring expressions in which they are instantiated as the head. However, it is not always the case that a proper noun has this function. Consider the expression

\[ a \text{ Reagan republican} \]

in which the proper noun Reagan functions as a modifier. Splitting the lexical representation of proper nouns from the constructions in which they are instantiated makes it possible for the lexical entry for proper nouns to be kept simple and unambiguous. The use of proper nouns as modifiers presents a serious challenge for predicate logic and extensions.

In Double-R Grammar, there are composite subcategories of proper noun to represent words like English (proper noun + noun), Arabic (proper noun + adjective) and Sunday (proper noun + nominal). These composite categories are not discussed in this paper.

**Pronoun**

The primary grammatical function of pronouns is to establish co-reference with another referring expression. To achieve this function, they encode a number of grammatical features which exhaust their meaning. However, they are certainly not meaningless.

Pronouns are divided into three parts of speech based on their distinctive grammatical behavior:

- Personal Pronoun (e.g. he, him, you, I, me)
- Reflexive Pronoun (e.g. himself, yourself, myself)
- Possessive Pronoun (e.g. his, your, yours, my, mine)

Personal pronouns can function as either the subject or object of a sentence:
he is going
Johnny likes him
*Johnny likes him
– non-matching index indicates disjoint reference
– matching index indicates co-reference, but the result is ungrammatical

In their object function, they cannot be co-referential with the subject.

Reflexive pronouns typically function as objects, and when they do they must be co-referential with the subject:

Johnny likes himself
*Johnny likes himself

Possessive pronouns typically function as the head of an object referring expression that functions as the specifier in a higher level object referring expression. The higher level object referring expression has either an explicit or implicit head, depending on the possessive pronoun:

her book is funny – explicit head book
*her is funny
yours is funny – implicit head
*yours book is funny

We will discuss pronouns in more detail in the section on co-reference.

**Nominal**

The nominal part of speech in Double-R Grammar is unique. It captures the grammatical behavior of words that function as the heads of object referring expressions, without separate specification, but which are not proper nouns, pronouns, plural nouns or mass nouns. Many such words are composite words that combine a determiner or quantifier and a noun into a single word (e.g. someone, anything, everytime, another, none). Double-R Grammar categorizes numerous words as nominals that are categorized as adverbs or pronouns in other treatments (e.g. today, tomorrow, here, there, then). All the wh-words in English (e.g. who, what, where) are categorized as wh-nominals, a subcategory of nominal, This treatments aligns with their basic grammatical function as variabilized referring expressions.

Nominals are not discussed further in this paper.

**Quantifier**

Quantifiers are words that express a quantity. This is essentially a semantic definition. Quantifiers exhibit a wide range of grammatical functions, making it difficult to categorize them strictly on the basis of grammatical function.

Quantifiers include words like all, some, and many, which have been widely studied in predicate logic and extensions. Since the quantifier part of speech is primarily determined on the basis of meaning, it is often missing in other grammatical treatments. Because of the
wide range of grammatical functions, words that are uniformly categorized as quantifiers in Double-R Grammar, on the basis of meaning, are ambiguously categorized as determiners, adverbs, pronouns and even nouns in other grammatical treatments, based primarily on their syntactic behavior. Double-R Grammar captures the range of behaviors of quantifiers by having them inherit from determiner and nominal, making quantifier a composite part of speech.

Two key grammatical behaviors include the function of quantifiers as specifiers and heads in object referring expressions:

- all men are mortal – all functions as the specifier
- ten are coming – ten functions as the head

As noted above, the word no is categorized as a determiner, despite its meaning. The reason for this is the existence of the word none which is categorized as a nominal. Together, these two words exhibit the behavior of a quantifier:

- no men are coming
- none are coming

Since the quantifier part of speech inherits from both determiner and nominal, the treatment of no as a determiner, and none as a nominal, works together to create a quantifier, which would normally be a single word.

Logical treatments tend to focus on the canonical behavior of quantifiers in their function as specifiers in object referring expressions. A classic example is

all men are mortal → ∀x[man(x) → mortal(x)]

where the quantifier all functions as the specifier in the object referring expression all men. It is frequently argued that quantified expressions are not even referring expressions.

However, quantifiers may occur without a noun head as in

- there are some on the table
- I want some

There is no head noun in these expressions to indicate what the quantifier is quantifying over. The kind of thing being quantified over can only be determined from the context. In Double-R Grammar, we leave it up to the interpretation function to determine what is being referred to and we treat quantified object referring expressions just like other object referring expressions.

Quantifiers can also occur with determiners as in

all the men are seated

where the quantifier all precedes the definite determiner the. This expression picks out a unique set of men (all the men in the situation model) and refers to that unique set. There is a clear difference in meaning between all the men and all men. All the men is relative to
some constrained space and time, whereas \textit{all men} is not. The existential and universal quantifiers operating over a single domain do not seem adequate to represent this difference. Double-R Grammar makes a distinction between the \textit{situation model} and the \textit{mental universe}. \textit{All the men} refers to the set of men in the situation model, which is a representation of the current state-of-affairs, whereas \textit{all men} encompasses the mental universe, which is not time and space specific. Note that \textit{all men} and \textit{the men} are nearly synonymous, perhaps with a difference in emphasis. On the other hand, \textit{all men} and \textit{the men} clearly differ in meaning.

Many logical treatments (cf. Gamut, 1991) acknowledge that the universal and existential quantifiers are insufficient to represent the meaning of the myriad quantifiers that occur in natural language. Generalized quantifiers (cf. Montague, 1973) were introduced to address this issue. In Double-R Grammar, quantifiers are recognized as a semantically motivated part of speech, and quantified object referring expressions are treated much like other object referring expressions. There is no special logical machinery for quantifiers. There are no logical quantifiers in Double-R Grammar (cf. Barwise and Perry, 1981). All quantifiers are non-logical. Of course, this means that issues of quantifier scope are not easily or explicitly addressed in Double-R Grammar.

\textbf{Demonstrative Pronoun}

Grammatically, demonstrative pronouns (e.g. \textit{this, that}) behave very much like quantifiers:

\begin{verbatim}
this book
some book
I want that
I want twenty
\end{verbatim}

However, they differ in meaning, with demonstrative pronouns functioning to point out objects (lexically indicating a distinction in distance), and quantifiers identifying the number of objects, either coarsely (e.g. \textit{some, many}) or finely (e.g. \textit{twenty, third}). The grammatical behavior of demonstrative pronouns is captured by having them inherit from determiner and nominal, just as quantifiers do. Since they do not inherit from pronoun, demonstrative pronouns are not true pronouns. Some grammatical treatments categorize demonstrative pronouns as demonstrative determiners when they function as specifiers.

\textbf{Adjective}

Adjectives are words that typically describe properties and states, where properties and states are types of unary relation.

The logical treatment of adjectives in Double-R Grammar differs significantly from logical treatments which lack any notion of modification. Given an object referring expression like \textit{an old man}

\begin{verbatim}
many logical treatments would represent this as
\exists x [\text{old}(x) \land \text{man}(x)]
\end{verbatim}
The basic idea is that \( \text{old}(x) \) corresponds to the set of individuals in the model that satisfy the predicate \( \text{old} \), \( \text{man}(x) \) corresponds to the set of individuals in the model that satisfy the predicate \( \text{man} \), and \( \text{old}(x) \land \text{man}(x) \) corresponds to the intersection of these two sets. The adjective \( \text{old} \) is called an intersective adjective to reflect this behavior. The existential quantifier indicates that this logical expression is true if there is at least one individual in the intersection of the two sets.

There is a lot that is wrong with this logical representation as a representation for \textit{an old man}. For one, it fails to capture the basic meaning of the determiner \textit{an}. For another, it treats the adjective \( \text{old} \) as a predicate (i.e. a sentence level grammatical function) that takes the existentially quantified variable \( x \) as its argument. It also treats the noun \( \text{man} \) as a predicate that takes the same variable \( x \) as its argument. In this representation \( \text{old} \) and \( \text{man} \) have the same logical status as sentence level predicates. The effect is to collapse any distinction between nouns and adjectives and treat them as conjoined, sentence level predicates. This would be OK if nouns and adjectives were grammatically equivalent and always functioned as sentence level predicates, but they do not.

From the perspective of Double-R Grammar, the noun \( \text{man} \) functions as the head of an object referring expression and denotes a type, \( \text{man} \), that is a subtype of object. The adjective \( \text{old} \) functions as a modifier within the object referring expression and denotes a property, \( \text{old} \). The basic function of a modifier is to specialize the type of the head it modifies, creating a subtype. \textit{Old man} combines a property, \( \text{old} \), and a type, \( \text{man} \), to create a subtype, \textit{old man}. Since \textit{old man} is unlikely to be encoded in the mental lexicon as a multi-word noun, this subtype must be created dynamically from the composition of the individual words. How does this occur in Double-R Grammar? Before answering this, we introduce the logical representation of adjectives.

The logical representation for an adjective (adj) is shown below:

\[
\text{adj: } \forall x. \forall P. P(x) \land \text{adj}(x) \land \text{index}(x, \text{Index}) \land \text{word}(x, \text{Word}) \land \text{adj-form}(x, \text{Comp})
\]

where
\[
\begin{align*}
x \text{ and } P & \text{ are variable handles} \\
x & \text{ is of type adjective (adj)} \\
\text{adjective} & \text{ is a part of speech} \\
P & \text{ is the specific property type denoted by the adjective} \\
\text{adj-form} & \text{ is a grammatical feature such that} \\
\text{Comp} & \in \{\text{base, comp, super}\}
\end{align*}
\]

The logical representation for the morphological variants of the adjective \( \text{old} \) are shown next:

\[
\begin{align*}
\text{old: } & \forall x. \text{old}(x) \land \text{adj}(x) \land \text{index}(x, \text{old}) \land \text{word}(x, \text{old}) \land \text{adj-form}(x, \text{base}) \\
\text{older: } & \forall x. \text{old}(x) \land \text{adj}(x) \land \text{index}(x, \text{old}) \land \text{word}(x, \text{older}) \land \text{adj-form}(x, \text{comp}) \\
\text{oldest: } & \forall x. \text{old}(x) \land \text{adj}(x) \land \text{index}(x, \text{old}) \land \text{word}(x, \text{oldest}) \land \text{adj-form}(x, \text{super})
\end{align*}
\]

where \( \forall P. P(x) \) reduces to \( \text{old}(x) \) by substitution of the index value, \( \text{old} \), for \( P \).
By itself, an adjective denotes a property. How does this property combine with an object type to create an object subtype? Through the intervention of an object head construction. How this works is discussed below when we consider the composition of object referring expressions. For now, we note that the attributive (i.e. modifying) use of adjectives creates challenges for predicate logic based representations of natural language.

Adjectives also function as key elements of predicates as in

\[ \text{the man } \text{is old} \]

where \text{the man} is the subject and \text{is old} is the predicate. In this function, adjectives are commonly called \textit{predicate adjectives}.

Predicate logic is well designed to represent predicate adjectives:

\[ \exists x: \text{man}(x)[\text{old}(x)] \]

In this representation, the noun \textit{man} is treated as a \textit{restriction} to distinguish it from the adjective \textit{old} which functions as the sentential \textit{predicate}. This simplified representation, although quite common, ignores the function and meaning of the auxiliary verb \textit{is}. Double-R Grammar aligns with logical treatments in treating the adjective as the head of a situation referring expression, but also represents the auxiliary verb as a specifier. Most other grammatical treatments would treat the auxiliary verb (e.g. \textit{is}) as the head, and the adjective \textit{old} as a complement, distancing the syntactic representation from the logical representation.

**Adverb**

Adverbs are words describing relations that typically function as modifiers of modifiers in object referring expressions. For example, in

\[ \text{the very old man} \]

the adverb \textit{very} modifies the adjective \textit{old} which modifies the noun \textit{man}.

Adverbs also function as modifiers of predicates and situations in situation referring expressions:

\[ \text{he ran quickly} \quad \text{quickly modifies the predicate ran} \]
\[ \text{fortunately, John won} \quad \text{fortunately modifies the situation referring expression John won} \]

In all these functions, adverbs present a representational challenge for standard predicate logic.

**Preposition**

Prepositions are words that typically establish a locative or temporal relationship between objects. However, they can also function to establish a locative or temporal relationship between a situation and an object, or between two situations. In these functions, they are
typically transitive (i.e. they relate two objects). They are also used to indicate the directionality of an object or action. In this function, they may be intransitive (i.e. they establish the direction of a single object or action). There are also a range of different relationships expressed by prepositions (e.g. recipient, beneficiary, instrument, manner). The wide range of possible relations expressed by prepositions complicates their grammatical and logical treatment.

In this paper, we limit the discussion of prepositions to their function as heads of situation referring expressions as in

_The book is on the table_

where _on_ is treated as the head of a situation referring expression, called a predicate preposition, that describes a locative relationship between _the book_ and _the table_. This discussion shows how Double-R Grammar aligns with logical treatments in making the preposition the head of the situation referring expression, in contrast to most other grammatical treatments which treat the auxiliary verb (e.g. _is_) as the head, and the prepositional phrase _on the table_ as a complement or adjunct.

**Verb**

Verbs are words that typically describe actions, processes or states, overlapping with adjectives to some extent in describing states. Actions, processes and states are types of relation. As relations, verbs combine with one or more arguments to form situation referring expressions.

Predicate logic is often used to represent the predicate-argument structure associated with verbs. For example, the expression

_John kicked a ball_

can be represented as

_∃x:ball(x)[kick(J, x)]_

using the restriction notation for _a ball_.

Double-R Grammar differs from predicate logic in treating the subject as an external argument of the verb, and other arguments (e.g. object, indirect object) as internal to the verb. This adds a level of structure that is typical of grammatical treatments of English (i.e. verb phrase or VP), but not predicate logic representations. It represents the division of a sentence into a subject and a predicate – where predicate includes the verb, any non-subject arguments, and predicate modifiers. It should be noted that logical treatments prior to predicate logic often assumed this sentence level division. It is also the case that early versions of Double-R Grammar (Ball, 1992) followed predicate logic in having flat predicate-argument representations. However, grammatical evidence for the division of a sentence into a subject and a predicate, in English, ultimately held sway, and it was introduced into Double-R Grammar.
It should be noted that verbs also occur in object referring expressions either as modifiers or heads:

- the running bull
- the running of the bull

When they function as modifiers, the arguments they normally relate are suppressed, but may be implied within the object referring expression. Thus, bull is the implicit subject of running, but since bull, by itself, is not an object referring expression, it cannot function as the argument of running. When verbs function as heads of object referring expressions, the action denoted by the verb is reified and treated as an object. Exactly how reified actions, and verbs functioning as modifiers, map into the situation model is an open research question, but the fact that Double-R Grammar allows reference to situations as well as objects should make an adequate treatment possible. Typical predicate logic representations simply ignore the grammatical function of verbs in object referring expressions, treating them as predicates. A predicate logic approximation for both expressions above is shown below:

$$\exists x[run(x) \land bull(x)]$$

This representation fails to represent the fact that bull is the head of the first expression and running is the head of the second expression. The first object referring expression refers to a bull, whereas the second object referring expression refers to a reified running event.

**Auxiliary Verb**

The grammatical behavior of auxiliary verbs is distinct from regular verbs. Whereas regular verbs typically function as the heads of situation referring expressions, describing relations between arguments, auxiliary verbs function as the specifier and do not relate any arguments. The specifier function of auxiliary verbs in situation referring expressions is similar to the specifier function of determiners and quantifiers in object referring expressions. That is, the auxiliary verb functions to situate the situation referring expression relative to other situations in the situation model. This situating function is made possible by the grammatical features associated with the auxiliary verb – especially, tense (past, present) and aspect (progressive, perfect). These situating features are the situation referring expression equivalent of the definiteness feature of object referring expressions.

Modal auxiliaries, a subtype of auxiliary, have the additional function of indicating the modality of situation referring expressions (e.g. possibility, necessity). In Double-R Grammar, modality is represented as a grammatical feature of situation referring expressions, not as a logical operator.

Since the auxiliary verb functions as the specifier in Double-R Grammar, it is not part of the predicate that functions as the head of a situation referring expression. This means that the head is a tenseless predicate. Consider the situation referring expression
John is kicking the ball

in which the auxiliary verb is functions as the specifier and kicking the ball functions as an untensed predicate that is the head of the situation referring expression. Note that John is the subject and external argument of kicking. John is not an argument of is, since auxiliary verbs functioning as specifier do not establish a relation between any arguments. This treatment of auxiliary verbs in Double-R Grammar is motivated, in part, by subject-auxiliary inversion. Consider the yes-no-question situation referring expression

is John kicking the ball?

in which the auxiliary verb is is inverted with the subject John and is separated from the untensed predicate kicking the ball. In a grammatical treatment in which the predicate is the tensed is kicking the ball, we have a discontinuous constituent. In Double-R Grammar, this is not a discontinuous constituent, since the auxiliary verb has a distinct grammatical function. On the other hand, subject-regular verb inversion would result in a discontinuous constituent, but is very awkward in English:

?kicks John the ball?

Instead, the auxiliary verb do is normally inserted to avoid a discontinuous constituent:

did John kick the ball?

The auxiliary verbs do and have can also function as the heads of situation referring expressions. Consider

John did the job
John has the ball

In these sentences, they are functioning as regular transitive verbs. Approximate logical representations are shown below:

∃o[do(J, o) ∧ job(o)]
∃o[have(J, o) ∧ ball(o)]

It is less clear that the auxiliary verb be has this function. In

John is a man

Double-R Grammar treats a man as the predicate head of the situation referring expression in what is called a predicate nominal construction. An approximate logical representation would be

∃o[man(o) ∧ ] = o]

A need to treat be as a full-fledged transitive verb has not been identified.
Part of Speech Summary

Double-R Grammar treats parts of speech as abstract semantic categories as in Cognitive Grammar (Langacker, 1987, 1991) and traditional grammar. As such, parts of speech are included as constituents of logico-grammatical representations. In modern generative grammar, parts of speech are assumed to be purely syntactic in accord with the autonomy of syntax hypothesis and the purported failure of semantic definitions. We reject that position. Determination of the part of speech of a word based purely on distributional (i.e. syntactic) criteria has two major flaws: 1) it creates an unnecessary divide between structure and meaning, and 2) it results in rampant ambiguity. Since most words exhibit multiple grammatical functions, purely distributional criteria lead to their categorization into multiple parts of speech. Consider

*John is running*
*The running bull*
*The running of the bull*
*The bull rider*

Based on purely distributional criteria, *running* can be a verb, adjective or noun, and *bull* can be a noun or adjective. This necessitates either three entries in the mental lexicon for *running* and two for *bull*, or some other method like type shifting to accommodate the different possibilities. The multiple lexical entry approach destroys any semantic basis for part of speech categories, which helps explain the position adopted in modern generative grammar. A type shifting approach might suggest that a word has a basic part of speech based on meaning (e.g. verb for *running*, noun for *bull*), with alternative parts of speech derived from the basic type, based on context. Double-R Grammar goes one step further by providing a constructional context with grammatical functions that words can fill without a shift in part of speech type. *Running*, the verb, can function as the predicate head of a situation referring expression, and as a modifier or head in an object referring expression. *Bull*, the noun, can function as the head or modifier in an object referring expression. It is the introduction of grammatical functions and constructions which exhibit these grammatical functions that makes this possible. As a consequence, phrasal categories like noun phrase (NP) and verb phrase (VP), which need not be headed by a noun or verb, become problematic and are avoided in Double-R Grammar.

Besides the introduction of constructions and grammatical functions to support alternative behaviors of words, parts of speech are organized into a semi-lattice. Higher levels in the semi-lattice capture generalities across lower levels. For example, proper nouns, nouns and pronouns are all subtypes of proto-noun. The common behavior of these parts of speech to function as the heads of object referring expressions is captured by their categorization as proto-nouns. There is no need to treat proper nouns and pronouns as though they are (common) nouns when they head object referring expressions, in order for the phrasal category noun phrase to remain coherent. In addition, Double-R Grammar provides composite categories to capture the behavior of words like quantifiers which exhibit a wide range of grammatical functions. Quantifiers inherit from both determiner and nominal. This allows them to function as either the specifier or head of an object referring
expression, without requiring multiple entries in the mental lexicon. Composite categories are useful when the range of functions of a word do not indicate a distinction between basic and derived functions. By contrast, the use of the verb *running* as a head in an object referring expression is assumed to be a derived function. Hence the verb *running* does not inherit from proto-noun and verbs are not proto-nouns in Double-R Grammar. In traditional grammar, the part of speech category *gerund* is sometimes used to categorize verbs when they function as heads of noun phrases and, especially, when they function as heads of clauses that function as the subject of sentences. For example, in

*giving a speech* is scary

the verb *giving* may be categorized as a gerund, where use of the term gerund indicates its dual status as a verb and a noun, or a derived noun. What is interesting about this example is that *giving* combines with the argument *a speech* just like a verb, and unlike a noun, arguing against its treatment as a derived noun. In Double-R Grammar *giving* is treated as a verb which functions as the (predicate) head of the situation referring expression *giving a speech*. In this situation referring expression, there is an implicit subject. Double-R Grammar allows embedded situation referring expressions to function as subjects of matrix situation referring expressions, as in this example. If Double-R Grammar had a gerund part of speech category, it would be a composite category that inherited from both noun and verb. The problem with this composite category is that it is not semantically coherent. What does it mean for a word to be at once a verb and a noun? It would be more coherent to have a verb and adjective composite part of speech category, since words which describe a state may either be categorized as a verb (e.g. *know*) or adjective (e.g. *sad*) in English. A composite verb-adjective category would support the use of *running* as a modifier above, but not its use as the head of an object referring expression. It would also facilitate an underspecified representation of words like *entertaining* in

*my parents are entertaining*

where *entertaining* may either be a verb as in

*my parents are entertaining our neighbors*

or an adjective as in

*my parents are very entertaining*

where *very* typically modifies adjectives and not verbs.

To capture the full range of uses of present participle verbs like *running*, we would need a composite verb-adjective-noun category. But this category is even less semantically coherent than a composite verb-noun category. In sum, we treat *running* as a verb and allow it to be coerced into the functions of modifier and head of object referring expressions in the appropriate constructional contexts. We leave the question of whether there are distinct entries for *entertaining* as an adjective and verb, or a composite verb-adjective part of speech, open.
With respect to quantifiers, the use of a quantifier as the head of an object referring expression is assumed to be a non-derived function that is supported by its composite categorization as a nominal. Quantifiers routinely function as specifiers and heads of object referring expressions. One use does not appear to be derived from the other.

A good example of the need for composite parts of speech is provided by words like *English*. This word is a proper noun, but it also functions like an adjective in contrast to most proper nouns:

- *English* is a difficult language
- The *English* language
- *The John* book

Grammatical treatments based on distribution would categorize this word as a proper noun in the first expression and as an adjective in the second expression. Neither use appears to be derived. In Double-R Grammar, *English* is categorized as a composite proper-noun adjective to capture the range of uses and its composite meaning, and to distinguish it from proper nouns that don’t normally function as modifiers. It is interesting to contemplate whether there still needs to be two entries for *English* in the mental lexicon: one corresponding to the language sense of *English*, and one corresponding to the people sense (e.g. the *English* people). If we allow for fine-grained semantic types that are composite, then the composite semantic type language-or-people would allow for a single entry. Such fine-grained, composite semantic types are called dot types (e.g. language•people) by Pustejovsky (1995) in his Generative Lexicon theory. Pustejovsky motivates dot types as a way of avoiding the need for multiple word senses, what he calls word sense enumeration, to model the polysemy of words, and as a mechanism for dynamically determining meaning in context. There is a close semantic connection between groups of people and the language they speak that makes this composite semantic type coherent. In the second example above, it is unclear if *English* denotes the language or the people, although the intent was to denote the language.

It is a basic reality that words exhibit multiple grammatical functions. There are at least 4 ways in which this reality can be managed:

1. Separate entries in the mental lexicon for the part of speech that is prototypical of each grammatical use
2. Composite parts of speech that support the range of grammatical uses in a single lexical entry
3. Type shifting from one part of speech to another
4. Constructional coercion

The creation of separate part of speech entries in the mental lexicon for each grammatical use wreaks havoc on any semantic basis for part of speech determination, and engenders extensive ambiguity which often cannot be resolved at the time the word is incrementally processed. Composite parts of speech reduce the number of lexical entries needed, and make sense so long as the composite categories are semantically coherent. Type shifting appears to be needed to handle nonce uses of word. For example, in
The paper boy **porched** the newspaper (Clark & Clark, 1979)

the noun *porch* is dynamically transformed into a verb meaning something like *threw the newspaper on the porch*. The result is a new word with a part of speech and meaning that is implied by the context of use. Constructional coercion allows words to exhibit non-prototypical uses within the context of constructions that coerce them into the non-prototypical uses, without creating new words in the process.

It is likely that all of these options are manifest to some extent. For example, if a verb is used often enough as the head of an object referring expression, it is likely to become encoded in the mental lexicon as a noun. However, there is likely to be a shift in the meaning of the verb associated with this encoding. For example, the noun use might come to be associated with the object of the verb as in words associated with bodily functions (it is difficult to think of a non-taboo example, but *spit* may not be too offensive). This divergence in meaning is made possible by the existence of separate entries. The introduction of composite parts of speech is an unusual feature of Double-R Grammar. Many grammatical treatments assume a flat listing of parts of speech (on the order of 9 parts of speech in traditional grammar), but once a hierarchy is introduced, the extension to a semi-lattice (or tangled hierarchy) and the introduction of composite parts of speech is a natural one. Composite parts of speech are particularly useful for supporting different grammatical uses of words, where no one grammatical use appears to be basic or derived. Dramatic uses of words in uncommon contexts suggest a need for dynamic creation of new parts of speech via a mechanism like type shifting. Such coinings create new words with new meanings that are derived from the old words. Finally, constructional coercion supports less dramatic uses of words in new contexts without creating new words in the process.

For any particular word, which options are most appropriate is an essentially empirical question, and the answer may well vary from individual to individual. Further, a more detailed empirical analysis may reveal the benefits of introducing additional parts of speech, or consolidating some of the existing parts of speech. The current ontology has proved useful for a relatively broad grammatical analysis of the major grammatical constructions of English. Analysis of minor and specialized grammatical constructions is likely to reveal the need for additional categories and distinctions.

**The Logical Representation and Processing of Object Referring Expressions in Double-R Grammar**

Constructions are the components of expressions that provide the scaffolding into which parts of speech and subexpressions are instantiated. Logical approaches lacking constructions, or something equivalent, must encode this scaffolding into the lexical items themselves (e.g. Montague Grammar). We have already seen the example of determiners like *a* which are represented as binary relations between properties in Montague Grammar:

\[
    a: \lambda Q[\lambda P[\exists x(Q(x) \land P(x))]]
\]
This representation captures the structure of an entire sentence. It has the advantage of supporting the representation of the wide scope of the determiner *a*. It has the disadvantage of suggesting that the determiner *a* is logically very complex. It also runs into problems when object referring expressions are used to refer without being part of a sentence. This use of object referring expressions is actually quite common. Another disadvantage is the inflexibility this representation introduces. For words that exhibit multiple grammatical functions, multiple lexical entries (or some kind of type shifting mechanism) are likely to be needed. This introduces ambiguity that can only be resolved by the context of use of the word. Often the decisive element of the context is not available at the time the word is processed, given the incremental processing assumption that underlies Double-R Grammar.

**Object Referring Expression Construction**

Object referring expression constructions, which are a type of referring expression construction, and, more generally, a type of construction, consist of an ordered sequence of grammatical functions followed by an unordered sequence of grammatical features.

\[
\text{ORE} \rightarrow \langle \text{Mod}_{ps}, \text{Spec}, \text{Mod}, \text{Head}, \text{Mod}_{ph} \rangle \land \text{GFeat}_1 \land \ldots \land \text{GFeat}_m
\]

Grammatical functions, which are semantically motivated, are instantiated by a single part of speech or expression (i.e. they are unary branching). Expressions are instantiated constructions.

\[
\begin{align*}
\text{Spec} & \rightarrow \text{POS} | \text{Expr} \quad (\text{POS} = \text{part of speech}; \text{Expr} = \text{Expression}) \\
\text{Head} & \rightarrow \text{POS} | \text{Expr} \\
\text{Mod} & \rightarrow \text{POS} | \text{Expr} \quad (\text{Mod}_{ps} = \text{pres-spec-mod}; \text{Mod}_{ph} = \text{post-head-mod}) \\
\text{Expr} & \rightarrow \text{ORE} | \text{SRE} | \text{ObjHead} | \text{ObjMod} | \text{Rel-SRE} \\
\text{POS} & \rightarrow \text{det} | \text{noun} | \text{pronoun} | \text{prop-noun} | \text{adjective} | \text{preposition} | \text{quantifier} | \ldots
\end{align*}
\]

Grammatical features, which are also semantically motivated, are type-value pairs.

\[
\begin{align*}
\text{GFeat}_{\text{def}} & \rightarrow \langle \text{def}, \text{def} \rangle | \langle \text{def}, \text{indef} \rangle | \langle \text{def}, \text{univ} \rangle \\
\text{GFeat}_{\text{num}} & \rightarrow \langle \text{num}, \text{sing} \rangle | \langle \text{num}, \text{plur} \rangle
\end{align*}
\]

They are logically represented as predicates corresponding to the type with an argument for the expression and an argument for the value:

\[
\begin{align*}
\text{def}(o, \text{def}) & - \text{where } o \text{ is a } \lambda \text{ variable for an object referring expression} \\
\text{num}(o, \text{sing}) & - \text{where } o \text{ is a } \lambda \text{ variable for an object referring expression}
\end{align*}
\]

Parts of speech, which are a subtype of construction, are instantiated by words. The part of speech is the type of the word that instantiates it. Parts of speech encode indexes that generalize over the morphological variants of a word – commonly called a lexeme, but also ambiguously called a word, and the specific morphological form of the word. Indexes are not abstract “concepts” (scare quotes intended) or logical constants as is typically assumed in other logical approaches. We see no advantage to calling uppercase words like MAN or words with a single quote (or prime) like man’, “concepts” (or logical constants) – at least,
not without a non-linguistic representation of the meaning of concepts, and a mechanism to map from words to concepts. Otherwise, uppercase words just masquerade as concepts. Non-linguistic concepts are parts of situation model representations, not parts of logico-grammatical representations.

As syntactic sugar and for comparison to predicate logic, we represent the index in predicate logic form as a predicate, for parts of speech that are typically represented as predicates:

\[ \text{index}(x, \text{man}) \leftrightarrow \text{man}(x) \leftrightarrow \text{type}(x, \text{man}) \]

Constructions encode an ordered sequence of grammatical functions followed by an unordered sequence of grammatical features. The values of the grammatical functions and features are uninstantiated. The subexpressions and parts of speech that are instantiated into the construction to fill grammatical functions result in an instantiated construction. The values for the grammatical features are projected from the parts of speech that fill grammatical functions in the instantiated construction. These grammatical features must be reconciled at the level of the construction.

The fully articulated logical representation for an uninstantiated object referring expression construction is shown below:

\[ \lambda o. \eta ps \eta s \eta m \eta h \eta pm . \text{ore}(o) \land \text{pre-spec-mod}(o, \text{ps}) \land \text{spec}(o, \text{s}) \land \text{mod}(o, \text{m}) \land \text{head}(o, \text{h}) \land \text{post-head-mod}(o, \text{pm}) \land \text{def}(o, \text{Def}) \land \text{num}(o, \text{Num}) \land \text{anim}(o, \text{Anim}) \land \text{gen}(o, \text{Gen}) \land \text{pers}(o, \text{Pers}) \land \text{case}(o, \text{Case}) \]

where

- \( \lambda \) indicates a handle for a referring expression
- \( \eta \) indicates a handle for an element that is not a referring expression
- \( o, ps, s, m, h \) and \( pm \) are typed variables
- \( o: \text{type} = \text{object referring expression} \)
- \( ps: \text{type} = \text{modifier grammatical function} (\text{GFunc}) \)
- \( s: \text{type} = \text{specifier GFunc} \)
- \( m: \text{type} = \text{modifier GFunc} \)
- \( h: \text{type} = \text{head GFunc} \)
- \( pm: \text{type} = \text{modifier GFunc} \)
- \( \text{ore} = \text{object referring expression type} (\text{construction}) \)
- \( \text{pre-spec-mod} (\text{pred-spec modifier}) = \text{GFunc} \)
- \( \text{spec} (\text{specifier}) = \text{GFunc} \)
- \( \text{mod} (\text{modifier}) = \text{GFunc} \)
- \( \text{head} = \text{GFunc} \)
- \( \text{post-head-mod} (\text{post-head modifier}) = \text{GFunc} \)
- \( \text{def} (\text{definite}) = \text{grammatical feature} (\text{GFeat}) \)
- \( \text{Def} \in \{\text{def}, \text{indef}, \text{univ}\} \)
- \( \text{num} (\text{number}) = \text{GFeat} \)
- \( \text{Num} \in \{\text{sing}, \text{plur}\} \)
- \( \text{anim} (\text{animacy}) = \text{GFeat} \)
Anim ∈ \{anim, inanim, human\}
gen (gender) = GFeat
  Gen ∈ \{male, female, none\}
pers (person) = GFeat
  Pers ∈ \{1st, 2nd, 3rd\}
case = GFeat
  Case ∈ \{subj, obj, none\}

The underlined element is the default value for each grammatical feature. In this logical representation, the fixed order of grammatical functions is implicit.

This representation is essentially a neo-Davidsonian representation in which \(\lambda\) abstraction provides a special handle for referring expressions (not just events), the \(\eta\) operator provides a handle for building more complex representations that are not referring expressions, grammatical functions replace semantic roles, and grammatical features are added to represent grammatically relevant semantic information.

**Object Head Construction**

An object head is a construction that has the following logical representation:

\[
\text{object-head: } \eta \text{oh } \eta \text{m } \eta \text{h } \eta \text{pm } \text{. obj-head(oh) } \land \text{mod(oh, m) } \land \text{head(oh, h) } \land \text{post-head-mod(oh, pm) } \land \text{def(o, Def) } \land \text{num(o, Num) } \land \text{anim(o, Anim) } \land \text{gen(o, Gen) } \\
\land \text{pers(o, Pers)}
\]

where
- oh is an object head handle
- m and pm are handles for modifiers
- h is a handle for the head
- obj-head is the object head construction type
- mod, head and post-head-mod are grammatical functions (GFunc)
- def (definite) = grammatical feature (GFeat)
  - Def ∈ \{def, indef, univ\}
- num (number) = GFeat
  - Num ∈ \{sing, plur\}
- anim (animacy) = GFeat
  - Anim ∈ \{anim, inanim, human\}
- gen (gender) = GFeat
  - Gen ∈ \{male, female, none\}
- pers (person) = GFeat
  - Pers ∈ \{1st, 2nd, 3rd\}

The object head construction is needed for the representation of object referring expressions in which a modifier of the head occurs. Two examples include

*an old man*

*the book of rules*
where *old* modifies *man* and *of rules* modifies *book*. Double-R Grammar also supports modification at the object referring expression level, in which case an object-head construction may not be needed. In Double-R Grammar, the preference is to treat prepositional phrases, other than those headed by *of*, and relative clauses as object referring expression modifiers, and not object head modifiers:

```
the book; that, I gave you to
the book on the table
```

In the case of relative clauses, this preference is motivated by the co-referential representation of the demonstrative pronoun *that*, and the implicit object, *t*, of *gave*, with the object referring expression *the book*. In the case of prepositional phrases, it results in a grammatical distinction between prepositional phrases headed by the preposition *of*, and prepositional phrases headed by other prepositions, while retaining the functional categorization of all prepositional phrases as modifiers. Other grammatical approaches often treat prepositional phrases headed by *of* as complements of the head noun. In Double-R Grammar, nouns are not relations and do not take complements. This is reflected in the grammatical fact that prepositional phrases headed by *of* like other prepositional phrases are optional:

```
the book
the book of rules
the book on the table
```

The optional nature indicates that they are adjuncts, or modifiers, and not complements. We will not belabor the point here, since this is not relevant to the focus of this paper.

**Composing Object Referring Expressions from Parts of Speech and Constructions in Double-R Grammar**

Double-R Grammar makes a commitment to the incremental, left-to-right composition of object referring expressions from the words in a written linguistic expression. As each word or multi-word unit is processed, its part of speech is determined, contextually appropriate constructions are projected, and the part of speech is instantiated into one of the projected constructions, according to the grammar. Lower level constructions may in turn be instantiated into higher level constructions, according to the grammar. The incremental processing commitment has important ramifications for the logical representations. There is no commitment to having a fully connected representation at each stage in processing, although only a limited number of distinct representational pieces can be maintained in working memory at any time. Although Double-R Grammar builds the single best contextually determined representation at each step in analysis, that representation may turn out to be incorrect in the wider context. Double-R Grammar includes a non-monotonic context accommodation mechanism that is capable of making modest adjustments to the evolving representation. The situation model that provides the interpretation of the logico-grammatical representation of linguistic expressions is dynamically updated as logico-grammatical representations are created. Overall, the behavior of Double-R Grammar is pseudo-deterministic – it builds the single best representation at each step in processing, but may subsequently adjust the evolving
representation, when the wider context indicates the need for adjustment. In general, adjustments are modest and easily accommodated.

**Combining Determiners and Nouns with Object Referring Expression Constructions**

We first consider the logical composition of a simple object referring expression which combines a determiner and a noun.

The object referring expression

\[
\text{the man}
\]

is composed together as follows:

\[
\text{ore: } \lambda o \ \eta s \ \eta h . \ \text{ore}(o) \land \text{spec}(o, s) \land \text{head}(o, h) \land \text{def}(o, \text{Def}) \land \text{num}(o, \text{Num}) \land \\
\text{anim}(o, \text{Anim}) \land \text{gen}(o, \text{Gen}) \land \text{pers}(o, \text{Pers}) \land \text{case}(o, \text{Case}) \\
\land \\
\text{the: } \eta x . \ \text{det}(x) \land \text{index}(x, \text{the}) \land \text{word}(x, \text{the}) \land \text{def}(x, \text{Def}) \land \text{num}(x, \text{Num}) \\
\land \\
\text{man: } \eta x . \ \text{man}(x) \land \text{noun}(x) \land \text{index}(x, \text{man}) \land \text{word}(x, \text{man}) \land \text{def}(x, \text{Def}) \land \text{num}(x, \text{sing}) \land \\
\text{anim}(x, \text{human}) \land \text{gen}(x, \text{male}) \land \text{pers}(x, \text{3rd}) \land \text{count}(x, \text{count}) \\
= \\
\text{the man: } \lambda o \ \eta s \ \eta h . \ \text{ore}(o) \land \text{spec}(o, s) \land \text{def}(s) \land \text{index}(s, \text{the}) \land \text{word}(s, \text{the}) \land \\
\text{head}(o, h) \land \text{man}(h) \land \text{noun}(h) \land \text{index}(h, \text{man}) \land \text{word}(h, \text{man}) \land \text{def}(o, \text{def}) \land \\
\text{num}(o, \text{sing}) \land \text{anim}(o, \text{human}) \land \text{gen}(o, \text{male}) \land \text{pers}(o, \text{3rd})
\]

How does this composition occur? The primary mechanism of composition is unification according to the grammar. By unification according to the grammar, we mean that grammatical functions can be filled by parts of speech (and expressions) that are compatible with the grammar. In particular, the determiner *the* is compatible with the spec (specifier) function of an object referring expression, and the noun *man* is compatible with the head function of an object referring expression. The object referring expression construction is projected by the determiner *the* according to the grammar. In addition to parts of speech projecting constructions and filling compatible grammatical functions, they project grammatical features to the encompassing construction. These grammatical features get reconciled at the level of the construction. In the simplest case, as in this example, this occurs by monotonic unification of features.

In the case of plural nouns and mass nouns which are not preceded by a determiner, it is the noun that projects an object referring expression construction. The representation of the expression

\[
\text{men}
\]
results from the composition of an object referring expression construction projected by *men* with the lexical entry for the noun *men*:

\[
\text{o}r\text{e}: \lambda o \, \eta s \, \eta h \cdot \text{ore}(o) \land \text{spec}(o, s) \land \text{head}(o, h) \land \text{def}(o, \text{Def}) \land \text{num}(o, \text{Num}) \land \\
\text{anim}(o, \text{Anim}) \land \text{gen}(o, \text{Gen}) \land \text{pers}(o, \text{Pers}) \land \text{case}(o, \text{Case})
\]

\[
+ 
\]

\[
\text{men}: \lambda x \cdot \text{man}(x) \land \text{noun}(x) \land \text{index}(x, \text{man}) \land \text{word}(x, \text{men}) \land \text{def}(x, \text{indef}) \land \\
\text{num}(x, \text{plur}) \land \text{anim}(x, \text{human}) \land \text{gen}(x, \text{male}) \land \text{pers}(x, 3rd) \land \text{count}(x, \text{count})
\]

\[
= 
\]

\[
\text{men}: \lambda o \, \eta s \, \eta h \cdot \text{ore}(o) \land \text{spec}(o, s) \land \text{head}(o, h) \land \text{man}(h) \land \\
\text{word}(h, \text{men}) \land \text{def}(o, \text{indef}) \land \text{num}(o, \text{plur}) \land \text{anim}(o, \text{human}) \land \text{gen}(o, \text{male}) \land \\
\text{pers}(o, 3rd)
\]

Note that there is no determiner in this representation and the specifier function is uninstantiated since there is no part of speech or expression associated with the specifier handle \(\eta s\).

The interpretation of bare plural object referring expressions differs from that of singular object referring expressions. In Double-R Grammar, bare plurals are interpreted as referring to a collection in the situation model, where collections are accorded the same first class status as individuals. Carlson (1977) and Schubert & Pelletier (1987) argue that bare plurals can denote kinds as well as individuals. Kinds are also accorded first class status in Double-R Grammar. Kinds have two different viewpoints, which we call types and classes. Types and classes are essentially two sides of the same coin. Types are atomic with no subparts. Classes are collections that denote all the individuals of a type – close to the predicate logic treatment of predicates. A representative subset of the members of a class is cognitively salient. Object referring expressions that refer to types are singular (e.g. *a dog is a type of animal*), reflecting the atomic viewpoint. Object referring expressions that refer to classes are plural (e.g. *all men, dogs are mammals*) (Ball, 2012), reflecting the saliency of representative class members.

**Combining Determiners, Adjectives and Nouns with the Object Head and Object Referring Expression Constructions**

We are now in a position to consider the logical composition of *old* with *man*. The adjective *old* projects an object head construction in which it functions as the modifier (mod). The noun *man* is subsequently instantiated as the head of this object head construction.

\[
\text{object-head}: \eta m \, \eta h \, \eta o h \cdot \text{obj-head}(oh) \land \text{mod}(oh, m) \land \text{head}(oh, h) \land \text{def}(o, \text{Def}) \land \\
\text{num}(o, \text{Num}) \land \text{anim}(o, \text{Anim}) \land \text{gen}(o, \text{Gen}) \land \text{pers}(o, \text{Pers})
\]

\[
+ 
\]

\[
\text{old}: \eta x \cdot \text{old}(x) \land \text{adj}(x) \land \text{index}(x, \text{old}) \land \text{word}(x, \text{old})
\]

\[
+
\]
\( \text{man} \): \( \eta x \cdot \text{man}(x) \land \text{noun}(x) \land \text{index}(x, \text{man}) \land \text{word}(x, \text{man}) \land \text{def}(x, \text{Def}) \land \text{num}(x, \text{sing}) \land \text{anim}(x, \text{human}) \land \text{gen}(x, \text{male}) \land \text{pers}(x, 3rd) \land \text{count}(x, \text{count}) \)

= 

\( \text{old man} \): \( \eta m \eta h \eta oh \cdot \text{obj-head}(oh) \land \text{mod}(oh, m) \land \text{old}(m) \land \text{adj}(m) \land \text{index}(m, \text{old}) \land \text{word}(m, \text{old}) \land \text{head}(oh, h) \land \text{man}(h) \land \text{noun}(h) \land \text{index}(h, \text{man}) \land \text{word}(h, \text{man}) \land \text{def}(oh, \text{Def}) \land \text{num}(oh, \text{sing}) \land \text{anim}(oh, \text{human}) \land \text{gen}(oh, \text{male}) \land \text{pers}(oh, 3rd) \)

For this composition to succeed, the variable \( x \) of \( \text{old} \) is unified with the variable \( m \) of the object-head and the variable \( x \) of \( \text{man} \) is unified with the variable \( h \) of the object-head.

Note that the definiteness feature remains the variable \( \text{Def} \). This is because such expressions do not normally function as object referring expressions without separate specification of definiteness:

* \( \text{old man} \) is here*

The object head \( \text{old man} \) can combine with the determiner \( \text{the} \) as follows:

\( \text{ore}: \lambda o \cdot \eta s \eta h \cdot \text{ore}(o) \land \text{spec}(o, s) \land \text{head}(o, h) \land \text{def}(o, \text{Def}) \land \text{num}(o, \text{Num}) \land \text{anim}(o, \text{Anim}) \land \text{gen}(o, \text{Gen}) \land \text{pers}(o, \text{Pers}) \land \text{case}(o, \text{Case}) \)

+ 

\( \text{the}: \eta x \cdot \text{det}(x) \land \text{index}(x, \text{the}) \land \text{word}(x, \text{the}) \land \text{def}(x, \text{def}) \land \text{num}(x, \text{Num}) \)

+ 

\( \text{old man}: \eta m \eta h \eta oh \cdot \text{obj-head}(oh) \land \text{mod}(oh, m) \land \text{old}(m) \land \text{adj}(m) \land \text{index}(m, \text{old}) \land \text{word}(m, \text{old}) \land \text{head}(oh, h) \land \text{man}(h) \land \text{noun}(h) \land \text{index}(h, \text{man}) \land \text{word}(h, \text{man}) \land \text{def}(oh, \text{def}) \land \text{num}(oh, \text{sing}) \land \text{anim}(oh, \text{human}) \land \text{gen}(oh, \text{male}) \land \text{pers}(oh, 3rd) \)

= 

\( \text{the old man}: \eta s \eta m \eta h \eta oh \cdot \lambda o \cdot \text{ore}(o) \land \text{spec}(o, s) \land \text{det}(s) \land \text{index}(s, \text{the}) \land \text{word}(s, \text{the}) \land \text{obj-head}(oh) \land \text{head}(o, oh) \land \text{mod}(oh, m) \land \text{old}(m) \land \text{adj}(m) \land \text{index}(m, \text{old}) \land \text{word}(m, \text{old}) \land \text{head}(oh, h) \land \text{man}(h) \land \text{noun}(h) \land \text{index}(h, \text{man}) \land \text{word}(h, \text{man}) \land \text{def}(o, \text{def}) \land \text{num}(o, \text{sing}) \land \text{anim}(o, \text{human}) \land \text{gen}(o, \text{male}) \land \text{pers}(o, 3rd) \)

In this representation, grammatical features are only represented at the level of the object referring expression, although all but the count, index and word features of \( \text{man} \) are projected to the obj-head as well.

A primary reason for treating adjectives as modifiers is because the alternative of treating them as being conjoined with the head noun in a flat logical representation is not grammatically coherent. Although intersective adjectives functioning as modifiers can be modeled in this manner:
most modifiers cannot. Consider the case of noun modifiers:

\[ \exists x [ \text{old}(x) \land \text{man}(x) ] \]

where a representation like

\[ \# \exists x [ \text{altitude}(x) \land \text{restriction}(x) ] \]

is semantically incorrect – an \textit{altitude restriction} is not an \textit{altitude}, it is a \textit{restriction} (the \# indicates semantic anomaly). There is no clear sense in which the set of individuals in the model that satisfy the predicate \textit{altitude} is relevant to the meaning of this expression. Even in the case of intersective adjectives, the approach breaks down in more complex expressions. Consider

\[ \exists x [ \text{old}(x) \land \text{time}(x) \land \text{party}(x) ] \]

is not semantically correct. There is no sense in which the individuals in the set \textit{party} are also in the set of individuals that are \textit{old}, since \textit{old} modifies \textit{time} and not \textit{party}.

It is also the case that the meaning of \textit{old}, which is a degree adjective, is relative to the thing being modified. An \textit{old} baby is still a \textit{young} person. A flat conjunction of predicates does not capture this relativity.

This example demonstrates that modifiers not only modify head nouns, they can modify other modifiers. In fact, this is the basic function of adverbs in object referring expressions. Consider the expression

\[ \text{the very old man} \]

in which the adverb \textit{very} modifies \textit{old}. There is no coherent logical representation which represents this as a flat conjunction of predicates:

\[ \# \exists x [ \text{very}(x) \land \text{old}(x) \land \text{man}(x) ] \]

Finally, consider the expression

\[ \text{a Reagan republican} \]

where representations like

\[ \# \exists x [ \text{Reagan}(x) \land \text{republican}(x) ] \]
\[ \# \exists x [ \text{republican}(x) \land x = R ] \]

are clearly wrong.
The use of a flat conjunction of predicates for representing the meaning of object referring expressions turns out to be seriously flawed when we consider more complex object referring expressions. An adequate logical representation needs to somehow take into account the grammatical structure and function of the words in the linguistic input. Double-R Grammar uses handle variables to do this.

The logical composition of the expression

\textit{altitude restriction}

which contains a noun modifier is shown below:

object-head: \( \eta m \ \eta h \ \eta oh \ . \ obj\text{-head}(oh) \land mod(oh, m) \land head(oh, h) \land \text{def}(o, \text{Def}) \land \text{num}(o, \text{Num}) \land \text{anim}(o, \text{Anim}) \land \text{gen}(o, \text{Gen}) \land \text{pers}(o, \text{Pers}) \)

+ 

altitude: \( \eta x . \ \text{altitude}(x) \land \text{noun}(x) \land \text{index}(x, \text{altitude}) \land \text{word}(x, \text{altitude}) \land \text{num}(x, \text{sing}) \land \text{anim}(x, \text{inanim}) \land \text{gen}(x, \text{none}) \land \text{pers}(x, 3\text{rd}) \land \text{count}(x, \text{count}) \)

+ 

\textit{restriction}: \( \eta x . \ \text{restriction}(x) \land \text{noun}(x) \land \text{index}(x, \text{restriction}) \land \text{word}(x, \text{restriction}) \land \text{num}(x, \text{sing}) \land \text{anim}(x, \text{inanim}) \land \text{gen}(x, \text{none}) \land \text{pers}(x, 3\text{rd}) \land \text{count}(x, \text{count}) \)

= 

\textit{altitude restriction}: \( \eta m \ \eta h \ \eta oh \ . \ obj\text{-head}(oh) \land mod(oh, m) \land \text{altitude}(m) \land \text{noun}(m) \land \text{index}(m, \text{altitude}) \land \text{word}(m, \text{altitude}) \land \text{head}(oh, h) \land \text{restriction}(h) \land \text{noun}(h) \land \text{index}(h, \text{restriction}) \land \text{word}(h, \text{restriction}) \land \text{def}(oh, \text{def}) \land \text{num}(oh, \text{sing}) \land \text{anim}(oh, \text{inanim}) \land \text{gen}(oh, \text{none}) \land \text{pers}(oh, 3\text{rd}) \)

The handle variables allow us to represent the fact that \textit{altitude} is functioning as a modifier in the object head construction that is headed by \textit{restriction}.

We next show the composition of the object referring expression construction and the determiner \textit{the} with the object head \textit{altitude restriction}:

\textit{ore}: \( \lambda o . \ \eta s \ \eta q . \ \text{ore}(o) \land \text{spec}(o, s) \land \text{head}(o, h) \land \text{def}(o, \text{Def}) \land \text{num}(o, \text{Num}) \land \text{anim}(o, \text{Anim}) \land \text{gen}(o, \text{Gen}) \land \text{pers}(o, \text{Pers}) \land \text{case}(o, \text{Case}) \)

+ 

\textit{the}: \( \eta x . \ \text{det}(x) \land \text{index}(x, \text{the}) \land \text{word}(x, \text{the}) \land \text{def}(x, \text{def}) \land \text{num}(x, \text{Num}) \)

+ 

\textit{altitude restriction}: \( \eta m \ \eta h \ \eta oh \ . \ obj\text{-head}(oh) \land mod(oh, m) \land \text{altitude}(m) \land \text{noun}(m) \land \text{index}(m, \text{altitude}) \land \text{word}(m, \text{altitude}) \land \text{head}(oh, h) \land \text{restriction}(h) \land \text{noun}(h) \land \text{index}(h, \text{restriction}) \land \text{word}(h, \text{restriction}) \land \text{def}(oh, \text{def}) \land \text{num}(oh, \text{sing}) \land \text{anim}(oh, \text{inanim}) \land \text{gen}(oh, \text{none}) \land \text{pers}(oh, 3\text{rd}) \)

= 
the altitude restriction: η s η m η h η oh λ o . ore(o) ∧ spec(o, s) ∧ det(s) ∧ index(s, the) ∧ word(s, the) ∧ obj-head(oh) ∧ mod(oh, m) ∧ altitude(m) ∧ noun(m) ∧ index(m, altitude) ∧ word(m, altitude) ∧ head(oh, h) ∧ restriction(h) ∧ noun(h) ∧ index(h, restriction) ∧ word(h, restriction) ∧ def(o, def) ∧ num(o, sing) ∧ anim(o, inanim) ∧ gen(o, none) ∧ pers(o, 3rd)

Unfortunately, there is a problem with this composition. From a global representation perspective, it makes sense to combine altitude with restriction before combining altitude restriction with the. However, Double-R Grammar makes a psychologically motivated commitment to incremental processing of referring expressions from left to right in written form. Given this commitment, the generation of the logical representation for the altitude restriction actually proceeds as follows:

ore: λ o η s η h . ore(o) ∧ spec(o, s) ∧ head(o, h) ∧ def(o, Def) ∧ num(o, Num) ∧ anim(o, Anim) ∧ gen(o, Gen) ∧ pers(o, Pers) ∧ case(o, Case)
+
the: η x . det(x) ∧ index(x, the) ∧ word(x, the) ∧ def(x, def) ∧ num(x, Num)
=
ore+ the: λ o η s η h . ore(o) ∧ spec(o, s) ∧ det(s) ∧ index(s, the) ∧ word(s, the) ∧ head(o, h) ∧ def(o, def) ∧ num(o, Num) ∧ anim(o, Anim) ∧ gen(o, Gen) ∧ pers(o, Pers) ∧ case(o, Case)
+
altitude: η x . altitude(x) ∧ noun(x) ∧ index(x, altitude) ∧ word(x, altitude) ∧ def(o, Def) ∧ num(x, sing) ∧ anim(x, inanim) ∧ gen(x, none) ∧ pers(x, 3rd) ∧ count(x, count) ∧ index(x, altitude) ∧ word(x, altitude)
=
the altitude: η s η h λ o . ore(o) ∧ spec(o, s) ∧ index(s, the) ∧ word(s, the) ∧ head(o, h) ∧ altitude(h) ∧ noun(h) ∧ index(h, altitude) ∧ word(h, altitude) ∧ def(o, def) ∧ num(x, sing) ∧ anim(x, inanim) ∧ gen(x, none) ∧ pers(x, 3rd)
+
obj-head: η m η h η oh . obj-head(oh) ∧ mod(oh, m) ∧ head(oh, h) ∧ def(o, Def) ∧ num(x, Num) ∧ anim(x, Anim) ∧ gen(x, Gen) ∧ pers(x, Pers)
+
restriction: η x . restriction(x) ∧ noun(x) ∧ index(x, restriction) ∧ word(x, restriction) ∧ num(x, sing) ∧ anim(x, inanim) ∧ gen(x, none) ∧ pers(x, 3rd) ∧ count(x, count) ∧ index(x, restriction) ∧ word(x, restriction)
=
the altitude restriction: η s η m η h η oh λ o . ore(o) ∧ spec(o, s) ∧ det(s) ∧ index(s, the) ∧ word(s, the) ∧ obj-head(oh) ∧ mod(oh, m) ∧ altitude(m) ∧ noun(m) ∧ index(m, altitude) ∧ word(m, altitude) ∧ head(oh, h) ∧ restriction(h) ∧ noun(h) ∧ index(h,
How does this composition occur? Since *altitude* is initially integrated as the head of object referring expression corresponding to *the altitude*, monotonic unification of *restriction* cannot succeed. Instead, the integration of *altitude* as the head must be retracted so that *restriction* can be the head. This is accomplished by shifting *altitude* from the head of the object referring expression construction to the modifier of an object head construction and integrating the object head construction as the head of the object referring expression. Finally, *restriction* is integrated as the head of the object head construction and the grammatical features of *restriction* are projected to the object referring expression, overriding the features of *altitude*. In Double-R Grammar, this non-monotonic form of composition is called context accommodation.

Reconciling Grammatical Features

Grammatical constructions encode an ordered sequence of grammatical functions that are instantiated by parts of speech and subexpressions. Parts of speech encode grammatical features which get projected to the grammatical constructions in which they are instantiated. At the level of the construction, the grammatical features projected by parts of speech must be reconciled. Reconciliation combines a monotonic unification process with two non-monotonic processes: feature blocking and feature overriding. Feature blocking and overriding come into play when monotonic unification fails. Consider the simple expression

*the books*

As noted above, the determiner *the* encodes the definiteness feature definite (def) and the plural noun *books* encodes the definiteness feature indefinite (indef). Since these two features cannot be reconciled via unification, they must be reconciled via feature blocking or overriding. The terms feature blocking and overriding are process dependent and assume incremental processing. Feature blocking occurs when a feature projected by a previously instantiated part of speech blocks a feature from a subsequently instantiated part of speech. Feature overriding is the reverse. Feature blocking and overriding are defined over the partial orderings shown below:

- Definiteness: Spec > Head > Mod
- Number: Head > Mod > Spec
- Animacy: Head > Mod > Spec
- Gender: Head > Mod > Spec
- Person: Head > Mod > Spec

The basic implication of these partial orderings is that definiteness is primarily determined by the specifier and all other grammatical features are primarily determined by the head. As a few examples consider

- *the books* – definite (def) definiteness value from the specifier *the* blocks indefinite (indef) value from the head *books*
• *a few books* – plural (plur) number value from the modifier *few* overrides singular (sing) value from the specifier *a* and is overridden by plural value from the head *books*
• *the Reagan library* – inanimate (inanim) animacy value, none gender value, and singular number value from the head *library* override human, male and singular values from the modifier *Reagan*

The need for feature blocking and overriding follows from a commitment to keeping ambiguity in the mental lexicon to a minimum, and the commitment to incremental processing. It is possible to have multiple entries in the mental lexicon to support the reconciling of grammatical features via monotonic unification, but doing so would add considerable ambiguity that would disrupt incremental processing. It would also negatively impact the semantic motivation underlying grammatical features. For example, if there were an entry in the mental lexicon for the determiner *a* with number = plural to handle *a few books*, then it is unclear what the semantic motivation for the number feature could be.

**The Logical Representation and Processing of Situation Referring Expressions in Double-R Grammar**

The full logical representation for a situation referring expression construction is shown below:

\[
\lambda \text{sit} \ \eta_\text{ps} \ \eta_\text{s} \ \lambda \text{subj} \ \eta_\text{m} \ \eta_\text{p} \ \eta_\text{pm} \ . \ \text{sre}(\text{sit}) \land \text{pre-spec}(\text{sit}, \text{ps}) \land \text{spec}(\text{sit}, \text{s}) \land \text{mod}(\text{sit}, \text{m}) \land \text{head}(\text{sit}, \text{p}) \land \text{post-mod}(\text{sit}, \text{pm}) \land \text{tense}(\text{sit}, \text{Tense}) \land \text{aspect}(\text{sit}, \text{Aspect}) \land \text{voice}(\text{sit}, \text{Voice}) \land \text{modality}(\text{sit}, \text{Modal}) \land \text{polarity}(\text{sit}, \text{Polar})
\]

where

\[\lambda\] is a handle for a referring expression
\[\eta\] is a handle for an element that is not a referring expression
sit, ps, s, subj, m, h and pm are typed variables
sit: situation referring expression
ps: pre-specifier modifier grammatical function (GFunc)
s: specifier GFunc
subj: subject GFunc
m: modifier GFunc
p: predicate (head) GFunc
pm: post-head modifier GFunc
sre = situation referring expression type (construction)
pre-spec (pre-spec-modifier) = GFunc
spec (specifier) = GFunc
subj (subject) = GFunc
mod (modifier) = GFunc
head = GFunc
post-mod (post-head modifier) = GFunc
tense = grammatical feature (GFeat)
Tense \in \{\text{pres, past, non-fin}\}
aspect = GFeat
    Aspect ∈ \{prog, perf, none\}
voice = GFeat
    Voice ∈ \{act, inact, pass\}
modality = GFeat
    Modal ∈ \{can, could, shall, should, will, would, may, must\}
polarity = GFeat
    Polar ∈ \{pos, neg\}

The underlined element is the default value for each grammatical feature.

**Combining Verbs with Situation Referring Expression and Predicate Constructions in Double-R Grammar**

The grammatical behavior of verbs depends on the verb frame (e.g. intransitive, transitive, ditransitive), or frames, that are associated with the verb. The verb frame indicates the number and default type of arguments associated with the verb. In Double-R Grammar, the verb frame is represented by adding a predicate to represent the preferred verb frame and a predicate to represent an alternative. For the three most common verb frames we have:

\[ \exists \ \eta x . \ \text{verb}(x) \land \text{verb-frame}(x, \text{intrans}) \land \text{verb-frame-alt}(x, \text{VF}) \]
\[ \exists \ \eta x . \ \text{verb}(x) \land \text{verb-frame}(x, \text{trans}) \land \text{verb-frame-alt}(x, \text{VF}) \]
\[ \exists \ \eta x . \ \text{verb}(x) \land \text{verb-frame}(x, \text{ditrans}) \land \text{verb-frame-alt}(x, \text{VF}) \]

where Double-R Grammar supports the encoding of a primary (verb-frame) and alternative (verb-frame-alt) verb frame. The logical representation for verbs is shown below:

Verb: \[ \exists \ \eta x \eta P . \ P(x) \land \text{verb}(x) \land \text{index}(x, \text{Index}) \land \text{word}(x, \text{Word}) \land \text{verb-frame}(x, \text{VF}_1) \land \text{verb-frame-alt}(x, \text{VF}_2) \land \text{tense}(x, \text{Tense}) \land \text{aspect}(x, \text{Aspect}) \land \text{voice}(x, \text{Voice}) \land \text{modality}(x, \text{Mod}) \land \text{polarity}(x, \text{Pol}) \]

where

verb = part of speech
index = lexeme
word = word form
verb-frame: GFeat
    \text{VF}_1 ∈ \{\text{intrans, trans, ditrans, ...}\}
verb-frame-alt: GFeat
    \text{VF}_2 ∈ \{\text{intrans, trans, ditrans, ...}\}
tense: grammatical feature (GFeat)
    \text{Tense} ∈ \{\text{pres, past, non-fin}\}
aspect: GFeat
    \text{Aspect} ∈ \{\text{prog, perf, none}\}
voice: GFeat
    \text{Voice} ∈ \{\text{act, inact, pass}\}
modality: GFeat
Although verb-frame and verb-frame-alt are categorized as grammatical features, they do not project to the situation referring expression, like other grammatical features. Instead, they determine which predicate construction is (or may be) projected.

Lexical representations for two morphological variants of the intransitive verb *laugh* are shown next:

**laughed:** \( \eta x \cdot \text{laugh}(x) \land \text{verb}(x) \land \text{index}(x, \text{laugh}) \land \text{word}(x, \text{laughed}) \land \text{verb-frame}(x, \text{intrans}) \land \text{verb-frame-alt}(x, \text{trans}) \land \text{tense}(x, \text{past}) \land \text{aspect}(x, \text{Asp}) \land \text{voice}(x, \text{act}) \land \text{modality}(x, \text{Mod}) \land \text{polarity}(x, \text{pos}) \)

**laughing:** \( \eta x \cdot \text{laugh}(x) \land \text{verb}(x) \land \text{index}(x, \text{laugh}) \land \text{word}(x, \text{laughing}) \land \text{verb-frame}(x, \text{intrans}) \land \text{verb-frame-alt}(x, \text{trans}) \land \text{tense}(x, \text{pres}) \land \text{aspect}(x, \text{prog}) \land \text{voice}(x, \text{act}) \land \text{modality}(x, \text{Mod}) \land \text{polarity}(x, \text{pos}) \)

In these representations, \( \eta P \cdot P(x) \) is reduced to \( \text{laugh}(x) \) where

\( \text{laugh}(x) \leftrightarrow \text{index}(x, \text{laugh}) \leftrightarrow \text{type}(x, \text{laugh}) \)

We are now in a position to consider the composition of a simple situation referring expression containing an intransitive verb:

**the man laughed**

\[ \lambda \text{sit} \lambda \text{subj} \eta \cdot \text{sre(sit)} \land \text{subj(sit, subj)} \land \text{head(sit, p)} \land \text{tense(sit, Tense)} \land \text{aspect(sit, Aspect)} \land \text{voice(sit, Voice)} \land \text{modality(sit, Modal)} \land \text{polarity(sit, Polar)} \]

\[ + \]

**the man:** \( \lambda o \eta s \eta h \cdot \text{ore(o)} \land \text{spec(o, s)} \land \text{det(s)} \land \text{index(s, the)} \land \text{word(s, the)} \land \text{head(o, h)} \land \text{man(h)} \land \text{noun(h)} \land \text{index(h, man)} \land \text{word(h, man)} \land \text{def(o, def)} \land \text{num(o, sing)} \land \text{anim(o, human)} \land \text{gen(o, male)} \land \text{pers(o, 3rd)} \]

\[ + \]

**laughed:** \( \eta x \cdot \text{laugh}(x) \land \text{verb}(x) \land \text{index}(x, \text{laugh}) \land \text{word}(x, \text{laughed}) \land \text{verb-frame}(x, \text{intrans}) \land \text{verb-frame-alt}(x, \text{trans}) \land \text{tense}(x, \text{past}) \land \text{aspect}(x, \text{none}) \land \text{voice}(x, \text{act}) \land \text{modality}(x, \text{Mod}) \land \text{polarity}(x, \text{pos}) \]

\[ = \]

**the man laughed:** \( \lambda \text{sit} \eta p \lambda \text{subj} \eta s \eta h \cdot \text{sre(sit)} \land \text{subj(sit, subj)} \land \text{ore(subj)} \land \text{spec(subj, s)} \land \text{det(s)} \land \text{index(s, the)} \land \text{word(s, the)} \land \text{head(subj, h)} \land \text{man(h)} \land \text{noun(h)} \land \text{index(h, man)} \land \text{word(h, man)} \land \text{def(subj, def)} \land \text{num(subj, sing)} \land \text{anim(subj, human)} \land \text{gen(subj, male)} \land \text{pers(subj, 3rd)} \land \text{head(sit, p)} \land \text{laugh(p)} \land \text{verb(p)} \land \text{index(p, laugh)} \land \text{word(p, laughed)} \land \text{verb-frame(p, intrans)} \land \text{verb-frame-alt(p, trans)} \land \text{tense(p, past)} \land \text{aspect(p, none)} \land \text{voice(p, act)} \land \text{modality(p, Mod)} \land \text{polarity(p, pos)} \)
frame-alt(x, trans) ∧ tense(sit, past) ∧ aspect(sit, none) ∧ voice(sit, act) ∧ modality(sit, Modal) ∧ polarity(sit, pos)

How does this composition work? First, we ignore the situation referring expression level grammatical functions pre-spec, spec, mod and post-mod which are not instantiated. For the instantiation of the subject (subj), the λ variable o of object referring expression the man must unify with the λ variable subj of the situation referring expression. For the instantiation of the (predicate) head, the η variable x of the verb laughed must unify with the η variable p of the situation referring expression. Finally, the grammatical features of the verb laughed must be projected to the situation referring expression. Note that the grammatical features of the object referring expression the man do not get projected to the situation referring expression.

The representation of a situation referring expression with an intransitive verb is kept relatively simple by allowing the intransitive verb to be instantiated as the (predicate) head of the situation referring expression. This is possible so long as the intransitive verb is not part of a more complex expression and because the subject, which is an external argument of the intransitive verb, is instantiated at the level of the situation referring expression.

When an intransitive verb participates in an expression as in

**the man laughed loud**

where loud modifies laughed, a predicate construction must be introduced to tie the parts of the expression together. This predicate construction is specific to intransitive verbs:

\[
\text{Pred-Intrans-Verb : } \eta p \eta m \eta h \eta pm . \text{pred-intrans-verb}(p) \land \text{mod}(p, m) \land \text{head}(p, h) \\
\land \text{verb}(h) \land \text{post-mod}(p, pm) \land \text{tense}(x, \text{Tense}) \land \text{aspect}(x, \text{Aspect}) \land \text{voice}(x, \text{Voice}) \\
\land \text{modality}(x, \text{Mod}) \land \text{polarity}(x, \text{Pol})
\]

where

- p, m, h and pm are handle variables
- p is of type predicate intransitive verb (pred-intrans-verb) expression
- m and pm are of type modifier grammatical function (GFunc)
- h is of type head GFunc
- pred-intrans-verb = subtype of predicate expression
- mod, head, post-mod = GFuncs
- tense, aspect, voice, modality, plurality = GFeats

In Double-R Grammar, loud is a composite adjective-adverb (adj-adv) part of speech (i.e. adjective-adverb inherits from both adjective and adverb). The lexical entry for loud is shown below:

\[
loud : \eta x . \text{loud}(x) \land \text{adj-adv}(x) \land \text{adj-form}(x, \text{base})
\]

The composition of the entire expression proceeds as follows:
the man: \lam o \ps h \l o e(o) \land spec(o, s) \land det(s) \land index(s, the) \land word(s, the) \land head(o, h) \land man(h) \land noun(h) \land index(h, man) \land word(h, man) \land def(o, def) \land num(o, sing) \land anim(o, human) \land gen(o, male) \land pers(o, 3rd)

+ 
\lam sit \lam subj \ps h \l se(sit) \land subj(sit, subj) \land head(sit, h) \land tense(sit, Tense) \land aspect(sit, Aspect) \land voice(sit, Voice) \land modality(sit, Modal) \land polarity(sit, Polar)

+ 
laughed: \eta x . \laugh(x) \land verb(x) \land index(x, laugh) \land word(x, laughed) \land verb-frame(x, intrans) \land verb-frame-alt(x, trans) \land tense(x, past) \land aspect(x, none) \land voice(x, act) \land modality(x, Mod) \land polarity(x, pos)

= 
\lam sit \lam subj \ps h \l se(sit) \land subj(sit, subj) \land ore(subj) \land spec(subj, s) \land det(s) \land index(s, the) \land word(s, the) \land head(subj, h) \land man(h) \land noun(h) \land index(h, man) \land word(h, man) \land def(subj, def) \land num(subj, sing) \land anim(subj, human) \land gen(subj, male) \land pers(subj, 3rd) \land head(sit, p) \land laugh(p) \land verb(p) \land index(p, laugh) \land word(p, laughed) \land tense(sit, past) \land aspect(sit, none) \land voice(sit, act) \land modality(sit, Modal) \land polarity(sit, pos)

+ 
Pred-Intrans-Verb: \eta p \ps h \eta pm . \pred-intrans-verb(p) \land head(p, h) \land verb(h) \land post-mod(p, pm) \land tense(p, Tense) \land aspect(p, Aspect) \land voice(p, Voice) \land modality(p, Mod) \land polarity(p, Pol)

+ 
loud: \eta x . \loud(x) \land adj-adv(x) \land adj-form(x, base) \land index(x, loud) \land word(x, loud)

= 
\lam sit \lam subj \ps h1 \ps h2 \eta pm . \se(sit) \land subj(sit, subj) \land ore(subj) \land spec(subj, s) \land det(s) \land index(s, the) \land word(s, the) \land head(subj, h1) \land man(h1) \land noun(h1) \land index(h1, man) \land def(subj, def) \land num(subj, sing) \land anim(subj, human) \land gen(subj, male) \land pers(subj, 3rd) \land head(sit, p) \land pred-intrans-verb(p) \land head(p, h2) \land laugh(h2) \land verb(h2) \land index(h2, laugh) \land word(h2, laughed) \land post-mod(p, pm) \land loud(pm) \land adj-adv(pm) \land index(pm, loud) \land word(pm, loud) \land tense(sit, past) \land aspect(sit, none) \land voice(sit, act) \land modality(sit, Modal) \land polarity(sit, pos)

Note that the intransitive verb *laughed* is initially integrated as the head of the situation referring expression based on the incremental processing commitment. When *loud* is processed, a predicate intransitive verb construction is projected and overrides *laughed* as the head of the situation referring expression. Finally, *laughed* is integrated as the head of the predicate intransitive verb construction and *loud* is integrated as a post-head modifier (post-mod).
The logical representation for a Transitive verb is shown below:

\[ \text{Trans-Verb: } \exists x \exists P \cdot P(x) \land \text{verb}(x) \land \text{index}(x, \text{Index}) \land \text{word}(x, \text{Word}) \land \text{verb-frame}(x, \text{trans}) \land \text{verb-frame-alt}(x, \text{VF}) \land \text{tense}(x, \text{Tense}) \land \text{aspect}(x, \text{Aspect}) \land \text{voice}(x, \text{Voice}) \land \text{modality}(x, \text{Mod}) \land \text{polarity}(x, \text{Pol}) \]

where

\[ \begin{align*}
\text{verb} & = \text{part of speech} \\
\text{index} & = \text{lexeme} \\
\text{word} & = \text{word form} \\
\text{Tense} & \in \{\text{pres, past}\} \\
\text{Aspect} & \in \{\text{prog, perf, none}\} \\
\text{Voice} & \in \{\text{act, inact, pass}\} \\
\text{Modality} & \in \{\text{can, could, shall, should, will, would, may, must}\} \\
\text{Polarity} & \in \{\text{pos, neg}\}
\end{align*} \]

The representation for the transitive verb *kicked* is next:

\[ \text{kicked: } \exists x \cdot \text{kick}(x) \land \text{verb}(x) \land \text{index}(x, \text{kick}) \land \text{word}(x, \text{kicked}) \land \text{verb-frame}(x, \text{trans}) \land \text{verb-frame-alt}(x, \text{ditrans}) \land \text{tense}(x, \text{past}) \land \text{aspect}(x, \text{none}) \land \text{voice}(x, \text{act}) \land \text{modality}(x, \text{Mod}) \land \text{polarity}(x, \text{pos}) \]

In this representation, \( \exists P \cdot P(x) \) is reduced to \( \text{kick}(x) \) where

\[ \text{kick}(x) \leftrightarrow \text{index}(x, \text{kick}) \leftrightarrow \text{type}(x, \text{kick}) \]

The predicate transitive verb construction that is projected by a transitive verb when it functions as the head of a situation referring expression is shown below:

\[ \text{Pred-Trans-Verb : } \exists p \exists m \exists h \exists \text{obj} \exists \text{pm} \cdot \text{pred-trans-verb}(p) \land \text{mod}(p, m) \land \text{head}(p, h) \land \text{verb}(h) \land \text{obj}(p, \text{obj}) \land \text{post-mod}(p, \text{pm}) \land \text{tense}(x, \text{Tense}) \land \text{aspect}(x, \text{Aspect}) \land \text{voice}(x, \text{Voice}) \land \text{modality}(x, \text{Mod}) \land \text{polarity}(x, \text{Pol}) \]

where

\[ \begin{align*}
p, m, h, \text{obj}, \text{pm} \text{ and } P \text{ are handle variables} \\
p \text{ is of type predicate transitive verb (pred-trans-verb) expression} \\
m, h, \text{obj}, \text{pm are of type grammatical function (GFunc)} \\
\text{pred-trans-verb} = \text{subtype of predicate expression} \\
\text{mod, head, obj, post-mod} = \text{GFuncs} \\
\text{tense, aspect, voice, modality, polarity} = \text{GFeats}
\end{align*} \]

The composition of the simple transitive verb expression

\[ \text{the man kicked the ball} \]

proceeds as follows:
the man: \( \lambda o \ \eta s \ \eta h \ . \ ore(o) \ \land \ spec(o, s) \ \land \ det(s) \ \land \ index(s, the) \ \land \ word(s, the) \ \land \ head(o, h) \ \land \ man(h) \ \land \ noun(h) \ \land \ index(h, man) \ \land \ word(h, man) \ \land \ def(o, def) \ \land \ num(o, sing) \ \land \ anim(o, human) \ \land \ gen(o, male) \ \land \ pers(o, 3rd) \)

+ 
\( \lambda \text{sit} \ \lambda \text{subj} \ \eta s \ \eta p \ . \ \text{sre(sit)} \ \land \ \text{subj}(sit, subj) \ \land \ \text{head}(sit, p) \ \land \ \text{tense}(sit, Tense) \ \land \ \text{aspect}(sit, Aspect) \ \land \ \text{voice}(sit, Voice) \ \land \ \text{modality}(sit, Modal) \ \land \ \text{polarity}(sit, Polar) \)

+ 
\text{Pred-Trans-Verb} : \ \eta p \ \eta h \ \lambda \text{obj} \ \eta pm \ . \ \text{pred-trans-verb(p)} \ \land \ \text{head(p, h)} \ \land \ \text{verb(h)} \ \land \ \text{obj(p, obj)} \ \land \ \text{post-mod(p, pm)} \ \land \ \text{tense(x, Tense)} \ \land \ \text{aspect(x, Aspect)} \ \land \ \text{voice(x, Voice)} \ \land \ \text{modality(x, Mod)} \ \land \ \text{polarity(x, Polar)} \)

+ 
\( \text{kicked} : \ \eta x \ . \ \text{kick(x)} \ \land \ \text{verb(x)} \ \land \ \text{index(x, kick)} \ \land \ \text{word(x, kicked)} \ \land \ \text{verb-frame(x, trans)} \ \land \ \text{verb-frame-alt(x, ditrans)} \ \land \ \text{tense(x, past)} \ \land \ \text{aspect(x, none)} \ \land \ \text{voice(x, act)} \ \land \ \text{modality(x, Mod)} \ \land \ \text{polarity(x, pos)} \)

= 
\( \lambda \text{sit} \ \lambda \text{subj} \ \eta s_1 \ \eta h_1 \ \eta p \ \eta h_2 \ \lambda \text{obj} \ . \ \text{sre(sit)} \ \land \ \text{subj}(sit, subj) \ \land \ \text{ore(subj)} \ \land \ \text{spec(subj, s)} \ \land \ \text{det(s)} \ \land \ \text{index(s, the)} \ \land \ \text{word(s, the)} \ \land \ \text{head(subj, h_1)} \ \land \ \text{man(h_1)} \ \land \ \text{noun(h_1)} \ \land \ \text{index(h_1, man)} \ \land \ \text{word(h_1, man)} \ \land \ \text{def(subj, def)} \ \land \ \text{num(subj, sing)} \ \land \ \text{anim(subj, human)} \ \land \ \text{gen(subj, male)} \ \land \ \text{pers(subj, 3rd)} \ \land \ \text{head(sit, p)} \ \land \ \text{pred-trans-verb(p)} \ \land \ \text{head(p, h_2)} \ \land \ \text{kick(h_2)} \ \land \ \text{verb(h_2)} \ \land \ \text{index(h_2, kick)} \ \land \ \text{obj(p, obj)} \ \land \ \text{tense(sit, past)} \ \land \ \text{aspect(sit, none)} \ \land \ \text{voice(sit, act)} \ \land \ \text{modality(sit, Modal)} \ \land \ \text{polarity(sit, pos)} \)

+ 
\( \text{the ball} : \ \lambda o \ \eta s \ \eta h . \ ore(o) \ \land \ spec(o, s) \ \land \ det(s) \ \land \ index(s, the) \ \land \ word(s, the) \ \land \ head(o, h) \ \land \ ball(h) \ \land \ noun(h) \ \land \ index(h, ball) \ \land \ word(h, ball) \ \land \ def(o, def) \ \land \ num(o, sing) \ \land \ anim(o, inanim) \ \land \ gen(o, none) \ \land \ pers(o, 3rd) \)

= 
\( \lambda \text{sit} \ \lambda \text{subj} \ \eta s_1 \ \eta h_1 \ \eta p \ \eta h_2 \ \eta s_3 \ \eta h_3 \ \lambda \text{obj} \ . \ \text{sre(sit)} \ \land \ \text{subj}(sit, subj) \ \land \ \text{ore(subj)} \ \land \ \text{spec(subj, s_1)} \ \land \ \text{det(s_1)} \ \land \ \text{index(s_1, the)} \ \land \ \text{word(s_1, the)} \ \land \ \text{head(subj, h_1)} \ \land \ \text{man(h_1)} \ \land \ \text{noun(h_1)} \ \land \ \text{index(h_1, man)} \ \land \ \text{def(subj, def)} \ \land \ \text{num(subj, sing)} \ \land \ \text{anim(subj, human)} \ \land \ \text{gen(subj, male)} \ \land \ \text{pers(subj, 3rd)} \ \land \ \text{head(sit, p)} \ \land \ \text{pred-trans-verb(p)} \ \land \ \text{head(p, h_2)} \ \land \ \text{kick(h_2)} \ \land \ \text{verb(h_2)} \ \land \ \text{index(h_2, kick)} \ \land \ \text{word(h_2, kicked)} \ \land \ \text{obj(p, obj)} \ \land \ \text{ore(obj)} \ \land \ \text{spec(obj, s_3)} \ \land \ \text{det(s_3)} \ \land \ \text{index(s_3, the)} \ \land \ \text{word(s_3, the)} \ \land \ \text{head(obj, h_3)} \ \land \ \text{ball(h_3)} \ \land \ \text{noun(h_3)} \ \land \ \text{index(h_3, ball)} \ \land \ \text{word(h_3, ball)} \ \land \ \text{def(obj, def)} \ \land \ \text{num(obj, sing)} \ \land \ \text{anim(obj, inanim)} \ \land \ \text{gen(obj, none)} \ \land \ \text{pers(obj, 3rd)} \ \land \ \text{tense(sit, past)} \ \land \ \text{aspect(sit, none)} \ \land \ \text{voice(sit, act)} \ \land \ \text{modality(sit, Modal)} \ \land \ \text{polarity(sit, pos)} \)

Because a transitive verb expects an object argument, a predicate transitive verb construction is always required. Hence the processing of the verb \textit{kicked} first projects a
predicate transitive verb construction and then a situation referring expression construction. Both of these constructions are then available to support the integration of the subject *the man*, the predicate head *kicked*, and the object *the ball*.

**Auxiliary Verb**

Auxiliary verbs, which function as specifiers in situation referring expressions, perform a different grammatical function than regular verbs, which function as predicate heads. The lexical entry for auxiliary verbs is shown below:

\[
\text{Aux-Verb: } \eta x \cdot \text{aux}(x) \land \text{index}(x, \text{Index}) \land \text{word}(x, \text{Word}) \land \text{tense}(x, \text{Tense}) \land \text{aspect}(x, \text{Aspect}) \land \text{voice}(x, \text{Voice}) \land \text{modality}(x, \text{Mod}) \land \text{polarity}(x, \text{Pol})
\]

where

- \text{aux} = \text{part of speech}
- \text{index} = \text{lexeme}
- \text{word} = \text{word form}
- \text{Tense} \in \{\text{pres, past}\}
- \text{Aspect} \in \{\text{prog, perf}\}
- \text{Voice} \in \{\text{act, inact, pass}\}
- \text{Modality} \in \{\text{can, could, shall, should,...}\}
- \text{Polarity} \in \{\text{pos, neg}\}

The lexical entry for the auxiliary verb *is* is shown next:

\[
is: \eta x \cdot \text{aux}(x) \land \text{index}(x, \text{be}) \land \text{word}(x, \text{is}) \land \text{tense}(x, \text{pres}) \land \text{aspect}(x, \text{Aspect}) \land \text{voice}(x, \text{inact}) \land \text{modality}(x, \text{Mod}) \land \text{polarity}(x, \text{pos})\]

The composition of the simple situation referring expression

*the man is laughing*

proceeds as follows:

\[
\text{the man: } \lambda o \eta s \eta h . \text{ore}(o) \land \text{spec}(o, s) \land \text{det}(s) \land \text{index}(s, \text{the}) \land \text{word}(s, \text{the}) \land \text{head}(o, h) \land \text{noun}(h) \land \text{index}(h, \text{man}) \land \text{word}(h, \text{man}) \land \text{def}(o, \text{def}) \land \text{num}(o, \text{sing}) \land \text{anim}(o, \text{human}) \land \text{gen}(o, \text{male}) \land \text{pers}(o, \text{3rd})
\]

\[
+ \lambda \text{sit} \lambda \text{subj} \eta s \eta h . \text{sre}(\text{sit}) \land \text{subj}(\text{sit, subj}) \land \text{spec}(\text{sit, s}) \land \text{head}(\text{sit, h}) \land \text{tense}(\text{sit, Tense}) \land \text{aspect}(\text{sit, Aspect}) \land \text{voice}(\text{sit, Voice}) \land \text{modality}(\text{sit, Modal}) \land \text{polarity}(\text{sit, Polar})
\]

\[
+ \eta x . \text{aux}(x) \land \text{index}(x, \text{be}) \land \text{word}(x, \text{is}) \land \text{tense}(x, \text{pres}) \land \text{aspect}(x, \text{Aspect}) \land \text{voice}(x, \text{inact}) \land \text{modality}(x, \text{Mod}) \land \text{polarity}(x, \text{pos})
\]

= 
\[ \lambda \text{sit} \lambda \text{subj} \eta_1 \eta_2 \eta_3 \eta_4 \text{. } \text{sre(sit)} \land \text{subj(sit, subj)} \land \text{ore(subj)} \land \text{spec(subj, s)} \land \text{det(s)} \land \text{index(s, the)} \land \text{word(s, the)} \land \text{head(subj, h)} \land \text{man(h)} \land \text{noun(h)} \land \text{index(h, man)} \land \text{word(h, man)} \land \text{def(subj, def)} \land \text{num(subj, sing)} \land \text{anim(subj, human)} \land \text{gen(subj, male)} \land \text{pers(subj, 3rd)} \land \text{spec(sit, s)} \land \text{aux(s)} \land \text{index(s, be)} \land \text{word(s, is)} \land \text{head(sit, h)} \land \text{tense(sit, pres)} \land \text{aspect(sit, none)} \land \text{voice(sit, inact)} \land \text{modality(sit, Modal)} \land \text{polarity(sit, pos)} \]

+ \[
\text{laughing} : \eta x . \text{laugh(x)} \land \text{verb(x)} \land \text{index(x, laugh)} \land \text{word(x, laughing)} \land \text{verb-frame(x, intrans)} \land \text{verb-frame-alt(x, trans)} \land \text{tense(x, pres)} \land \text{aspect(x, prog)} \land \text{voice(x, act)} \land \text{modality(x, x, Mod)} \land \text{polarity(x, pos)}
\]

= \[
\lambda \text{sit} \lambda \text{subj} \eta_1 \eta_2 \eta_3 \eta_4 \text{. } \text{sre(sit)} \land \text{subj(sit, subj)} \land \text{ore(subj)} \land \text{spec(subj, s)} \land \text{det(s)} \land \text{index(s, the)} \land \text{word(s, the)} \land \text{head(subj, h)} \land \text{man(h)} \land \text{noun(h)} \land \text{index(h, man)} \land \text{word(h, man)} \land \text{def(subj, def)} \land \text{num(subj, sing)} \land \text{anim(subj, human)} \land \text{gen(subj, male)} \land \text{pers(subj, 3rd)} \land \text{spec(sit, s)} \land \text{aux(s)} \land \text{index(s, be)} \land \text{word(s, is)} \land \text{head(sit, h)} \land \text{tense(sit, pres)} \land \text{aspect(sit, prog)} \land \text{voice(sit, act)} \land \text{modality(sit, Mod)} \land \text{polarity(sit, pos)}
\]

Note that the auxiliary verb \textit{is} functions as the (clausal) specifier, whereas the regular verb \textit{laughing} functions as the predicate head. In Double-R Grammar, the (clausal) specifier and subject are distinct grammatical functions. The (clausal) specifier functions to situate the situation referring expression, whereas the subject functions as an external argument of the predicate head.

**Combining Adjectives and Prepositions with Situation Referring Expression and Predicate Constructions in Double-R Grammar**

The treatment of adjectives and prepositions at the situation referring expression level in Double-R Grammar differs significantly from other grammatical treatments, and aligns with common logical treatments. In particular, adjectives and prepositions may function as the heads of situation referring expressions, whereas they are treated as complements and adjuncts in other grammatical approaches. Consider the sentences

\[
\begin{align*}
\text{John is happy} \\
\text{the book is on the table}
\end{align*}
\]

for which the following predicate logic representations are suggested:

\[
\begin{align*}
\text{happy(J)} \\
\exists x \exists y [\text{on}(x, y) \land \text{book}(x) \land \text{table}(y)]
\end{align*}
\]

In these predicate logic representations, the adjective \textit{happy} and the preposition \textit{on} are both translated into predicates, and the contribution of the auxiliary verb \textit{is} is ignored. These logical representations capture the fact that \textit{happy} and \textit{on} are the primary relational elements in these sentences. In Double-R Grammar terms, they function as the predicate
heads of the situation referring expressions, with the auxiliary verb *is* having a subordinate grammatical function (i.e. specifier). The logical representations also capture the fact that *happy* is an intransitive relation (i.e. a property), whereas *on* is a transitive relation.

The logical representation of these sentences in Double-R Grammar parallels the predicate logic representations with the extensions that have already been introduced.

For the first sentence, the composition proceeds as follows:

\[
\text{John: } \lambda o \ \eta s \ \eta h \ . \ \text{ore}(o) \land \text{spec}(o, s) \land \text{head}(o, h) \land \text{prop-noun}(h) \land \text{index}(h, \text{John}) \land \\
\text{word}(h, \text{John}) \land \text{def}(o, \text{def}) \land \text{num}(o, \text{sing}) \land \text{anim}(o, \text{human}) \land \text{gen}(o, \text{male}) \land \\
\text{pers}(o, 3\text{rd}) + \\
\lambda s \ \text{sit} \ \eta s \ \eta h \ . \ \text{sre}(\text{sit}) \land \text{subj}(\text{sit}, \text{subj}) \land \text{spec}(\text{sit}, s) \land \text{head}(\text{sit}, h) \land \text{tense}(\text{sit}, \text{Tense}) \land \text{aspect}(\text{sit}, \text{Aspect}) \land \text{voice}(\text{sit}, \text{Voice}) \land \text{modality}(\text{sit}, \text{Modal}) \land \\
\text{polarity}(\text{sit}, \text{Polar}) + \\
is: \ \eta x \ . \ \text{aux}(x) \land \text{index}(x, \text{be}) \land \text{word}(x, \text{is}) \land \text{tense}(x, \text{pres}) \land \text{aspect}(x, \text{Aspect}) \land \\
\text{voice}(x, \text{inact}) \land \text{modality}(x, \text{Modal}) \land \text{polarity}(x, \text{pos}) = \\
\lambda s \ \text{sit} \ \lambda \text{subj} \ \eta s_1 \ \eta h_1 \ \eta s_2 \ \eta h_2 \ . \ \text{sre}(\text{sit}) \land \text{subj}(\text{sit}, \text{subj}) \land \text{ore}(\text{subj}) \land \text{spec}(o, s) \land \\
\text{head}(\text{subj}, h_1) \land \text{prop-noun}(h_1) \land \text{index}(h_1, \text{John}) \land \text{word}(h_1, \text{John}) \land \text{def}(\text{subj}, \text{def}) \land \text{num}(\text{subj}, \text{sing}) \land \text{anim}(\text{subj}, \text{human}) \land \text{gen}(\text{subj}, \text{male}) \land \text{pers}(\text{subj}, 3\text{rd}) \land \\
\text{spec}(\text{sit}, s_2) \land \text{aux}(s_2) \land \text{index}(s_2, \text{be}) \land \text{word}(s_2, \text{is}) \land \text{head}(\text{sit}, h_2) \land \text{tense}(\text{sit}, \text{pres}) \land \text{aspect}(\text{sit}, \text{none}) \land \text{voice}(\text{sit}, \text{inact}) \land \text{modality}(\text{sit}, \text{Modal}) \land \\
\text{polarity}(\text{sit}, \text{pos}) + \\
happy: \ \eta x \ . \ \text{happy}(x) \land \text{adj}(x) \land \text{index}(x, \text{happy}) \land \text{word}(x, \text{happy}) \land \text{adj-form}(x, \text{base}) = \\
\lambda s \ \text{sit} \ \lambda \text{subj} \ \eta s_1 \ \eta h_1 \ \eta s_2 \ \eta h_2 \ . \ \text{sre}(\text{sit}) \land \text{subj}(\text{sit}, \text{subj}) \land \text{ore}(\text{subj}) \land \text{spec}(o, s) \land \\
\text{head}(\text{subj}, h_1) \land \text{noun}(h_1) \land \text{index}(h_1, \text{John}) \land \text{word}(h_1, \text{John}) \land \text{def}(\text{subj}, \text{def}) \land \\
\text{num}(\text{subj}, \text{sing}) \land \text{anim}(\text{subj}, \text{human}) \land \text{gen}(\text{subj}, \text{male}) \land \text{pers}(\text{subj}, 3\text{rd}) \land \\
\text{spec}(\text{sit}, s_2) \land \text{aux}(s_2) \land \text{index}(s_2, \text{be}) \land \text{word}(s_2, \text{is}) \land \text{head}(\text{sit}, h_2) \land \text{happy}(h_2) \land \\
\text{adj}(h_2) \land \text{index}(h_2, \text{happy}) \land \text{word}(h_2, \text{happy}) \land \text{tense}(\text{sit}, \text{pres}) \land \text{aspect}(\text{sit}, \text{Asp}) \land \text{voice}(\text{sit}, \text{inact}) \land \text{modality}(\text{sit}, \text{Modal}) \land \\
\text{polarity}(\text{sit}, \text{pos})
\]

Note that in this example, the grammatical features of the situation referring expression are exclusively projected by the auxiliary verb. The adjective *happy* does not project any grammatical features.

For the second sentence, the composition proceeds as follows:
the book: \(\lambda o \ \eta s \ \eta h. \ \text{ore}(o) \land \text{spec}(o, s) \land \text{det}(s) \land \text{index}(s, \text{the}) \land \text{word}(s, \text{the}) \land \text{head}(o, h) \land \text{book}(h) \land \text{noun}(h) \land \text{index}(h, \text{book}) \land \text{word}(h, \text{book}) \land \text{def}(o, \text{def}) \land \text{num}(o, \text{sing}) \land \text{anim}(o, \text{anim}) \land \text{gen}(o, \text{none}) \land \text{pers}(o, \text{3rd})\)

+ \(\lambda \text{sit} \ \lambda \text{subj} \ \eta s \ \eta h. \ \text{sre}(\text{sit}) \land \text{subj}(\text{sit}, \text{subj}) \land \text{spec}(\text{sit}, s) \land \text{head}(\text{sit}, h) \land \text{tense}(\text{sit}, \text{Tense}) \land \text{aspect}(\text{sit}, \text{Aspect}) \land \text{voice}(\text{sit}, \text{Voice}) \land \text{modality}(\text{sit}, \text{Modal}) \land \text{polarity}(\text{sit}, \text{Polar})\)

+ \(\eta x. \ \text{aux}(x) \land \text{index}(x, \text{be}) \land \text{word}(x, \text{is}) \land \text{tense}(x, \text{pres}) \land \text{aspect}(x, \text{Aspect}) \land \text{voice}(x, \text{inact}) \land \text{modality}(x, \text{Modal}) \land \text{polarity}(x, \text{pos})\)

= \(\lambda \text{sit} \ \lambda \text{subj} \ \eta s_1 \ \eta h_1 \ \eta s_2 \ \eta h_2. \ \text{sre}(\text{sit}) \land \text{subj}(\text{sit}, \text{subj}) \land \text{ore}(\text{subj}) \land \text{spec}(\text{subj}, s_1) \land \text{det}(s_1) \land \text{index}(s_1, \text{the}) \land \text{word}(s_1, \text{the}) \land \text{head}(\text{subj}, h_1) \land \text{book}(h_1) \land \text{noun}(h_1) \land \text{index}(h_1, \text{book}) \land \text{word}(h_1, \text{book}) \land \text{def}(\text{subj}, \text{def}) \land \text{num}(\text{subj}, \text{sing}) \land \text{anim}(\text{subj}, \text{anim}) \land \text{gen}(\text{subj}, \text{none}) \land \text{pers}(\text{subj}, \text{3rd}) \land \text{spec}(\text{sit}, s_2) \land \text{aux}(s_2) \land \text{index}(s_2, \text{be}) \land \text{word}(s_2, \text{is}) \land \text{head}(\text{sit}, h_2) \land \text{tense}(\text{sit}, \text{pres}) \land \text{aspect}(\text{sit}, \text{Asp}) \land \text{voice}(\text{sit}, \text{inact}) \land \text{modality}(\text{sit}, \text{Modal}) \land \text{polarity}(\text{sit}, \text{pos})\)

+ \(\text{Pred-Trans}: \ \eta p \ \eta h. \ \lambda \text{obj} \ \eta p m. \ \text{pred-trans}(p) \land \text{head}(p, h) \land \text{obj}(p, \text{obj}) \land \text{post-mod}(\text{p}, \text{pm}) \land \text{tense}(\text{x}, \text{Tense}) \land \text{aspect}(\text{x}, \text{Aspect}) \land \text{voice}(\text{x}, \text{Voice}) \land \text{modality}(\text{x}, \text{Mod}) \land \text{polarity}(\text{x}, \text{Pol})\)

+ \(\text{on}: \ \eta x. \ \text{on}(x) \land \text{prep}(x) \land \text{index}(x, \text{on}) \land \text{word}(x, \text{on})\)

= \(\lambda \text{sit} \ \lambda \text{subj} \ \eta s_1 \ \eta h_1 \ \eta p \ \eta h_2 \ \lambda \text{obj} \ \eta p m. \ \text{sre}(\text{sit}) \land \text{subj}(\text{sit}, \text{subj}) \land \text{ore}(\text{subj}) \land \text{spec}(\text{subj}, s_1) \land \text{det}(s_1) \land \text{index}(s_1, \text{the}) \land \text{word}(s_1, \text{the}) \land \text{head}(\text{subj}, h_1) \land \text{book}(h_1) \land \text{noun}(h_1) \land \text{index}(h_1, \text{book}) \land \text{word}(h_1, \text{book}) \land \text{def}(\text{subj}, \text{def}) \land \text{num}(\text{subj}, \text{sing}) \land \text{anim}(\text{subj}, \text{anim}) \land \text{gen}(\text{subj}, \text{none}) \land \text{pers}(\text{subj}, \text{3rd}) \land \text{head}(\text{sit}, p) \land \text{pred-trans}(p) \land \text{head}(p, h_2) \land \text{on}(h_2) \land \text{prep}(h_2) \land \text{index}(h_2, \text{on}) \land \text{word}(h_2, \text{on}) \land \text{obj}(p, \text{obj}) \land \text{tense}(\text{sit}, \text{pres}) \land \text{aspect}(\text{sit}, \text{Asp}) \land \text{voice}(\text{sit}, \text{inact}) \land \text{modality}(\text{sit}, \text{Modal}) \land \text{polarity}(\text{sit}, \text{pos})\)

+ \(\text{the table}: \ \lambda o \ \eta s \ \eta h. \ \text{ore}(o) \land \text{spec}(o, s) \land \text{det}(s) \land \text{index}(s, \text{the}) \land \text{word}(s, \text{the}) \land \text{head}(o, h) \land \text{table}(h) \land \text{noun}(h) \land \text{index}(h, \text{table}) \land \text{word}(h, \text{table}) \land \text{def}(o, \text{def}) \land \text{num}(o, \text{sing}) \land \text{anim}(o, \text{anim}) \land \text{gen}(o, \text{none}) \land \text{pers}(o, \text{3rd})\)

= \(\lambda \text{sit} \ \lambda \text{subj} \ \eta s_1 \ \eta h_1 \ \eta p \ \eta h_2 \ \eta h_3 \ \lambda \text{obj} \ \eta p m. \ \text{sre}(\text{sit}) \land \text{subj}(\text{sit}, \text{subj}) \land \text{ore}(\text{subj}) \land \text{spec}(\text{subj}, s_1) \land \text{det}(s_1) \land \text{index}(s_1, \text{the}) \land \text{word}(s_1, \text{the}) \land \text{head}(\text{subj}, h_1) \land \text{table}(h_1) \land \text{noun}(h_1) \land \text{index}(h_1, \text{table}) \land \text{word}(h_1, \text{table}) \land \text{def}(\text{subj}, \text{def}) \land \text{num}(\text{subj}, \text{sing}) \land \text{anim}(\text{subj}, \text{anim}) \land \text{gen}(\text{subj}, \text{none}) \land \text{pers}(\text{subj}, \text{3rd}) \land \text{table}(h_2) \land \text{on}(h_2) \land \text{prep}(h_2) \land \text{index}(h_2, \text{on}) \land \text{word}(h_2, \text{on}) \land \text{obj}(p, \text{obj}) \land \text{tense}(\text{sit}, \text{pres}) \land \text{aspect}(\text{sit}, \text{Asp}) \land \text{voice}(\text{sit}, \text{inact}) \land \text{modality}(\text{sit}, \text{Modal}) \land \text{polarity}(\text{sit}, \text{pos})\)


Likes predicate adjectives, predicate prepositions do not project any grammatical features to the situation referring expressions they head. Nonetheless, they are the main semantic element of the situation referring expression. In its function as a specifier, the auxiliary verb performs a peripheral grammatical function and is not the head, even though it projects the situation referring expression and provides all the grammatical features.

**Combining the Object Referring Expression, Predicate Nominal and Situation Referring Expression Constructions in Double-R Grammar**

We are now in a position to consider the representation and processing of predicate nominals like 

*John is a man*

\[ John: \lambda o \eta s \eta h. \text{ore}(o) \land \text{spec}(o, s) \land \text{head}(o, h) \land \text{prop-noun}(h) \land \text{index}(h, John) \land \text{word}(h, John) \land \text{def}(o, def) \land \text{num}(o, sing) \land \text{anim}(o, human) \land \text{gen}(o, male) \land \text{pers}(o, 3rd) \]

\[ + \]

\[ \lambda \text{sit} \lambda \text{subj} \eta s \eta h. \text{sre}(\text{sit}) \land \text{subj}(\text{sit}, \text{subj}) \land \text{spec}(\text{sit}, s) \land \text{head}(\text{sit}, h) \land \text{tense}(\text{sit}, Tense) \land \text{aspect}(\text{sit}, Aspect) \land \text{voice}(\text{sit}, Voice) \land \text{modality}(\text{sit}, \text{Modal}) \land \text{polarity}(\text{sit}, \text{Polar}) \]

\[ + \]

\[ is: \eta x. \text{aux}(x) \land \text{index}(x, \text{be}) \land \text{word}(x, \text{is}) \land \text{tense}(x, \text{pres}) \land \text{aspect}(x, Aspect) \land \text{voice}(x, \text{inact}) \land \text{modality}(x, \text{Modal}) \land \text{polarity}(x, \text{pos}) \]

\[ = \]

\[ \lambda \text{sit} \lambda \text{subj} \eta s_1 \eta h_1 \eta s_2 \eta h_2. \text{sre}(\text{sit}) \land \text{subj}(\text{sit}, \text{subj}) \land \text{ore}(\text{subj}) \land \text{spec}(o, s_1) \land \text{head}(\text{subj}, h_1) \land \text{prop-noun}(h_1) \land \text{index}(h_1, John) \land \text{word}(h_1, John) \land \text{def}(\text{subj}, \text{def}) \land \text{num}(\text{subj}, \text{sing}) \land \text{anim}(\text{subj}, \text{human}) \land \text{gen}(\text{subj}, \text{male}) \land \text{pers}(\text{subj}, 3rd) \land \text{spec}(\text{sit}, s_2) \land \text{aux}(s_2) \land \text{index}(s_2, \text{be}) \land \text{word}(s_2, \text{is}) \land \text{head}(\text{sit}, h_2) \land \text{tense}(\text{sit}, \text{pres}) \land \text{aspect}(\text{sit}, Aspect) \land \text{voice}(\text{sit}, \text{inact}) \land \text{modality}(\text{sit}, \text{Modal}) \land \text{polarity}(\text{sit}, \text{pos}) \]

\[ + \]
The proper noun *John* projects an object referring expression construction in which it is integrated as the head. The auxiliary verb *is* projects a situation referring expression in which it is integrated as the specifier. The object referring expression with head *John* (in the subject buffer) is simultaneously integrated as the subject. The object referring expression *a man*, is integrated as the (predicate) head of the situation referring expression. The handle variables for the two object referring expressions are equated to indicate co-reference (i.e. subj = h₂). Even though *a man* is functioning as a predicate nominal, it is still a referring expression. In terms of the situation model, the processing of *a man* either introduces a referent of type *man* into the situation model that is equated with the referent for *John*, or there is a single referent that is associated with both the name *John* and the type *man*.

**Situation Referring Expression Summary**

We have now considered the logical representation and processing of the four basic types of predicate that occur in situation referring expressions:

1. predicate verb (intransitive and transitive)
2. predicate adjective
3. predicate preposition
4. predicate nominal

There are a number of additional predicate verb types that are discussed in Ball et al. (2014). Since verbs often participate in multiple verb frames (cf. Levin, 1993), Double-R Grammar provides mechanisms for handling these alternatives which do not require multiple entries in the mental lexicon. It is a basic fact of English that the non-subject arguments of a verb are not normally available at the time the verb itself is processed. As such, the incremental processing mechanism must prefer the most common verb frame for a verb, but be capable of handling the alternatives. Double-R Grammar uses a form of context accommodation to achieve this objective.

The four types of predicate are all untensed with one exception – predicate verbs may be tensed:
In the first example, *kicked* is in the past tense. In the second example, the past tense is carried by the auxiliary verb *did*, which functions as a specifier and is not part of the predicate. The use of a tensed main verb is the situation referring expression equivalent of object referring expressions headed by proper nouns, pronouns and mass and plural nouns without separate specification.

**Co-reference in Predicate Logic and Extensions**

The topic of co-reference in predicate logic is an important one that becomes even more important when we consider discourses that involve multiple independent sentences or sentences with embedded clauses. In this section we will only touch on this topic, primarily using Discourse Representation Theory (DRT) as the logical formalism.

The simple discourse

*John likes Mary. She is nice.*

can be represented in predicate logic as

\[ \text{like}(J, M) \land \text{nice}(M) \]

where \(J\) and \(M\) are logical constants. However, this representation presupposes that the pronoun *she* has somehow been resolved to *Mary*. If we introduce a variable \(x\) to represent the pronoun *she*, and use the equality operator to equate the variable with the logical constant \(M\), we have instead

\[ \exists x [\text{like}(J, M) \land \text{nice}(x) \land M = x] \]

In this representation, the logical constant \(M\) is set equal to the existentially quantified variable \(x\), indicating co-reference.

DRT goes a step further in introducing discourse referents for the proper nouns *John* and *Mary*, as well. Adding these discourse referents and using existential quantification results in the following predicate logic representation:

\[ \exists x \exists y \exists z [\text{like}(x, y) \land x = \text{John} \land y = \text{Mary} \land \text{nice}(z) \land y = z] \]

Abstracting from the existential quantification and conjunction leads to the following Discourse Representation Structure (DRS):

\[ [(x, y, z) \text{ like}(x, y), x = \text{John}, y = \text{Mary}, \text{nice}(z), y = z] \]

In this representation, discourse referents are introduced for the object referring expressions *John, Mary* and *she*, and the discourse referent for *she* is set equal to the discourse referent for *Mary*, establishing co-reference.
Early descriptions of DRT theory (Kamp, 1981; Kamp & Reyle, 1993) did not explain how co-reference is actually determined. Although it was acknowledged that grammatical features play an important role in co-reference resolution, they were not logically represented. There is nothing in these early formulations of DRT that would preclude the following representation in which she is set equal to John:

\[(x, y, z) \text{ like}(x, y), x=\text{John}, y=\text{Mary}, \text{nice}(z), x = z\]

More recent descriptions of DRT are beginning to address this issue (Kamp, van Genabith & Reyle, 2011). Not only that, but the version of DRT described in Kamp, van Genabith & Reyle (2011) adopts an incremental processing perspective that necessitates a context accommodation mechanism like that used in Double-R Grammar. For example, the simple discourse

*A delegate* arrived. *She* registered.

requires accommodation of the gender grammatical feature to resolve the co-reference of she with a delegate, since the gender feature of a delegate is not female. If the gender feature of a delegate is represented as a variable, simple unification of a variable with a value suffices. If the gender feature is represented as the value neuter (or none), then the female gender feature of she must override the neuter (or none) gender feature of a delegate.

The determination of co-reference between pronouns and antecedents is subject to accessibility constraints in DRT. For example, the discourse referents introduced by the antecedent of a conditional are accessible to the discourse referents introduced by the consequent, but not otherwise. Given the expression

*If Pedro owns a donkey, he beats it*

the discourse referents introduced by Pedro and a donkey in the antecedent Pedro owns a donkey are accessible to the discourse referents introduced by he and it in the consequent he beats it. This supports establishment of co-reference between he and Pedro and between it and a donkey. On the other hand, in the expression

*If he reads the book, John will like it*

the discourse referent introduced by John in the consequent is not accessible to he in the antecedent unless it “escapes” to an accessible position (e.g. if there is a discourse referent outside the conditional expression that is equated with the discourse referent for it. We saw an example of this in our earlier discussion of

*Every man loves a woman*

where a discourse referent \(v\) is introduced in the outer box to support a wide scope interpretation of a woman:
The discourse referent in the outer box is accessible to discourse referents in the inner boxes, but not vice versa. The mechanism by which discourse referents “escape” is only discussed for proper nouns in Kamp, van Genabith & Reyle (2011).

**Co-reference in Double-R Grammar**

In this section, we only provide a brief overview of co-reference in Double-R Grammar. A fuller description, which is not logically oriented, is available as Ball (2014a). The treatment of co-reference in Double-R Grammar is inextricably entwined with processing and memory considerations. A purely declarative discussion is not available.

An important strength of the integrated logico-grammatical representations of Double-R Grammar is the support that this provides for determining co-reference. It is well known that there are grammatical constraints on the co-reference of pronouns and anaphors (cf. Chomsky, 1981). For example, consider the expressions

\[
\begin{align*}
&\text{John likes himself} \\
&\text{John likes him}
\end{align*}
\]

It is clear that the reflexive pronoun *himself* in the first sentence is co-referential with *John*, whereas the personal pronoun *him* in the second sentence is not co-referential with *John*. In these examples, co-reference depends on the grammatical categorization of the words *himself* and *him* and their grammatical function. In Double-R Grammar, reflexive pronoun is a distinct part of speech from personal pronoun. This distinction is made possible by the multi-level part of speech lattice used in Double-R Grammar. In addition, the grammatical functions subject and object are explicitly represented, and the subject is retained in a subject buffer to support co-reference determination when the object is processed. (An object buffer and an indirect object buffer are also available.)

Besides the distinction between reflexive and personal pronoun, and the subject and object grammatical functions, the encoding of the grammatical features expressed by pronouns is crucial for determining co-reference. Consider the expressions
John likes Mary. **She** is nice.
John likes Mary. **He** is nice.

The co-reference of the pronoun *she* with *Mary* in the first sentence, and *he* with *John* in the second sentence depends on the gender match between *Mary* and *she*, on the one hand, and *John* and *he*, on the other hand. Based on an examination of pronouns, the following grammatical features have been identified:

- **Number** – *he* vs. *they*
- **Animacy** – *he* vs. *it*
- **Gender** – *he* vs. *she* vs. *it*
- **Person** – *he* vs. *I* vs. *you*
- **Case** – *he* vs. *him*
- **Possessive** – *he* vs. *his*

In addition to these pronoun based distinctions, the distinction between definiteness is also important for co-reference.

- **Definiteness** – *the man* vs. *a man*

Examples where most of these grammatical features are definitive for co-reference determination are shown below:

- **John likes books. They are funny.**
  - they ← books (plural)
  - he ← John (singular)

- **John likes the book. It was funny.**
  - the book ← it (inanimate)
  - John ← he (animate)

- **John likes Mary. She is funny.**
  - Mary ← she (female)
  - John ← he (male)

- **I like John. He is funny.**
  - John ← he (3rd person)

- **I like you. You are funny.**
  - l ← I (1st person)

- **John read a book. The book was funny.**
  - a book ← the book (indef ← definite)

At a minimum, these grammatical features are needed for resolving the co-reference of pronouns and definite descriptions.

The case and possessive grammatical features serve other grammatical purposes. In some grammatical treatments, the possessive grammatical feature is treated as a kind of case – called genitive case in contrast to subjective and objective case. In Double-R Grammar, case is a viewed as a grammatical feature of pronouns functioning as the subject and object of situation referring expressions (and as object of oblique referring expressions headed by prepositions). The possessive grammatical feature does not have a comparable function, and there are actually two kinds of possessive pronoun:

- **John likes my book**
- **John likes mine**
The possessive pronoun *my* behaves like a determiner in functioning as the specifier of an object referring expression, except that it is also an object referring expression. The function of the possessive pronoun *mine* is even more complex. In order to resolve the co-reference of expressions involving possessive pronouns, two object referring expressions must be represented – both projected by the possessive pronoun.

The need for two object referring expressions becomes apparent when we consider discourse:

\[
\begin{align*}
I & \text{ want } [\text{her book}]. \text{ It is funny. } \\
I & \text{ want } [\text{hers}]. \text{ It is funny. } \\
I & \text{ want } [\text{her book}]. \text{ She has already read it. }
\end{align*}
\]

In the first example, the pronoun *it* is co-referential with the object referring expression *her book*. In the second example, *it* is co-referential with an unexpressed object referring expression that *hers* implies. This object referring expression is lacking a type specification, but a generic object type is nonetheless implied. Note, in particular, that the implicit animacy feature is inanimate and not human. Unlike personal pronouns like *he*, which project a single object referring expression in which they are instantiated as the head, possessive pronouns project an object referring expression in which they are instantiated as the head – supporting reference to an object which corresponds to the possessive pronoun, and the resulting possessive object referring expression (called a possessive object specifier) in turn projects an object referring expression construction in which it is instantiated as the specifier. In the first example, with *her book*, the noun *book* is instantiated as the head of the higher level object referring expression construction. In the second example, the head of the higher level object referring expression is implied. The type specification for the higher level object referring expression is either a generic object type or is determined by the actual referent. Nothing explicitly in the linguistic input represents this, nor is it grammatical to do so:

\[I \text{ want hers object}\]

Possessive pronouns differ from personal pronouns in projecting a second object referring expression construction, functioning as the specifier of that second construction, and not encoding the case grammatical feature. However, they also behave in some respects like personal pronouns. For example, they encode the person, number, animacy and gender grammatical features that are characteristic of personal pronouns. To capture these similarities, they both inherit from the pronoun super type.

Now that grammatical features have been introduced, we can consider how co-reference resolution is managed in the processing of the following discourse:

\[John \text{ likes Mary. She is nice.}\]

The entries in the mental lexicon for *John, Mary* and *she* are shown below:

\[
\text{John}: \eta_{pn1} \cdot \text{prop-noun(pn1)} \land \text{index(pn1, John)} \land \text{word(pn1, John)} \land \text{def(pn1, def)} \land \\
\text{anim(pn1, human)} \land \text{gen(pn1, male)} \land \text{num(pn1, sing)} \land \text{pers(pn1, 3rd)}
\]
When \textit{she} is processed, an object referring expression with head \textit{she} and \( \lambda \) variable \( o_3 \) is projected and placed in the matrix subject buffer. Its grammatical features are compared against the grammatical features of the object referring expressions in the discourse subject and discourse object buffers. Since the features of the object referring expression in the discourse object buffer match the object referring expressions are shifted from the matrix subject to the discourse subject and matrix object to discourse object buffers. Since the features of the object referring expression in the discourse subject buffer match the grammatical features of the object referring expressions in the discourse subject buffer, their \( \lambda \) variables are set equal (i.e. \( o_2 = o_3 \)), indicating co-reference.

If the input had been

\textit{John likes Mary. He thinks she is nice.}

the object referring expression headed by \textit{he} in the matrix subject buffer would match the object referring expression in the discourse subject buffer and their \( \lambda \) variables would be set equal (i.e. \( o_1 = o_3 \)). The pronoun \textit{she} also occurs in this example. The object referring expression headed by \textit{she} with \( \lambda \) variable \( o_4 \) functions as the subject of the embedded clause \textit{she is nice}. To support determination of co-reference from object referring expressions in embedded clauses, embedded subject, object and indirect object buffers are provided. In this example, the object referring expression headed by \textit{she} is placed in the embedded subject buffer. Since the grammatical features of the object referring expression in the embedded subject buffer match the grammatical features of the object referring expression in the discourse object buffer, their \( \lambda \) variables are set equal (i.e. \( o_2 = o_4 \)).

Double-R Grammar provides grammatical function specific buffers for the subject, object and indirect object at the discourse, matrix, and embedded clause levels. In addition, a stack of the three most recent object referring expressions is provided to support co-

\[
Mary: \eta p_{n_2} \cdot \text{prop-noun}(p_{n_2}) \land \text{index}(p_{n_2}, Mary) \land \text{word}(p_{n_2}, Mary) \land \text{def}(p_{n_2}, \text{def}) \land \text{anim}(p_{n_2}, \text{human}) \land \text{gen}(p_{n_2}, \text{female}) \land \text{num}(p_{n_2}, \text{sing}) \land \text{pers}(p_{n_2}, \text{3rd})
\]

\[
she: \eta pp \cdot \text{pers-pron}(pp) \land \text{index}(pp, she) \land \text{word}(pp, she) \land \text{def}(pp, \text{def}) \land \text{anim}(pp, \text{human}) \land \text{gen}(pp, \text{female}) \land \text{num}(pp, \text{sing}) \land \text{pers}(pp, \text{3rd})
\]
reference determination in the context of modifiers and possessive object specifiers. For example, in

*John likes Mary’s book. She lent it to him.*

doing the object referring expression *Mary’s* – called a possessive object specifier – which is headed by *Mary* and functions as a specifier of the object referring expression headed by *book* is retained in the most recent object referring expression stack. This supports the establishment of co-reference between the object referring expressions headed by *she* and *Mary*, even though *Mary* is not in a grammatical function buffer. It is important to note that the object referring expression headed by *book* (i.e. *Mary’s book*) is also in the most recent object referring expression stack. However, the grammatical features of this object referring expression do not match. It is also important to note that the object referring expression *Mary’s book* is in the discourse object buffer as well as the most recent object referring expression stack. Once the most recent object referring expression stack contains three referring expressions, addition of a new object referring expression expels the oldest object referring expression from the stack, limiting the number of object referring expressions that are available in working memory. The combination of three levels of grammatical function specific object referring expression buffers and a three buffer stack of the most recent object referring expressions is adequate to handle many, but not all, examples of nominal co-reference. Additional mechanisms that support the retrieval of object referring expressions from declarative memory are needed to handle more complex examples, but have not yet been implemented.

The proper treatment of plural and conjoined object referring expressions has been a challenging area of research in predicate logic. When we consider co-reference, the challenges increase. A simple discourse like

*John and Mary laughed. They were happy.*

is problematic since typical logical representations for conjoined names like

\[
\text{laugh}(J) \land \text{laugh}(M)
\]

provide no variable or constant that the plural pronoun *they* can be co-referential with. A recent variant of DRT (Kamp, van Genabith & Reyle, 2011) addresses this problem by allowing the introduction of a discourse referent for the conjunction in the context of *they*. This discourse referent is subcategorized as plural (e.g. \(x^\text{pl}\)) so that the plural discourse referent representing the plural pronoun *they* is compatible with it. The introduction of this plural discourse referent is necessary, because the initial logical representation does not encode any logical relationship between *John* and *Mary*. However, grammatically, *John* and *Mary* are conjoined. Double-R Grammar encodes this grammatical relationship explicitly, encodes a plural number feature, and provides a handle variable for the conjunction. This handle variable and the plural number feature support co-reference with *they*. There is no need to introduce a \(\lambda\) variable when *they* is processed, as in the DRT approach, since it is already available.
Although Kamp, van Genabith & Reyle (2011) allow plural discourse referents for expressions like *John and Mary*, they argue against doing so for quantified plural expressions like

*some boys*

For all quantified expressions, a non-plural discourse referent is introduced. This has the unfortunate side effect of forcing the discourse referent for the pronoun *they*, which is co-referential with *some boys* in the following discourse, to also be non-plural:

*Some boys laughed. They were happy.*

In Double-R Grammar, we avoid this kind of reinterpretation of the grammatical characteristics of object referring expressions, since such reinterpretations complicate the creation of fully explicated logico-grammatical representations (i.e. with grammatical features), as well as the mapping into the situation model (cf. Ball, 2010). A plural object referring expression is assumed to refer to a collection of some sort in all its uses, including uses involving quantifiers. That’s why it is plural. Kamp, van Genabith & Reyle (2011) only accept the introduction of plural discourse referents reluctantly, since they result in a significant expansion of the power of the logical system of representation. In Double-R Grammar they are embraced.

The buffers available in Double-R Grammar make it possible to handle co-reference given expressions like

*If he reads the book, John will like it*

in which the pronoun *he* occurs before the antecedent *John*. At the end of processing of the antecedent clause *he reads the book*, the object referring expression headed by *he* will be in the embedded subject buffer, and the object referring expression *the book* will be in the embedded object buffer. That *he reads the book* is an embedded clause is grammatically indicated by the subordinating conjunction *if*. When *John* is processed, an object referring expression is projected and placed in the matrix subject buffer. At this point, co-reference between *John* and *he* can be established. Likewise, when *it* is processed, an object referring expression is projected and placed in the matrix object buffer. Since the object referring expression *the book* is in the embedded object buffer, it is accessible and co-reference between *it* and *the book* can be established.

There is an assumption underlying Double-R Grammar that humans are capable of learning how to buffer needed information. This is an important element of expertise. It is also the basic mechanism for efficiently handling *long-distance dependencies* in Double-R Grammar – co-reference being a prime example. The grammatical function specific buffers and the object referring expression stack that are available in Double-R Grammar are not assumed to be innate. They are learned. The assumption that humans can learn how to buffer information is especially important in that it suggests an ability to modify the cognitive architecture, which is typically assumed to be comprised of only the invariants of human cognition (cf. Anderson, 2007).
Although this section is focused on the treatment of co-reference between object referring expressions, co-reference involving situation referring expressions is also possible. Consider the discourse

*John kicked the ball hard. That was how he relieved stress.*

It seems clear that the object referring expression projected by the demonstrative pronoun *that* is co-referential with the situation referring expression corresponding to *John kicked the ball hard*. Exactly how this co-reference is determined has not yet been worked out, but the association of a $\lambda$ variable with the situation referring expression makes this possible, so long as it can be determined that the object referring expression headed by the demonstrative pronoun *that* is co-referential with the preceding situation referring expression.

**Encoding Temporal Relations between Situation Referring Expressions in Predicate Logic with Extensions**

A primary relation between situation referring expressions is their temporal alignment. The temporal relationship between situation referring expressions may be expressed grammatically via grammatical features like tense (e.g. past, present) and aspect (progressive, perfect), or lexically via prepositions (e.g. *before, after*) and situation modifiers (e.g. *yesterday, tomorrow, now*).

Numerous temporal logics have been developed to represent the temporal relations between situation referring expressions. In this section we consider the DRT approach of Kamp, van Genabith & Reyle (2011) as a representative example. Kamp, van Genabith & Reyle note that the temporal relationships between situation referring expressions are essentially discourse level phenomena. The authors state that “the starting point for DRT was an attempt in the late seventies to come to grips with certain problems in the theory of tense and aspect” (Ibid. p. 71). In particular, temporal logics at that time provided no mechanism for referring to times, since the representation of tense was meta-linguistic. What this boils down to is that there were no variables corresponding to times that could be referenced within logical representations, despite the fact that English contains many expressions like

*on the first of February 2011* (Ibid. p. 72)

which appear to refer to times.

In Kamp, van Genabith & Reyle’s variant of DRT, discourse referents are introduced to support reference to times. Two important times that may need to be referenced are the utterance time and the time of the described situation. With discourse referents for these times, the different between past, present and future tense can be represented as (Ibid. p. 75)

\[
\begin{align*}
t & \lessdot n \text{ (past tense)} \\
t & = n \text{ (present tense)} \\
n & \lessdot t \text{ (future tense)}
\end{align*}
\]
where \( t \) corresponds to the described situation time interval, \( n \) corresponds to the utterance time interval (i.e. now), and the symbol \( \prec \) indicates before in time.

In addition, Kamp, van Genabith & Reyle add discourse referents for the eventualities that reference these times. The term eventuality comes from Bach (1981), and encompasses events, states and processes. It corresponds to the term situation in Double-R Grammar. Kamp, van Genabith & Reyle simplify somewhat by treating processes as states. They also introduce a predicate PROG that transforms an event into a state. This predicate corresponds to the progressive verb form as in

\[ \text{The man was pulling out a gun} \]

which they represent with the condition

\[ s: \text{PROG}(\wedge e: \text{"pull-gun"}(x)) \] (Ibid. p. 80)

In this condition, \( s \) is a state discourse referent, \( e \) is an event discourse referent (the \( \wedge \) indicates an intensional reference), and “pull-gun” is a simplified predicate representation for the expression pull out a gun. In this condition, PROG transforms the event “pull-gun” into a state (in their terminology) which might be paraphrased as “pulling-gun”. The past tense + progressive aspect that is expressed by was pulling is represented by the following conditions:

\[
\begin{align*}
  t & \prec n \\
  t & \subseteq s
\end{align*}
\]

As noted above, the first condition indicates past tense. In the second condition, \( s \) is the state discourse referent introduced above. The second condition indicates that the time interval \( t \) occurs within the period that the state \( s \) holds (\( t \subseteq s \)). Since time \( t \) is before now, the state \( s \) held in the past, but may extend to the present. Together, these two conditions represent the past progressive status of was pulling (out a gun).

Although Kamp, van Genabith & Reyle represent progressive aspect as a predicate that transforms an event into a state, they represent perfect aspect, which is traditionally viewed as indicating the completion or boundedness of an event (in contrast to progressive aspect), in a manner similar to past tense. Consider the past perfect expression

\[ \text{The man had pulled out a gun} \]

In this expression, the past tense is indicated by had, and perfect aspect is indicated by pulled (or had pulled) – where pulled is the past participle and not the past tense of pull. To represent this, two time intervals are needed:

\[
\begin{align*}
  t_1 & \prec n \\
  t_2 & \prec t_1 \\
  e & \subseteq t_2
\end{align*}
\]
In this case, the discourse referent $e$ corresponds to an event which occurs within the time interval $t_2$, where $t_2$ is before $t_1$ which is before $now$. This may make more sense in the following context:

*Before John turned around, the man had pulled out a gun.*

In this discourse, the event of *John turning around* occurred before *now* and the event of *the man pulling out a gun* occurred before *John turning around*. This view of the past perfect is commonly called a past-in-the-past.

We have only scratched the surface in the discussion in this section. The important message is the need to introduce discourse referents for times, events, and states, as well as special aspectual predicates (e.g. PROG), to more fully model tense and aspect.

**Encoding Temporal Relations between Situation Referring Expressions in Double-R Grammar**

There is currently no theory or computational implementation of the temporal relationships between situation referring expressions in Double-R Grammar. What Double-R Grammar does provide is a representation of the grammatical features expressed by verbs, including tense and aspect. These features are needed in any theory or computational implementation. Double-R Grammar also encodes the referential type of referring expressions, including a value of **time** for temporal object referring expressions (e.g. *tomorrow, last week*) and a corresponding semantic type of **time** for words which head temporal object referring expressions (e.g. *tomorrow, last week*). However, as Kamp, van Genabith & Reyle note, determining the temporal relationship between situation referring expressions requires consideration of fine-grained semantic and pragmatic knowledge which is not currently encoded in Double-R Grammar. In this respect, they cite three examples of temporal relations between situation referring expressions (Ibid. 81):

*Chris had a fantastic meal. He ate salmon.* (overlapping in time)
*Max fell. John pushed him.* (pushing occurs before falling)
*John turned the light off. The room was pitch black.* (turning off light occurs first)

Grammatically, all these sentences are past tense, but the temporal relationships between the sentence pairs vary. Determining the temporal relationship between the pairs of sentences in each example requires fine-grained semantic knowledge (e.g. *salmon is something one eats during a meal, pushing someone may cause them to fall, turning off the lights may make a room dark*).

Double-R Grammar supports the representation of two grammatical features for tense and a single aspect grammatical feature. The first tense grammatical feature distinguishes between finite and non-finite verb forms. For finite verbs, the second tense feature distinguishes between present tense and past tense. (In the previous discussion, these two features were combined.) There is no grammatical feature for future tense in Double-R Grammar. The tense features of the verb are projected to the encompassing situation referring expression (sre):
John wants to kick the ball – kick is non-finite
- $\lambda\text{srt. sre}(\text{srt}) \land \text{tense-fin}(\text{srt}, \text{non-fin})$

John kicks the ball – kicks is finite, present tense
- $\lambda\text{srt. sre}(\text{srt}) \land \text{tense-fin}(\text{srt}, \text{fin}) \land \text{tense}(\text{srt}, \text{pres})$

John kicked the ball – kicked is finite, past tense
- $\lambda\text{srt. sre}(\text{srt}) \land \text{tense-fin}(\text{srt}, \text{fin}) \land \text{tense}(\text{srt}, \text{past})$

The lack of a future tense grammatical feature is reflected in the treatment of all modal auxiliaries as encoding present tense, including the modal auxiliary will. Assuming that sentences with modal verbs encode a tense feature, the grammatically most consistent value is present tense:

John can go (*yesterday, now, tomorrow)
John could go (yesterday, now, tomorrow)
John may go (*yesterday, now, tomorrow)
John must go (*yesterday, now, tomorrow)
John will go (*yesterday, now, tomorrow)
John would go (?yesterday, now, tomorrow)
John shall go (*yesterday, now, tomorrow)
John should go (*yesterday, now, tomorrow)

All these modal auxiliaries are compatible with both present and future time, which is expressed in English by the present tense (which might be better called non-past). The modal auxiliary could also allows a past time reading, perhaps reflecting the historical present vs. past tense grammatical contrast between can and could, will and would, and shall and should. Although all modal auxiliaries encode a present tense grammatical feature, they differ in terms of modality. In Double-R Grammar, the value of the modality feature is the modal auxiliary itself. For the sentence above with can (showing only the situation referring expression, tense and modality), we have

- $\lambda\text{srt. sre}(\text{srt}) \land \text{tense-fin}(\text{srt}, \text{fin}) \land \text{tense}(\text{srt}, \text{pres}) \land \text{modality}(\text{srt}, \text{can})$

Grammatical features function like logical operators. However, they are represented as predicates whose first argument is a variable corresponding to the situation referring expression over which they take scope, and whose second argument is the value for the grammatical feature.

The grammatical evidence described above suggests that future time is not encoded as a grammatical contrast that should be encoded as a grammatical feature. It is part of the meaning of words like yesterday, now and tomorrow, or perhaps will, that they express past time, present time and future time. How this meaning should be grammatically represented is an open research question, but it is likely to require the use of temporal variables along the lines of Kamp, van Genabith & Reyle (e.g. $t_{sre} \prec \text{now}$ equals past time, $t_{sre} = \text{now}$ equals present time, $\text{now} \prec t_{sre}$ equals future time). Double-R Grammar currently encodes a time semantic type in the lexical entries for yesterday, now and tomorrow, but does not characterize this semantic type further. That characterization is part of the fine-grained meaning of these words.
As noted above, Double-R Grammar provides a single grammatical aspect feature. This feature distinguishes between progressive and perfect aspect:

\begin{align*}
\text{John is } & \text{ kicking the ball} \quad – \text{kicking is progressive aspect} \\
\lambda \text{sit. sre(sit)} & \land \text{aspect(sit, prog)} \\
\text{John has } & \text{ kicked the ball} \quad – \text{kicked is perfect aspect} \\
\lambda \text{sit. sre(sit)} & \land \text{aspect(sit, perf)}
\end{align*}

Whereas progressive aspect indicates the continuation or lack of bounds of an action, perfect aspect indicates its completion or boundedness.

Note that there are two verbs in these examples, an auxiliary verb and a main verb. The auxiliary verb encodes the tense grammatical features and the main verb encodes the aspect grammatical feature. Both tense and aspect get projected to the encompassing situation referring expression:

\begin{align*}
\lambda \text{sit. sre(sit)} & \land \text{tense-fin(sit, fin)} \land \text{tense(sit, pres)} \land \text{aspect(sit, prog)} \\
\lambda \text{sit. sre(sit)} & \land \text{tense-fin(sit, fin)} \land \text{tense(sit, pres)} \land \text{aspect(sit, perf)}
\end{align*}

There is one verbal construction that the tense and aspect grammatical features in Double-R Grammar cannot handle:

\begin{align*}
\text{John has been kicking the ball} \quad – \text{present + perfect + progressive}
\end{align*}

If \textit{been} expresses perfect aspect and \textit{kicking} expresses progressive aspect, then there are two aspects expressed and only one grammatical feature available to represent the two values.

\begin{align*}
\#\lambda \text{sit. sre(sit)} \land \text{tense-fin(sit, fin)} \land \text{tense(sit, pres)} \land \text{aspect(sit, perf)} \land \text{aspect(sit, prog)}
\end{align*}

Treating \textit{been} as expressing a perfect tense does not solve this problem, since there will be a conflict between present and perfect tense.

\begin{align*}
\#\lambda \text{sit. sre(sit)} \land \text{tense-fin(sit, fin)} \land \text{tense(sit, pres)} \land \text{aspect(sit, perf)} \land \text{aspect(sit, prog)}
\end{align*}

With a single aspect grammatical feature and a single (finite) tense grammatical feature, we either need a composite present-perfect tense or perfect-progressive aspect. Since it is unclear what a combined perfect (bounded) plus progressive (unbounded) aspect could mean, a composite present-perfect tense that functions much like a past tense is preferred. Note that neither this example, nor

\begin{align*}
\text{John was } & \text{ kicking the ball} \quad – \text{past + progressive}
\end{align*}

necessarily entail completion or boundedness of the action, although both imply it. The unboundedness of the progressive \textit{kicking} weakens the boundedness of the past tense and perfect aspect. However, allowing the progressive aspect of \textit{kicking} to simply override the perfect aspect of \textit{been} fails to capture the present-perfect meaning of \textit{has been}.
Treating perfect as a composite form of tense in expressions with *has been* (and *had been*) does not mean it should never function as an aspectual feature. In the expressions

\[
\text{John is } \textit{kicking} \text{ the ball} \quad \text{present + progressive}
\]
\[
\text{John has } \textit{kicked} \text{ the ball} \quad \text{present + perfect}
\]

The contrast between *kicking* (unbounded) and *kicked* (bounded) is treated as a contrast in aspect in Double-R Grammar.

The representation of temporal relations between situation referring expressions in Double-R Grammar is a work in progress. Double-R Grammar supports the representation of the grammatical features tense and aspect for verbs, along with a time semantic type for words and a time referential type for referring expressions. However, Double-R Grammar does not show how this grammatical information is used to compute the temporal relationships between situation referring expressions.

### The Model and Possible Worlds in Predicate Logic with Extensions

Standard model-theoretic predicate logic uses a set-theoretic model of the real world to support the interpretation of logical expressions. The set-theoretic model consists of a domain of atomic individuals, sets of individuals corresponding to properties that are true of the individuals, and ordered tuples of individuals corresponding to relations that hold between individuals. The interpretation of logical operators is also expressed in set-theoretic terms. The conjunction of two logical expressions is interpreted as set intersection. The disjunction of two logical expressions is interpreted as set union. Negation is interpreted as set difference. Existential and universal quantification are interpreted as quantification over the individual elements of sets.

The model is intended to be a complete model of the real world. It contains all the individuals in the real world and sets corresponding to all the properties and relations in which individuals and ordered tuples of individuals participate. On the other hand, it does not contain any imaginary individuals. For example, there are no individuals corresponding to unicorns in the model, since unicorns do not exist in the real world. In the model, there is a single individual corresponding to *the morning star* and *the evening star*, where both these expressions describe the planet Venus which is not a real star. Since humans use expressions like *the unicorn* to refer to imaginary unicorns and may not realize that the morning star and the evening star refer to the same planet and are not stars, it is difficult to explain this cognitive behavior in terms of the model.

To address the fact that *the morning star* and *the evening star* differ in meaning, if not in reference, Frege (1992) introduced an important distinction between sense and reference. The sense of these expressions is what allows them to pick out or denote the same individual, even though they differ in meaning.

Whereas Frege assumed that “properties” like MAN and DOG denoted sets of individuals, he assumed that logical sentences denoted truth values, at least in extensional contexts. For example, the denotation of the logical sentence corresponding to
John is a man -> man(J)

is the value true if the individual denoted by John is a member of the set of men, and false otherwise. However, in intensional contexts, Frege assumed that the denotation of a logical sentence is its sense. For example, in

John believes Mary likes Bill

The denotation of Mary likes Bill is its sense and not a truth value. A treatment along these lines is necessary since Mary may or may not actually like Bill in the real world and we do not want to say that this sentence means that John believes true or John believes false. The two possible values for the reference of a logical sentence are inadequate to capture meaning in intensional contexts.

In possible worlds semantics, the distinction between sense and reference is transformed into a distinction between intension and extension which is made explicit in terms of possible worlds. The intension of a logical sentence is the set of possible worlds in which the extension of the logical sentence denotes the value true. Possible worlds are complete models that model alternative worlds, only one of which is the real world. With the introduction of possible worlds it is possible to model imaginary objects like unicorns by allowing them to denote individuals in possible worlds other than the real world. It is also possible to have a possible world in which the morning star and the evening star denote different individuals. Possible worlds also make it possible to represent the meaning of modal expressions like

John may go to the store

Such expressions can be represented in terms of possible worlds in which John goes to the store denotes the value true. Finally, intensional contexts can also be handled in terms of possible worlds. For example, in the world of John’s beliefs, Mary likes Bill denotes the value true, whether or not Mary likes Bill denotes the value true in the real world.

We can extend the support for intensional contexts in a possible world semantics by making situations and propositions first class entities - i.e. by treating them as individuals. If we make situations and propositions first class entities, then there will be a quantifiable variable associated with the situation of John believing that Mary likes Bill and a quantifiable variable associated with the proposition that Mary likes Bill. With these variables, we can represent the sentence John believes Mary likes Bill as

∃s∃p[believe(s, J, p) ∧ like(p, M, B)]

where s = situation variable and p = proposition variable. What does this mean in terms of a possible worlds semantics? The expression is true in those worlds in which John believes p is true and p = Mary likes John is either true or false. (Some approaches add a constraint that the proposition Mary likes John cannot be false in worlds that are compatible with John’s beliefs.) How is the referent (or denotation) of the proposition variable to be treated? Either the referent of p is just another individual in an untyped domain (or set of individuals), or individuals are sorted into types such that the referent of p is an individual
proposition (or set of individual propositions). If we assume the latter, do individual propositions have internal structure in possible worlds? This is an unanswered question, even in Situation Semantics (cf. Kratzer, 2014) where situations and propositions are treated as first class entities. If propositions do have internal structure in possible worlds, how is that structure represented without exceeding the capabilities of set theory (or multi-sorted logic on top of set theory now that we are assuming sorts)? If they don’t have internal structure, doesn’t the representation of a proposition as an atomic individual abstract away from much of the meaning of the proposition? And how is the atomic proposition associated with the arguments that it relates? Is the set of ordered pairs of individuals corresponding to liking events in a world the same set or a different set from the set of atomic liking individual propositions in that world? Presumably they are different sets, since they are different sorts (i.e. ordered pairs of individual objects vs. individual propositions). These are just some of the issues that arise in the treatment of propositional attitudes in model-theoretic predicate logic with possible worlds.

**The Situation Model and the Mental Universe in Double-R Grammar**

The term *situation model* is borrowed from the psycholinguistic research of van Dijk and Kintsch (1983). It originally referred to a propositional representation of a text, elaborated with inferences derived from the text. More recently, situation models have been argued to include non-propositional representations with imaginal characteristics (Zwann & Radvansky, 1998), aligning with Barsalou’s perceptual symbol systems (Barsalou, 1999) in this respect.

The situation model in Double-R Grammar is a non-linguistic, conceptual stage on which objects and situations are created, referenced and modified during discourse processing. At the beginning of a discourse, the stage is empty. As language is processed, the stage becomes populated. Some of the objects and situations on the stage may refer to objects and situations in the mental universe of experience of an individual, in which case they may be perceptually grounded. The mental universe is the repository of all the experiences of an individual accumulated during a lifetime, including the current experience that is contemporaneous with the evolving situation model. It contains many objects and situations. There is no notion of the actual world in Double-R Grammar, just the world (or universe) of experience of an individual (Ball, 2012), although it is assumed that the experienced world mirrors the actual world, constrained by the capabilities of human perception and cognition. Cognitively speaking, the situation model is actively maintained in working memory, whereas the mental universe resides in long-term memory.

One view of the situation model in Double-R Grammar is that it represents both the sense and reference of the linguistic expressions from which it is dynamically constructed. There are referents in the situation model corresponding to the referring expressions in the linguistic input. There are also representations of the relations between those referents. From these relational representations, it is possible to create sets of individuals with various properties, and ordered tuples of individuals who participate in various relations, but that is not the end game.
From the perspective of predicate logic, non-linguistic situation model representations (ignoring their imaginal characteristics and perceptual grounding) look very much like logico-semantic representations, and the logico-grammatical representations look like syntactic representations, albeit syntactic representations that are closely aligned with logico-semantic representations. We do not subscribe to the view that Double-R Grammar’s logico-grammatical representations are syntactic, although they are grammatical. Viewing them as syntactic does simplify the alignment of Double-R Grammar with many predicate logic approaches. On this view, the mapping from logico-grammatical representations to situation model representations is a translation from syntax to semantics, and the associations between situation model representations and the mental universe correspond to the interpretation of logical representations in a model.

An alternative perspective, and the one we prefer, is that model representations need to be richly structured, eliminating set theory as an adequate representational formalism, by itself. There is some recognition of the need for structured representations to model propositional attitudes and other intensional contexts. For example, Cresswell (1985) introduces structured meanings as a way of dealing with propositional attitudes (Israel, 1987). Lewis (1972) proposes that senses correspond to structured representations (i.e. syntactic representations) in which the terminal nodes (i.e. words) are replaced with their intensions. Situation model representations in Double-R Grammar are derived from logico-grammatical representations with which they are closely aligned. Situation model representations are richly structured, and that structure is not limited to intensional contexts.

The combination of the situation model and mental universe supports a distinction between referential and attributive uses of object referring expressions in Double-R Grammar (cf. Donnellan, 1966), although the term attributive is somewhat problematic. Consider the expression

John’s murderer, whoever he is, is insane

The definite object referring expression John’s murderer, refers to a person in the situation model, and the predicate is insane asserts a property of that person. However, the expression whoever he is makes it clear that reference to John’s murderer is actually indefinite, even though the expression John’s murderer, by itself, is definite (i.e. the definite reference point John’s makes the overall expression definite). Since both John’s murderer and whoever refer to the same person, the indefiniteness of whoever overrides the definiteness of John’s murderer. The indefiniteness of whoever indicates that there is not another referent in the situation model to which John’s murderer is co-referential. Further, it indicates that there is no individual in the mental universe that is associated with the referent of John’s murderer, either. Nonetheless, once a referent is introduced into the situation model, that referent can be referred to by the object referring expression headed by the pronoun he, via co-reference with John’s murderer and whoever.

If the expression had been

John’s murderer, Bill, is insane
There is a strong implication that the referent of Bill is either available in the situation model or is salient in the mental universe. If Bill is only available in the situation model, then this reference continues to be attributive. The same is true for the pronoun he above. Co-referential attributive uses remain attributive. However, if Bill is associated with someone in the mental universe, then the use is referential in the sense of Donnellan (1966). Further, attributive uses may be ground in the mental universe after the fact:

*John’s murderer, whoever he is, is insane. Bill did it.*

The combination of a situation model and mental universe is not sufficient to handle propositional attitudes and modal expressions, like those discussed above, without adding more structure. To represent propositional attitudes, a capability to create subspaces appears to be needed. Given the expression

*John believes Mary likes Bill*

creation of a belief subspace for John’s beliefs, in which Mary likes Bill is suggested. This is the approach espoused by Fauconnier (1994) in his Mental Spaces theory, which is a key subtopic of Cognitive Linguistics. Unlike possible worlds which are complete worlds, mental spaces are partial worlds which go hand in hand with the partial nature of the situation model (and the situations of Situation Semantics). In fact, the situation model can be viewed as the base space within a mental spaces treatment.

With the introduction of mental spaces, we are now in a position to consider the examples discussed in the previous section. The referring expressions the morning star and the evening star may have distinct referents in the situation model and even in the mental universe, even though they refer to the same planet in objective reality. Humans may simply not know that they refer to the same planet. Imaginary objects like unicorns and golden mountains may also have referents in the situation model and mental universe. If a human knows they are imaginary objects, that knowledge may be represented using a predicate like Hobbs’ (2003) rexists (really exists):

∃x[unicorn(x) ∧ ¬rexists(x)] (i.e. there is a unicorn that doesn’t really exist)

As discussed above, intensional contexts can be handled via the introduction of mental spaces for the beliefs of an individual. That leaves modal expressions like:

*John may go to the store*

Does the introduction of mental spaces also support the representation of modal expressions, especially possibility and necessity? In possible world semantics, possibility and necessity are defined across worlds. A proposition is possible if it is true in at least one world. A proposition is necessary if it is true in all worlds. Although there can be more than one subspace in a mental spaces approach, there is nothing equivalent to the unbounded number of possible worlds in possible worlds semantics. It might be possible to tag a subspace as a possibility subspace, but it is not entirely clear what this means. It is even less clear what it would mean to tag a subspace as necessary. Given these challenges, we
will leave the representation of possibility and necessity, and other modalities, as an open research question.

The situation model is populated with more kinds of things than is allowed in standard set-theory based model-theoretic semantics. At a minimum, there are kinds of objects and collections of objects in addition to individuals. In addition to objects, situations are also represented as structured individuals, and include kinds of situations and collections of situations. Allowing situations to be structured individuals means that individuals need not be atomic. Situations are composed of objects and relations between objects. Properties are situations involving a single object. The standard predicate logic representation of situations as sets of ordered tuples of individuals fails to capture much of the meaning of situations. In Double-R Grammar, the word *run* does not denote the set of all individuals who run, and the word *kick* does not denote the set of ordered pairs of individuals who kick and are kicked. Rather, *run* denotes running events, and when used as a predicate in a situation referring expression, a specific running event. Likewise for *kick*. The set theoretic view of the meaning of relations in terms of ordered tuples of individuals who participate in relations, is foreign to Double-R Grammar. Relations combine with their arguments to refer to situations, where those situations involve individuals, but relations are not defined in terms of the individuals. In Double-R Grammar properties and relations, and the situations they refer to in combination with their arguments, are first class, structured entities in the situation model.

Most predicate logic approaches which add additional types of things either treat them as ordinary atomic individuals in order to retain a set-theoretic model, or allow them to be sorted, and use multi-sorted logic on top of set-theory for the model. Link (1983) introduces a semi-lattice into the model to support the representation of plural and mass individuals. However, Link uses the semi-lattice to generalize the distinction between plural, mass and single individuals, rather than treating them as distinct sorts. Chierchia (1988) extends set theory to property theory to better represent the intensionality of properties, relations and propositions. As noted above, Cresswell (1985) introduces structured meanings as a way of dealing with propositional attitudes (but ultimately explicates these structured meanings in set-theoretic terms), and Lewis (1972) treats senses as structured representations with the intensions of words as the terminal nodes. Double-R Grammar goes beyond these logical approaches in assuming a rich ontology of entities and inherently structured representations in accord with its roots in cognitive linguistics and conceptual semantics. In this respect, Double-R Grammar aligns with Hobbs (1985) ontologically promiscuous approach to meaning representation and is inspired by the philosophical musings of Meinong. Double-R Grammar also assumes that situation model representations have imaginal characteristics and are ground in a perceptual chain of experience. However, a computational implementation of these imaginal and perceptual characteristics remains elusive.

**Conclusions**

This paper presents a logical description of Double-R Grammar. That description is unique in various respects. Most importantly, the logico-grammatical representations capture the
full grammatical structure of the linguistic input. They do not abstract away from that structure. The basic claim is that representing most, if not all, of the grammatical structure is important for an adequate semantic representation. Another important representational difference is that the representations are linguistic. They contain actual words and not abstract concepts (or their logical equivalents). They are object language representations, not meta-language representations. Since words and grammatical structures can be ambiguous, the logico-grammatical representations cannot be complete representations of meaning. They are not. These logico-grammatical representations are mapped into situation model representations which encode the objects and situations which are the referents of the referring expressions in the logico-grammatical representations. This mapping resolves many of the ambiguities inherent in the logico-grammatical representations. It is the combination of the logico-grammatical representations with situation model representations (and associations to the mental universe) which constitutes the representation of meaning in Double-R Grammar. Situation model representations encode both the reference and the sense of the logico-grammatical representations, necessitating a more powerful representational formalism than set theory. Situation model representations are structured conceptual representations. To large extent, the structure of these conceptual representations is mirrored in the linguistic representations from which they are derived, but also in the perceptual experience of the objects and situations described by the linguistic expressions. How this conceptual integration happens in the brain is an open research question, but the research of Corso and colleagues (cf. Kumar, Dhiman & Corso, 2014) suggests one possibility based on the composition of sparsely coded multi-modal representations.

From a practical perspective, the logico-grammatical representations are derived from ACT-R chunk (i.e. frame) representations in the following way: a handle variable for the chunk is created. For each slot/value pair of the chunk, a two argument predicate is created where the slot name is the predicate, the first argument of the predicate references the chunk handle variable, and the second argument is the slot value. The predicates created in this manner are conjoined together. The chunk handle variable ties the predicates together, distinguishing them from predicates created from other chunks. For this to work, a large number of handle variables must be created. The essential function of these handle variables is to capture structure in what is otherwise a flat conjunction of predicates. This approach has roots in Davidson’s introduction of event variables to support reference to events within logical representations, in neo-Davidsonian extensions, and in Minimal Recursion Semantics which adds handle variables for all elementary predications to support (under)specification of quantifier scope. Indeed, the trend in compositional semantics is toward the introduction of more and more variables to support reference to more and more kinds of individuals, including worlds, times, propositions, situations, kinds, collections, etc. Double-R Grammar extends this trend to the introduction of variables to support the full representation of grammatical structure. The result is highly articulated logico-grammatical representations, as is obvious from the examples presented in this paper.

Double-R Grammar includes a computational implementation in the ACT-R cognitive architecture which demonstrates many of the representation and processing commitments.
The development of the computational implementation has informed Double-R Grammar in important ways, supporting formalization of the grammar and revealing many inconsistencies that had to be corrected. Of course, inconsistencies remain, but the computational implementation has attained relatively broad coverage of the core grammatical constructions of English, and contains a mental lexicon that is near the size of some estimates of the human mental lexicon, although the entries in the mental lexicon are impoverished as full representations of meaning (as well as lacking any phonologic or phonetic representations).

Having attained relatively broad coverage of the core grammatical constructions of English, we are now in a position to begin to consider the representation and processing of the fine-grained semantic and pragmatic knowledge that is needed for an adequate compositional as well as lexical semantics. That fine-grained knowledge is largely encoded in the meaning of words and expressions. It is not grammatically encoded and is, hence, not fully compositional. Complete representations of meaning must ultimately integrate grammatically encoded compositional, non-compositional and lexically encoded meaning. We look forward to pursuing this research objective.

References


