HPSG Tips

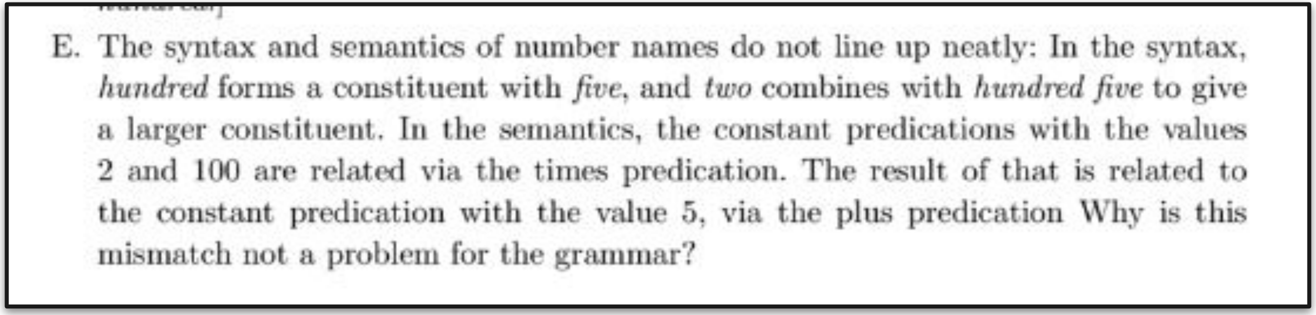
10/19/2023

Overview

- Semantics of number names
- Components of the grammar (again…)

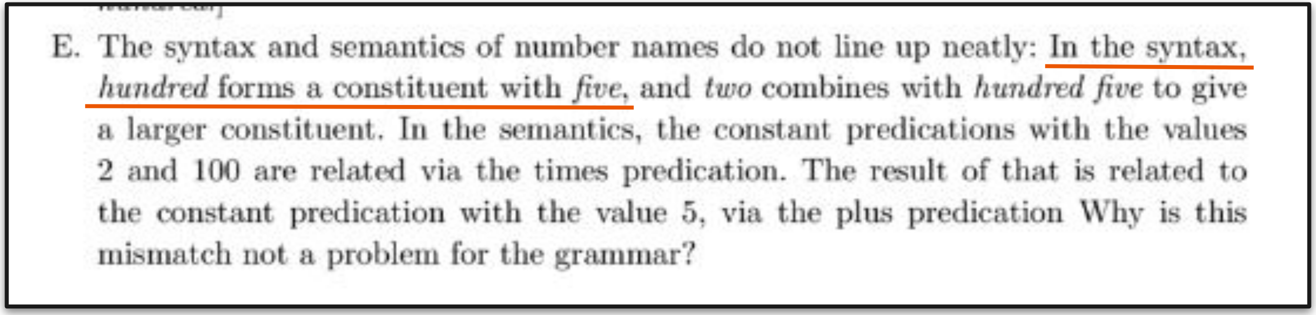
Semantics of number names

5.3.E: Mismatch between syntax and semantics

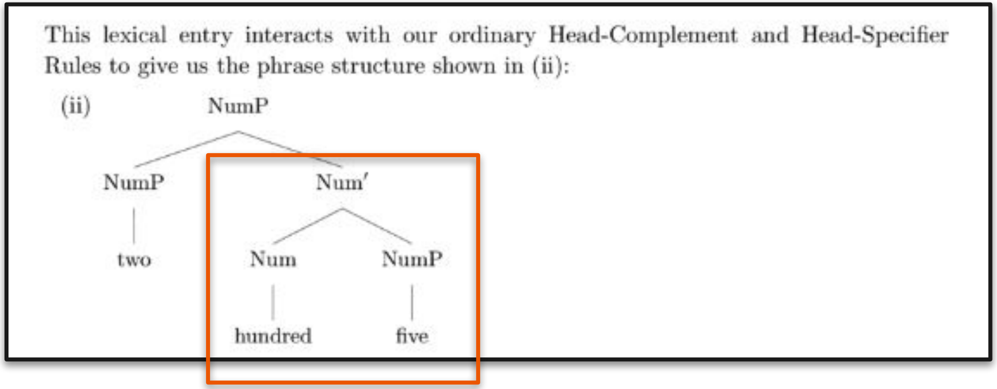


E. The syntax and semantics of number names do not line up neatly: In the syntax, *hundred* forms a constituent with *five*, and *two* combines with *hundred five* to give a larger constituent. In the semantics, the constant predications with the values 2 and 100 are related via the times predication. 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?

5.3.E: Mismatch between syntax and semantics

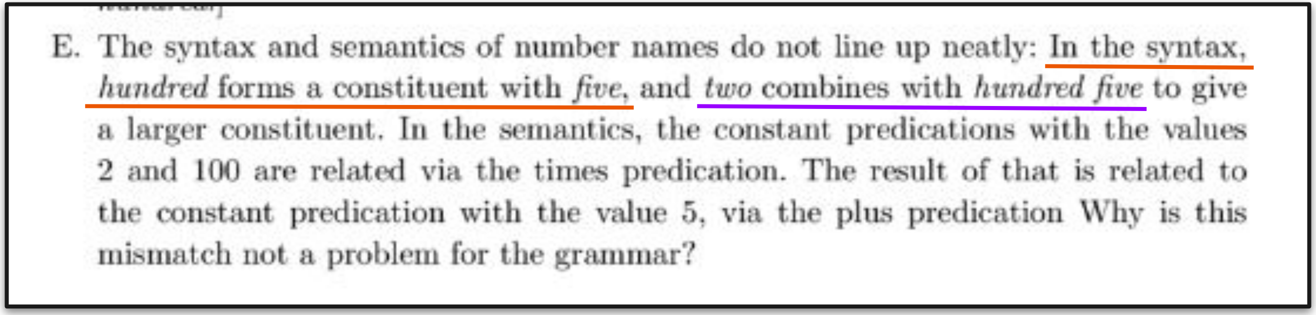


E. The syntax and semantics of number names do not line up neatly: In the syntax, *hundred* forms a constituent with *five*, and *two* combines with *hundred five* to give a larger constituent. In the semantics, the constant predications with the values 2 and 100 are related via the times predication. 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?

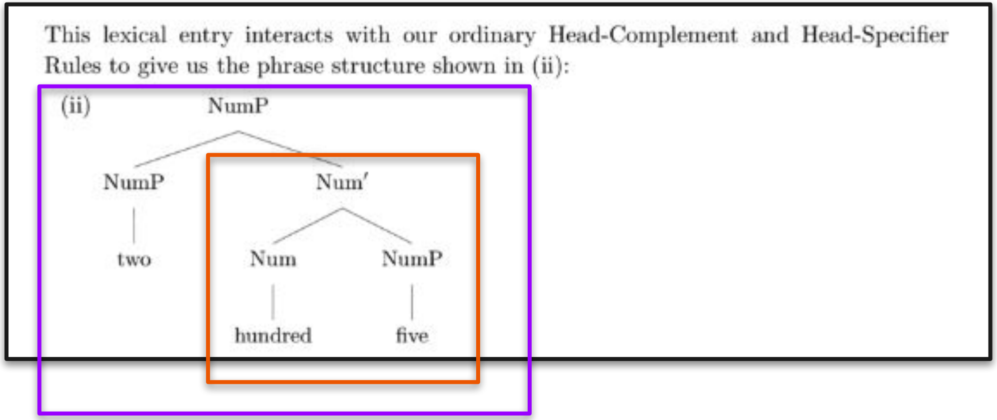


This lexical entry interacts with our ordinary Head-Complement and Head-Specifier Rules to give us the phrase structure shown in (ii):

5.3.E: Mismatch between syntax and semantics



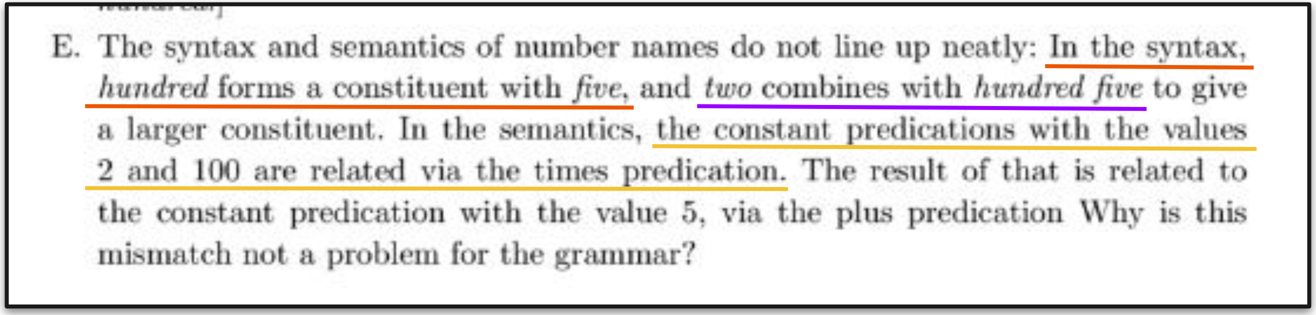
E. The syntax and semantics of number names do not line up neatly: In the syntax, *hundred* forms a constituent with *five*, and *two* combines with *hundred five* to give a larger constituent. In the semantics, the constant predications with the values 2 and 100 are related via the times predication. 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?



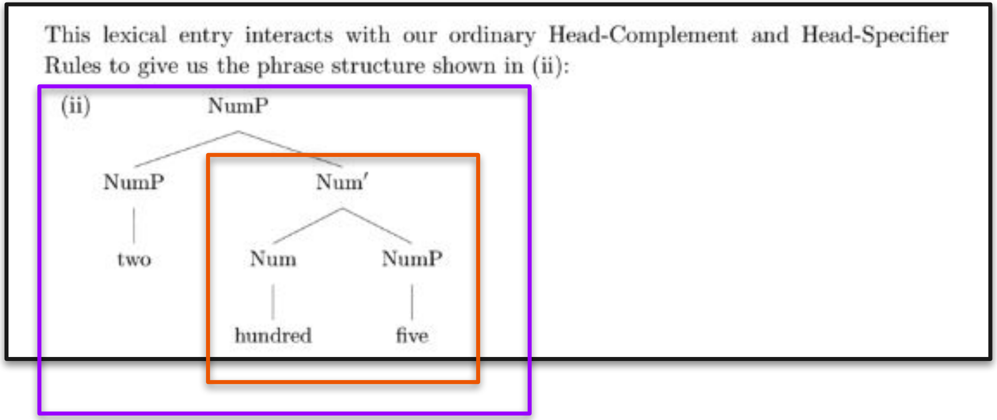
This lexical entry interacts with our ordinary Head-Complement and Head-Specifier Rules to give us the phrase structure shown in (ii):

(ii)

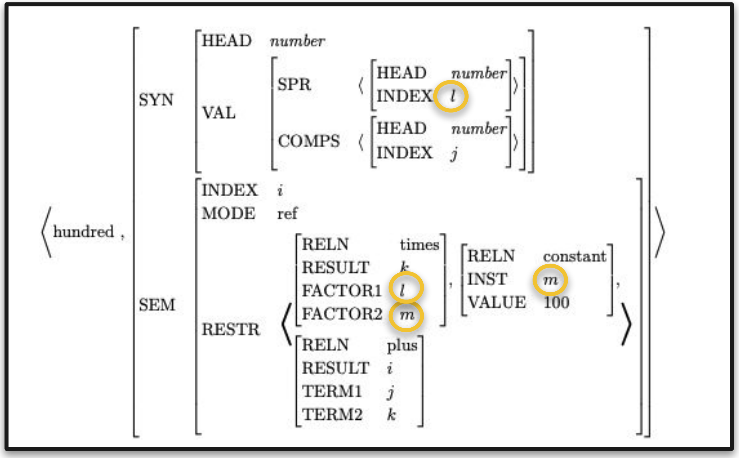
5.3.E: Mismatch between syntax and semantics



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?

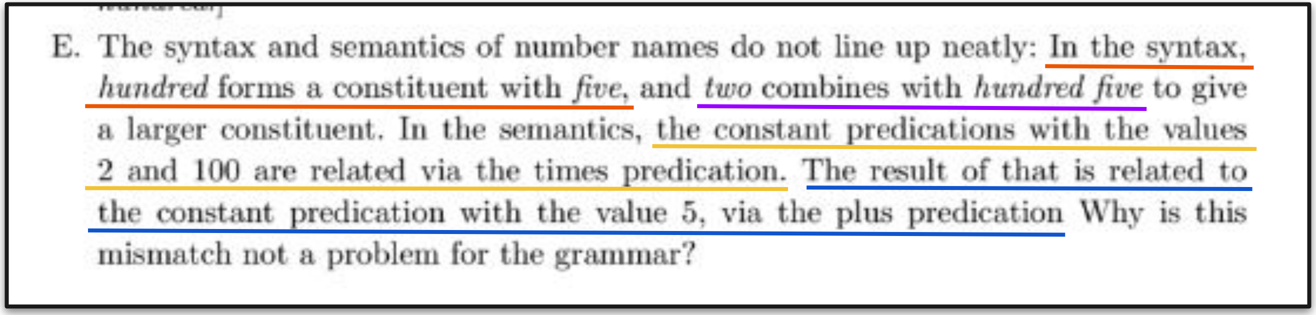


This lexical entry interacts with our ordinary Head-Complement and Head-Specifier Rules to give us the phrase structure shown in (ii):

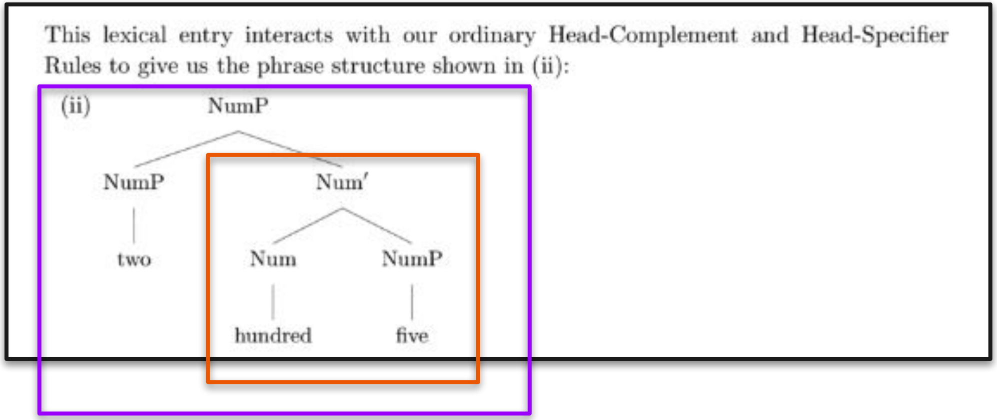


$$\left\langle \text{hundred}\;, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{number} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix} \text{HEAD} & \textit{number} \\ \text{INDEX} & l \end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{HEAD} & \textit{number} \\ \text{INDEX} & j \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{MODE} & \text{ref} \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \text{times} \\ \text{RESULT} & k \\ \text{FACTOR1} & l \\ \text{FACTOR2} & m \end{bmatrix}, \begin{bmatrix} \text{RELN} & \text{constant} \\ \text{INST} & m \\ \text{VALUE} & 100 \end{bmatrix}, \begin{bmatrix} \text{RELN} & \text{plus} \\ \text{RESULT} & i \\ \text{TERM1} & j \\ \text{TERM2} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

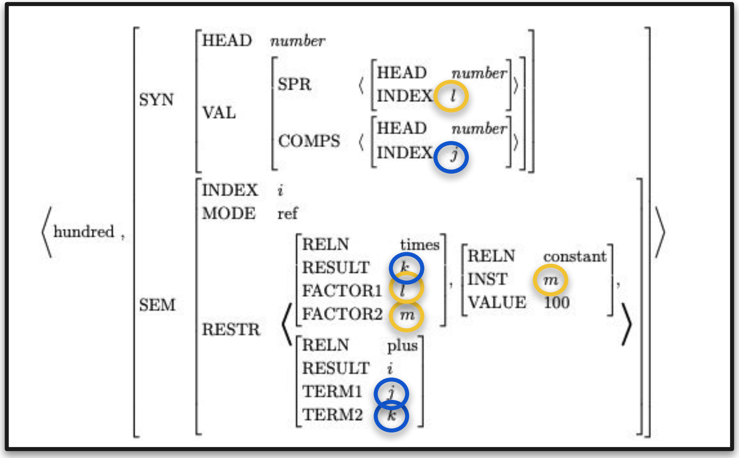
5.3.E: Mismatch between syntax and semantics



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?



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

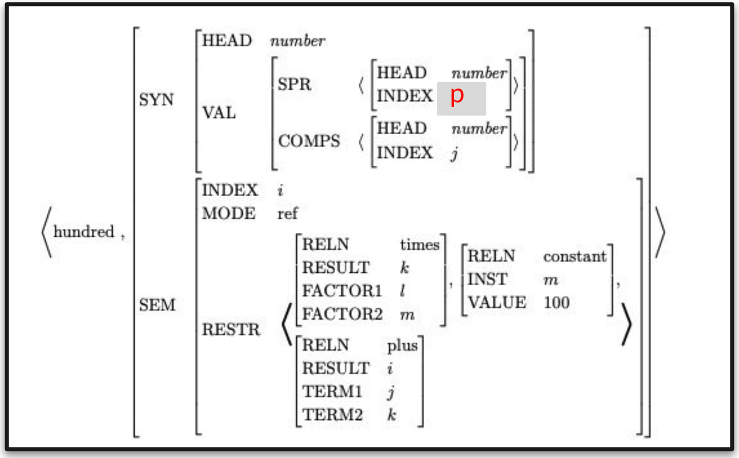


$$\left\langle \text{hundred}, \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{HEAD} & \textit{number} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \begin{bmatrix}\text{HEAD} & \textit{number}\\ \text{INDEX} & l\end{bmatrix}\right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix}\text{HEAD} & \textit{number}\\ \text{INDEX} & j\end{bmatrix}\right\rangle \end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & i \\ \text{MODE} & \text{ref} \\ \text{RESTR} & \left\langle \begin{bmatrix}\text{RELN} & \text{times}\\ \text{RESULT} & k \\ \text{FACTOR1} & l \\ \text{FACTOR2} & m\end{bmatrix}, \begin{bmatrix}\text{RELN} & \text{constant}\\ \text{INST} & m\\ \text{VALUE} & 100\end{bmatrix}, \begin{bmatrix}\text{RELN} & \text{plus}\\ \text{RESULT} & i\\ \text{TERM1} & j\\ \text{TERM2} & k\end{bmatrix}\right\rangle \end{bmatrix}\end{bmatrix}\right\rangle$$

5.3.E: Mismatch between syntax and semantics

- NOT the Semantic Compositionality Principle
  - If the lexical entry for *hundred* did not link the INDEX of the item on its SPR list to FACTOR1, then even if we had all of the predications in the RESTR list, then the semantic structure would still not be correct

- NOT the Semantic Inheritance Principle
  - Similarly, passing the INDEX up the tree is not what does the work of ensuring the items in the RESTR list are linked to the right things



⟨ hundred ,
SYN
HEAD *number*
VAL
SPR ⟨ HEAD *number*, INDEX p ⟩
COMPS ⟨ HEAD *number*, INDEX *j* ⟩
SEM
INDEX *i*
MODE ref
RESTR ⟨ [RELN times, RESULT *k*, FACTOR1 *l*, FACTOR2 *m*], [RELN constant, INST *m*, VALUE 100], [RELN plus, RESULT *i*, TERM1 *j*, TERM2 *k*] ⟩ ⟩

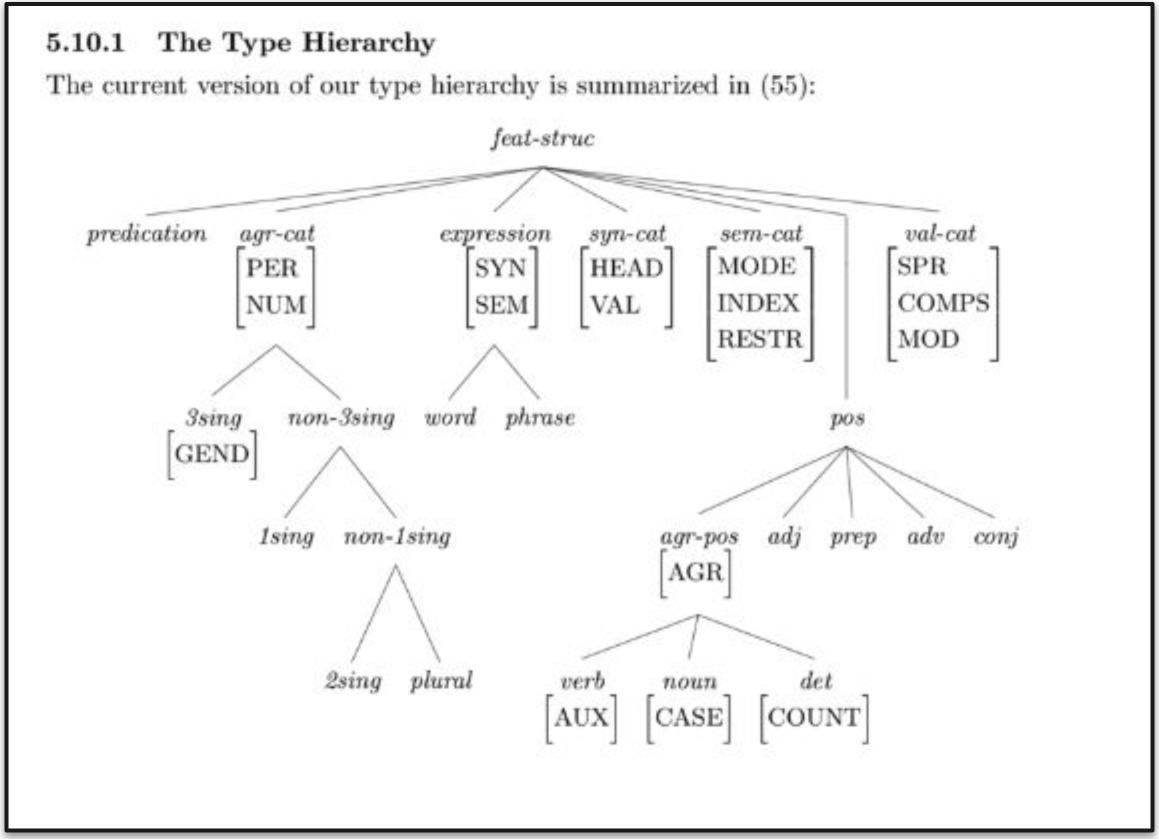
Components of the grammar

Components of the grammar

- Type hierarchy
  - The statement of what we say exists in the model. Definitional specification of the kinds of things we have to play with. Some of those things are the “big pieces” we pick up and put together … some are properties of those bigger things
- Lexical entries
  - Partial descriptions of word structures … partial descriptions of what can be going on at the bottom of the tree
- Grammar rules
  - Partial descriptions of how constituents can be put together… a constituent can be a word structure or a phrase licensed by a grammar rule

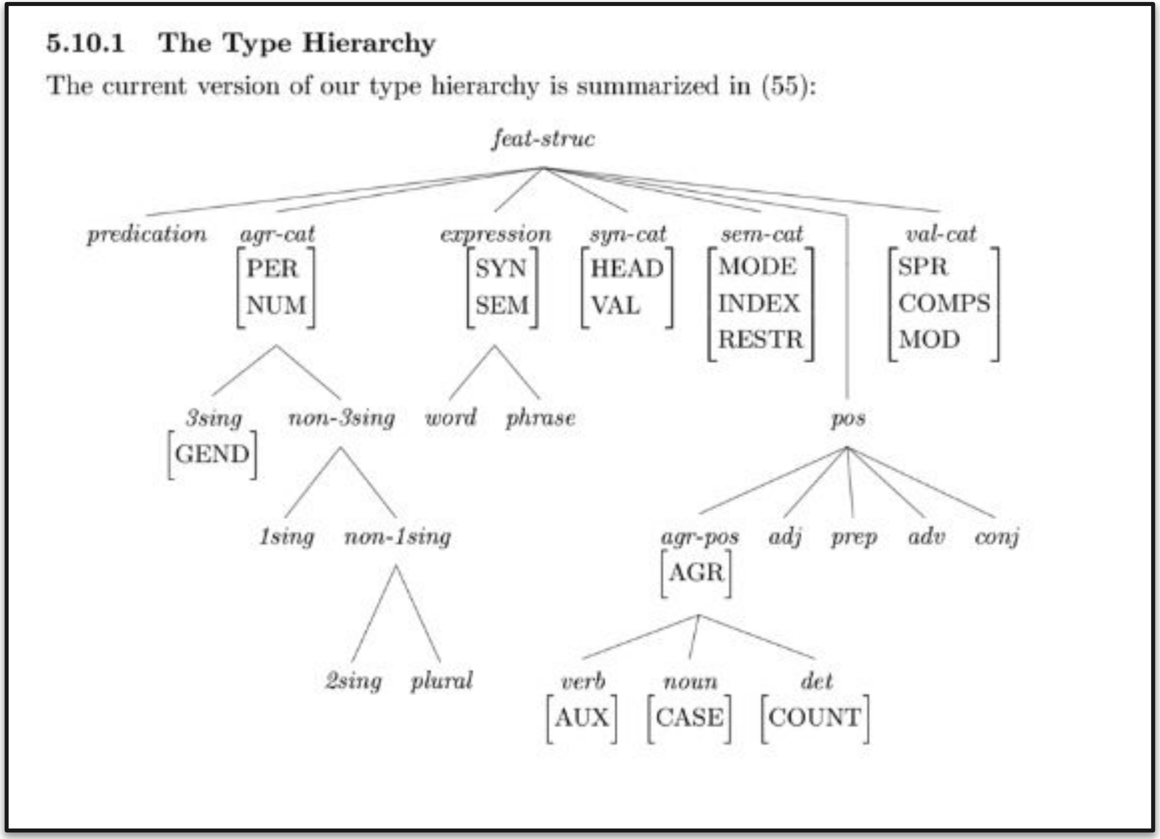
Components of the grammar

- Principles
  - Partial descriptions of well-formed structures, but instead of descriptions of things at the bottom of the tree or how you put together pieces of the tree it’s descriptions of **things that have to be true of the tree** for it to be well-formed
- Initial symbol
  - Constraints that have to be true of the top node of the tree for it to be well-formed
- Abbreviations (N, NOM, NP, etc)
  - Syntactic sugar, conveniences
  - also partial descriptions of feature structures



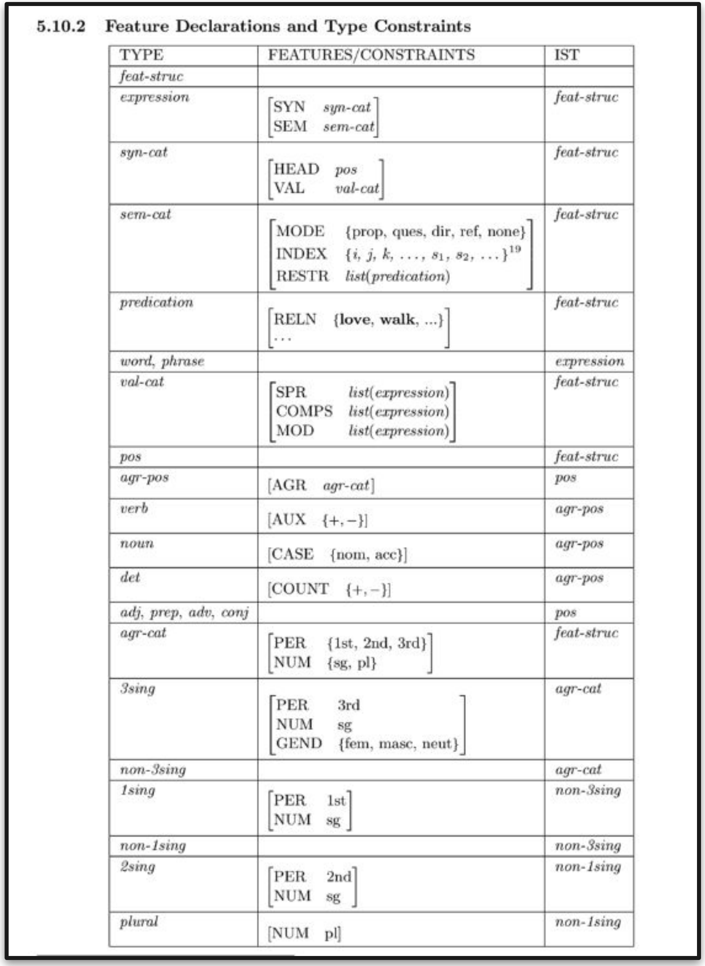
5.10.1 The Type Hierarchy

The current version of our type hierarchy is summarized in (55):



5.10.1 The Type Hierarchy

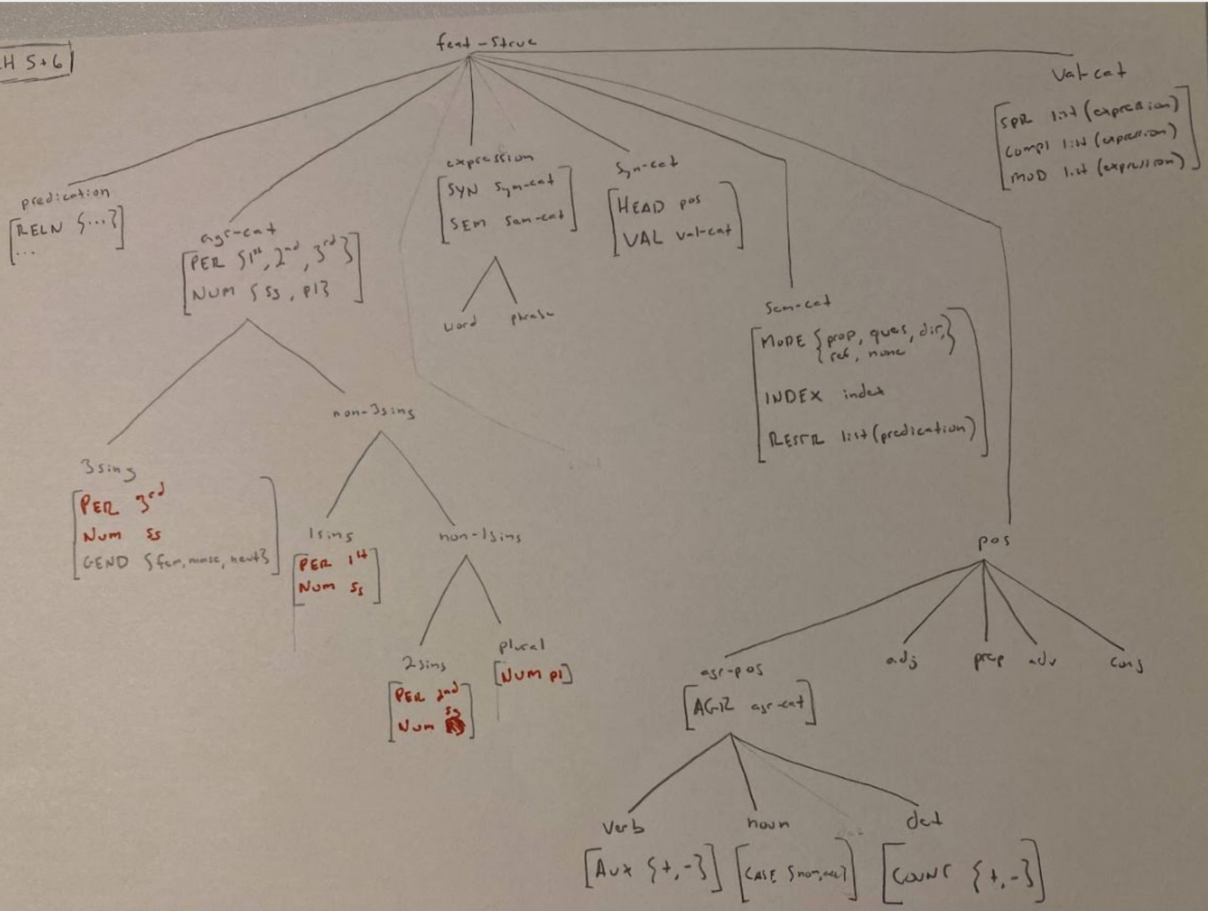
The current version of our type hierarchy is summarized in (55):



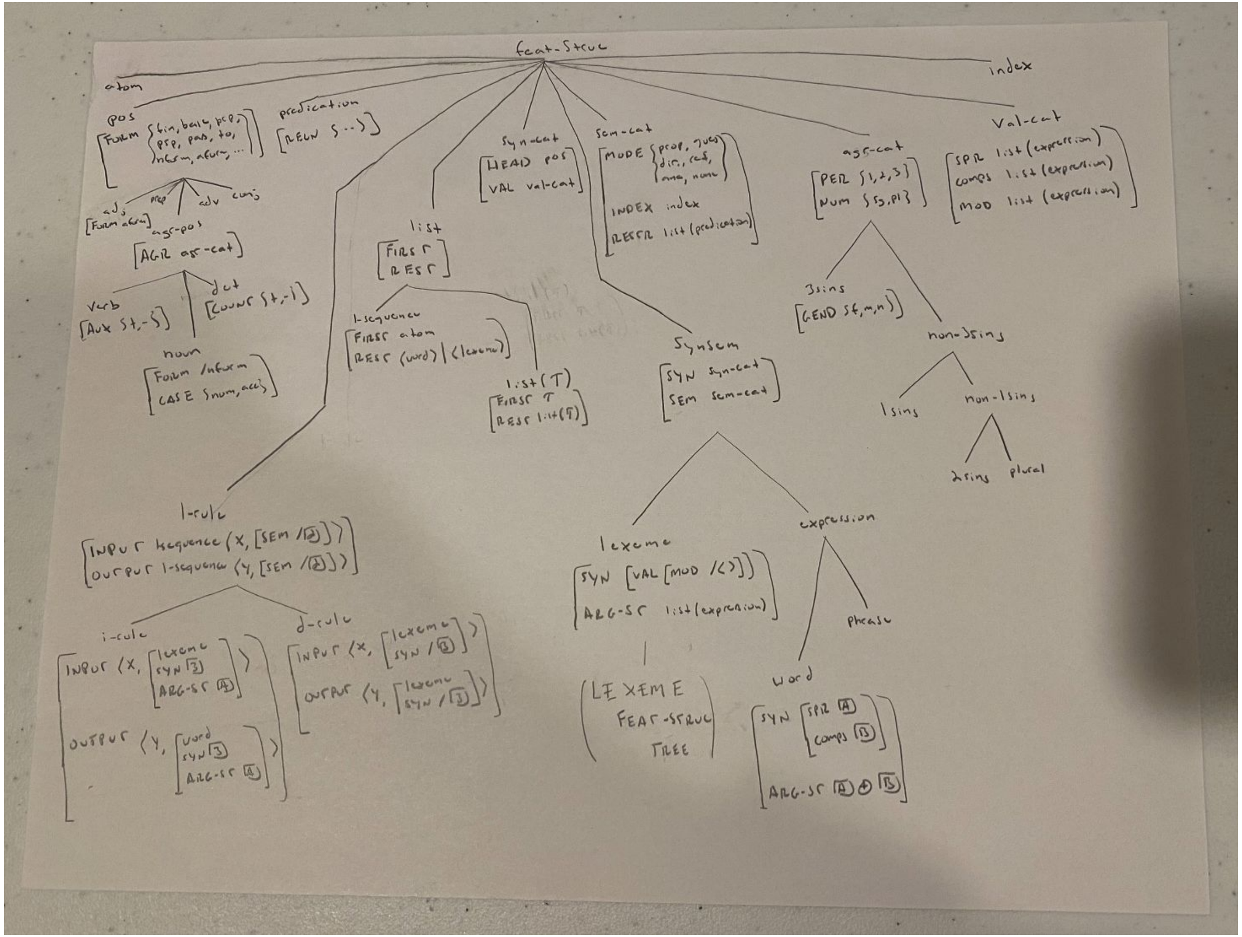
5.10.2 Feature Declarations and Type Constraints

TYPE	FEATURES/CONSTRAINTS	IST
feat-struc		
expression	$\begin{bmatrix}\text{SYN} & \textit{syn-cat}\\ \text{SEM} & \textit{sem-cat}\end{bmatrix}$	*feat-struc*
syn-cat	$\begin{bmatrix}\text{HEAD} & \textit{pos}\\ \text{VAL} & \textit{val-cat}\end{bmatrix}$	*feat-struc*
sem-cat	$\begin{bmatrix}\text{MODE} & \{\text{prop, ques, dir, ref, none}\}\\ \text{INDEX} & \{i, j, k, \ldots, s_1, s_2, \ldots\}^{19}\\ \text{RESTR} & \textit{list(predication)}\end{bmatrix}$	*feat-struc*
predication	$\begin{bmatrix}\text{RELN} & \{\textbf{love}, \textbf{walk}, \ldots\}\\ \ldots & \end{bmatrix}$	*feat-struc*
word, phrase		*expression*
val-cat	$\begin{bmatrix}\text{SPR} & \textit{list(expression)}\\ \text{COMPS} & \textit{list(expression)}\\ \text{MOD} & \textit{list(expression)}\end{bmatrix}$	*feat-struc*
pos		*feat-struc*
agr-pos	$\begin{bmatrix}\text{AGR} & \textit{agr-cat}\end{bmatrix}$	*pos*
verb	$\begin{bmatrix}\text{AUX} & \{+, -\}\end{bmatrix}$	*agr-pos*
noun	$\begin{bmatrix}\text{CASE} & \{\text{nom, acc}\}\end{bmatrix}$	*agr-pos*
det	$\begin{bmatrix}\text{COUNT} & \{+, -\}\end{bmatrix}$	*agr-pos*
adj, prep, adv, conj		*pos*
agr-cat	$\begin{bmatrix}\text{PER} & \{\text{1st, 2nd, 3rd}\}\\ \text{NUM} & \{\text{sg, pl}\}\end{bmatrix}$	*feat-struc*
3sing	$\begin{bmatrix}\text{PER} & \text{3rd}\\ \text{NUM} & \text{sg}\\ \text{GEND} & \{\text{fem, masc, neut}\}\end{bmatrix}$	*agr-cat*
non-3sing		*agr-cat*
1sing	$\begin{bmatrix}\text{PER} & \text{1st}\\ \text{NUM} & \text{sg}\end{bmatrix}$	*non-3sing*
non-1sing		*non-3sing*
2sing	$\begin{bmatrix}\text{PER} & \text{2nd}\\ \text{NUM} & \text{sg}\end{bmatrix}$	*non-1sing*
plural	$\begin{bmatrix}\text{NUM} & \text{pl}\end{bmatrix}$	*non-1sing*

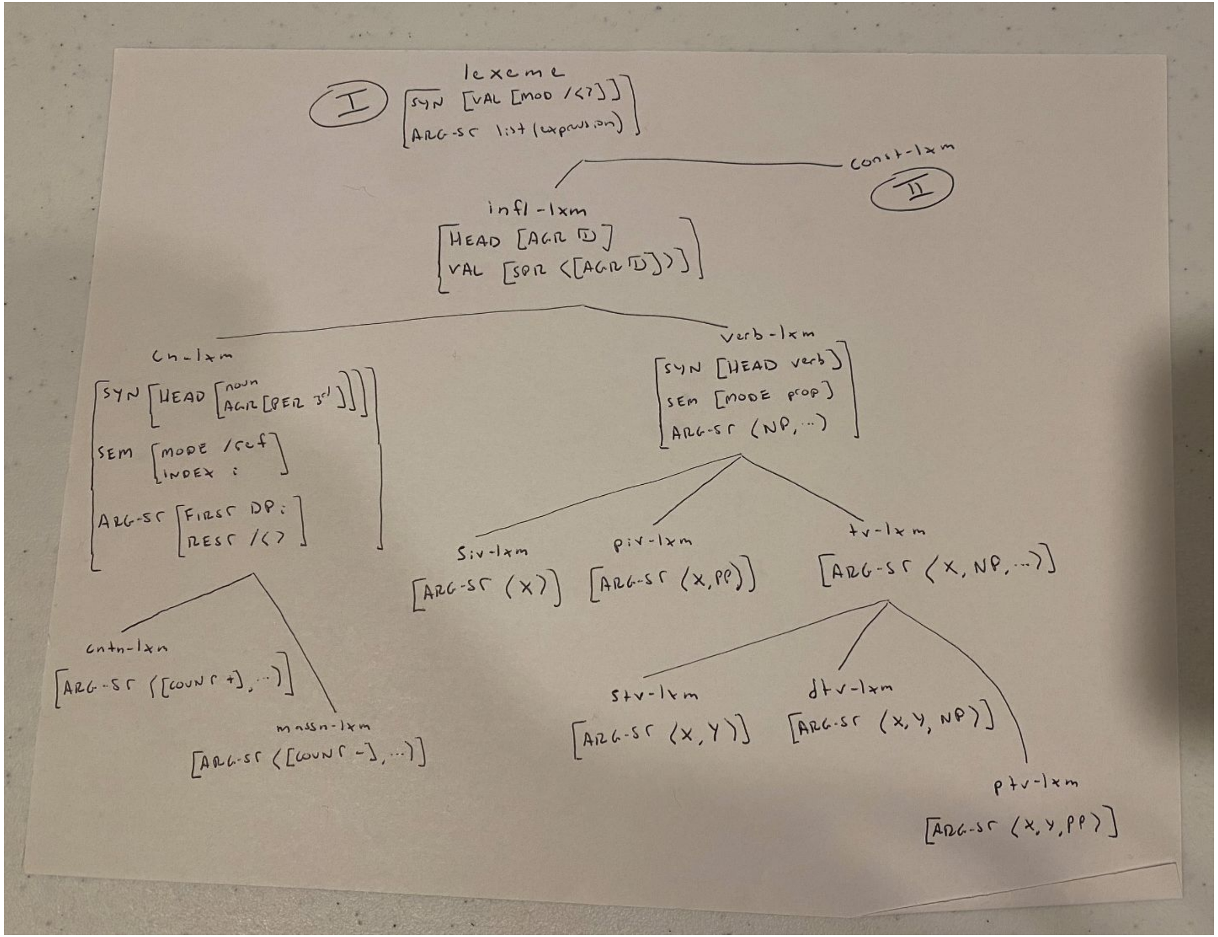
CH 5+6


feat-struc
predication
[RELN {...}
...]
agr-cat
[PER {1st, 2nd, 3rd}
NUM {sg, pl}]
expression
[SYN syn-cat
SEM sem-cat]
word
phrase
syn-cat
[HEAD pos
VAL val-cat]
sem-cat
[MODE {prop, ques, dir, ref, none}
INDEX index
RESTR list(predication)]
val-cat
[SPR list(expression)
COMPS list(expression)
MOD list(expression)]
3sing
[PER 3rd
NUM sg
GEND {fem, masc, neut}]
non-3sing
1sing
[PER 1st
NUM sg]
non-1sing
2sing
[PER 2nd
NUM sg]
plural
[NUM pl]
pos
agr-pos
[AGR agr-cat]
adj
prep
adv
conj
verb
[AUX {+, -}]
noun
[CASE {nom, acc}]
det
[COUNT {+, -}]

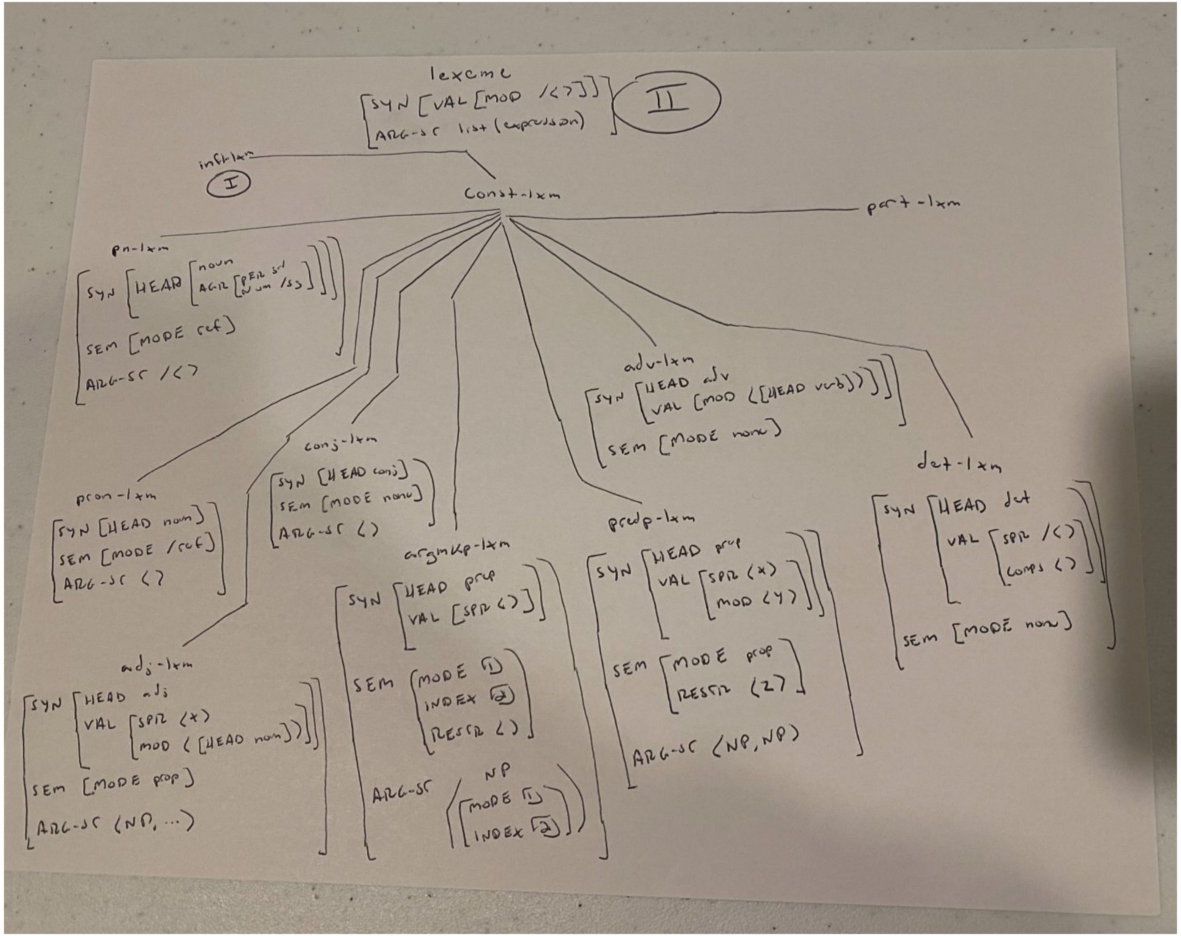
Type Hierarchy by the end of the book (1/3)



Type Hierarchy by the end of the book (2/3)



Type Hierarchy by the end of the book (3/3)



Lexical entries vs. Trees

- Lexical entries are **independent** of one another
  - They are underspecified and only include what information is necessary
  - When creating a grammar, we write these and we need to write them such that they will help us license the trees that we want
- Trees are **fully specified**
  - Every tree you could ever possibly come up with exists out in the Tree Universe (Emily calls it soup...)
  - But only a subset of those trees are legitimate trees for English, and those are the ones we want our grammar to license

Lexical entries vs. Trees

- Lexical entries are **independent** of one another
  - They are underspecified and only include what information is necessary
  - When creating a grammar, we write these and we need to write them such that they will help us license the trees that we want

- Trees are **fully specified**
  - Every tree you could ever possibly come up with exists out in the Tree Universe (Emily calls it soup...)
  - But only a subset of those trees are legitimate trees for English, and those are the ones we want our grammar to license

Question for Emily: How can we decide what information is "necessary" for a lexical entry?