
1. English Mini-Fragment 1.

1.0 Introduction

In this handout we present a small and partial sample English grammar (a “fragment”, in MG terminology), that is, an explicit description of the syntax and semantics of a small part of English. A real fragment (of which this is just a “mini” or “preview” version) can serve several purposes: making certain aspects of formal semantics more explicit, including (and illustrating) more of the basics of the background logic. A reasonable-sized fragment is of interest in its own right in exploring the syntax and semantics of English. This mini-fragment, with its very minimal lexicon, mainly just puts together what we’ve seen so far, making some of it slightly more explicit. If we don’t get through it all today, we’ll continue with it in Lecture 6 next Wednesday.

The semantics of the fragment will be given via translation into predicate logic supplemented with the iota-operator for definite descriptions. In later fragments in parts II and III we’ll have the added power of the lambda calculus and higher types. Here we’ll show in fine print some of the things we’re NOT including yet, including the kinds of adjectives that need to be interpreted as functions, and the treatment of quantified noun phrases in higher types. We’ll come back to those things later.

Now we introduce the fragment of English: first the syntactic categories and the category-type correspondence, then the basic syntactic rules and the principles of semantic interpretation, and then a small lexicon and some meaning postulates. In Section 2 we present some examples. Certain rules of the fragment are postponed to Section 3 where they receive separate discussion; these are rules that go beyond the simple phrase structure rule schemata of Section 1.

1.1. Syntactic categories and their semantic types.

<table>
<thead>
<tr>
<th>Syntactic category</th>
<th>Semantic type</th>
<th>Expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>t</td>
<td>sentence</td>
</tr>
<tr>
<td>ProperN</td>
<td>e</td>
<td>name (John)</td>
</tr>
</tbody>
</table>
1.2. Syntactic Rules and Semantic Rules.

Two different approaches to semantic interpretation of natural language syntax (both compositional, both formalized, and illustrated, by Montague):

**A. Direct Model-theoretic interpretation:** Semantic values of natural language expressions (or their “underlying structure” counterparts) are given directly in model-theoretic terms; no intermediate language like Montague’s intensional logic (but for some linguists there is a syntactic level of “logical form” to which this model-theoretic interpretation applies, so the distinction between the two strategies is not always sharp.) This is the direct “English as a formal language” strategy. For illustration, see Heim and Kratzer (1998). Also see the discussion in Larson’s chapter 12.

**B. Interpretation via translation:** Stage 1: compositional translation from natural language to a language of semantic representation, such as Montague’s intensional logic or, for now, into a slightly enriched version of first-order predicate logic. For an expression $\gamma$ of category C formed from expressions $\alpha$ of category A and $\beta$ of category B, determine $\text{TR}(\gamma)$ as a function of $\text{TR}(\alpha)$ and $\text{TR}(\beta)$. Stage 2: Apply the compositional model-theoretic interpretation rules to the intermediate language.

We will follow a mixture of the two strategies, using Predicate Logic slightly enriched as our logical language where we can, and giving model theoretic rules where predicate logic can’t express what we need to express.
Some abbreviations and notational conventions:

We will sometimes write $\alpha \forall$ as a shorthand for $\text{TR}(\alpha)$. And sometimes we use the category name in place of a variable over expressions of that category, writing $\text{TR}(A)$, or $A'$, in place of $\text{TR}(\alpha)$ when $\alpha$ is an expression of category A. And we will write some of our syntactic rules like simple phrase structure rules. Here is an example of a syntactic rule and corresponding translation rule, and their abbreviations as they will appear below.

**Official Syntactic Rule:** If $\alpha$ is an expression of category DET and $\beta$ is an expression of category CNP, then $F_0(\alpha, \beta)$ is an expression of category NP, where $F_0(\alpha, \beta) = \alpha \beta$.

**Official Semantic Rule:** If $\text{TR}(\alpha) = \alpha'$ and $\text{TR}(\beta) = \beta'$, then $\text{TR}(F_0(\alpha, \beta)) = \alpha'(\beta')$.

**Abbreviated Syntactic Rule:** NP $\rightarrow$ DET CNP

**Abbreviated Semantic Rule:** NP' = DET'(CNP')

### 1.2.1. Basic syntactic rules

#### Basic rules, phrasal:

- $S \rightarrow \text{NP } \text{VP}$
- $S \rightarrow \text{NP } \text{TV } \text{NP}$ (later we’ll introduce rules for building a VP here instead)
- $\text{NP } \rightarrow \text{DET } \text{CNP}$
- $\text{NP } \rightarrow \text{DET } \text{TCNP}$
- $\text{CNP } \rightarrow \text{ADJP } \text{CNP}$ (we could add a rule for adjectives with TCNPs)
- $\text{CNP } \rightarrow \text{CNP } \text{REL}$ (it’s debatable whether REL ever combines with TCNP)
- $\text{DET } \rightarrow \text{NP 's}$ (NP plus possessive morpheme; examples limited to proper nouns)
- $\text{VP } \rightarrow \text{is ADJP}$
- $\text{VP } \rightarrow \text{is NP}_{\text{pred}}$

#### Basic rules, non-branching rules introducing lexical categories:

- $\text{NP } \rightarrow \text{ProperN}$
- $\text{CNP } \rightarrow \text{CN}$
- $\text{TCNP } \rightarrow \text{TCN}$
- $\text{ADJP } \rightarrow \text{ADJ}$
- $\text{VP } \rightarrow \text{IV}$

### 1.2.2. Semantic interpretation of the basic rules.

- $S \rightarrow \text{NP } \text{VP}$: $||S|| = 1 \text{ iff } ||\text{NP}|| \in ||\text{VP}||$; in terms of translation, $S' = \text{VP'}(\text{NP'})$

- $\text{NP } \rightarrow \text{DET } \text{CNP}$

  We have only two DETs: an indefinite article $a/an$ as used in predicate nominals ($John \text{ is a student}, Fido \text{ is a brown dog}, Dumbo \text{ is an elephant who flies}$), and a definite article $the$ used to form definite referring expressions. We also have Possessives as DETs, from Lecture 5. (See Larson 1995 for a preview of what to do with quantificational DETs.)

  In each case the DET will be interpreted as a function that applies to the interpretation of the CNP. But the types of the two DETs are different, and the types of the two resulting NPs are different. (Later on we’ll see how to put these special cases into a more general picture.)

  $||\text{NP}|| = ||\text{DET}||(||\text{CNP}||)$ (the function is applied to the argument to yield the value)
**Ling 310 The Structure of Meaning, Lecture 6**

**B. H. Partee, March 8, 2006**


DET → NP ’s : limited to NPs of type e for now. We will follow Jensen and Vikner’s analysis and give just one interpretation, for possessives that combine with TCNPs.

||DET|| = that function f which applies to a relation R and gives as result the unique entity a such that a bears R to ||NP|| (and is undefined if there isn’t one). E.g. Mary’s denotes the function f such that f (||mother||) is the unique individual a who bears the mother-relation to Mary, if there is indeed one and only one such individual. (This is the prenominal possessive, which combines the semantics of the postnominal possessive with the semantics of the definite article.)

CNP → ADJP CNP: For the intersective adjectives, which are all we can deal with explicitly so far, each of these expressions is of type e → t, set-denoting.

||CNP1|| = ||ADJP|| ∩ ||CNP2||

We can’t do a “translation” variant of this using the First-order Predicate Calculus, because we have no way to “conjoin” two predicates – we can only conjoin formulas. When we have the lambda calculus, we’ll be able to express this semantic rule in our logic as well.

CNP → CNP REL

Relative clauses are semantically just like intersective adjectives. So:

||CNP1|| = ||REL|| ∩ ||CNP2||

S → NP TVP NP

||S|| = 1 iff <||NP1||,||NP2||> ∈ ||VP||;  in terms of translation, S’ = VP’(NP1’, NP2’)

VP → is ADJP

||VP|| = ||ADJP||: Both are of type e → t, and we just ignore “is”.

VP → is NP_{pred}

||VP|| = ||NP_{pred}||: Again both are of type e → t, and we just ignore “is”. This predicative use of NPs is different from their entity-denoting use. Examples below.

**Non-branching nodes:** NP → ProperN, CNP → CN, TCNP → TCN, ADJP → ADJ, VP → IV. In all these cases, the interpretation is just carried from the lower node to the higher node.

**1.2.3. Syntax and semantics of Relative clauses, primitive version**

Relative clauses are in effect clause-sized adjectives: the relative clause construction is one construction that gives us an infinite set of modifiers. (Question for the class: what are some other ways that we get from a finite set of lexical adjectives to an infinite set of adjectival modifiers?)

CNP: man who Mary loves [e3]

CNP  REL: who Mary loves [e3]

| CN  | S: Mary loves him3 |

| man |
The types for CN, CNP, and REL are all $e \rightarrow t$; the rule for combining CNP and REL that we gave above amounts to predicate conjunction. Model-theoretically it’s set intersection.

The relative clause itself is a predicate formed by “abstracting” on the variable corresponding to the WH-word. Since we don’t have lambda-abstraction yet, we’ll represent it as a kind of set-formation. We will include in our lexicon below indexed pronouns $hei$/him, that are interpreted as variables $x_i$, and we treat them as “underlying” our relative pronouns.

A syntactically very crude and informal version of the relative clause rule, with its semantic interpretation, can be stated as follows:

**Rel Clause Rule, syntax:** If $\varphi$ is an S and $\varphi$ contains an indexed pronoun $hei$/him, in relativizable position, then the result of adjoining who(m) to S and leaving a trace $e_i$ in place of $hei$/him, is a REL.

**Rel Clause Rule, semantics:** $||REL|| = \{d | ||\varphi||^{d/x_i} = 1\}$.

(This just means the set of individuals that satisfy the open sentence underlying the relative clause. We will be able to express it more cleanly when we have lambdas.)

Semantic derivation corresponding to the syntactic derivation above; compositional model-theoretic interpretation (read it bottom-to-top):

$$\{x | man(x)\} \cap \{x_3 | love(Mary, x_3)\}, \text{ equivalently: } \{y | man(y) \& love(Mary, y)\}$$

$$\{x | man(x)\} \quad \{d | ||love(Mary, x_3)||^{d/x_3} = 1\}, \text{ i.e. } \{x_3 | love(Mary, x_3)\}$$

1.3. **Lexicon.**

First we simply list some lexical items of various syntactic categories; aside from the category DET, these are all open classes. Then we discuss their semantics.

In later lectures we will be concerned with how best to enrich the semantic information associated with the lexicon in ways compatible with a compositional semantics.

**ProperN:** John, Mary, Bill, ...

**DET:** a, the  (Later we will add every, no, some.)

**ADJ:** carnivorous, happy, tall, ...  (not treating skillful, former, alleged)

**CN:** man, king, violinist, surgeon, fish, senator, teacher ...

**TCN:** mother, teacher, friend, ...

**TV:** sees, loves, catches, eats, ...

**IV:** walks, talks, runs, ...

**Semantics of Lexicon (MG):**

Open class lexical items (nouns, adjectives, verbs) translated into constants of appropriate type (notation: English expressions man, tall translated into IL constants man, tall, etc.). Interpretation of these constants a central task of lexical semantics. See earlier remarks about meaning postulates as one way to capture aspects of lexical semantics. A few open class words (e.g. be, entity, former) sometimes treated as part of the "logical vocabulary".
Closed class lexical items: some treated like open class items (e.g. most prepositions), others (esp. "logical" words) given explicit interpretations, as illustrated below.

**Determiners:**

*the*: One DET is the definite article *the*, interpreted as the Iota operator, which applies to a set and yields an entity (if its presuppositions are satisfied; otherwise it’s undefined and the semantic derivation “crashes”.) It is defined as follows:

\[ \iota_\alpha = d \text{ iff there is one and only one entity } d \text{ in the set denoted by } \|\alpha\|. \]

\[ \iota_\alpha \text{ is undefined otherwise.} \]

*a*: We will interpret the *a/an* that forms predicate nominals as an identity function on sets: it applies to any set as argument and gives the very same set as value.

So \[a \text{ student}|| = a (||\text{ student}||) = ||\text{ student}|| = \text{ the set of individuals in the model who are students.}\]

2. **Examples**

**Compositional semantics for these simple constructions:**

We can do some more examples on the whiteboard, such as:

*Mary talks.*

*Mary loves the student.*

*The student who Mary loves is happy.*

*John is a happy student.*

*Mary’s mother loves Mary.*

3. **Representations of meanings of English sentences in predicate logic vs. compositional interpretation of English syntax**

For this part, see the last section of the Lecture 4 handout, which we didn’t get to last week.

And we might also include here some more discussion of scope ambiguity, from the handout for Lecture 2, page 4, item 2.