JMORF/MORS — Morpho-Syntax

Complex Feature Values
Valence, Agreement, Case

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

Lecture 4
Location: KB-1.11

Overview

- List-valued valence features
  - A problem with the Chapter 3 grammar
  - Generalize COMPS and SPR
  - The Valence Principle
- The heads of S and NP
- Complements and modifiers
- Agreement: the SHAC and count/mass
- Building a tree, step by step
- Problems 4.1, 4.5, 4.6

Review

Pizza review

- Unification is an operation for combining constraints from different sources.
- What are those sources in the pizza example?
- Why do we need to combine information from different sources in our grammars?

Pizza review

- Unification is an operation for combining constraints from different sources.
- What are those sources in the pizza example?
  The preferences of the different people sharing the pizza.
- Why do we need to combine information from different sources in our grammars?
  A tree has to satisfy the lexical entries, the grammar rules *and* the general principles (like the HFP) all at once. None of them builds the tree on its own.

Reminder: Where We Are

- Attempting to model English with CFG led to problems with the granularity of categories, e.g.
  - Need to distinguish various subtypes of verbs
  - Need to identify properties common to all verbs
- So we broke categories down into feature structures and began constructing a hierarchy of types of feature structures.
- This allows us to schematize rules and state cross-categorial generalizations, while still making fine distinctions.

Heads

- Intuitive idea: A phrase typically contains a word that determines its most essential properties, including
  - where it occurs in larger phrases
  - what its internal structure is
- This is called the **head**
- The term **head** is used both for the head word in a phrase and for all the intermediate phrases containing that word
- NB: Not all phrases have heads (e.g. coordination: ***Kim and Sandy***)

Formalizing the Notion of Head

- Expressions have a feature HEAD
- HEAD's values are of type *pos*
- For HEAD values of type *agr-pos*, HEAD's value also includes the feature AGR
- Well-formed trees are subject to the **Head Feature Principle**:

  In any headed phrase, the HEAD value of the mother and the head daughter must be identical.

A Tree is Well-Formed if ...

- It and each subtree are licensed by a grammar rule or lexical entry
- All general principles (like the HFP) are satisfied
- NB: Trees are part of our model of the language, so all their features have values (even though we will often be ~~lazy~~ compact and leave out the values irrelevant to our current point).

List-Valued Valence Features

But it's still not quite right ...

There's still too much redundancy: the rules and features encode the same information in different ways.

Rule 1: $\begin{bmatrix} \textit{phrase} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \textit{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \textit{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$

Rule 2: $\begin{bmatrix} \textit{phrase} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \textit{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \textit{str} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$ NP

Rule 3: $\begin{bmatrix} \textit{phrase} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \textit{itr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} \\ \text{VAL} \begin{bmatrix} \text{COMPS} & \textit{dtr} \\ \text{SPR} & - \end{bmatrix} \end{bmatrix}$ NP NP

The value *str* and the single NP in Rule 2 say the same thing twice: "this verb takes one object". Likewise *dtr* and the two NPs in Rule 3.

Solution: More Elaborate Valence Feature Values

- The rules just say that heads combine with whatever their lexical entries say they can (or must) combine with.

- The information about what a word can or must combine with is encoded in list-valued valence features.
  - The elements of the lists are themselves feature structures
  - The elements are "cancelled" off the lists once heads combine with their complements and specifiers.

How to Read a Rule

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

- Left of the arrow is the mother; on the right are the daughters, in the order they are pronounced
- **H** marks the **head daughter**
- A boxed number (a **tag**) means the very same thing: $\boxed{1}$ on the COMPS list and the daughter $\boxed{1}$ are one object
- $\langle\rangle$ is the empty list: nothing left to find
- Out loud: "A word that needs complements $\boxed{1}$ to $\boxed{n}$, followed by $\boxed{1}$ to $\boxed{n}$, makes a phrase that needs no more complements"

Head-Complement Rule (Final Version)

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

- This allows for arbitrary numbers of complements, but only applies when there is at least one.
  - The possible complements are specified lexically
  - Heads in English probably never have more than 3 or 4 complements
  - This doesn't apply where Head-Complement Rule 1 would. (Why?)
- This can cover lots of cases not covered by the old Head-Complement Rules 1–3. (Examples?)

Head-Complement Rule (Final Version)

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

- This allows for arbitrary numbers of complements, but only applies when there is at least one.
  - The possible complements are specified lexically
  - Heads in English probably never have more than 3 or 4 complements
  - This doesn't apply where Head-Complement Rule 1 would. (Why?)
    We don't need it: a word with an empty COMPS list can be the head of the Head-Specifier Rule directly.

- This can cover lots of cases not covered by the old Head-Complement Rules 1–3. (Examples?)
  put $\langle \text{NP PP} \rangle$, ***fond*** $\langle \text{PP} \rangle$, ***bet*** $\langle \text{NP NP S} \rangle$, ***tell*** $\langle \text{NP S} \rangle$ ...

The valence comes from the lexicon

$$\left\langle \text{devour}, \begin{bmatrix} \textit{word} \\ \textsc{head} & \textit{verb} \\ \textsc{val} & \left[\textsc{comps} \left\langle \text{NP} \right\rangle\right] \end{bmatrix} \right\rangle \quad \left\langle \text{put}, \begin{bmatrix} \textit{word} \\ \textsc{head} & \textit{verb} \\ \textsc{val} & \left[\textsc{comps} \left\langle \text{NP PP} \right\rangle\right] \end{bmatrix} \right\rangle$$

$$\left\langle \text{eat}, \begin{bmatrix} \textit{word} \\ \textsc{head} & \textit{verb} \\ \textsc{val} & \left[\textsc{comps} \left\langle (\text{NP}) \right\rangle\right] \end{bmatrix} \right\rangle \quad \left\langle \text{bet}, \begin{bmatrix} \textit{word} \\ \textsc{head} & \textit{verb} \\ \textsc{val} & \left[\textsc{comps} \left\langle \text{NP } (\text{NP}) (\text{S}) \right\rangle\right] \end{bmatrix} \right\rangle$$

$$\left\langle \text{dine}, \begin{bmatrix} \textit{word} \\ \textsc{head} & \textit{verb} \\ \textsc{val} & \left[\textsc{comps} \left\langle \right\rangle\right] \end{bmatrix} \right\rangle \quad \left\langle \text{fond}, \begin{bmatrix} \textit{word} \\ \textsc{head} & \textit{adjective} \\ \textsc{val} & \left[\textsc{comps} \left\langle \text{PP}\textit{:of} \right\rangle\right] \end{bmatrix} \right\rangle$$

Question: What if English had postpositions?

How would the grammar change if English had postpositions (***the bushes from***) instead of prepositions?

Question: What if English had postpositions?

How would the grammar change if English had postpositions (***the bushes from***) instead of prepositions?

Split the Head-Complement Rule into a head-first and a head-last version:

$$\begin{bmatrix} \textit{phrase} \\ \text{VAL} \ \begin{bmatrix} \text{COMPS} \ \langle\rangle \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} \\ \text{HEAD} \ \ \textit{verb} \mid \textit{adj} \mid \textit{noun} \\ \text{VAL} \ \ \begin{bmatrix} \text{COMPS} \ \langle \boxed{1}, \ \ldots, \ \boxed{n} \rangle \end{bmatrix} \end{bmatrix} \ \boxed{1}, \ \ldots, \ \boxed{n}$$

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

The lexical entries stay the same: only word order is in the rules.

Specifiers

In English, nouns place constraints on their specifiers.

- In Number:

  (1) *This dog barked.*
  (2) **This dogs barked.*
  (3) **These dog barked.*
  (4) *These dogs barked.*

- In Countability

  (5) *Much furniture was broken.*
  (6) **A furniture was broken.*
  (7) **Much chair was broken.*
  (8) *A chair was broken.*

Head-Specifier Rule (Version I)

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

- Combines the rules expanding S and NP.
- In principle also generalizes to other categories.
- Question: Why is SPR list-valued?

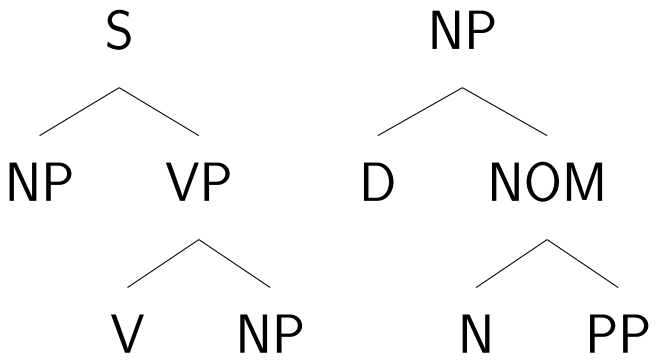
Head-Specifier Rule (Version I)

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

- Combines the rules expanding S and NP.
- In principle also generalizes to other categories.
- Question: Why is SPR list-valued?
  For consistency with COMPS: an empty list always means "satisfied" (**saturated**).
  It is a design choice, not a deep fact about language.

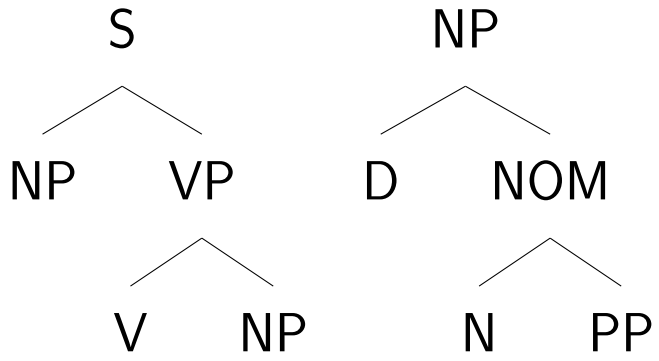
Question: Why Right-Branching?

Why are these right-branching? That is, what formal property of our grammar forces the COMPS to be lower in the tree than the SPR?



Question: Why Right-Branching?

Why are these right-branching? That is, what formal property of our grammar forces the COMPS to be lower in the tree than the SPR?



1. The head daughter of the Head-Complement Rule must be a *word*, so complements attach first, right next to the head

2. The head daughter of the Head-Specifier Rule must be COMPS $\langle\rangle$, so the specifier can only attach once all the complements are found

Another Question ...

- What determines the VAL value of phrasal nodes?

- **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.

More on the Valence Principle

- Intuitively, the VAL features list the contextual requirements that haven't yet been found.
- This way of thinking about it (like talk of "cancellation") is bottom-up and procedural.
- But formally, the Valence Principle (like the rest of our grammar) is just a well-formedness constraint on trees, without inherent directionality.

The Valence Principle in Action

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

- The Head-Complement Rule says nothing about SPR
  - so the mother's SPR is the same as the head daughter's
  - ***devour*** has SPR $\langle \text{NP} \rangle$, so ***devour the pizza*** does too
- It *does* give COMPS for both mother and head daughter
  - so here they can differ: this is what "unless the rule says otherwise" means

Why Do We Want VAL on All Nodes?

- Why not leave SPR out for prepositions, say?
- Then a preposition's SPR would be unspecified, which is compatible with anything: it could go through the Head-Specifier Rule and pick up a specifier
- We would license lots of trees we don't want
- There is no magic in syntax:
  - a rule doesn't fail to apply because we can see it is irrelevant
  - we have to enforce the right behaviour with constraints

So far, we have:

- Replaced atomic-valued VAL features with list-valued ones.
- Generalized Head-Complement and Head-Specifier rules, to say that heads combine with whatever their lexical entries say they should combine with.
- Introduced the Valence Principle to keep the information on the COMPS and SPR lists until it gets "cancelled" by the Head-Complement and Head-Specifier rules.

The Heads of S and NP

The Parallelism between S and NP

- Motivation:
  - pairs like
    (9) *Chris lectured about syntax.*
    (10) *Chris's lecture about syntax*
  - both S and NP exhibit agreement
    (11) *The bird sings/*sing.*
    (12) *The birds sing/*sings.*
    (13) *this/*these bird*
    (14) *these/*this birds*

- So we treat NP as the saturated category of type *noun* and S as the saturated category of type *verb*.
  (**saturated** = both SPR and COMPS are empty)

Abbreviations

NP, VP, S ... are now just short names for feature structures:

	noun	*verb*
a single word	$\text{N} = \begin{bmatrix} \textit{word} \\ \text{HEAD} \quad \textit{noun} \end{bmatrix}$	$\text{V} = \begin{bmatrix} \textit{word} \\ \text{HEAD} \quad \textit{verb} \end{bmatrix}$
needs a specifier	$\text{NOM} = \begin{bmatrix} \text{HEAD} \quad \textit{noun} \\ \text{VAL} \quad \begin{bmatrix} \text{COMPS} \quad \langle\rangle \\ \text{SPR} \quad \langle \text{X} \rangle \end{bmatrix} \end{bmatrix}$	$\text{VP} = \begin{bmatrix} \text{HEAD} \quad \textit{verb} \\ \text{VAL} \quad \begin{bmatrix} \text{COMPS} \quad \langle\rangle \\ \text{SPR} \quad \langle \text{X} \rangle \end{bmatrix} \end{bmatrix}$
saturated	$\text{NP} = \begin{bmatrix} \text{HEAD} \quad \textit{noun} \\ \text{VAL} \quad \begin{bmatrix} \text{COMPS} \quad \langle\rangle \\ \text{SPR} \quad \langle\rangle \end{bmatrix} \end{bmatrix}$	$\text{S} = \begin{bmatrix} \text{HEAD} \quad \textit{verb} \\ \text{VAL} \quad \begin{bmatrix} \text{COMPS} \quad \langle\rangle \\ \text{SPR} \quad \langle\rangle \end{bmatrix} \end{bmatrix}$

X is any feature structure. None of these say *phrase*: ***walks*** is a VP *and* a V.

Any other reason to treat V as the head of S?

- In standard English, sentences must have verbs.
  (How about non-standard English or other languages?)

- Verbs taking S complements can influence the form of the verb in the complement:

  (15) *I insist/*recall (that) you be here on time.*

- Making V the head of S helps us state such restrictions formally

A possible formalization of the restriction

$$\left\langle \text{insist}, \begin{bmatrix} \textit{word} \\ \textsc{head} & \textit{verb} \\ \textsc{val} & \begin{bmatrix} \textsc{spr} & \langle \textsc{np} \rangle \\ \textsc{comps} & \langle \textsc{s} \begin{bmatrix} \textsc{head} & \begin{bmatrix} \textsc{mood} & \textit{subjunctive} \end{bmatrix} \end{bmatrix} \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Note that this requires that the verb be the head of the complement. We don't have access to the features of the other constituents of the complement.

$$\text{Recall: } \textsc{s} = \begin{bmatrix} \textsc{head} & \textit{verb} \\ \textsc{val} & \begin{bmatrix} \textsc{spr} & \langle\rangle \\ \textsc{comps} & \langle\rangle \end{bmatrix} \end{bmatrix}$$

Complements and Modifiers

Complements vs. Modifiers

- Intuitive idea: **Arguments** (specifiers and complements) introduce essential participants in the situation denoted; modifiers refine the description.

- Generally accepted distinction, but disputes over individual cases.

- Linguists rely on heuristics to decide how to analyze questionable cases (usually PPs).

Heuristics for Complements vs. Modifiers

- Obligatory PPs are usually complements.
- Temporal & locative PPs are usually modifiers.
- An entailment test:
  If *X Ved (NP) PP* $\not\Rightarrow$ *X did something PP*, then the PP is a complement.

  (16) *Pat relied on Chris* $\not\Rightarrow$ *Pat did something on Chris*
  (17) *Pat put nuts in a cup* $\not\Rightarrow$ *Pat did something in a cup*
  (18) *Pat slept until noon* $\Rightarrow$ *Pat did something until noon*
  (19) *Pat ate lunch at the canteen* $\Rightarrow$ *Pat did something at the canteen*

Agreement

Agreement

- Two kinds so far (namely?)
- Both initially handled via stipulation in the Head-Specifier Rule
- But if we want to use this rule for categories that don't have the AGR feature (such as PPs and APs, in English), we can't build it into the rule.

Agreement

- Two kinds so far (namely?)
  subject–verb (***the birds sing***) and determiner–noun (***these birds***)

- Both initially handled via stipulation in the Head-Specifier Rule

- But if we want to use this rule for categories that don't have the AGR feature (such as PPs and APs, in English), we can't build it into the rule.

The Specifier-Head Agreement Constraint (SHAC)

Verbs and common nouns must be specified as:

$$\begin{bmatrix} \textsc{head} & \left[\textsc{agr}\ \boxed{1}\right] \\ \textsc{val} & \left[\textsc{spr}\ \left\langle \left[\textsc{head}\ \left[\textsc{agr}\ \boxed{1}\right]\right]\right\rangle\right] \end{bmatrix}$$

- Why is this lexical?

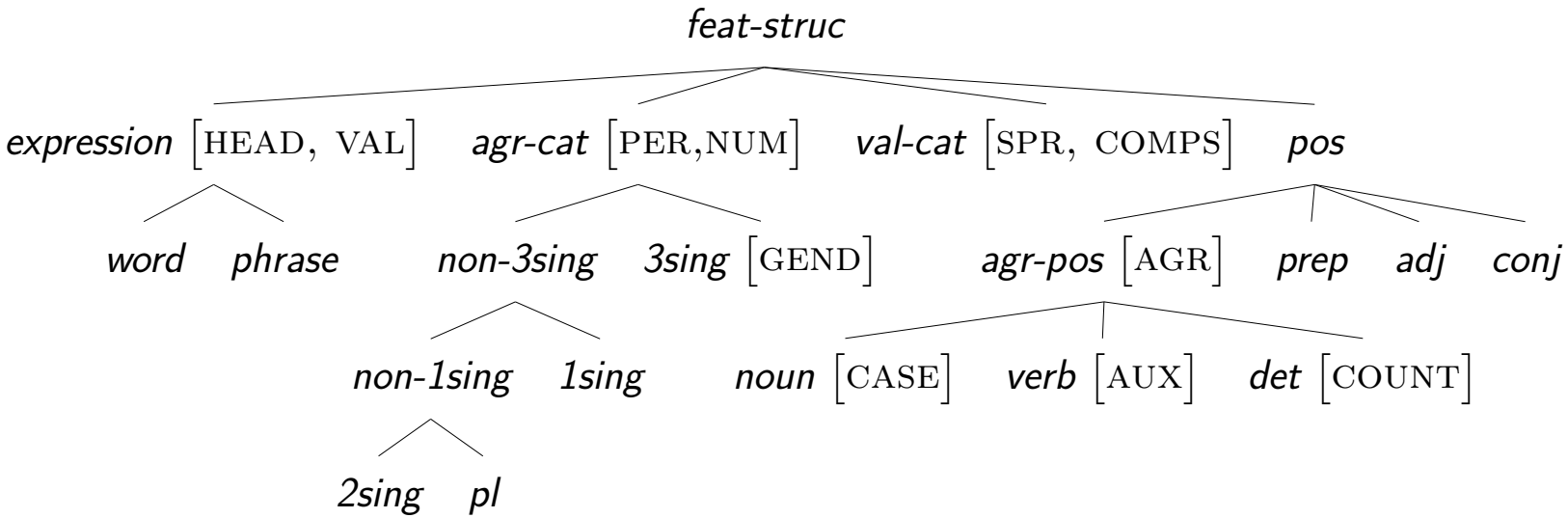
The Specifier-Head Agreement Constraint (SHAC)

Verbs and common nouns must be specified as:

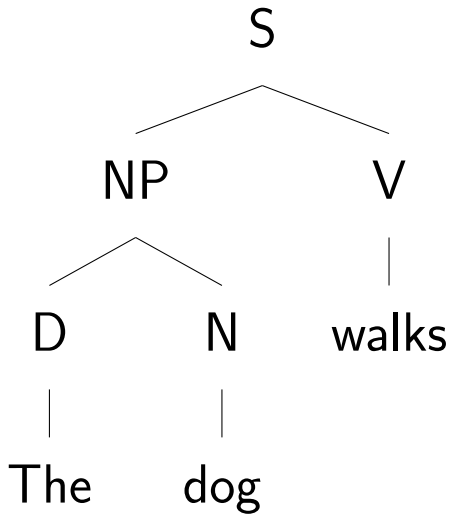
$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

- Why is this lexical?
  So the Head-Specifier Rule can stay general. Prepositions and adjectives also take specifiers (***With [them on our team] ...***), but they have no AGR, so agreement can't be stated in the rule.

Chapter 4 Type Hierarchy



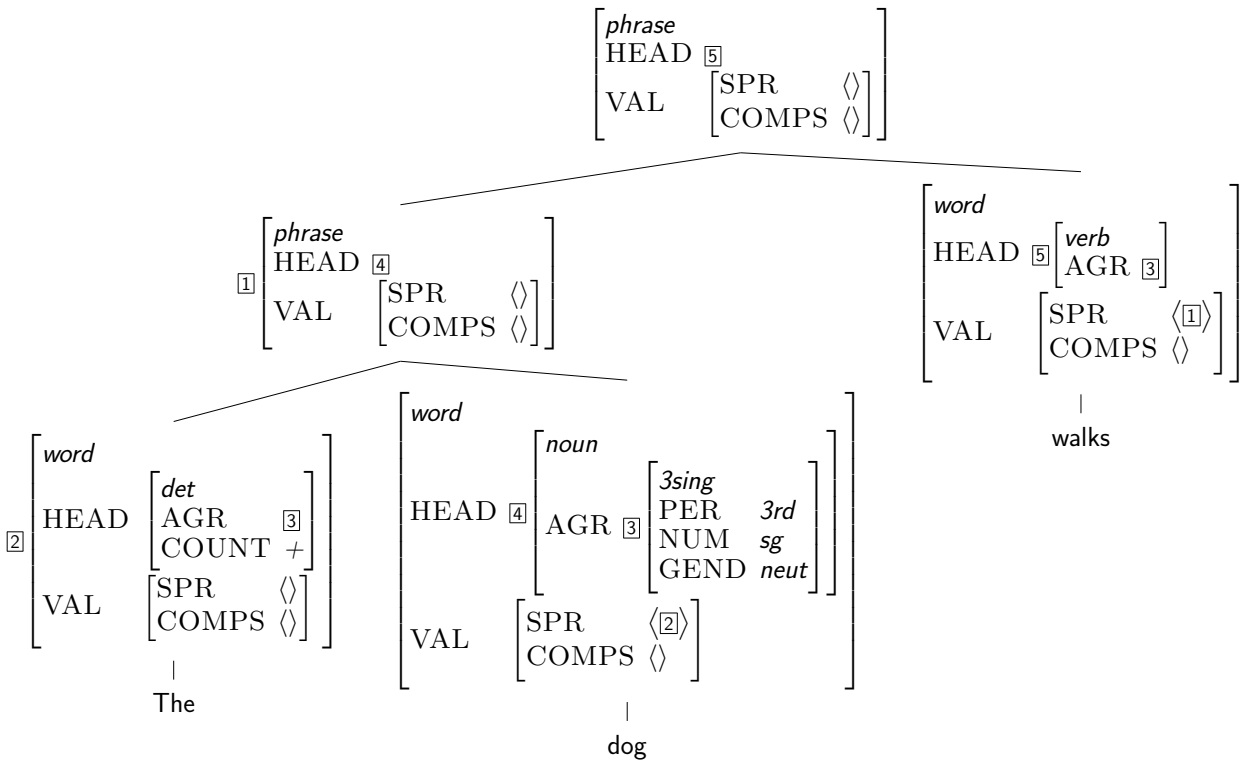
The dog walks: The Big Picture



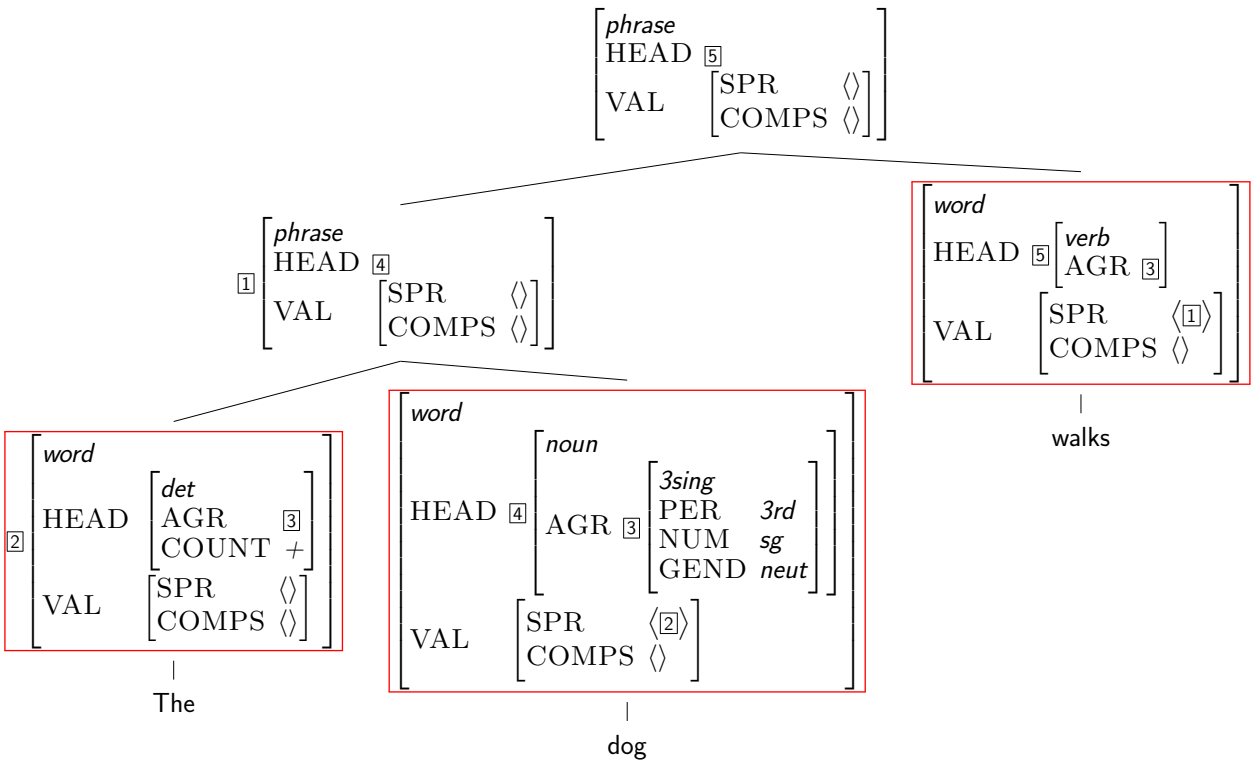
- Each label is an abbreviation for a feature structure
- There is no VP node: ***walks*** is a word with COMPS $\langle\rangle$, so it can be the head of the Head-Specifier Rule directly
- Next: the same tree with all the features filled in

The dog walks: The Full Tree

What licenses each node?



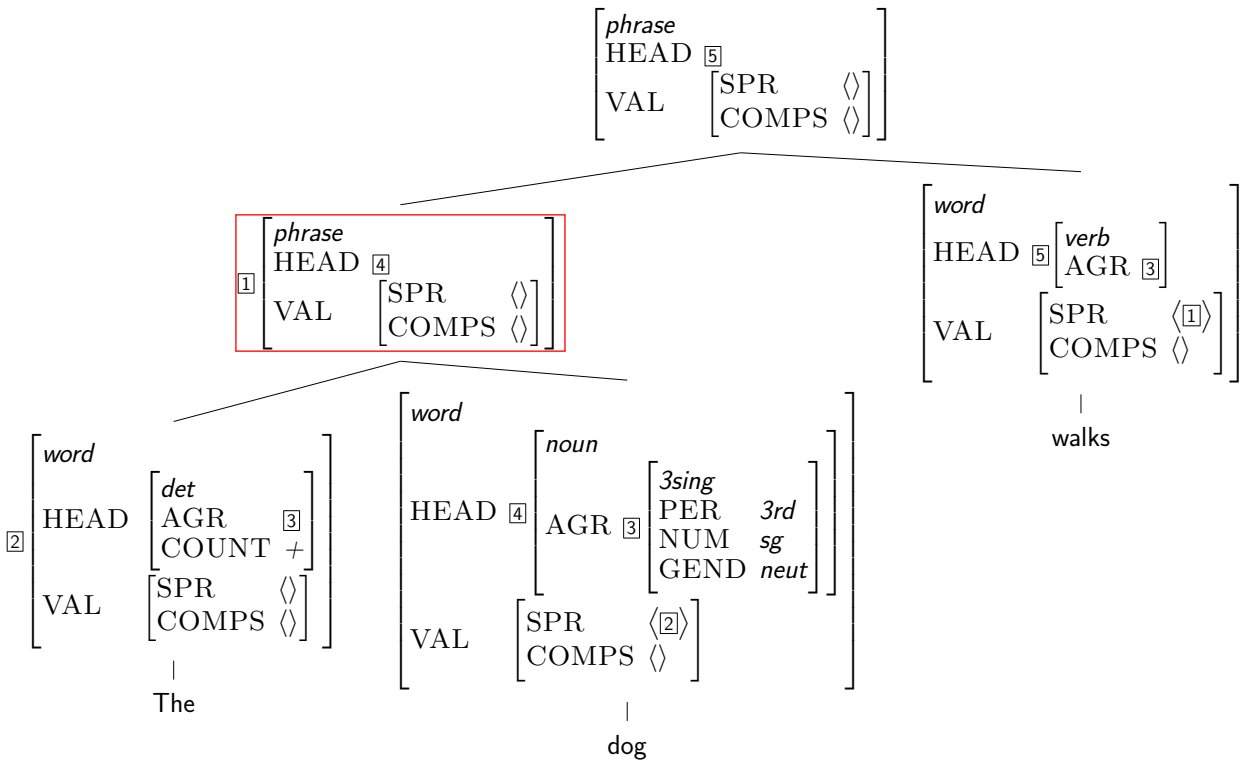
The dog walks: The Full Tree



What licenses each node?

- The three words: their lexical entries, plus the SHAC for ***dog*** and ***walks***

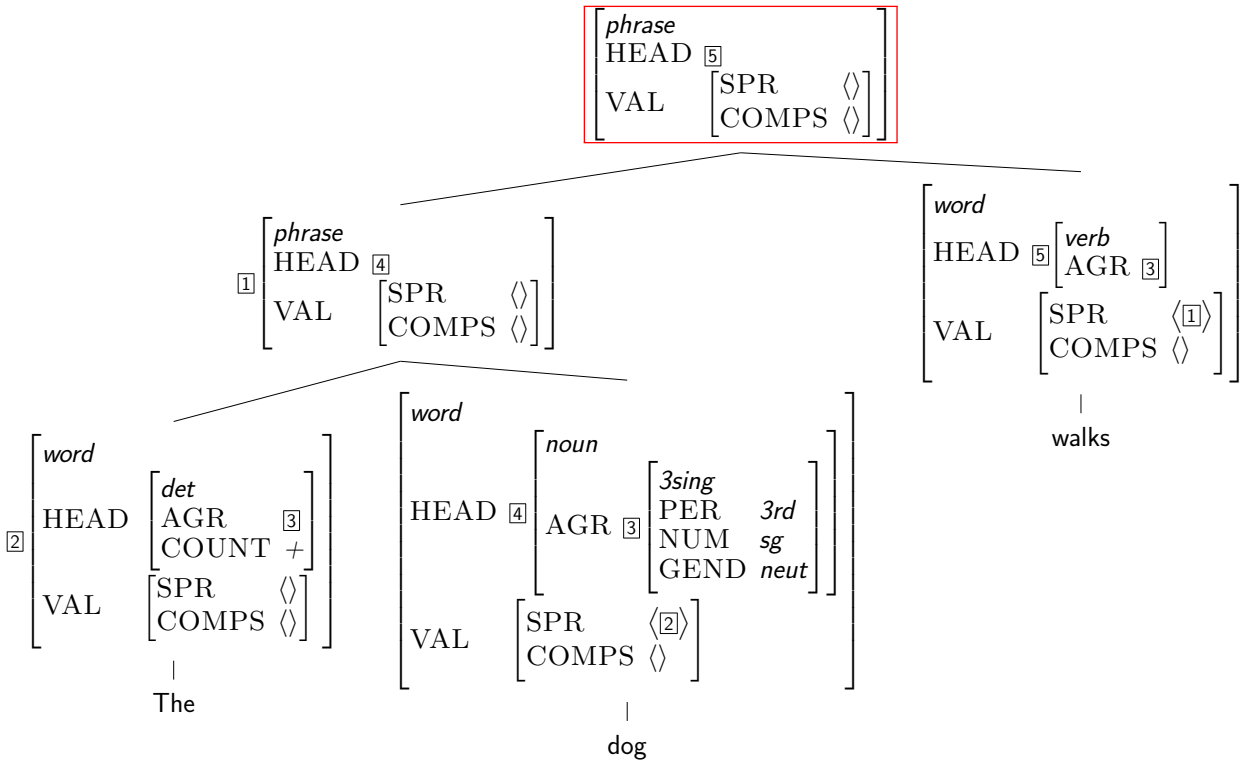
The dog walks: The Full Tree



What licenses each node?

- The three words: their lexical entries, plus the SHAC for ***dog*** and ***walks***
- ***the dog***: the Head-Specifier Rule, with ***dog*** as head; HFP and Valence Principle

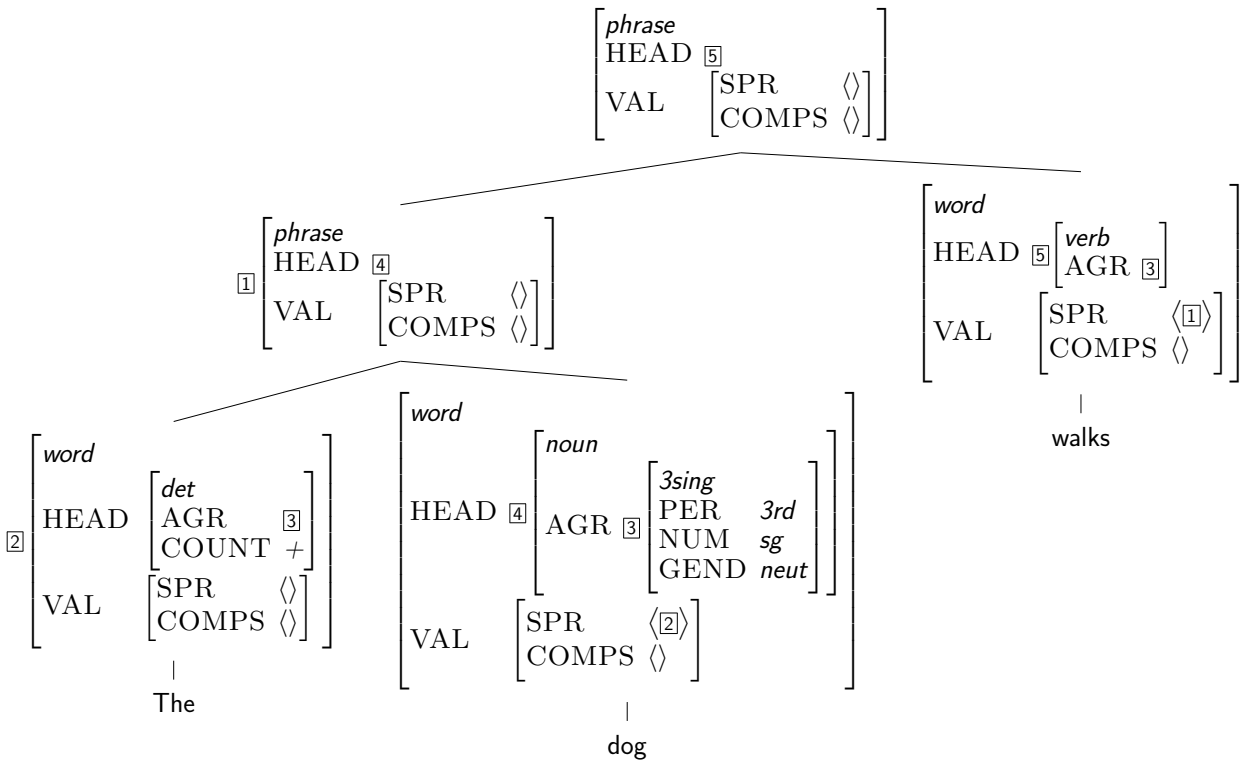
The dog walks: The Full Tree



What licenses each node?

- The three words: their lexical entries, plus the SHAC for ***dog*** and ***walks***
- ***the dog***: the Head-Specifier Rule, with ***dog*** as head; HFP and Valence Principle
- The S: the Head-Specifier Rule again, with the *word* ***walks*** as head

The dog walks: What Do the Tags Mean?



1. ***the dog*** is the specifier ***walks*** is looking for
2. ***the*** is the specifier ***dog*** is looking for
3. one AGR value shared by all three words (the SHAC, twice)
4. NP and ***dog*** share HEAD (HFP)
5. S and ***walks*** share HEAD (HFP)

The phrases' empty COMPS lists come from the Valence Principle, but we don't usually tag empty lists.

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 individual entities (***bird, cookie, insult, fact***)

- But there are exceptions:
  - ***muesli*** (mass) denotes a mix of oats, nuts & fruit, so it's not undifferentiated.
  - ***furniture, footwear, cutlery, ...*** are mass nouns, although they refer to individual artifacts
  - ***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, ...***)
  - lexically underspecified (***the, all, some, no, ...***)

- 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.

Building a Tree

Algorithm for Building a Tree

Create the words (leaf nodes)

1. Copy all the information from each word's lexical entry
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) 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

Build a Tree!

The cat sleeps in the park.

$$\left\langle \text{the}, \begin{bmatrix} \textit{word} \\ \textsc{head} & \textit{det} \\ \textsc{val} & \begin{bmatrix} \textsc{spr} & \langle\rangle \\ \textsc{comps} & \langle\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{sleeps}, \begin{bmatrix} \textit{word} \\ \textsc{head} & \begin{bmatrix} \textit{verb} \\ \textsc{agr} & \textit{3sing} \end{bmatrix} \\ \textsc{val} & \begin{bmatrix} \textsc{spr} & \langle \text{NP} \rangle \\ \textsc{comps} & \langle\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{cat}, \begin{bmatrix} \textit{word} \\ \textsc{head} & \begin{bmatrix} \textit{noun} \\ \textsc{agr} & \textit{3sing} \end{bmatrix} \\ \textsc{val} & \begin{bmatrix} \textsc{spr} & \left\langle \text{D} \left[\textsc{count} \ + \right] \right\rangle \\ \textsc{comps} & \langle\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{park}, \begin{bmatrix} \textit{word} \\ \textsc{head} & \begin{bmatrix} \textit{noun} \\ \textsc{agr} & \begin{bmatrix} \textit{3sing} \\ \textsc{gend} & \textit{neut} \end{bmatrix} \end{bmatrix} \\ \textsc{val} & \begin{bmatrix} \textsc{spr} & \left\langle \text{D} \left[\textsc{count} \ + \right] \right\rangle \\ \textsc{comps} & \langle\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

$$\left\langle \text{in}, \begin{bmatrix} \textit{word} \\ \textsc{head} & \textit{prep} \\ \textsc{val} & \begin{bmatrix} \textsc{spr} & \langle\rangle \\ \textsc{comps} & \langle \text{NP} \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Head-Modifier Rule:

$$\left[\textit{phrase}\right] \rightarrow \textbf{H} \left[\textsc{val} \ \left[\textsc{comps} \ \langle\rangle\right]\right] \ \text{PP}$$

Build a Tree: One More Question

One feature in the Chapter 4 type hierarchy isn't constrained by anything yet. Which one? What does that mean for our trees?

Build a Tree: One More Question

One feature in the Chapter 4 type hierarchy isn't constrained by anything yet. Which one? What does that mean for our trees?

CASE on nouns (AUX on verbs is the same so far). No rule or lexical entry mentions it, so a tree is licensed whatever value it has: nothing yet rules out **Him sleeps*. Problems 4.5 and 4.6 fix this.

Summary

- List-valued valence features
  - A problem with the Chapter 3 grammar
  - Generalize COMPS and SPR
  - The Valence Principle
- The heads of S and NP
- Complements and modifiers
- Agreement: the SHAC and count/mass
- Building a tree, step by step
- Problems 4.1, 4.5, 4.6

Problems

4.1 Valence Variations

Write lexical entries (including HEAD, SPR, and COMPS values). You may use NP, VP, etc. as abbreviations for the feature structures on COMPS lists.

As you do this problem, keep the following points in mind: (1) In Chapter 4 COMPS became a list-valued feature, and (2) heads select for their specifier and complements (if they have any); the elements on the SPR and COMPS lists do not simultaneously select for the head.

[*Hint: For the purposes of this problem, assume that adjectives and prepositions all have empty SPR lists.*]

A. Write lexical entries for the words *here* and *there* as they are used in (i).

(i) *Kim put the book here/there.*

[*Hint: Compare (i) to (7) on p. 97.*]

C. (Part B of the book's problem is not set.) Assume that motion verbs like *jump*, *move*, etc. take an optional PP complement:

$$\left[\text{COMPS } \left\langle (\text{PP}) \right\rangle\right]$$

Write the lexical entries for the prepositions *out*, *from* and *of*:

(i) *Kim jumped out of the bushes.*
(ii) *Bo jumped out from the bushes.*
(iii) *Lee moved from under the bushes.*
(iv) *Leslie jumped out from under the bushes.*
(v) *Dana jumped from the bushes.*
(vi) *Chris ran out the door.*
(vii) **Kim jumped out of from the bushes.*
(viii) *Kim jumped out.*
(ix) **Kim jumped from.*

D. Based on the following data, write the lexical entries for the words *grew* (in the 'become' sense, not the 'cultivate' sense), *seemed*, *happy*, and *close*.

(i) *They seemed happy (to me).*
(ii) *Lee seemed an excellent choice (to me).*
(iii) **They seemed (to me).*
(iv) *They grew happy.*
(v) **They grew a monster (to me).*
(vi) **They grew happy to me.*
(vii) *They grew close to me.*
(viii) *They seemed close to me to Sandy.*

[*Note: APs have an internal structure analogous to that of VPs. Though no adjectives select NP complements (in English), there are some adjectives that select PP complements (e.g. to me), and some that do not.*]

E. Using the lexical entries you wrote for part (D), draw a tree (showing the values of HEAD, SPR, and COMPS at each node, using tags as appropriate) for *They seemed close to me to Sandy*.

4.5 Facts of English case

- For each of the following positions, determine which case the pronouns in that position must have:
  - Subject of a sentence
  - Direct object of a verb
  - Second object of a verb like ***give***
  - Object of a preposition

- Give examples

4.6 A lexicalist analysis

- Section 4.8 hinted that case marking can be limited in the same way that we handle agreement, i.e., without any changes to the grammar rules. Show how this can be done. Your answer should include lexical entries for ***they***, ***us***, ***likes***, and ***with***.

- Hint: Assume that there is a feature CASE with the values *acc* and *nom*, and assume that English pronouns have CASE features specified in their lexical entries.

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 4)

References

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