Overview

• Leftovers: Count v. Mass
• Some notes on the linguist’s stance
• Which aspects of semantics we’ll tackle
• Our formalization; Semantics Principles
• Building semantics of phrases
• Modification, coordination
• Structural ambiguity
The Count/Mass Distinction

• Partially semantically motivated
  – mass terms tend to refer to undifferentiated substances (air, butter, courtesy, information)
  – count nouns tend to refer to individuatable entities (bird, cookie, insult, fact)
• But there are exceptions:
  – succotash (mass) denotes a mix of corn & lima beans, so it’s not undifferentiated.
  – furniture, footwear, cutlery, etc. refer to individuatable artifacts with mass terms
  – cabbage can be either count or mass, but many speakers get lettuce only as mass.
  – borderline case: data
Our Formalization of the Count/Mass Distinction

• Determiners are:
  – [COUNT −] (much and, in some dialects, less),
  – [COUNT +] (a, six, many, etc.), or
  – lexically underspecified (the, all, some, no, etc.)

• Nouns select appropriate determiners
  – “count nouns” say SPR <[COUNT +]>
  – “mass nouns” say SPR <[COUNT −]>

• Nouns themselves aren’t marked for the feature COUNT

• So the SHAC plays no role in count/mass marking.
The Linguist's Stance: Building a precise model

• Some of our statements are statements about how the model works:

  “[prep] and [AGR 3sing] can’t be combined because AGR is not a feature of the type prep.”

• Some of our statements are statements about how (we think) English or language in general works.

  “The determiners *a* and *many* only occur with count nouns, the determiner *much* only occurs with mass nouns, and the determiner *the* occurs with either.”

• Some are statements about how we code a particular linguistic fact within the model.

  “All count nouns are [SPR < [COUNT +]>].”
The Linguist's Stance:
A Vista on the Set of Possible English Sentences

- ... as a background against which linguistic elements (words, phrases) have a distribution
- ... as an arena in which linguistic elements “behave” in certain ways
So far, our grammar has no semantic representations. We have, however, been relying on semantic intuitions in our argumentation, and discussing semantic contrasts where they line up (or don't) with syntactic ones.

Examples?

- structural ambiguity
- S/NP parallelism
- count/mass distinction
- complements vs. modifiers
Our Slice of a World of Meanings
Aspects of meaning we won’t account for

• Pragmatics
• Fine-grained lexical semantics:

The meaning of life is life’, or, in our case,

\[
\begin{bmatrix}
\text{RELN} & \text{life} \\
\text{INST} & i
\end{bmatrix}
\]
“... the linguistic meaning of Chris saved Pat is a proposition that will be true just in case there is an actual situation that involves the saving of someone named Pat by someone named Chris.” (p. 140)
Our Slice of a World of Meanings

What we are accounting for is the compositionality of sentence meaning.

• How the pieces fit together

  Semantic arguments and indices

• How the meanings of the parts add up to the meaning of the whole.

  Appending RESTR lists up the tree
Semantics in Constraint-Based Grammar

• Constraints as (generalized) truth conditions
  • proposition: what must be the case for a proposition to be true
  • directive: what must happen for a directive to be fulfilled
  • question: the kind of situation the asker is asking about
  • reference: the kind of entity the speaker is referring to

• Syntax/semantics interface: Constraints on how syntactic arguments are related to semantic ones, and on how semantic information is compiled from different parts of the sentence.
Feature Geometry

\[
\begin{align*}
\text{SYN} & \quad \begin{bmatrix}
\text{HEAD} & \text{pos} \\
\text{VAL} & \begin{bmatrix}
\text{SPR} & \text{list(}\text{expression}\text{)} \\
\text{COMPS} & \text{list(}\text{expression}\text{)}
\end{bmatrix}
\end{bmatrix} \\
\text{SEM} & \quad \begin{bmatrix}
\text{MODE} & \{\text{prop, ques, dir, ref, none}\} \\
\text{INDEX} & \{\text{i, j, k, ... s}_1, s_2, ...\} \\
\text{RESTR} & \text{list(}\text{pred}\text{)}
\end{bmatrix}
\end{align*}
\]
How the Pieces Fit Together

\[
\langle \text{Dana} , \rangle
\]

\[
\begin{align*}
\text{word} & \quad \left[ \begin{array}{c}
\text{index} \\
\text{head} \\
\text{val} \\
\text{sem} \\
\end{array} \right] \\
\text{syn} & \quad \left[ \begin{array}{c}
\text{noun} \\
\text{agr} \\
\text{spr} \\
\text{comps} \\
\end{array} \right] \\
\text{val} & \quad \left[ \begin{array}{c}
\text{spr} \\
\text{comps} \\
\end{array} \right] \\
\end{align*}
\]

\[
\begin{align*}
\text{sem} & \quad \left[ \begin{array}{c}
\text{reln} \\
\text{name} \\
\text{named} \\
\end{array} \right] \\
\text{index} & \quad \left[ \begin{array}{c}
\text{ref} \\
\end{array} \right] \\
\text{head} & \quad \left[ \begin{array}{c}
\text{3sing} \\
\end{array} \right] \\
\end{align*}
\]
How the Pieces Fit Together

<slept,>

[SYN]

[sleep,]

(word)

[HEAD verb]

[VAL]

[SPR ⟨ NP j ⟩]

[COMPS ⟨ ⟩]

[INDEX s₁]

[MODE prop]

[SEM]

[RESTR]

[RELN sleep]

[SIT s₁]

[SLEEPER j]

, …}
The Pieces Together

S

[SEM [INDEX i]]

[INP]

Dana

VP

[SYN [VAL [SPR ⟨1⟩]]]

SEM

RESTR

[RELN sleep]

[SIT s₁]

[SLEEPER i]

,...

slept
A More Detailed View of the Same Tree

```
S
  SEM
   MODE
      RESTR

[1]NP
  INDEX
     i
  SEM
     RESTR
        RELN name
        NAME Dana
        NAMED i

VP
  SYN
     [ VAL
         SPR ⟨ 1 ⟩ ]
     SEM
     RESTR
        RELN sleep
        SIT s1
        SLEEPER i
```
To Fill in Semantics for the S-node

We need the Semantics Principles

- The Semantic Inheritance Principle:

  In any headed phrase, the mother's MODE and INDEX are identical to those of the head daughter.

- The Semantic Compositionality Principle:
Semantic Inheritance Illustrated

\[
S \left[ \begin{array}{c}
\text{SEM} \\
\text{INDEX} \ s_1 \\
\text{MODE} \ prop \\
\text{RESTR}
\end{array} \right]
\]

\[
\text{NP} \left[ \begin{array}{c}
\text{SEM} \\
\text{INDEX} \ i \\
\text{RESTR} \\
\left( \begin{array}{c}
\text{RELN} \ name \\
\text{NAME} \ Dana \\
\text{NAMED} \ i
\end{array} \right)
\end{array} \right]
\]

\[
\text{VP} \left[ \begin{array}{c}
\text{SEM} \\
\text{RESTR} \\
\left( \begin{array}{c}
\text{RELN} \ sleep \\
\text{SIT} \ s_1 \\
\text{SLEEPER} \ i
\end{array} \right)
\end{array} \right]
\]

\[
\text{SYN} \left[ \begin{array}{c}
\text{VAL} \\
\left[ \text{SPR} \left( \begin{array}{c}1 \end{array} \right) \right]
\end{array} \right]
\]
To Fill in Semantics for the S-node

We need the Semantics Principles

• The Semantic Inheritance Principle:

  In any headed phrase, the mother's MODE and INDEX are identical to those of the head daughter.

• The Semantic Compositionality Principle:

  In any well-formed phrase structure, the mother's RESTR value is the sum of the RESTR values of the daughter.
Semantic Compositionality Illustrated
What Identifies Indices?

[Diagram of a tree structure representing the sentence:]

S

NP

D the

NOM_i cat

VP

SPR

<1>

RESTR

RELN sleep

SIT

SLEEPER_i

slept

PP on the mat
Summary: Words ...

• contribute predications
• ‘expose’ one index in those predications, for use by words or phrases
• relate syntactic arguments to semantic arguments

\[
\begin{align*}
\text{SYN} & \quad \text{HEAD} \quad \text{verb} \\
\text{VAL} & \quad \text{SPR} \quad \langle \text{NP}_j \rangle \\
\text{COMPS} & \quad \langle \rangle \\
\text{INDEX} & \quad s_1 \\
\text{MODE} & \quad \text{prop} \\
\text{SEM} & \quad \text{RELN} \quad \text{sleep} \\
\text{SIT} & \quad s_1 \\
\text{SLEEPER} & \quad j \\
\end{align*}
\]
Summary: Grammar Rules ...

• identify feature structures (including the INDEX value) across daughters

Head Specifier Rule

\[
\begin{align*}
\text{phrase} & \\
\text{SYN} [\text{VAL} [\text{SPR} \langle \rangle]] & \rightarrow H \text{ SYN} [\text{VAL} [\text{SPR} \langle \rangle]]
\end{align*}
\]

Head Complement Rule

\[
\begin{align*}
\text{phrase} & \\
\text{SYN} [\text{VAL} [\text{COMPS} \langle \rangle]] & \rightarrow H \text{ word} \\
\text{SYN} [\text{VAL} [\text{COMPS} \langle \rangle]] & \rightarrow H \text{ SYN} [\text{VAL} [\text{COMPS} \langle 1, \ldots, n \rangle]]
\end{align*}
\]

Head Modifier Rule

\[
\begin{align*}
\text{phrase} & \rightarrow H [1] \text{ SYN} [\text{COMPS} \langle \rangle] [\text{SYN} [\text{VAL} [\text{COMPS} \langle \rangle]] [\text{SYN} [\text{VAL} [\text{MOD} \langle \rangle]]]
\end{align*}
\]
Summary: Grammar Rules ...

- identify feature structures (including the INDEX value) across daughters
- license trees which are subject to the semantic principles
  - SIP ‘passes up’ MODE and INDEX from head daughter

```
S

[ INDEX \textcolor{red}{s_1} ]

 MODE prop

SEM

RESTR \langle [ RELN \textcolor{red}{name} ] , [ RELN \textcolor{red}{sleep} ] \rangle

NAME Dana

NAMED i

[ SLEEPER i ] , \ldots

[ INDEX i ]

MODE prop

SEM

RESTR \langle [ RELN \textcolor{red}{name} ] \rangle

NAME Dana

NAMED i

[ SLEEPER i ]

[ INDEX \textcolor{red}{s_1} ]

MODE prop

SEM

RESTR \langle [ RELN \textcolor{red}{sleep} ] \rangle

SIT \textcolor{red}{s_1}

SLEEPER i

[ SLEEPER i ] , \ldots
```

© 2003 CSLI Publications
Summary: Grammar Rules ...

- identify feature structures (including the INDEX value) across daughters
- license trees which are subject to the semantic principles
  - SIP ‘passes up’ MODE and INDEX from head daughter
  - SCP: ‘gathers up’ predications (RESTR list) from all daughters

\[
\begin{align*}
S &\left[\begin{array}{c}
\text{INDEX } s_1 \\
\text{MODE } \text{prop} \\
\text{SEM} \\
\text{RESTR} \\
\text{RELN } \text{name} \\
\text{NAME } \text{Dana} \\
\text{NAMED } i \\
\text{RESTR} \\
\text{RELN } \text{sleep} \\
\text{SIT } s_1 \\
\text{SLEEPER } i \\
\end{array}\right] \\
\end{align*}
\]
Other Aspects of Semantics

- Tense, Quantification (only touched on here)
- Modification
- Coordination
- Structural Ambiguity
Evolution of a Phrase Structure Rule

Ch. 2:  NOM ---> NOM PP  
       VP ---> VP PP

Ch. 3:  \[
\begin{array}{c}
\text{phrase} \\
\text{VAL} \\
\text{SPR}
\end{array}
\] \rightarrow
\[
\begin{array}{c}
\text{H} \\
\text{VAL} \\
\text{SPR}
\end{array}
\]

Ch. 4:  \[
\begin{array}{c}
\text{[phrase]} \\
\text{COMPS} \\
\text{SYN}
\end{array}
\] \rightarrow
\[
\begin{array}{c}
\text{H} \\
\text{VAL} \\
\text{COMPS}
\end{array}
\]

Ch. 5:  \[
\begin{array}{c}
\text{[phrase]} \\
\text{COMPS} \\
\text{MOD}
\end{array}
\] \rightarrow
\[
\begin{array}{c}
\text{H} \\
\text{COMPS} \\
\text{MOD}
\end{array}
\]

Ch. 5 (abbreviated): \[
\begin{array}{c}
\text{[phrase]} \\
\text{COMPS}
\end{array}
\] \rightarrow
\[
\begin{array}{c}
\text{COMPS} \\
\text{MOD}
\end{array}
\]
Evolution of Another Phrase Structure Rule

Ch. 2: \( X \rightarrow X^+ \text{ CONJ } X \)

Ch. 3: \( 1 \rightarrow 1^+ \left[ \begin{array}{c} \text{word} \\ \text{HEAD} \end{array} \right] \text{ conj } \left[ \begin{array}{c} 1 \\ \end{array} \right] \)

Ch. 4: \( \left[ \begin{array}{c} \text{VAL} \end{array} \right] \rightarrow \left[ \begin{array}{c} \text{VAL} \end{array} \right]^+ \left[ \begin{array}{c} \text{word} \\ \text{HEAD} \end{array} \right] \text{ conj } \left[ \begin{array}{c} \text{VAL} \\ \end{array} \right] \)

Ch. 5: \( \left[ \begin{array}{c} \text{SYN} \\ \text{SEM} \end{array} \right] \left[ \begin{array}{c} \text{VAL} \end{array} \right] \) \( \rightarrow \)

\( \left[ \begin{array}{c} \text{SYN} \\ \text{SEM} \end{array} \right] \left[ \begin{array}{c} \text{VAL} \end{array} \right] \) \( \rightarrow \) \( \left[ \begin{array}{c} \text{SYN} \\ \text{SEM} \end{array} \right] \left[ \begin{array}{c} \text{VAL} \\ \end{array} \right] \)

Ch. 5 (abbreviated):

\( \left[ \begin{array}{c} \text{VAL} \\ \text{IND} \end{array} \right] \rightarrow \left[ \begin{array}{c} \text{VAL} \end{array} \right]^\ast \left[ \begin{array}{c} \text{VAL} \\ \text{IND} \end{array} \right] \left[ \begin{array}{c} \text{HEAD} \end{array} \right] \text{ conj } \left[ \begin{array}{c} \text{IND} \\ \text{RESTR} \end{array} \right] \left( \left[ \begin{array}{c} \text{ARGS} \end{array} \right] \left( s_1 \ldots s_n \right) \right) \left[ \begin{array}{c} \text{IND} \\ \end{array} \right] \left[ \begin{array}{c} \text{VAL} \\ \end{array} \right] \)
Combining Constraints and Coordination

Coordination Rule

\[
\begin{bmatrix}
\text{VAL } 0 \\
\text{IND } s_0
\end{bmatrix} \rightarrow \begin{bmatrix}
\text{VAL } 0 \\
\text{IND } s_1
\end{bmatrix} \cdots \begin{bmatrix}
\text{VAL } 0 \\
\text{IND } s_{n-1}
\end{bmatrix} \begin{bmatrix}
\text{HEAD } \text{conj} \\
\text{IND } s_0 \\
\text{RESTR } \langle [\text{ARGS } \langle s_1 \ldots s_n \rangle] \rangle
\end{bmatrix} \begin{bmatrix}
\text{VAL } 0 \\
\text{IND } s_n
\end{bmatrix}
\]

Lexical Entry for a Conjunction

\[
\langle \text{and} , \begin{bmatrix}
\text{SYN} \\
\text{SEM}
\end{bmatrix} \begin{bmatrix}
\text{HEAD } \text{conj} \\
\text{INDEX } s \\
\text{MODE } \text{none} \\
\text{RESTR } \langle [\text{RELN } \langle \text{and} \rangle] \rangle \\
\text{SIT } s
\end{bmatrix} \rangle
\]
Combining Constraints and Coordination

Lexical Entry for *and*

\[
\begin{bmatrix}
\text{SYN} & [\text{HEAD} \ conj] \\
\text{INDEX} & s \\
\text{MODE} & \text{none} \\
\text{RESTRICTION} & \left\langle \text{RELN and SIT} \atop s \right\rangle
\end{bmatrix}
\]

Coordination Rule

\[
\begin{bmatrix}
[\text{VAL} \ 0] \\
\text{IND} \ s_0
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
[\text{VAL} \ 0] \\
\text{IND} \ s_1 \\
\ldots \\
[\text{VAL} \ 0] \\
\text{IND} \ s_{n-1}
\end{bmatrix}
\begin{bmatrix}
\text{HEAD} \ conj \\
\text{IND} \ s_0 \\
\text{RESTRICTION} \ (\text{ARGS} \ s_1 \ldots s_n)
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
[\text{VAL} \ 0] \\
\text{IND} \ s_n
\end{bmatrix}
\]

Pat sings

*and*

Lee dances
Structural Ambiguity, Tree I

[IND $s_0$]

[CONJ]

[IND $s_1$]

[IND $s_2$]

[frequently]

NP

VP

NP

VP

Pat

sings

Lee

dances

IND prop

MODE prop

RESTR

RELN name

NAME Pat

NAMED k

RELN name

NAME Lee

NAMED j

RELN sing

SIT $s_1$

SINGER k

RELN dance

SIT $s_2$

DANCER j

RELN and

SIT $s_0$

ARGS $s_1$, $s_2$

RELN frequently

ARG $s_0$
Structural Ambiguity, Tree II

- **S**
  - [IND \( s_0 \)]
  - **CONJ**
  - [IND \( s_1 \)]
  - and
  - [IND \( s_2 \)]

- **NP**
  - *Pat*
  - **VP**
  - *sings*

- **NP**
  - *Lee*
  - **VP**
  - *dances*

**RESTR**

- [RELN name]
  - [NAME \( k \)]
  - [NAMED \( j \)]

- [RELN sing]
  - [SIT \( s_1 \)]
  - [DANCER \( j \)]

- [RELN dance]
  - [SIT \( s_2 \)]
  - [ARG SIT \( s_2 \)]

- [RELN frequently]
  - [ARG \( s_2 \)]

© 2003 CSLI Publications
Question About Structural Ambiguity

Why isn’t this a possible semantic representation for the string *Pat sings and Lee dances frequently*?
Semantic Compositionality

\[
\begin{align*}
\text{IND } s_0 \\
\text{MODE prop} \\
\text{RESTR} &\begin{cases}
\text{RELN name} &\text{NAME Pat} \\
\text{NAMED k} &\text{SIT s_1} \\
\text{RELN name} &\text{NAME Lee} \\
\text{NAMED j} &\text{SIT s_2} \\
\end{cases} \\
\text{RELN name} &\text{NAME Lee} \\
\text{NAMED j} &\text{DANCER j} \\
\end{align*}
\]
Overview

• Some notes on the linguist’s stance
• Which aspects of semantics we’ll tackle
• Our formalization; Semantics Principles
• Building semantics of phrases
• Modification, coordination
• Structural ambiguity
• Next time: How the grammar works
Reading Questions

• What is our analogue of theta roles in GB?
• What set are the feature names on predications drawn from?
• How do you build a practical lexicon like this?
• How is MODE assigned to lexical entries? In *Pat left?* does `?` have to be the head?
• Does the SCP mean that we are now working bottom-up?
Reading Questions

• How would you deal with sarcasm?
• How do we handle sentences with implied elements from previous sentences?
• If love requires [COMP j] how do we handle Kim loves?
• What about selectional restrictions? (#The book barked)
Reading Questions

• What’s the deal with QSCOPE and BV? (See Copestake et al 2005)

• The coordination Semantic compositionality principle states that the mother's RESTR value is the sum of the all the daughters. So in coordination, the mother will have the RESTR values from both conjuncts and the conjunction as well. With the way we have defined the conjunction rule (42), would this mean that the index values s1 and s2 are in the mother's RESTR value in two places because they are in the RESTR value for the conjunction? if so does this cause any issues?