JMORF/MORS — Morpho-Syntax

Semantics

Francis Bond
Palacký University
https://fcbond.github.io/
bond@ieee.org

Lecture 5
Location: KB-1.11

Overview

- The linguist's stance
- Which aspects of meaning we model
- Semantic features and principles: building the semantics of phrases
- Modification and coordination
- Structural ambiguity
- Building a tree (updated)
- Problems: two kinds of modifiers, number names

Review

Last Week

- Complex feature structures allow us to
  - Write more general rules
  - Constrain them lexically

Head-Complement Rule

$$\begin{bmatrix} \textit{phrase} \\ \text{VAL} \ \left[\text{COMPS} \ \langle\rangle\right] \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} \\ \text{VAL} \ \left[\text{COMPS} \ \left\langle \boxed{1}, \ \ldots, \ \boxed{n} \right\rangle\right] \end{bmatrix} \quad \boxed{1}, \ \ldots, \ \boxed{n}$$

- The possible complements are specified lexically

$$\left\langle \text{devour}, \begin{bmatrix} \textit{word} \\ \text{HEAD} \ \textit{verb} \\ \text{VAL} \ \ \left[\text{COMPS} \ \langle \text{NP} \rangle\right] \end{bmatrix} \right\rangle \quad \left\langle \text{put}, \begin{bmatrix} \textit{word} \\ \text{HEAD} \ \textit{verb} \\ \text{VAL} \ \ \left[\text{COMPS} \ \langle \text{NP PP} \rangle\right] \end{bmatrix} \right\rangle$$

Head-Specifier Rule

$$\begin{bmatrix} \textit{phrase} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \langle\rangle \\ \text{SPR} & \langle\rangle \end{bmatrix} \end{bmatrix} \rightarrow \boxed{2} \quad \mathbf{H} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{COMPS} & \langle\rangle \\ \text{SPR} & \langle \boxed{2} \rangle \end{bmatrix} \end{bmatrix}$$

- Combines the rules expanding S and NP (and potentially other categories).
- Again, restrictions on specifiers come from the lexicon.

$$\left\langle \text{dog}, \begin{bmatrix} \textit{word} \\ \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{D} \rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \right\rangle \quad \left\langle \text{eat}, \begin{bmatrix} \textit{word} \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \text{NP} \rangle \\ \text{COMPS} & \langle (\text{NP}) \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Two Principles

- **The Valence Principle**

  Unless the rule says otherwise, the mother's values for the VAL features (SPR and COMPS) are identical to those of the head daughter.

- **The Specifier-Head Agreement Constraint (SHAC)**
  Verbs and common nouns must be specified as:

$$\begin{bmatrix} \text{HEAD} & \left[\text{AGR } \boxed{1}\right] \\ \text{VAL} & \left[\text{SPR } \left\langle \left[\text{HEAD } \left[\text{AGR } \boxed{1}\right]\right] \right\rangle\right] \end{bmatrix}$$

  In plain English: a verb or noun agrees with its specifier

Overview

- The linguist's stance
- Which aspects of meaning we model
- Semantic features and principles: building the semantics of phrases
- Modification and coordination
- Structural ambiguity
- Building a tree (updated)
- Problems: two kinds of modifiers, number names

The Linguist's Stance

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 Possible Sentences

We look at 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

In plain English: we learn what a word is by looking at where it can and can't appear.

Our Slice of Meaning

Semantics: Where's the Beef?

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?

Semantics: Where's the Beef?

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 (in this course)

- Pragmatics
- Fine-grained lexical semantics
  The meaning of *life* is
  - `life` (or life′)
  - $\begin{bmatrix} \text{RELN} & \textit{life} \\ \text{INST} & i \end{bmatrix}$
  - Not like WordNet: $\texttt{life}_1 \subset \texttt{being}_1 \subset \texttt{state}_1$ …

We use the word itself as the name of its meaning: we don't try to define what *life* is.

Our Slice of a World of Meanings

$$\begin{bmatrix} \text{MODE} & \textit{prop} \\ \text{INDEX} & s \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{save} \\ \text{SIT} & s \\ \text{SAVER} & i \\ \text{SAVED} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \textit{Chris} \\ \text{NAMED} & i \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \textit{Pat} \\ \text{NAMED} & j \end{bmatrix} \right\rangle \end{bmatrix}$$

"...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." (Sag et al., 2003, p. 140)

Read it as: there is a situation s that is a saving; the saver i is named Chris; the saved j is named Pat.

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

The value of **restriction** is the set of conditions that must hold (in some possible world) for the expression to be applicable.

Semantics in Constraint-Based Grammar

- Constraints as (generalized) truth conditions
  - **proposition**: what must be the case for a proposition to be true (*Kim slept.*)
  - **directive**: what must happen for a directive to be fulfilled (*Sleep!*)
  - **question**: the kind of situation the asker is asking about (*Did Kim sleep?*)
  - **reference**: the kind of entity the speaker is referring to (*the cat*)

- **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.

Semantic Features and Principles

Feature Geometry

$$\begin{bmatrix} \textit{expression} & \\ \text{SYN} & \begin{bmatrix} \textit{syn-cat} & \\ \text{HEAD} & \begin{bmatrix} \textit{pos} \\ \dots \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle \dots \rangle \\ \text{COMPS} & \langle \dots \rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \textit{sem-cat} & \\ \text{MODE} & \{\textit{prop}, \textit{ques}, \textit{dir}, \textit{ref}, \textit{none}\} \\ \text{INDEX} & \{i, j, k, \dots, s_1, s_2, \dots\} \\ \text{RESTR} & \langle \dots \rangle \end{bmatrix} \end{bmatrix}$$

SEM in Plain English

MODE	What kind of meaning: *prop* (a statement), *ques* (a question), *dir* (a command), *ref* (something referred to), or *none*
INDEX	A label for the thing or situation the expression is about: $i, j, k, \ldots$ for individuals, s for situations
RESTR	A list of facts (**predications**) that must be true of it

Each predication has a RELN (what kind of fact) and **roles** that say who is involved,

e.g. $\begin{bmatrix} \text{RELN} & \textbf{sleep} \\ \text{SIT} & s \\ \text{SLEEPER} & i \end{bmatrix}$: in situation s, i sleeps

How the Pieces Fit Together: A Name

$$\left\langle \textit{Kim}, \left[\begin{array}{ll} \textsc{syn} & \left[\begin{array}{ll} \textsc{head} & \left[\begin{array}{ll} \textit{noun} & \\ \textsc{agr} & \textit{3sing} \end{array} \right] \\ \textsc{val} & \left[\begin{array}{ll} \textsc{spr} & \langle\rangle \\ \textsc{comps} & \langle\rangle \end{array} \right] \end{array} \right] \\ \textsc{sem} & \left[\begin{array}{ll} \textsc{mode} & \textit{ref} \\ \textsc{index} & \boxed{i} \\ \textsc{restr} & \left\langle \left[\begin{array}{ll} \textsc{reln} & \textbf{name} \\ \textsc{name} & \textit{Kim} \\ \textsc{named} & \boxed{i} \end{array} \right] \right\rangle \end{array} \right] \end{array} \right] \right\rangle$$

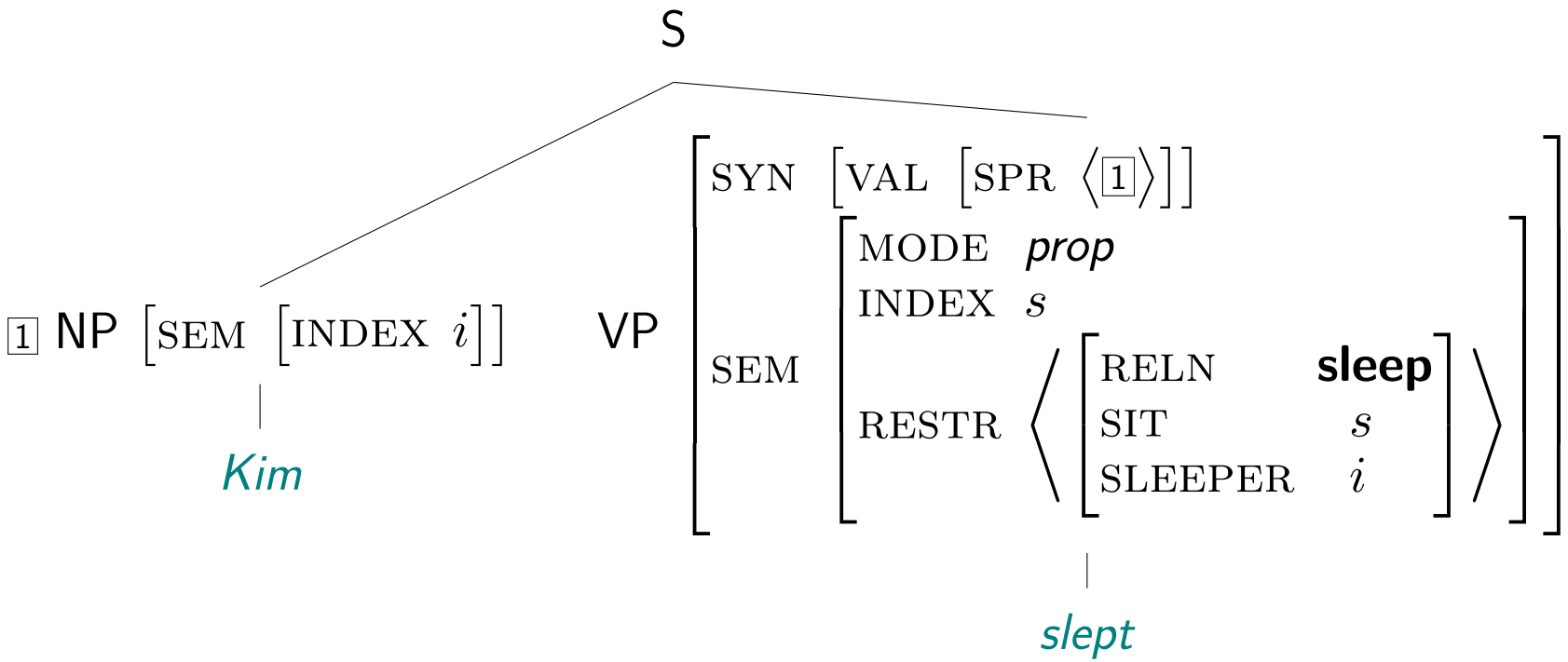
- INDEX i: the person this word is about
- NAMED i: that same person is called Kim

How the Pieces Fit Together: A Verb

$$\left\langle \textit{slept} \;,\; \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \text{NP}_{\boxed{i}} \right\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \textit{prop} \\ \text{INDEX} & \boxed{s} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{sleep} \\ \text{SIT} & \boxed{s} \\ \text{SLEEPER} & \boxed{i} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

- INDEX s: the sleeping situation
- NP_i is short for an NP whose INDEX is i
- So whoever is the specifier (the subject) is the SLEEPER

The Pieces Together



- The Head-Specifier Rule makes the NP the VP's specifier 1
- So ***Kim***'s index i is the SLEEPER

How to Share Semantic Information

We need two more principles

- **The Semantic Inheritance Principle (SIP)**

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

  A phrase is about what its head is about.

- **The Semantic Compositionality Principle (SCP)**

  In any well-formed phrase structure, the mother's RESTR value is the sum of the RESTR values of the daughters.

  A phrase means everything its parts mean.

List Sum

The SCP adds lists together with $\oplus$ (technically, concatenation): just join the lists, in order

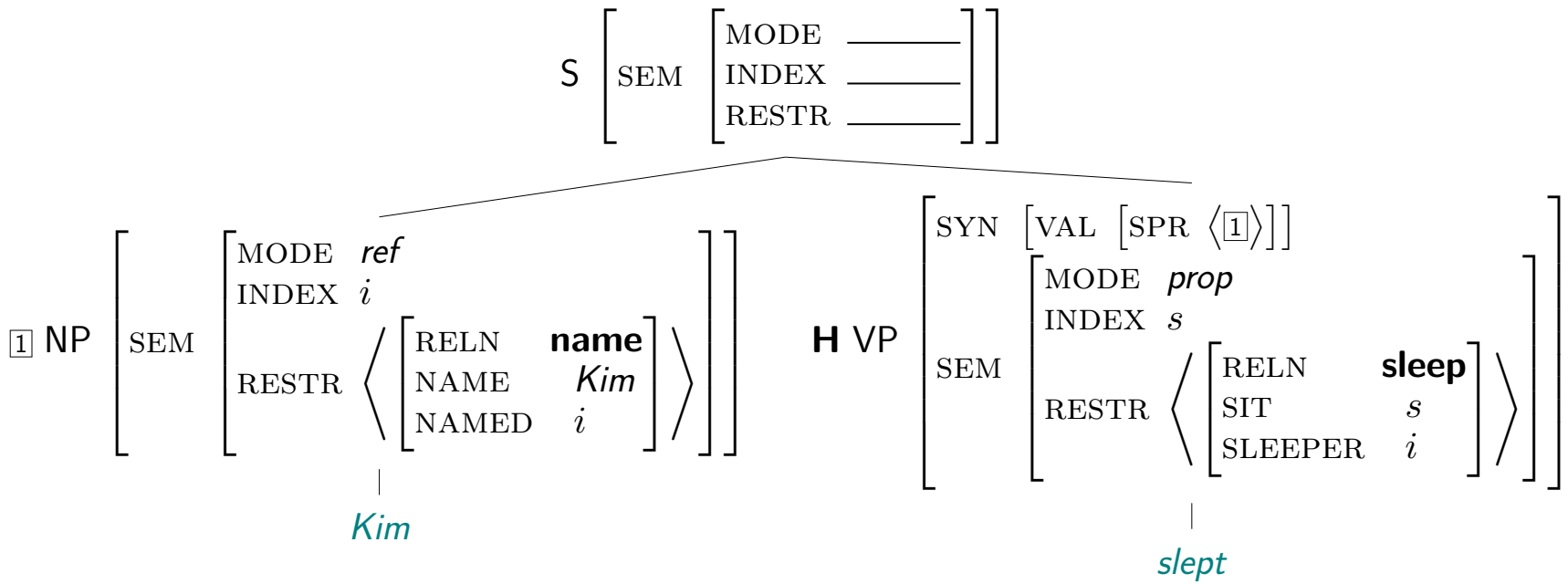
$$\langle \mathrm{A},\ \mathrm{B} \rangle \oplus \langle \mathrm{C} \rangle = \langle \mathrm{A},\ \mathrm{B},\ \mathrm{C} \rangle$$

Order matters:

$$\langle \mathrm{C} \rangle \oplus \langle \mathrm{A},\ \mathrm{B} \rangle = \langle \mathrm{C},\ \mathrm{A},\ \mathrm{B} \rangle$$

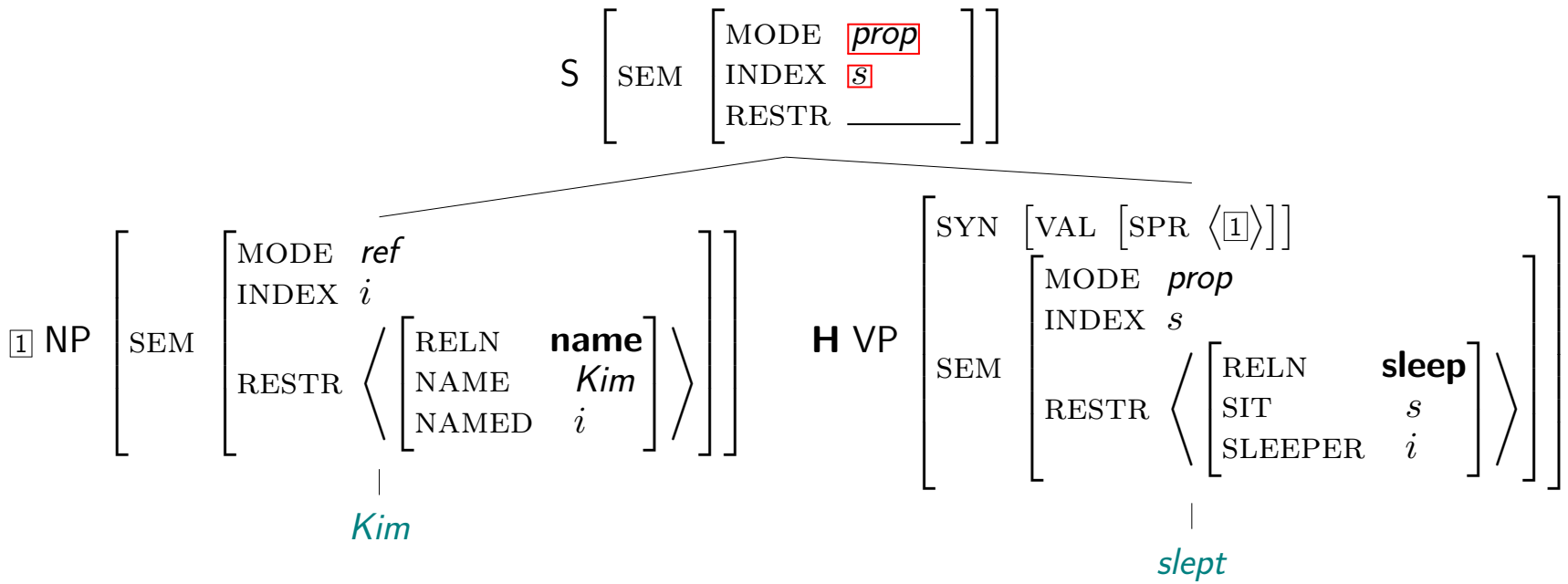
In general: $\langle a_1, \ldots, a_n \rangle \oplus \langle b_1, \ldots, b_m \rangle = \langle a_1, \ldots, a_n, b_1, \ldots, b_m \rangle$

Filling in the S Node



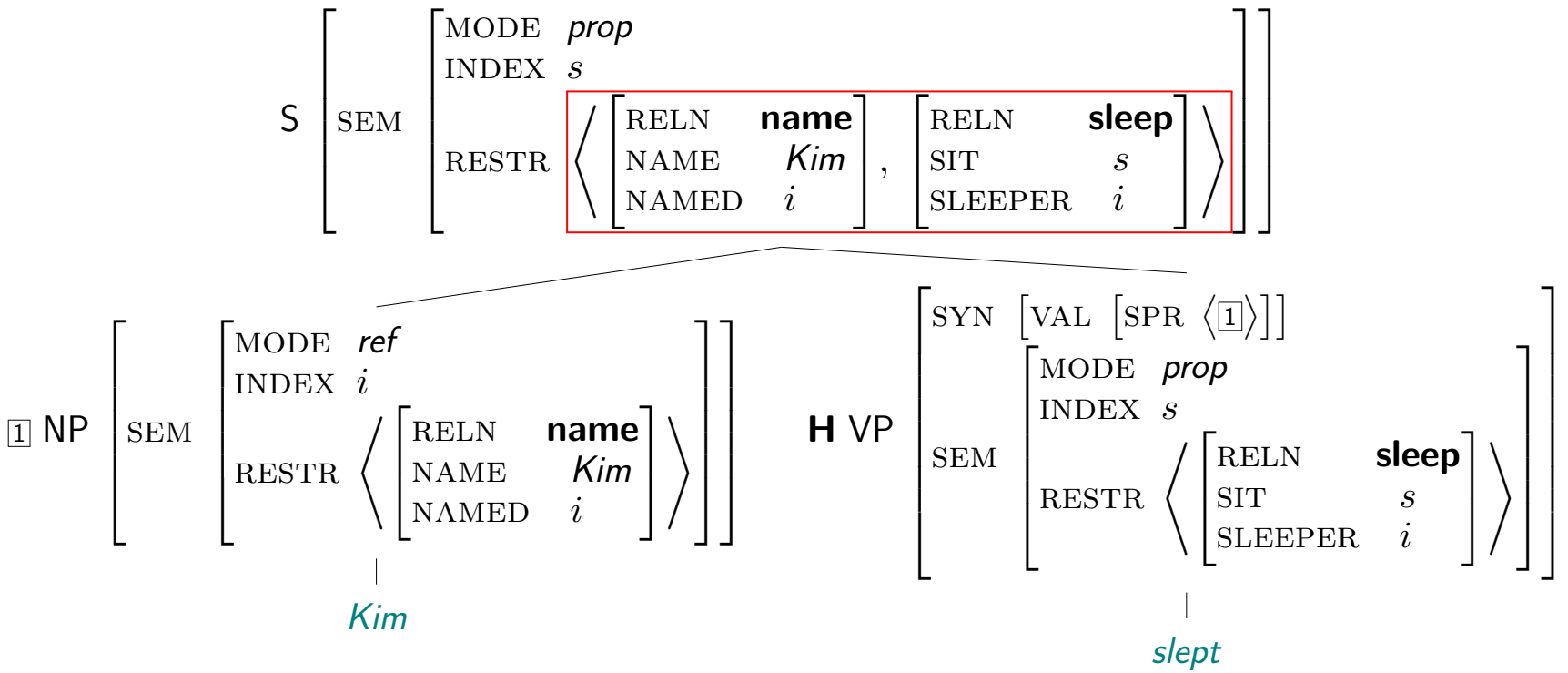
Kim and ***slept*** get their SEM from their lexical entries. S has no lexical entry: where do its MODE, INDEX and RESTR come from?

Filling in the S Node: SIP



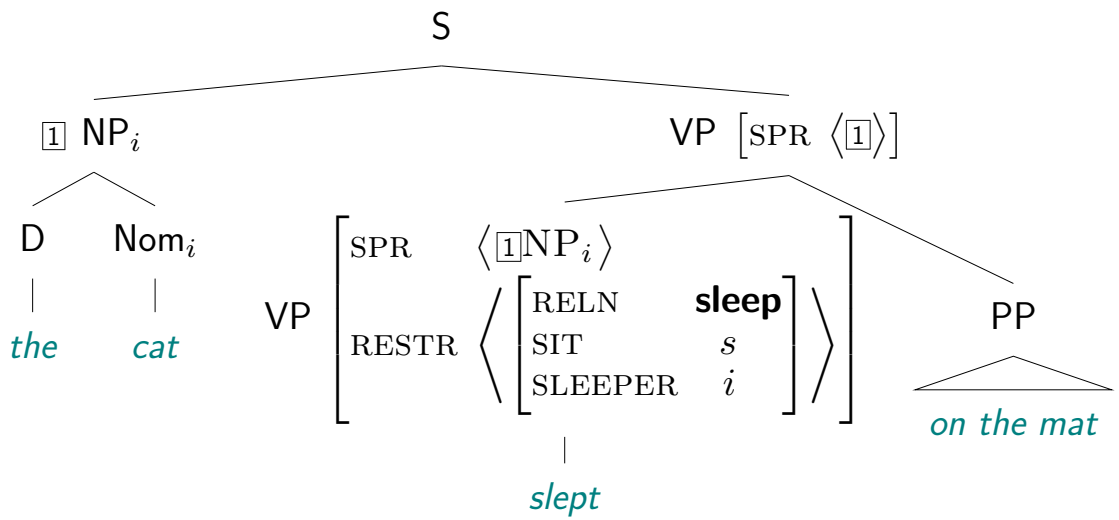
SIP: VP is the head daughter, so S has the same MODE and INDEX as the VP

Filling in the S Node: SCP



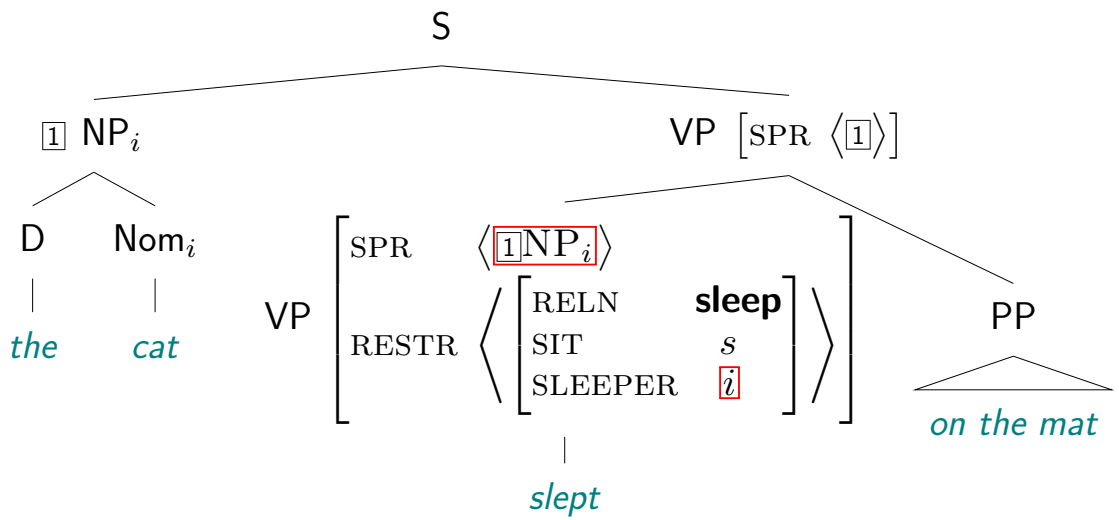
SCP: S's RESTR is the NP's RESTR $\oplus$ the VP's RESTR

What Identifies Indices?



Why is *cat*'s index the sleeper?

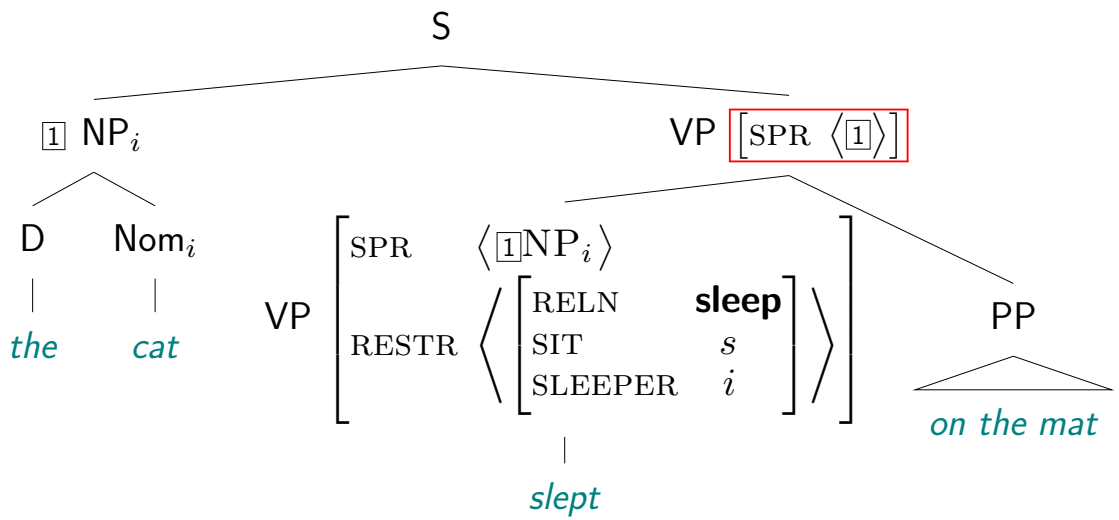
What Identifies Indices?



Why is *cat*'s index the sleeper?

1. ***slept***'s lexical entry: its specifier's index is the SLEEPER

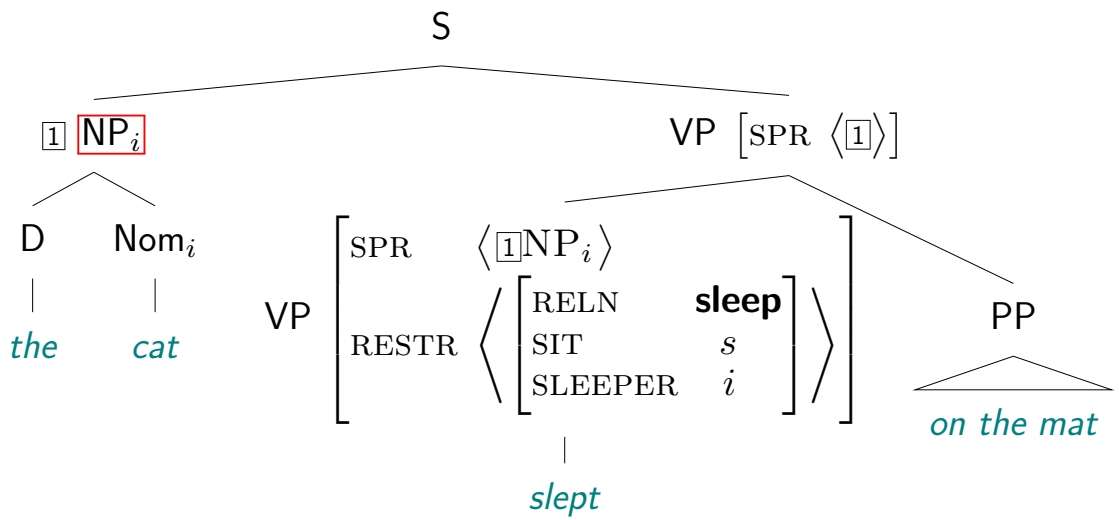
What Identifies Indices?



Why is *cat*'s index the sleeper?

1. ***slept***'s lexical entry: its specifier's index is the SLEEPER
2. Valence Principle: the Head-Modifier Rule doesn't change SPR, so the upper VP has the same SPR ⟨1⟩

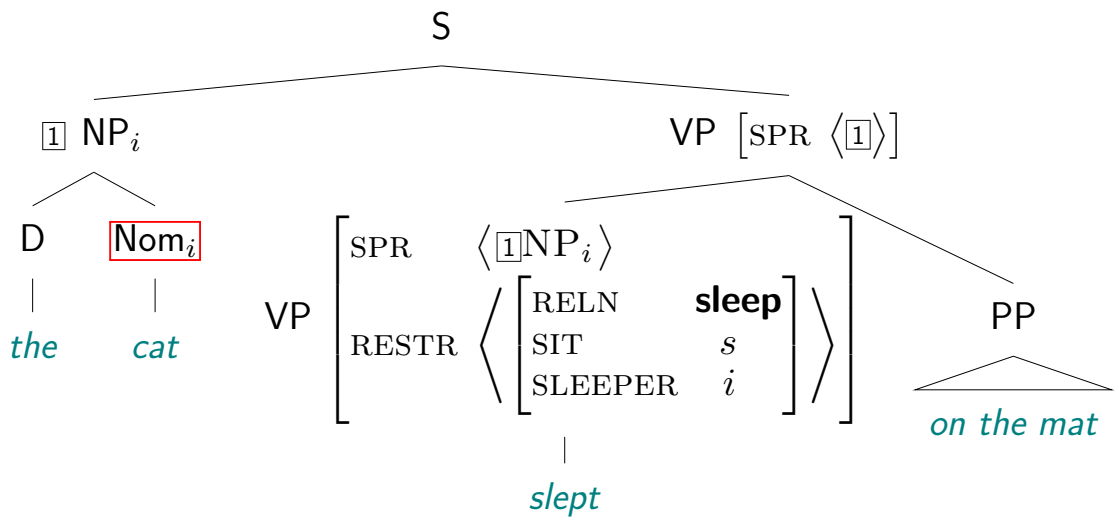
What Identifies Indices?



Why is *cat*'s index the sleeper?

1. ***slept***'s lexical entry: its specifier's index is the SLEEPER
2. Valence Principle: the upper VP has the same SPR $\langle \boxed{1} \rangle$
3. Head-Specifier Rule: the NP is that specifier $\boxed{1}$, so its index is i

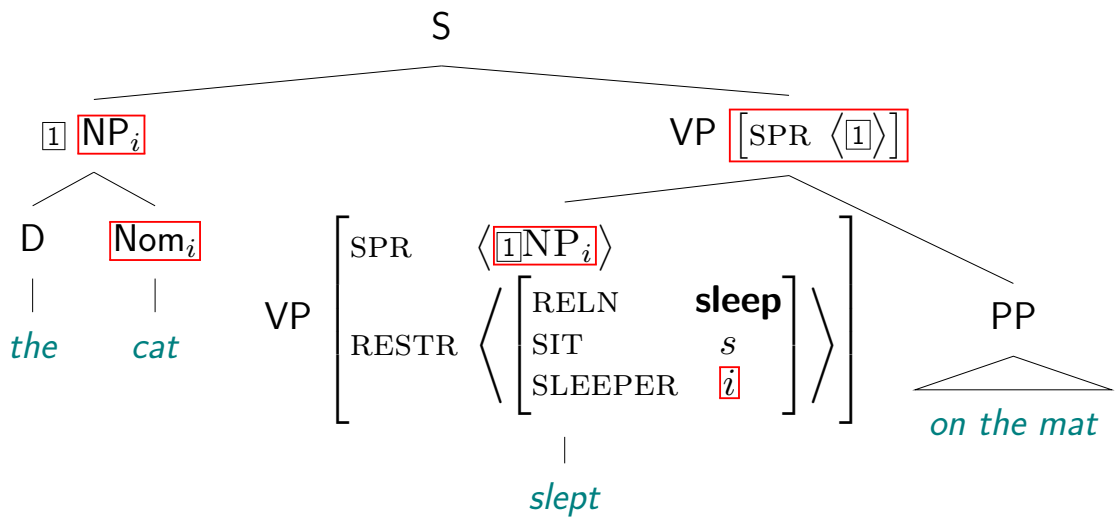
What Identifies Indices?



Why is *cat*'s index the sleeper?

1. ***slept***'s lexical entry: its specifier's index is the SLEEPER
2. Valence Principle: the upper VP has the same SPR ⟨1⟩
3. Head-Specifier Rule: the NP is that specifier 1, so its index is i
4. SIP: the NP gets its INDEX from its head, the Nom ***cat***

Answer: A Chain of Identities



Why is *cat*'s index the sleeper?

A chain of identities makes the INDEX of ***cat*** the SLEEPER in the **sleep** predication of ***slept***.

The cat is the one who slept.

Summary: Words ...

- contribute predications (**sleep**)
- 'expose' one index in those predications, for use by words or phrases (INDEX s = SIT s)
- relate syntactic arguments to semantic arguments (NP_i = SLEEPER i)

$$\left\langle \textit{slept} \;, \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \textit{verb} \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle \text{NP}_{\boxed{i}} \rangle \\ \text{COMPS} & \langle\rangle \end{array} \right] \end{array} \right] \\ \text{SEM} & \left[\begin{array}{ll} \text{MODE} & \textit{prop} \\ \text{INDEX} & \boxed{s} \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{sleep} \\ \text{SIT} & \boxed{s} \\ \text{SLEEPER} & \boxed{i} \end{array} \right] \right\rangle \end{array} \right] \end{array} \right] \right\rangle$$

Summary: Grammar Rules ...

- Identify feature structures (including the INDEX value) across daughters
  - Head-Specifier Rule

$$\begin{bmatrix} \textit{phrase} \\ \text{VAL} \ \begin{bmatrix} \text{SPR} \ \langle\rangle \end{bmatrix} \end{bmatrix} \rightarrow \boxed{1} \quad \mathbf{H} \begin{bmatrix} \text{VAL} \ \begin{bmatrix} \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix}$$

  - Head-Complement Rule

$$\begin{bmatrix} \textit{phrase} \\ \text{VAL} \ \begin{bmatrix} \text{COMPS} \ \langle\rangle \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} \\ \text{VAL} \ \begin{bmatrix} \text{COMPS} \ \langle \boxed{1}, \ \ldots, \ \boxed{n} \rangle \end{bmatrix} \end{bmatrix} \quad \boxed{1}, \ \ldots, \ \boxed{n}$$

  - Head-Modifier Rule

$$\begin{bmatrix} \textit{phrase} \end{bmatrix} \rightarrow \mathbf{H} \ \boxed{1} \ \begin{bmatrix} \text{VAL} \ \begin{bmatrix} \text{COMPS} \ \langle\rangle \end{bmatrix} \end{bmatrix} \quad \begin{bmatrix} \text{VAL} \ \begin{bmatrix} \text{COMPS} & \langle\rangle \\ \text{MOD} & \langle \boxed{1} \rangle \end{bmatrix} \end{bmatrix}$$

Summary: Grammar Rules ...

- Identify feature structures (including the INDEX value) across daughters
- License trees that are subject to the semantic principles
  - SIP: 'passes up' MODE and INDEX from head daughter
  - SCP: 'gathers up' predications (RESTR list) from all daughters
- The semantics is strictly compositional — all of the meaning comes from the words, rules and principles.
  - We then enrich this with pragmatic inference — but we need a base to infer from

Modification and Coordination

Other Aspects of Semantics

- Tense, quantification (only touched on here)
- Modification
- Coordination
- Structural ambiguity

Evolution of a Phrase Structure Rule

C2 NOM $\rightarrow$ NOM PP; VP $\rightarrow$ VP PP

C3 $\begin{bmatrix} \textit{phrase} \\ \text{VAL} \begin{bmatrix} \text{SPR} & \text{-} \\ \text{COMPS} & \textit{itr} \end{bmatrix} \end{bmatrix} \rightarrow \textbf{H} \begin{bmatrix} \textit{phrase} \\ \text{VAL} \begin{bmatrix} \text{SPR} & \text{-} \end{bmatrix} \end{bmatrix}$ PP

C4 $\left[\textit{phrase}\right] \rightarrow \textbf{H} \left[\text{VAL} \left[\text{COMPS} \left\langle\right\rangle\right]\right]$ PP

C5 $\left[\textit{phrase}\right] \rightarrow \textbf{H} \boxed{1} \left[\text{SYN} \left[\text{VAL} \left[\text{COMPS} \left\langle\right\rangle\right]\right]\right] \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{COMPS} & \left\langle\right\rangle \\ \text{MOD} & \left\langle \boxed{1} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$

$= \left[\textit{phrase}\right] \rightarrow \textbf{H} \boxed{1} \left[\text{COMPS} \left\langle\right\rangle\right] \begin{bmatrix} \text{COMPS} & \left\langle\right\rangle \\ \text{MOD} & \left\langle \boxed{1} \right\rangle \end{bmatrix}$

Evolution of Another Phrase Structure Rule

C2 $\text{X} \rightarrow \text{X}+ \text{ CONJ X}$

C3 $\boxed{1} \rightarrow \boxed{1}+ \begin{bmatrix} \textit{word} \\ \text{HEAD} \quad \textit{conj} \end{bmatrix} \boxed{1}$

C4 $\begin{bmatrix} \text{VAL} \quad \boxed{1} \end{bmatrix} \rightarrow \begin{bmatrix} \text{VAL} \quad \boxed{1} \end{bmatrix}+ \begin{bmatrix} \textit{word} \\ \text{HEAD} \quad \textit{conj} \end{bmatrix} \begin{bmatrix} \text{VAL} \quad \boxed{1} \end{bmatrix}$

C5 $\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \boxed{0} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{IND} & s_0 \end{bmatrix} \end{bmatrix} \rightarrow$

$$\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \boxed{0} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{IND} & s_1 \end{bmatrix} \end{bmatrix} \ldots \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \boxed{0} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{IND} & s_{n-1} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{conj} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{IND} & s_0 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{ARGS} & \langle s_1, \ldots, s_{n-1}, s_n \rangle \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & \boxed{0} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{IND} & s_n \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} \text{VAL} & \boxed{0} \\ \text{IND} & s_0 \end{bmatrix} \rightarrow \begin{bmatrix} \text{VAL} & \boxed{0} \\ \text{IND} & s_1 \end{bmatrix} \ldots \begin{bmatrix} \text{VAL} & \boxed{0} \\ \text{IND} & s_{n-1} \end{bmatrix} \begin{bmatrix} \text{HEAD} & \textit{conj} \\ \text{IND} & s_0 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{ARGS} & \langle s_1, \ldots, s_{n-1}, s_n \rangle \end{bmatrix} \right\rangle \end{bmatrix} \begin{bmatrix} \text{VAL} & \boxed{0} \\ \text{IND} & s_n \end{bmatrix}$$

Combining Constraints and Coordination

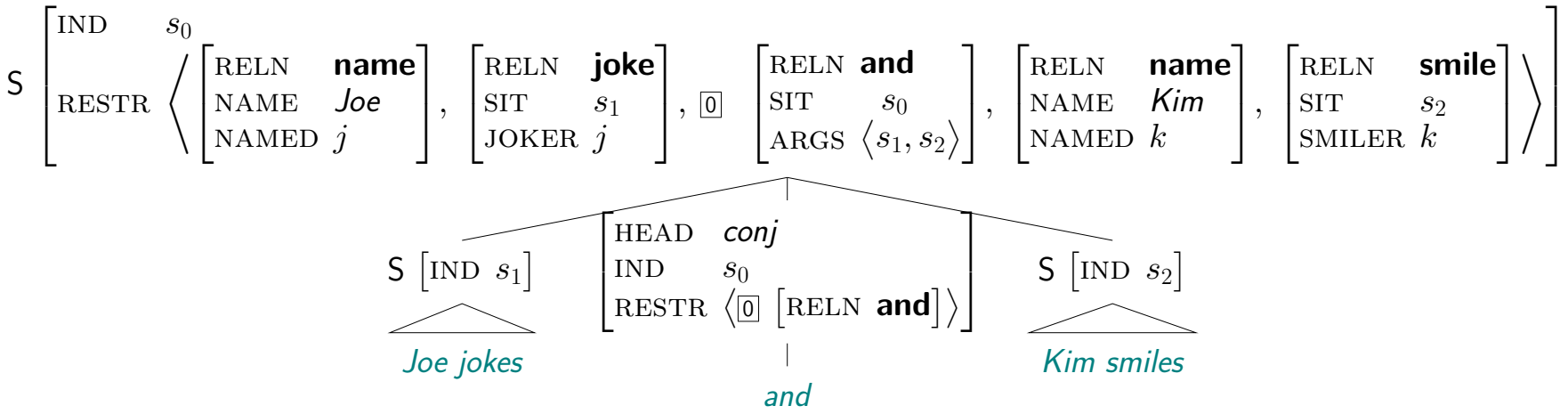
- Coordination Rule

$$\begin{bmatrix}\text{VAL} & \boxed{0}\\ \text{IND} & s_0\end{bmatrix} \rightarrow \begin{bmatrix}\text{VAL} & \boxed{0}\\ \text{IND} & s_1\end{bmatrix} \ldots \begin{bmatrix}\text{VAL} & \boxed{0}\\ \text{IND} & s_{n-1}\end{bmatrix} \begin{bmatrix}\text{HEAD} & \textit{conj}\\ \text{IND} & s_0\\ \text{RESTR} & \left\langle \left[\text{ARGS} \left\langle s_1, \ldots, s_{n-1}, s_n \right\rangle\right]\right\rangle\end{bmatrix} \begin{bmatrix}\text{VAL} & \boxed{0}\\ \text{IND} & s_n\end{bmatrix}$$

- Lexical Entry for ***and***

$$\left\langle \textit{and}\ , \begin{bmatrix}\text{SYN} & \left[\text{HEAD} \ \textit{conj}\right]\\ \text{SEM} & \begin{bmatrix}\text{MODE} & \textit{none}\\ \text{IND} & s\\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \textbf{and}\\ \text{SIT} & s\end{bmatrix}\right\rangle\end{bmatrix}\end{bmatrix}\right\rangle$$

Combining Constraints and Coordination



How to Read the Coordination Rule

- The rule is not headed, so the SIP doesn't apply: the rule itself has to say what the mother's IND is
  - The SCP still applies: the mother's RESTR is all the daughters' RESTRs added together
- $\boxed{0}$: the mother and all the conjuncts have the same VAL — you can only coordinate things that need the same things
- The mother's IND s_0 is the conjunction's IND
- The conjunction's predication links the conjuncts: its ARGS are their indices $\langle s_1, \ldots, s_n \rangle$

Where Does Each Index Come From?

In *Joe jokes and Kim smiles*:

- s_1 and s_2?
- s_0?
- $\textsc{args}\ \langle s_1, s_2 \rangle$?
- The whole $\textsc{restr}$ list?

Where Does Each Index Come From?

In *Joe jokes and Kim smiles*:

- s_1 and s_2? From ***jokes*** and ***smiles***, passed up to each S by the SIP
- s_0? From the lexical entry for ***and***; the Coordination Rule makes it the mother's IND
- ARGS $\langle s_1, s_2 \rangle$? The Coordination Rule: the indices of the two conjuncts
- The whole RESTR list? The SCP: everything the daughters' RESTRs contain, in order

Structural Ambiguity

Structural Ambiguity

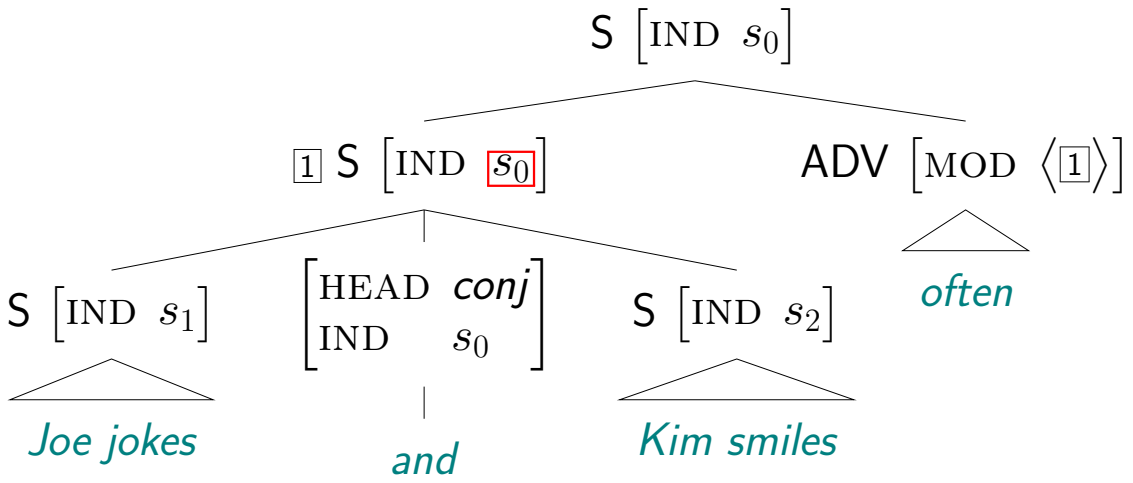
Joe jokes and Kim smiles often has two readings:

I ***often*** modifies the whole coordination
It is often the case that Joe jokes and Kim smiles

II ***often*** modifies only *Kim smiles*
Joe jokes, and Kim often smiles

Our grammar gives each reading its own tree, and the two trees build different RESTR lists

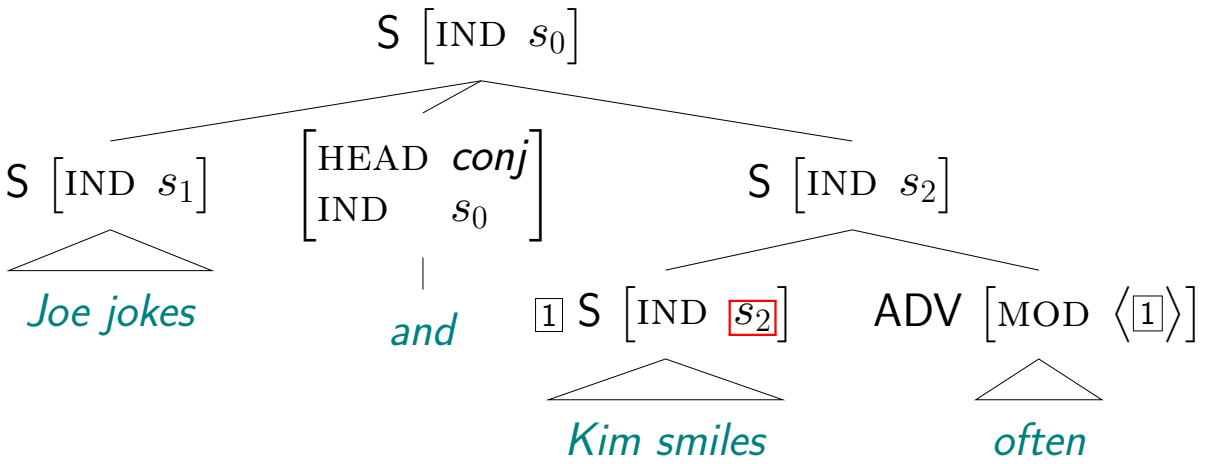
Tree I: *often* Modifies the Coordination



The top S:

$$\left[\begin{array}{ll} \text{IND} & s_0 \\ \text{MODE} & \textit{prop} \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{name} \\ \text{NAME} & \textit{Joe} \\ \text{NAMED} & j \end{array}\right], \left[\begin{array}{ll} \text{RELN} & \textbf{joke} \\ \text{SIT} & s_1 \\ \text{JOKER} & j \end{array}\right], \left[\begin{array}{ll} \text{RELN} & \textbf{and} \\ \text{SIT} & s_0 \\ \text{ARGS} & \langle s_1, s_2 \rangle \end{array}\right], \left[\begin{array}{ll} \text{RELN} & \textbf{name} \\ \text{NAME} & \textit{Kim} \\ \text{NAMED} & k \end{array}\right], \left[\begin{array}{ll} \text{RELN} & \textbf{smile} \\ \text{SIT} & s_2 \\ \text{SMILER} & k \end{array}\right], \left[\begin{array}{ll} \text{RELN} & \textbf{often} \\ \text{SIT} & s_3 \\ \text{ARG} & \boxed{s_0} \end{array}\right] \right\rangle \end{array}\right]$$

Tree II: *often* Modifies *Kim smiles*



The top S:

$$\begin{bmatrix} \text{IND} & s_0 \\ \text{MODE} & \textit{prop} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \textit{Joe} \\ \text{NAMED} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{joke} \\ \text{SIT} & s_1 \\ \text{JOKER} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{and} \\ \text{SIT} & s_0 \\ \text{ARGS} & \langle s_1, s_2 \rangle \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \textit{Kim} \\ \text{NAMED} & k \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{smile} \\ \text{SIT} & s_2 \\ \text{SMILER} & k \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{often} \\ \text{SIT} & s_3 \\ \text{ARG} & \boxed{s_2} \end{bmatrix} \right\rangle \end{bmatrix}$$

Question About Structural Ambiguity

Why isn't this a possible semantic representation for the string *Joe jokes and Kim smiles often*?

$$\begin{bmatrix} \text{IND} & s_0 \\ \text{MODE} & \textit{prop} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \textit{Joe} \\ \text{NAMED} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{joke} \\ \text{SIT} & s_1 \\ \text{JOKER} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{and} \\ \text{SIT} & s_0 \\ \text{ARGS} & \langle s_1, s_2 \rangle \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \textit{Kim} \\ \text{NAMED} & k \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{smile} \\ \text{SIT} & s_2 \\ \text{SMILER} & k \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{often} \\ \text{SIT} & s_3 \\ \text{ARG} & \boxed{s_1} \end{bmatrix} \right\rangle \end{bmatrix}$$

Question About Structural Ambiguity

Why isn't this a possible semantic representation for the string *Joe jokes and Kim smiles often*?

$$\begin{bmatrix} \text{IND} & s_0 \\ \text{MODE} & \textit{prop} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \textit{Joe} \\ \text{NAMED} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{joke} \\ \text{SIT} & s_1 \\ \text{JOKER} & j \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{and} \\ \text{SIT} & s_0 \\ \text{ARGS} & \langle s_1, s_2 \rangle \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{name} \\ \text{NAME} & \textit{Kim} \\ \text{NAMED} & k \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{smile} \\ \text{SIT} & s_2 \\ \text{SMILER} & k \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{often} \\ \text{SIT} & s_3 \\ \text{ARG} & \boxed{s_1} \end{bmatrix} \right\rangle \end{bmatrix}$$

often's ARG is s_1, the index of *Joe jokes*. But ***often*** can only modify a phrase it combines with, and the only ones next to it are *Kim smiles* (s_2) and the whole coordination (s_0). No tree puts ***often*** with *Joe jokes* alone.

Some Standard Extensions

- Quantification
  - typically expressed as restrictions on scope
  - Minimal Recursion Semantics goes further
- Pragmatics
  - typically expressed as another feature: CONTEXT
  - contains things like *speaker*, *hearer*, *audience*
  - used for pronominal reference, politeness

What About Scope?

Every dog chases a cat has two readings:

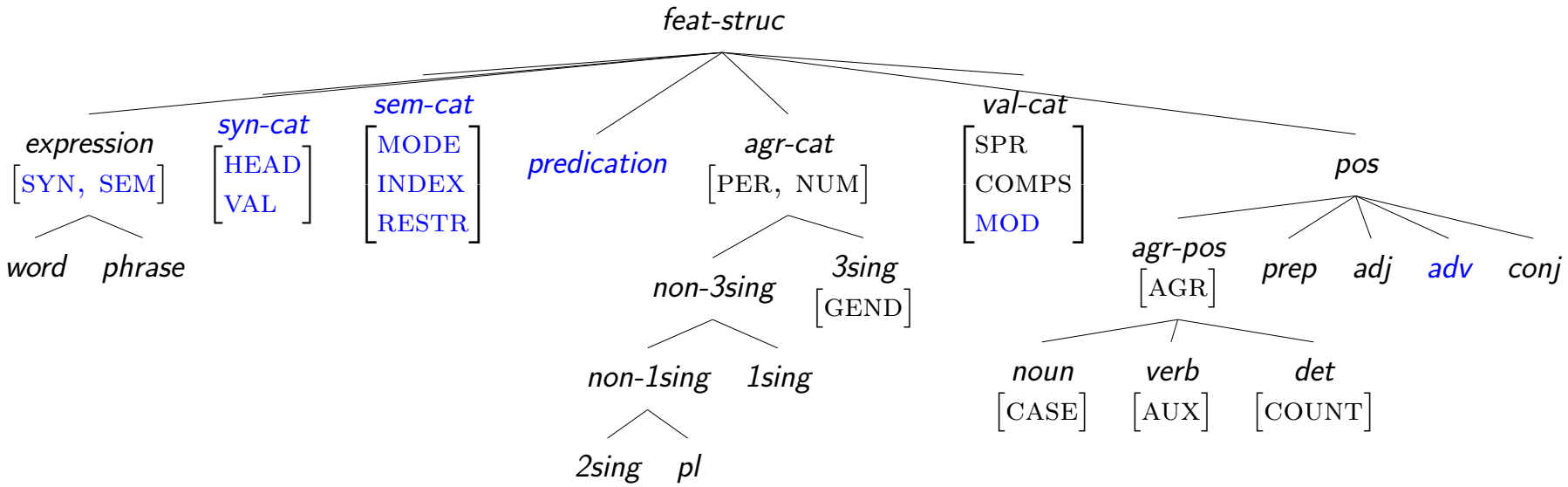
- Each dog chases a cat, maybe a different cat for each dog
- There is one particular cat, and every dog chases it

The difference is which quantifier is "on top" (has **wide scope**)

- We won't deal with scope in this course
- Implemented HPSG grammars (with Minimal Recursion Semantics) leave it **underspecified**: one representation that covers both readings
- Usually we don't need to decide which reading is meant

Building a Tree (Updated)

Chapter 5 Type Hierarchy



Algorithm for Building a Tree (Updated)

Create the words (leaf nodes)

1. Copy all the information from each word's lexical entry, including its SEM
2. Enforce the SHAC (on verbs and common nouns)
3. Check the type hierarchy and add every appropriate feature (leave the value blank if it is underspecified)

Create the phrases

4. Find a rule that fits; add everything it says
5. Apply the principles (HFP, Valence Principle, SIP, SCP) to update features
6. Repeat 4–5 until you reach the S node
7. Check: every feature you left blank should now have a value

Problems

Problem 5.1: Two Kinds of Modifiers in English

In English, modifiers of nouns can appear either before or after the noun, although any given modifier is usually restricted to one position or the other.

(i) *The red dog on the roof*

(ii) **The on the roof dog*

(iii) **The dog red*

Our current Head-Modifier Rule only licenses post-head modifiers (like *on the roof* in (i)).

Problem 5.1: Two Kinds of Modifiers (cont.)

A. Write a second Head-Modifier Rule that licenses pre-head modifiers (e.g., *red* in (i)).

B. Modify the Head-Modifier 1 and Head-Modifier 2 Rules so that they are sensitive to which kind of modifier is present and don't generate (ii) or (iii). [*Hint: Use a feature* [*POST-HEAD* $\{+,-\}$] *to distinguish red and on the roof.*]

C. Is POST-HEAD a HEAD feature? Why or why not?

D. Give lexical entries for *red* and *on* that show the value of POST-HEAD. (You may omit the SEM features in these entries.)

E. Is (i) ambiguous according to your grammar (i.e. the Chapter 5 grammar modified to include the two Head-Modifier Rules, instead of just one)? Explain your answer.

Problem 5.1: Two Kinds of Modifiers (cont.)

This problem assumed that we don't want to make the two Head-Modifier Rules sensitive to the part of speech of the modifier. One reason for this is that modifiers of the same part of speech can occur before and after the head, even though individual modifiers might be restricted to one position or the other.

F. Provide three examples of English NPs with adjectives or APs after the noun.

G. Provide three examples of adverbs that can come before the verbs they modify.

H. Provide three examples of adverbs that can come after the verbs they modify.

Problem 5.3: Semantics of Number Names

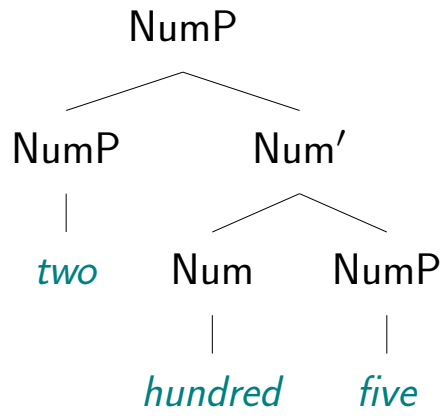
In Chapter 3, we considered the syntax of English number names, and in particular how to find the head of a number name expression. Based on the results of that problem, the lexical entry for *hundred* in a number name like *two hundred five* should include the constraints in (i). (Here we are assuming a new subtype of *pos*, *number*, which is appropriate for number name words.)

(i) $$\left\langle \textit{hundred} \ , \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{number} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle [\text{HEAD} \ \textit{number}] \rangle \\ \text{COMPS} & \langle [\text{HEAD} \ \textit{number}] \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle$$

Problem 5.3: Semantics of Number Names (cont.)

This lexical entry interacts with our ordinary Head-Complement and Head-Specifier Rules to give us the phrase structure shown in (ii):

(ii)



Problem 5.3: Semantics of Number Names (cont.)

Smith (1999) provides a compositional semantics of number names. The semantics of this NP should be (iii):

(iii)
$$\begin{bmatrix} \text{INDEX} & i \\ \text{MODE} & \textit{ref} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{constant} \\ \text{INST} & l \\ \text{VALUE} & \textit{2} \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{times} \\ \text{RESULT} & k \\ \text{FACTOR1} & l \\ \text{FACTOR2} & m \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{constant} \\ \text{INST} & m \\ \text{VALUE} & \textit{100} \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{plus} \\ \text{RESULT} & i \\ \text{TERM1} & j \\ \text{TERM2} & k \end{bmatrix}, \begin{bmatrix} \text{RELN} & \textbf{constant} \\ \text{INST} & j \\ \text{VALUE} & \textit{5} \end{bmatrix} \right\rangle \end{bmatrix}$$

This expresses "(two times one hundred) plus five" (i.e. 205) as a FS.

Problem 5.3: Semantics of Number Names (cont.)

A. Assume that the two constant predications with the values 2 and 5 are contributed by the lexical entries for *two* and *five*. What predications must be on the RESTR list of the lexical entry for *hundred* in order to build (iii) as the SEM value of *two hundred five*?

B. The lexical entry for *hundred* will identify the indices of its specifier and complement with the value of some feature of a predication on its RESTR list. Which feature of which predication is the index of the specifier identified with? What about the index of the complement?

C. The lexical entry for *hundred* will identify its own INDEX with the value of some feature of some predication on its RESTR list. Which feature of which predication must this be, in order for the grammar to build (iii) as the SEM value of *two hundred five*?

Problem 5.3: Semantics of Number Names (cont.)

D. Based on your answers in parts (A)–(C), give a lexical entry for *hundred* that includes the constraints in (i) and a fully specified SEM value. [*Note: Your lexical entry need only account for hundred as it is used in two hundred five. Don't worry about other valence possibilities, such as two hundred, two hundred and five, or a hundred.*]

E. The syntax and semantics of number names do not line up neatly: In the syntax, *hundred* forms a constituent with *five*, and *two* combines with *hundred five* to give a larger constituent. In the semantics, the constant predications with the values 2 and 100 are related via the times predication. The result of that is related to the constant predication with the value 5, via the plus predication. Why is this mismatch not a problem for the grammar?

Summary

- We model how the meanings of the parts add up (**compositionality**), not pragmatics or fine word meanings
- SEM: MODE (what kind of meaning), INDEX (what it is about) and RESTR (the facts that must hold)
- Words link their syntactic arguments to semantic roles ($\text{NP}_i = \text{SLEEPER}\ i$)
- SIP: a phrase is about what its head is about
  SCP: a phrase means everything its parts mean
- Coordination has no head, so its rule states the semantics
- Structurally ambiguous strings get different RESTR lists

Acknowledgments and References

- Course design and slides borrow heavily from Emily Bender's course: *Linguistics 566: Introduction to Syntax for Computational Linguistics* http://courses.washington.edu/ling566

- This lecture complements Sag et al. (2003, Chapter 5)

*

References

Ivan A. Sag, Tom Wasow, and Emily Bender. 2003. *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, second edition.