Ling 566
Nov 3, 2015

Lexical Rules

Overview

- How lexical rules fit in
- Three types of lexical rules, constraints
- Example: Plural noun lexical rule
- Advice on writing lexical rules
- Constant lexemes
- ARG-ST & ARP
- The feature FORM

Lexical Types & Lexical Rules

- Lexemes capture the similarities among *run*, *runs*, *running*, and *ran*
- The lexical type hierarchy captures the similarities among *run*, *sleep*, and *laugh*, among those and other verbs like *devour* and *hand*, and among those and other words like *book*.
- Lexical rules capture the similarities among *runs*, *sleeps*, *devours*, *hands*, ...

Parsimony & Plausibility

- Lexical rules capture **productive** generalizations.
- There may be some ‘precompiling’ going on as well.

Three Kinds of Lexical Rules

- Inflectional: *lexeme* to *word*

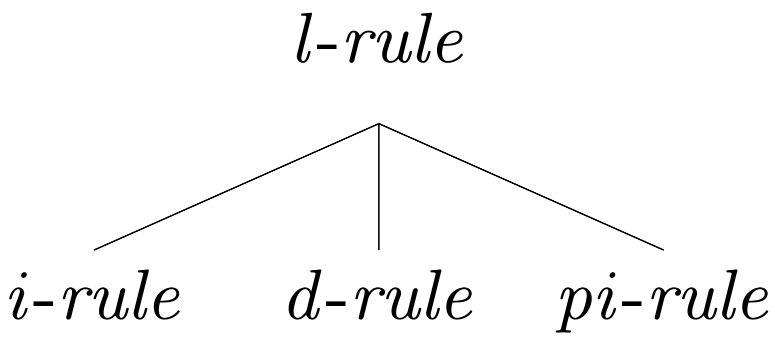
  Examples?

- Derivational: *lexeme* to *lexeme*

  Examples?

- Post-Inflectional: *word* to *word*
  (Chapters 11, 13, 14)

Three Subtypes of *l-rule*



$$\textit{l-rule} : \left[\begin{array}{ll} \text{INPUT} & \textit{l-sequence}\left\langle \text{X} , \left[\,\text{SEM} \quad / \; \boxed{2}\,\right] \right\rangle \\ \text{OUTPUT} & \textit{l-sequence}\left\langle \text{Y} , \left[\,\text{SEM} \quad / \; \boxed{2}\,\right] \right\rangle \end{array}\right]$$

$$\textit{i-rule} : \left[\begin{array}{ll} \text{INPUT} & \left\langle \text{X} , \left[\begin{array}{ll} \textit{lexeme} & \\ \text{SYN} & \boxed{3} \\ \text{ARG-ST} & \boxed{\text{A}} \end{array}\right] \right\rangle \\ & \\ \text{OUTPUT} & \left\langle \text{Y} , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \boxed{3} \\ \text{ARG-ST} & \boxed{\text{A}} \end{array}\right] \right\rangle \end{array}\right]$$

$$\textit{d-rule} : \left[\begin{array}{ll} \text{INPUT} & \left\langle \text{X} , \left[\begin{array}{ll} \textit{lexeme} & \\ \text{SYN} & / \; \boxed{3} \end{array}\right] \right\rangle \\ & \\ \text{OUTPUT} & \left\langle \text{Y} , \left[\begin{array}{ll} \textit{lexeme} & \\ \text{SYN} & / \; \boxed{3} \end{array}\right] \right\rangle \end{array}\right]$$

Plural Noun LR

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ ,\ \textit{cntn-lxm} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{NPL}(\boxed{1})\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Plural Noun LR with Inherited Constraints

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ ,\ \begin{bmatrix} \textit{cntn-lxm} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{NPL}(\boxed{1})\ ,\ \begin{bmatrix} \textit{word} & \\ & \begin{bmatrix} \text{HEAD} & [\text{AGR} & [\text{NUM} & \text{pl}]] \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Plural Noun LR with Inherited Constraints

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ ,\ \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SEM} & \boxed{2} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{NPL}(\boxed{1})\ ,\ \begin{bmatrix} \textit{word} & \\ & \begin{bmatrix} \text{HEAD} & [\text{AGR} & [\text{NUM} & \text{pl}]] \end{bmatrix} \\ \text{SEM} & \boxed{2} \end{bmatrix} \right\rangle \end{bmatrix}$$

Plural Noun LR with Inherited Constraints

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ , \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SEM} & \boxed{2} \\ \text{ARG-ST} & \boxed{\text{B}} \oplus \boxed{\text{C}} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{NPL}(\boxed{1})\ , \begin{bmatrix} \textit{word} & \\ & \begin{bmatrix} \text{HEAD} & [\text{AGR} & [\text{NUM} & \text{pl}]] \end{bmatrix} \\ \text{SEM} & \boxed{2} \\ \text{ARG-ST} & \boxed{\text{B}} \oplus \boxed{\text{C}} \end{bmatrix} \right\rangle \end{bmatrix}$$

Plural Noun LR with Inherited Constraints

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\,, \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \boxed{3} \\ \text{SEM} & \boxed{2} \\ \text{ARG-ST} & \boxed{\text{B}} \oplus \boxed{\text{C}} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{NPL}(\boxed{1})\,, \begin{bmatrix} \textit{word} & \\ \text{SYN} & \boxed{3} \begin{bmatrix} \text{HEAD} \quad [\text{AGR} \quad [\text{NUM} \quad \text{pl}]] \\ \\ \end{bmatrix} \\ \text{SEM} & \boxed{2} \\ \text{ARG-ST} & \boxed{\text{B}} \oplus \boxed{\text{C}} \end{bmatrix} \right\rangle \end{bmatrix}$$

Plural Noun LR with Inherited Constraints

$$
\begin{bmatrix}
\textit{i-rule} & \\
\text{INPUT} & \left\langle \boxed{1}\, , \begin{bmatrix}
\textit{cntn-lxm} & & \\
\text{SYN} & \boxed{3} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix}\textit{noun} & \\ \text{AGR} & \boxed{4}[\text{PER} \ \ \text{3rd}]\end{bmatrix}\\
\text{VAL} & \begin{bmatrix}\text{SPR} & \left\langle \overset{\text{DP}}{\begin{bmatrix}\text{COUNT} & + \\ \text{AGR} & \boxed{4}\end{bmatrix}} \right\rangle\end{bmatrix}
\end{bmatrix}\\
\text{SEM} & \boxed{2}[\text{MODE} \ / \ \text{ref}] & \\
\text{ARG-ST} & \boxed{B} \oplus \boxed{C} &
\end{bmatrix}\right\rangle\\
\text{OUTPUT} & \left\langle \text{F}_{NPL}(\boxed{1})\, , \begin{bmatrix}
\textit{word} & & \\
\text{SYN} & \boxed{3} & \begin{bmatrix}\text{HEAD} & [\text{AGR} \ \ [\text{NUM} \ \ \text{pl}]]\end{bmatrix} \\
\text{SEM} & \boxed{2} & \\
\text{ARG-ST} & \boxed{B}\oplus\boxed{C} &
\end{bmatrix}\right\rangle
\end{bmatrix}
$$

Plural Noun LR with Inherited Constraints

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\,, \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \boxed{3} \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{4}[\text{PER} \quad \text{3rd}] \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \overset{\text{DP}}{\begin{bmatrix} \text{COUNT} & + \\ \text{AGR} & \boxed{4} \end{bmatrix}} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \boxed{2}[\text{MODE} \;/\; \text{ref}] \\ \text{ARG-ST} & \boxed{B} \oplus \boxed{C} \end{bmatrix} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{NPL}(\boxed{1})\,, \begin{bmatrix} \textit{word} & \\ \text{SYN} & \boxed{3} \begin{bmatrix} \text{HEAD} & [\text{AGR} \quad [\text{NUM} \quad \text{pl}]] \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \boxed{B} \\ \text{COMPS} & \boxed{C} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \boxed{2} \\ \text{ARG-ST} & \boxed{B} \oplus \boxed{C} \end{bmatrix} \right\rangle \end{bmatrix}$$

Practicalities - Applying Lexical Rules

- INPUT is a family of lexical sequences.
- OUTPUT is another family of lexical sequences.
  - ...usually a smaller family
  - ...usually a disjoint one
- The only differences between the families are those stipulated in the rule (or the rule's type).
- Similarities are handled by the constraints on *l-rule* and its subtypes.
- If we've written the LRs correctly, nothing is left underconstrained.

Example: Lexical Entry for *cat*

$$\left\langle \text{cat} \; , \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \text{cat} \\ \text{INST} & k \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

Example: *cat*, with inheritance

$$\left\langle \text{cat}\ ,\ \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \begin{bmatrix} & \\ \text{VAL} & \begin{bmatrix} \\ \\ \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} & \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \text{cat} \\ \text{INST} & k \end{bmatrix} \right\rangle \end{bmatrix} \\ & \end{bmatrix} \right\rangle$$

Example: *cat*, with inheritance

$$\left\langle \text{cat} \; , \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \begin{bmatrix} & \\ \text{VAL} & \begin{bmatrix} & \\ \text{SPR} & \left\langle \left[\text{COUNT} \quad + \right] \right\rangle \\ & \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} & \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \text{cat} \\ \text{INST} & k \end{bmatrix} \right\rangle \end{bmatrix} \\ & \end{bmatrix} \right\rangle$$

Example: *cat*, with inheritance

$$\left\langle \text{cat}\,,\ \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix}\textit{noun} & \\ \text{AGR} & [\,\text{PER}\quad \text{3rd}\,]\end{bmatrix} \\ \text{VAL} & \begin{bmatrix}\text{SPR} & \left\langle \overset{\text{DP}}{\begin{bmatrix}\text{COUNT} & +\end{bmatrix}} \right\rangle\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \text{ref}\\ \text{INDEX} & k \\ \text{RESTR} & \left\langle\begin{bmatrix}\text{RELN} & \text{cat}\\ \text{INST} & k\end{bmatrix}\right\rangle\end{bmatrix} \\ \text{ARG-ST} & \langle \text{X}\rangle \end{bmatrix} \right\rangle$$

Example: *cat*, with inheritance

$$\left\langle \text{cat}\ ,\ \begin{bmatrix} \multicolumn{2}{l}{\textit{cntn-lxm}} \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \multicolumn{2}{l}{\textit{noun}} \\ \text{AGR} & \boxed{7}\,[\ \text{PER} \quad \text{3rd}\ ] \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \overset{\text{DP}}{\begin{bmatrix} \text{COUNT} & + \\ \text{AGR} & \boxed{7} \end{bmatrix}} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & k \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \text{cat} \\ \text{INST} & k \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \langle \text{X} \rangle \end{bmatrix} \right\rangle$$

Plural Noun LR

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle \boxed{1}\ ,\ \textit{cntn-lxm} \right\rangle \\ \text{OUTPUT} & \left\langle \text{F}_{NPL}(\boxed{1})\ ,\ \begin{bmatrix} \textit{word} \\ \text{SYN} \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{pl} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Licensing *cats*

$$
\begin{bmatrix}
\textit{i-rule} & & \\
\text{INPUT} & \left\langle \boxed{1}\text{cat}\ , & \begin{bmatrix}
\textit{cntn-lxm} & & \\
\text{SYN} & \boxed{3} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{7}\left[\ \text{PER} \quad \text{3rd}\ \right] \end{bmatrix} \\
\text{VAL} & \begin{bmatrix} \text{SPR} & \left\langle \overset{\text{DP}}{\begin{bmatrix} \text{COUNT} & + \\ \text{AGR} & \boxed{7} \end{bmatrix}} \right\rangle \end{bmatrix}
\end{bmatrix} \\
\text{SEM} & \boxed{2} & \begin{bmatrix}
\text{MODE} & \text{ref} \\
\text{INDEX} & k \\
\text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \text{cat} \\ \text{INST} & k \end{bmatrix} \right\rangle
\end{bmatrix} \\
\text{ARG-ST} & \boxed{\text{B}}\langle\ \text{X}\ \rangle \oplus \boxed{\text{C}}\langle\ \rangle &
\end{bmatrix} \right\rangle \\
\text{OUTPUT} & \left\langle \text{F}_{NPL}(\boxed{1})\ , & \begin{bmatrix}
\textit{word} & & \\
\text{SYN} & \boxed{3} & \begin{bmatrix}
\text{HEAD} & \left[\text{AGR} \quad \left[\text{NUM} \quad \text{pl}\right]\right] \\
\text{VAL} & \begin{bmatrix} \text{SPR} & \boxed{\text{B}} \\ \text{COMPS} & \boxed{\text{C}} \end{bmatrix}
\end{bmatrix} \\
\text{SEM} & \boxed{2} & \\
\text{ARG-ST} & \boxed{\text{B}} \oplus \boxed{\text{C}} &
\end{bmatrix} \right\rangle
\end{bmatrix}
$$

cats: The Lexical Sequences

$$\left\langle \text{cats}\ ,\ \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix}\textit{noun} & \\ \text{AGR} & \textit{3pl}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \boxed{\text{B}}\left\langle \overset{\text{DP}}{\begin{bmatrix}\text{COUNT} & + \\ \text{AGR} & \boxed{7}\end{bmatrix}} \right\rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{MODE} & \text{ref}\\ \text{INDEX} & k \\ \text{RESTR} & \left\langle\begin{bmatrix}\text{RELN} & \text{cat}\\ \text{INST} & k\end{bmatrix}\right\rangle\end{bmatrix} \\ \text{ARG-ST} & \boxed{\text{B}} \end{bmatrix} \right\rangle$$

Practicalities -- Writing Lexical Rules

- Determine the type of the LR.
- Determine the class of possible inputs.
- Determine what should change.
  - If INPUT and OUTPUT values are identified (by default or otherwise) and only OUTPUT value is mentioned, then...

    information is added.

    (Lexical sequences incompatible with that value are not possible inputs)
  - If INPUT and OUTPUT values are identified by default, but different values are given on the INPUT and OUTPUT of the rule, then...

    information is changed.
  - If INPUT and OUTPUT values are identified by an inviolable constraint, but different values are given on the INPUT and OUTPUT of the rule, then...

    there is no well-formed output

Constant lexemes

- What kinds of words are constant lexemes in our grammar?
- Why do we need a rule for these words?
- What would be an alternative analysis?

Constant Lexeme LR

$$\begin{bmatrix} \textit{i-rule} & \\ \text{INPUT} & \left\langle\, \boxed{1}\, ,\ \textit{const-lxm} \,\right\rangle \\ \text{OUTPUT} & \begin{bmatrix} \text{FIRST} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

- What keeps this from applying to, say, verb lexemes?

- Why is this an *i-rule*?

ARG-ST & ARP

- Given the ARP, what do we need to specify about the valence properties of words?
- Why isn’t the ARP a constraint on the type *lexeme*?

The Feature FORM

- Different inflected forms of verbs show up in different syntactic environments. Examples?
- These different forms are syntactically distinguished by the feature FORM, as assigned by lexical rules.
- FORM is also useful in our analyses of coordination and PP selection.

What rules these out?

- *Kim eat pizza.
- *Kim seems to eats pizza.
- *Dana helped Leslie [pack and moved].
- *Kim relies for Sandy.
- *Dana walked and Kim.

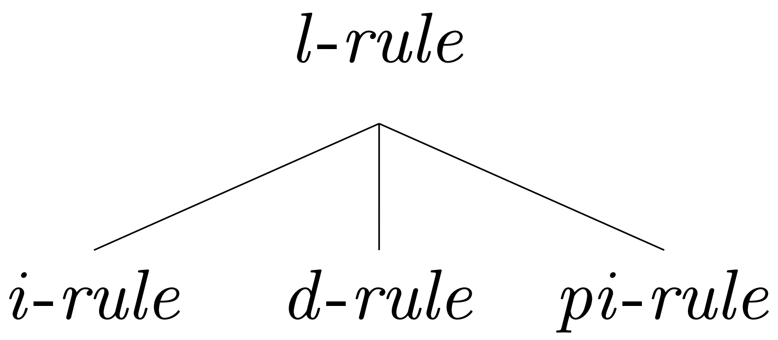
Overview

- How lexical rules fit in
- Three types of lexical rules, constraints
- Example: Plural noun lexical rule
- Advice on writing lexical rules
- Constant lexemes
- ARG-ST & ARP
- The feature FORM
- Reading Questions

Reading Questions

- If everything can be represented by derivational rules, why not just represent everything using d-rules, and have a simple i-rule that can convert any lexeme to word?
- Why are the participles d-rules and not i-rules?
- I'm a little curious why the i-rule formulated on page 252 isn't defeasible for the features SYN and ARG-ST for the input and output, respectively, similarly to the l-rule and d-rule specifications? Why would or wouldn't we want the i-rule formulation on 252 to be defeasible?

Three Subtypes of *l-rule*



$$l\text{-}rule : \begin{bmatrix} \text{INPUT} & l\text{-}sequence\left\langle X , \left[\,\text{SEM} \quad / \ \boxed{2}\,\right]\right\rangle \\ \text{OUTPUT} & l\text{-}sequence\left\langle Y , \left[\,\text{SEM} \quad / \ \boxed{2}\,\right]\right\rangle \end{bmatrix}$$

$$i\text{-}rule : \begin{bmatrix} \text{INPUT} & \left\langle X , \begin{bmatrix} lexeme & \\ \text{SYN} & \boxed{3} \\ \text{ARG-ST} & \boxed{A} \end{bmatrix}\right\rangle \\ \\ \text{OUTPUT} & \left\langle Y , \begin{bmatrix} word & \\ \text{SYN} & \boxed{3} \\ \text{ARG-ST} & \boxed{A} \end{bmatrix}\right\rangle \end{bmatrix}$$

$$d\text{-}rule : \begin{bmatrix} \text{INPUT} & \left\langle X , \begin{bmatrix} lexeme & \\ \text{SYN} & / \ \boxed{3} \end{bmatrix}\right\rangle \\ \\ \text{OUTPUT} & \left\langle Y , \begin{bmatrix} lexeme & \\ \text{SYN} & / \ \boxed{3} \end{bmatrix}\right\rangle \end{bmatrix}$$

Reading Questions

- Why does the author include "SEM" when discussing about i-rule since it does not seem to be needed at all for the discussion? (e.g. pg 253 (62))

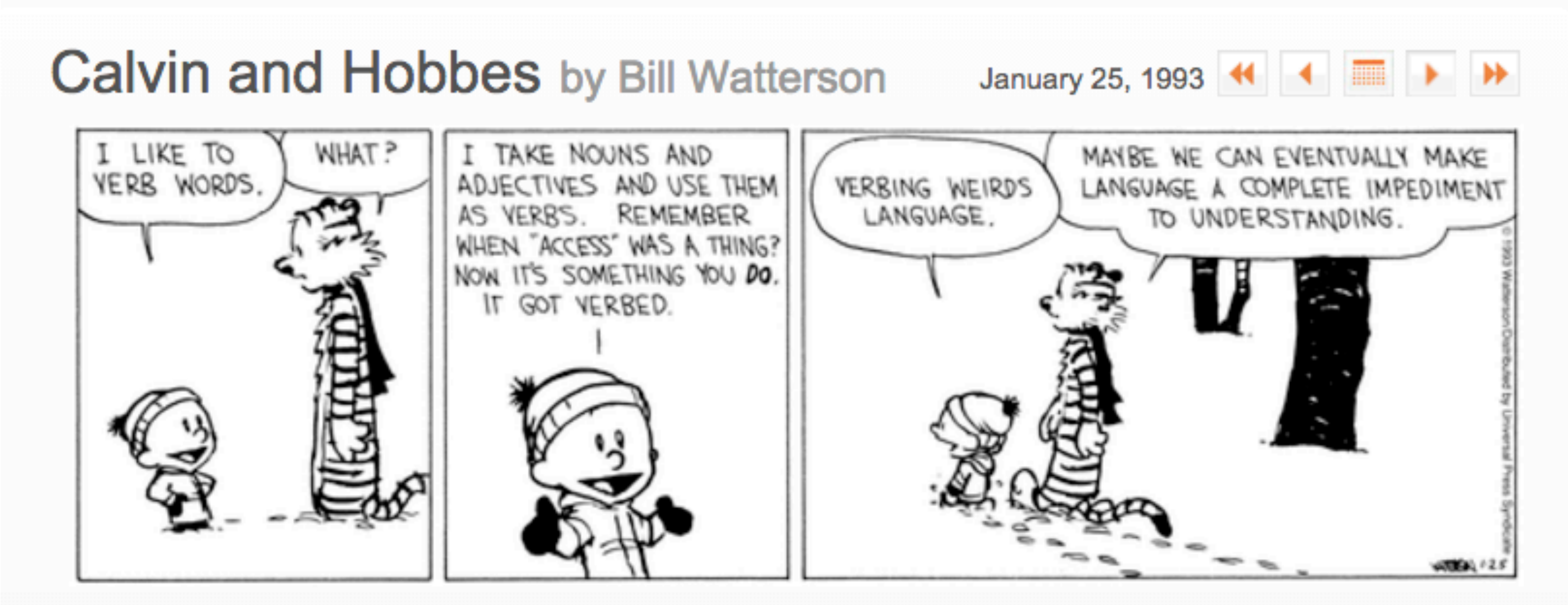
- Why does the i-rule not specify that the SEM value remains the same in the input and output? At least, why does it not specify that the SEM value can remain the same optionally; it changes in some of the verb lexical rule examples, like the non-3rd Singular Verb Lexical Rule.

Reading Questions

- However, for derivational changes on a word, don't the semantics typically change? If so, then why bother marking SEM with a defeasible default constraint that the SEM of INPUT and OUTPUT match?

Reading Questions

- One thought that I had about d-rules was for something like verbification (and then I saw example 79 on zero derivation). I have a noun 'paper clip', but I can start using 'paper clip' as a verb. What criteria is used to decide whether these are really different lexemes as opposed to creating a transformational rule? My guess is that many (or most) count nouns make silly verbs and a rule like this would open the flood gates for funny sentences. "I glassed my drink."


Calvin and Hobbes by Bill Watterson
January 25, 1993
I LIKE TO VERB WORDS.
WHAT?
I TAKE NOUNS AND ADJECTIVES AND USE THEM AS VERBS. REMEMBER WHEN "ACCESS" WAS A THING? NOW IT'S SOMETHING YOU DO. IT GOT VERBED.
VERBING WEIRDS LANGUAGE.
MAYBE WE CAN EVENTUALLY MAKE LANGUAGE A COMPLETE IMPEDIMENT TO UNDERSTANDING.
© 1993 Watterson/Distributed by Universal Press Syndicate

http://www.gocomics.com/calvinandhobbes/1993/01/25

Reading Questions

- It seems like the number of derivational rules would expand very quickly to account for derivational morphology that applies to only a few lexemes. Is there a way to generalize these rules, or do we need separate ones for each different morphological phenomenon?

Reading Questions

- How do the lexical rules fit into the trees?
- The book states INPUT and OUTPUT are metaphorical terms used to describe relationships. If this is the case, would it matter if we reversed INPUT and OUTPUT in our derivational lexical rules? I'm assuming that by convention the "broader" lexeme generally takes the INPUT role.
- Is there anything else other than in the lexicon that the input/output format is used for? It seems very useful.

Reading Questions

- On page 259, we have clarification that "despite the metaphor suggested by the feature names INPUT and OUTPUT, and the informal procedural language we use to describe them, lexical rules do not change or otherwise operate on lexical sequences." The statement strikes me as parallel to the essentially directionless nature of grammar, a directionlessness that changes in implementation. This makes me wonder about the implementation of lexical rules. How and when do they get implemented in parser or generator? And if they DO get implemented, is their characterization as rules still entirely metaphorical or is it part of the mechanism of implementation?

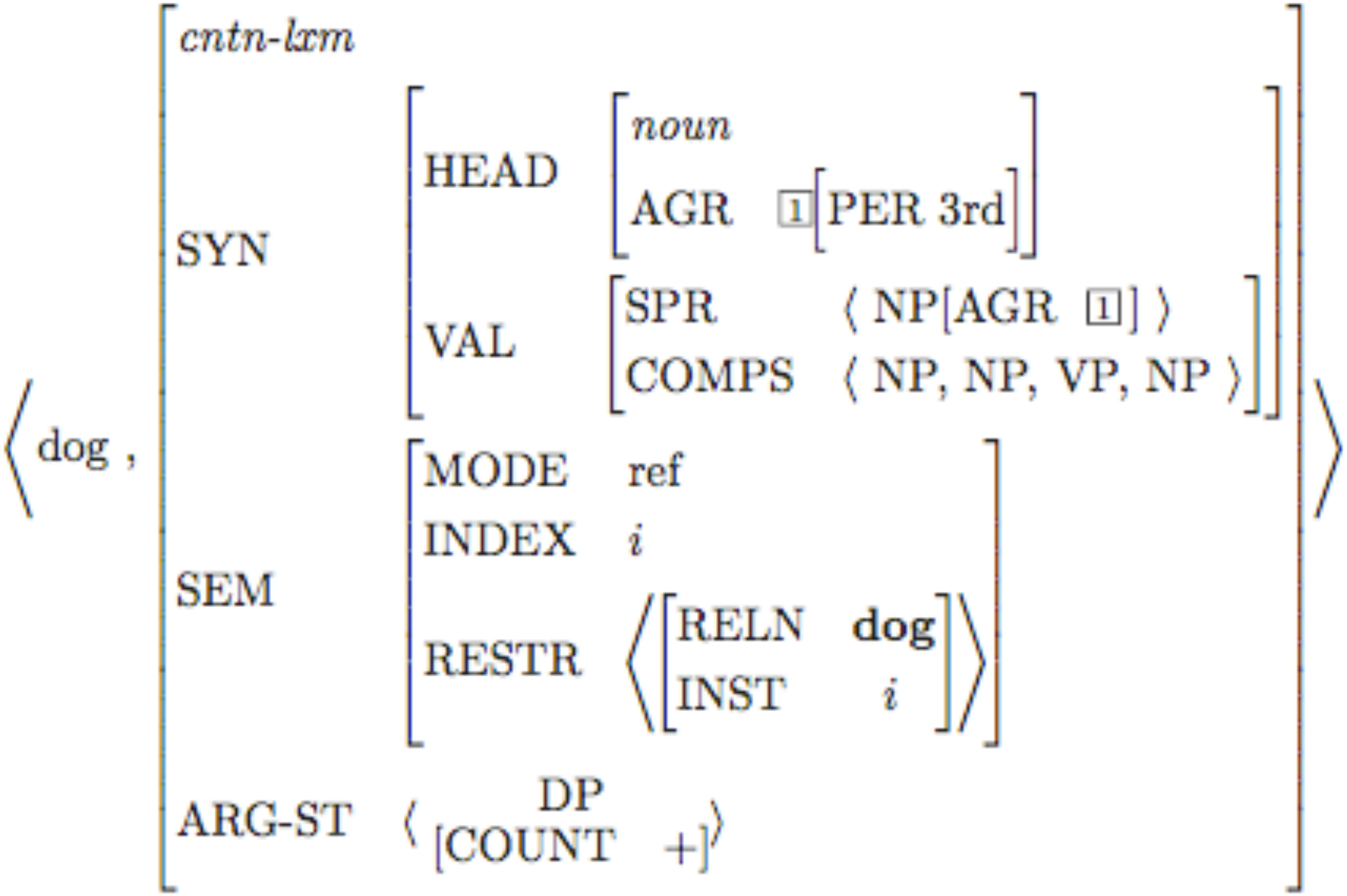
Reading Questions

- I'm a little confused by (74) on page 259. Why can the input vale of the Singular Noun Lexical Rule never be resolved to one of the lexical sequences depicted in (74)? Also, I'm confused as to what the lexical sequences are that are depicted in the example. Why are there more than one lexical sequence depicted?

- What exactly is the difference between lexical rule instantiations and lexical sequences? What would an example be?

Reading Questions

- In other words: Why doesn't this lexical sequence give rise to any words?



$$\left\langle \text{dog}\,, \begin{bmatrix} \textit{cntn-lxm} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{1}\begin{bmatrix}\text{PER 3rd}\end{bmatrix}\end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\, \text{NP[AGR } \boxed{1}] \,\rangle \\ \text{COMPS} & \langle\, \text{NP, NP, VP, NP} \,\rangle \end{bmatrix}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{MODE} & \text{ref} \\ \text{INDEX} & i \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{RELN} & \textbf{dog} \\ \text{INST} & i \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{matrix}\text{DP} \\ [\text{COUNT} \;\; +] \end{matrix} \right\rangle \end{bmatrix} \right\rangle$$

Reading Questions

- Since ARG-ST is SPR+COMPS, is there anything stopping it from theoretically being fulfilled by something with no SPR and 1+ COMPS? Wouldn't it be better to unambiguously write the specifier specification in SPR rather than ARG-ST?

Reading Questions

- I wonder if the cntn-lxm constraint is required for the plural noun lexical rule? I noticed that we can have F(X) = X in the table in 66 (F(fish) = fish), so why don't we remove the constraint from the lexical rule and leave it to F(X) to handle the mass nouns for instance?

- A footnote on page 258 mentions exceptions to lexical rules, such as the verb "be," and suggests that we should either "stipulate the individual forms in the lexicon or posit highly specialized lexical rules for the forms of be." How does a fully-fledged grammar handle exceptions to lexical rules? Are there advantages to making more entries in the lexicon vs. writing rules for special cases?

Reading Questions

- What would the Past-Tense Verb Lexical Rule look like for past tense verbs was and were?
- With some thoughts towards problem 8.7, I'm wondering how we restrict the input of d-rules. I'm fairly sure that we do so by specifying the lexical class or a FORM value. Going back to the ditransitive rules, would we create subclasses of dtv-lxm to prevent the constructions we do not wish to license or would we want something at the lexical level?

Reading Questions

- The footnote [27] at the bottom of the page mentions that the fact that English singular nouns have no inflectional endings (unlike plural nouns) is an accidental feature of English morphology. What does that mean?
- Will the morphology be encoded on the lexemes in order to generalize the lexical entries for different endings?

Reading Questions

- I'm curious to see a fuller explanation of some of the morphological functions posited in the chapter. It seems like they would be a bit unwieldy, especially for something like generating irregular word forms. I thought that the whole point of formulating the grammar to include defeasible constraints and storing most of the information in the lexicon was so that lexemes themselves could store information about whether they make exceptions to the rules. Isn't positing these functions as essentially long lists of irregular forms -or- a regular form just sweeping that problem under a different rug? And thinking in terms of programmatic usefulness, wouldn't it be good to at least have some way the lexeme can specify whether it is regular or irregular, so that, if a morphological function is required, a program needn't traverse an entire list of irregular forms before it can decide that the lexeme is indeed regular?

Reading Questions

- What would derivational rules look like for more complex word formations?
- For example, affixing the complement to the verb to derive the specifier (killjoy, pickpocket), infixing 'fucking' or its euphemisms as an intensifier, or blend words where morphemes (or even phonemes) from different words are combined (cyborg, smog).

Reading Questions

- The analysis of FORM as a headed feature is great for resolving verbal forms that require auxiliaries, but I wonder if treating the auxiliary as the head will be problematic. The auxiliary has to be the head in 'Kim may like Sandy' in order for FORM fine to be passed up and license a stand alone sentence. But from a semantic perspective do we want may to be the head? It seems more intuitive for the head to be like.