

1 Some Important Concepts

- Object language an metalanguage: the conventions regarding “Ann”, $\llbracket Ann \rrbracket$, and Ann
- Truth conditions vs. Truth values
- Locality of Semantic Composition: the denotation of any non-terminal node is computed from the denotations of its daughter nodes.
- Frege’s Conjecture: Semantic composition is function application.

2 Transitive Verbs

- (1) Britney loves Chris.

- New semantic objects:
functions from individuals to functions
- New semantic rules.

3 Schönfinkelization

The domain D consists of the three goats *Leopold*, *Sebastian*, and *Dimitri*.

Sebastian > *Dimitri* > *Leopold*

The ‘is-bigger-than’ relation on this domain is:

$Big = \{(S, D), (S, L), (D, L)\}$

We can represent this relation by the characteristic function $f_{big} : (D \times D) \rightarrow \{0, 1\}$, where $f_{big}((S, D)) = f_{big}((S, L)) = f_{big}((D, L)) = 1$, and f_{big} maps all other members of $D \times D$ to 0.

Now f_{big} takes ordered pairs of individuals as arguments.

Using a process called Currying/Schönfinkelization, we can convert f_{big} into equivalent functions which take individuals as arguments.

Since there are two elements in the ordered pair, we have choice regarding which argument we want first:

- (2) a. Left-to-Right Currying:
first the left argument, then the right argument
- b. Right-to-Left Currying:
first the right argument, then the left argument

For syntactic reasons, we will adopt the function we get from Right-to-Left Currying.

4 Types

Consider the following sets:

$A = \{\text{Kate, Christen, Harry, Lance}\}$
 $B = \{\{\text{Kate, Christen}\}, \{\text{Harry, Lance}\}\}$
 $C = \{(\text{Kate, Christen}), (\text{Harry, Lance})\}$

Compare these sets with:

$A' = \{\{\text{Kate, Christen}\}, \text{Harry, Lance}\}$
 $B' = \{(\text{Kate, Christen}), \{\text{Harry, Lance}\}\}$
 $C' = \{(\text{Kate, Christen}), \text{Harry, Lance}\}$

What's odd about A' , B' and C' ?

Heterogeneity and the notion of **type**

- **e** is for entity (you, me, my grey cat, Harry, Lance, Kate, ... are all entities)
- **t** is for truth values (0, 1 are truth values)

Let us refer to D_e as the set of all entities.

$D_t = \{0, 1\}$

Now we can explicitly state the fact that A , B , and C contain elements of the **same type**.

$A \subseteq D_e$
 $B \subseteq Pow(D_e)^1$
 $C \subseteq D_e \times D_e$

¹This can also be written as $B \subseteq 2^{D_e}$

4.1 Functions and Relations: Types

We know that a function $f : A \rightarrow B$ (or a relation from A to B) can be represented as a set of tuples.

Let f_1, f_2, \dots, f_n all be functions from A to B . Assume that $A \subseteq D_1$ and that $B \subseteq D_2$, and let $F = \{f_1, \dots, f_n\}$, then
 $F \subseteq D_1 \times D_2$

However, this characterization forces us to look at functions as sets. Sometimes this is not what we want. Can we talk about the **types** of functions *qua* functions?

We can, and this is how:

- as before we start with the basic types: e and t
- if σ and τ are types, then $\langle \sigma, \tau \rangle$ is a type.

For any type σ and τ , $D_{\langle \sigma, \tau \rangle}$ is the set of all functions from D_σ to D_τ .

- How many functions are there from A to B ? How many relations?

5 Type-driven Interpretation

So far we posited particular rules for specific pieces of phrase structures. Even as we did this, the redundancy of our enterprise seemed clear because most of our rules were essentially identical. We can reduce this redundancy drastically by adopting the principle of type-driven translation.

Basic Assumption: Function application is the only non-trivial means of semantic composition.

Once we make this assumption, all we need are the following three general principles.

- (3) *Terminal Nodes* (TN)
If α is a terminal node, $\llbracket \alpha \rrbracket$ is specified in the lexicon.
- (4) *Non-branching Nodes* (NN)
If α is a non-branching node, and β is its daughter node, then $\llbracket \alpha \rrbracket = \llbracket \beta \rrbracket$.
- (5) *Functional Application* (FA)
If α is branching node, and $\{\beta, \gamma\}$ is the set of α 's daughters, and $\llbracket \beta \rrbracket$ is a function whose domain contains $\llbracket \gamma \rrbracket$, then $\llbracket \alpha \rrbracket = \llbracket \beta \rrbracket(\llbracket \gamma \rrbracket)$.

6 Well-formedness and Interpretability

Why are the following sentences ungrammatical?

- (6) a. *Ann laughed Jan.
b. *It is not the case that greeted Ann.

One explanation: they are uninterpretable - not in the domain of $\llbracket \cdot \rrbracket$.

TN, NN, and FA modified:

- (7) *Terminal Nodes* (TN)
If α is a terminal node, then α is in the domain of $\llbracket \cdot \rrbracket$ if $\llbracket \alpha \rrbracket$ is specified in the lexicon.
- (8) *Non-branching Nodes* (NN)
If α is a non-branching node, and β is its daughter node, then $\llbracket \alpha \rrbracket$ is the domain of $\llbracket \cdot \rrbracket$ if β is. In this case, $\llbracket \alpha \rrbracket = \llbracket \beta \rrbracket$.
- (9) *Functional Application* (FA)
If α is branching node, and $\{\beta, \gamma\}$ is the set of α 's daughters, then α is in the domain of $\llbracket \cdot \rrbracket$ if both β and γ are and $\llbracket \beta \rrbracket$ is a function whose domain contains $\llbracket \gamma \rrbracket$. In this case, $\llbracket \alpha \rrbracket = \llbracket \beta \rrbracket(\llbracket \gamma \rrbracket)$.
- (10) *Principle of Interpretability*
All nodes in a phrase structure tree must be in the domain of the interpretation function $\llbracket \cdot \rrbracket$.

7 The θ -Criterion

- (11) *θ -Criterion*
Each argument bears one and only one θ -role and each θ -role is assigned to one and only one argument.

Do we really need the θ -Criterion as an autonomous part of our grammar?

- (12) a. *Ann laughed Jan.
b. *Greeted Ann.

Most cases where the θ -criterion does some work seem to be handled just as well by our Principle of Interpretability.

Some potential problems for the θ -criterion:

rich assigns a θ -role:

- (13) Madonna is rich.

But what about:

- (14) The rich are different.

One argument, but two θ roles:

- (15) Björk sings and acts.