

A decorative border in the top-left corner of the slide, composed of a grid of small orange squares and rectangles, creating a geometric pattern.

Ling 566

Nov 18, 2025

Auxiliaries cont: NICE

Overview

- NICE properties of auxiliaries
- The auxiliary *do*
- NICE properties (lexical rules)
- Reading questions

Descriptive Summary of the NICE Properties

Negation

Sentences are negated by putting *not* after the first auxiliary verb; they can be reaffirmed by putting *too* or *so* in the same position

Inversion

Questions are formed by putting an auxiliary verb before the subject NP

Contraction

Auxiliary verbs take negated forms, with *n't* affixed

Ellipsis

Verb phrases immediately following an auxiliary verb can be omitted

Negation (and Reaffirmation)

- Polar adverbs (sentential *not*, *so*, and *too*) appear immediately following an auxiliary

Pat will not leave

Pat will SO leave

Pat will TOO leave



Kim is SO going to the party



Ungrammatical/meaningless

0

Unambig: responding to Kim is not...

0

Unambig: expressing certainty

0

Ambiguous between those two

0

What's the difference?

0

Means yet something else

0

Negation (and Reaffirmation)

- Polar adverbs (sentential *not*, *so*, and *too*) appear immediately following an auxiliary
Pat will not leave
Pat will SO leave
Pat will TOO leave
- What about examples like *Not many people left*?
- What happens when you want to deny or reaffirm a sentence with no auxiliary?

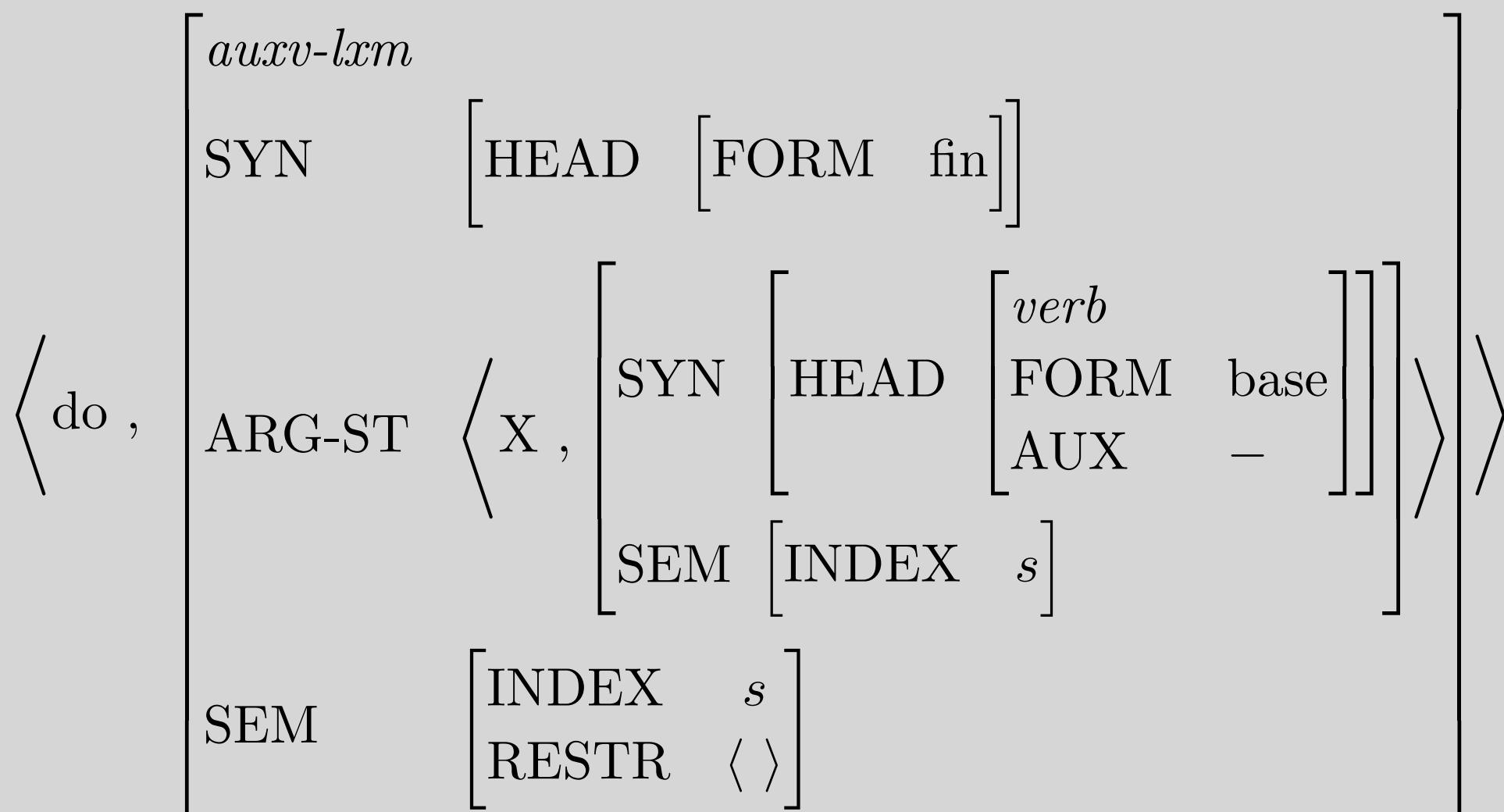
Pat left

Pat did not leave

Pat did TOO leave

The Auxiliary *do*

- Like modals, auxiliary *do* only occurs in finite contexts:
**Pat continued to do not leave*
- Unlike modals, *do* cannot be followed by other auxiliaries:
**Pat did not have left*



The ADV_{pol} -Addition Lexical Rule

pi-rule

INPUT	$\left\langle \mathbf{X}, \right.$	$\left[\begin{array}{l} \text{SYN} \\ \text{ARG-ST} \\ \text{SEM} \end{array} \right.$	$\left[\begin{array}{l} \text{HEAD} \\ \langle \boxed{1} \rangle \oplus \boxed{A} \\ \text{INDEX } s_1 \end{array} \right.$	$\left[\begin{array}{l} \left[\begin{array}{l} \text{verb} \\ \text{FORM } \text{fin} \\ \text{POL } - \\ \text{AUX } + \end{array} \right] \\ \left[\begin{array}{l} \text{INDEX } s_2 \\ \text{RESTR } \langle \boxed{A} \rangle \end{array} \right] \end{array} \right]$	$\rangle$
OUTPUT	$\left\langle \mathbf{Y}, \right.$	$\left[\begin{array}{l} \text{SYN} \\ \text{ARG-ST} \\ \text{SEM} \end{array} \right.$	$\left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{POL } + \\ \text{SPR } \langle \mathbf{Z} \rangle \end{array} \right] \\ \langle \boxed{1} \rangle \oplus \left\langle \left[\begin{array}{l} \text{INDEX } s_2 \\ \text{RESTR } \langle \boxed{A} \rangle \end{array} \right] \right\rangle \oplus \boxed{A} \\ \text{INDEX } s_2 \end{array} \right.$	$\rangle$	

What does the type *pi-rule* mean?

- It maps words to words (hence, “post-inflectional”)
- It preserves MOD values, HEAD values as a default, and (like other lexical rule types) SEM values as a default

$$\left[\begin{array}{l} \text{INPUT} \\ \text{OUTPUT} \end{array} \left\langle / \boxed{0}, \left[\begin{array}{l} \text{word} \\ \text{SYN} \left[\begin{array}{l} \text{HEAD} / \boxed{1} \\ \text{VAL} \left[\text{MOD} \boxed{A} \end{array} \right] \\ \text{SEM} / \boxed{2} \end{array} \right] \right\rangle \right]$$

Why doesn't ADV_{pol} -Addition LR mention VAL?

pi-rule

INPUT	$\langle$	X ,	$\left[\begin{array}{l} \text{SYN} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{verb} \\ \text{FORM} \quad \text{fin} \\ \text{POL} \quad - \\ \text{AUX} \quad + \end{array} \right] \end{array} \right] \\ \text{ARG-ST} \quad \langle \boxed{1} \rangle \oplus \boxed{A} \\ \text{SEM} \quad \left[\text{INDEX} \quad s_1 \right] \end{array} \right]$	$\rangle$
OUTPUT	$\langle$	Y ,	$\left[\begin{array}{l} \text{SYN} \left[\begin{array}{l} \text{HEAD} \left[\text{POL} \quad + \right] \\ \text{VAL} \left[\text{SPR} \quad \langle \text{Z} \rangle \right] \end{array} \right] \\ \text{ARG-ST} \quad \langle \boxed{1} \rangle \oplus \left\langle \begin{array}{l} \text{INDEX} \quad s_2 \\ \text{RESTR} \quad \left\langle \left[\text{ARG} \quad s_1 \right] \right\rangle \end{array} \right\rangle \oplus \boxed{A} \\ \text{SEM} \quad \left[\text{INDEX} \quad s_2 \right] \end{array} \right]$	$\rangle$

What is the role of these indices?

$$\begin{array}{c}
 \text{INPUT} \\
 \left[\begin{array}{c} \text{pi-rule} \\ \left\langle X, \left[\begin{array}{c} \text{SYN} \left[\begin{array}{c} \text{HEAD} \left[\begin{array}{c} \text{verb} \\ \text{FORM} \quad \text{fin} \\ \text{POL} \quad - \\ \text{AUX} \quad + \end{array} \right] \right] \\ \text{ARG-ST} \quad \langle \boxed{1} \rangle \oplus \boxed{A} \\ \text{SEM} \quad [\text{INDEX} \quad \textcircled{s_1}] \end{array} \right] \right\rangle \end{array} \right. \\
 \left. \begin{array}{c} \text{OUTPUT} \\ \left\langle Y, \left[\begin{array}{c} \text{SYN} \left[\begin{array}{c} \text{HEAD} \left[\begin{array}{c} \text{POL} \quad + \end{array} \right] \\ \text{VAL} \left[\begin{array}{c} \text{SPR} \quad \langle Z \rangle \end{array} \right] \end{array} \right] \\ \text{ARG-ST} \quad \langle \boxed{1} \rangle \oplus \left\langle \begin{array}{c} \text{INDEX} \quad \textcircled{s_2} \\ \text{RESTR} \quad \left\langle \begin{array}{c} \text{ADV}_{pol} \\ \left[\text{ARG} \quad \textcircled{s_1} \right] \end{array} \right\rangle \end{array} \right\rangle \oplus \boxed{A} \\ \text{SEM} \quad [\text{INDEX} \quad \textcircled{s_2}] \end{array} \right] \right\rangle \end{array} \right.
 \end{array}
 \right]
 \end{array}$$

Which *nots* does the rule license?

$$\begin{array}{c}
 \text{INPUT} \\
 \text{OUTPUT}
 \end{array}
 \left[\begin{array}{c}
 \text{pi-rule} \\
 \left\langle X, \left[\begin{array}{c}
 \text{SYN} \left[\begin{array}{c} \text{HEAD} \left[\begin{array}{c} \text{verb} \\ \text{FORM} \text{ fin} \\ \text{POL} - \\ \text{AUX} + \end{array} \right] \right] \\
 \text{ARG-ST} \langle \boxed{1} \rangle \oplus \boxed{A} \\
 \text{SEM} \left[\text{INDEX} \ s_1 \right]
 \end{array} \right] \right\rangle \\
 \left\langle Y, \left[\begin{array}{c}
 \text{SYN} \left[\begin{array}{c} \text{HEAD} \left[\text{POL} + \right] \\ \text{VAL} \left[\text{SPR} \ \langle Z \rangle \right] \end{array} \right] \\
 \text{ARG-ST} \langle \boxed{1} \rangle \oplus \left\langle \begin{array}{c} \text{ADV}_{pol} \\ \left[\text{INDEX} \ s_2 \right] \\ \left[\text{RESTR} \ \left\langle \left[\text{ARG} \ s_1 \right] \right\rangle \right] \end{array} \right\rangle \oplus \boxed{A} \\
 \text{SEM} \left[\text{INDEX} \ s_2 \right]
 \end{array} \right] \right\rangle
 \end{array} \right]
 \end{array}$$

Andy must not have been sleeping?



Andy must have not been sleeping?



Andy must have been not sleeping?



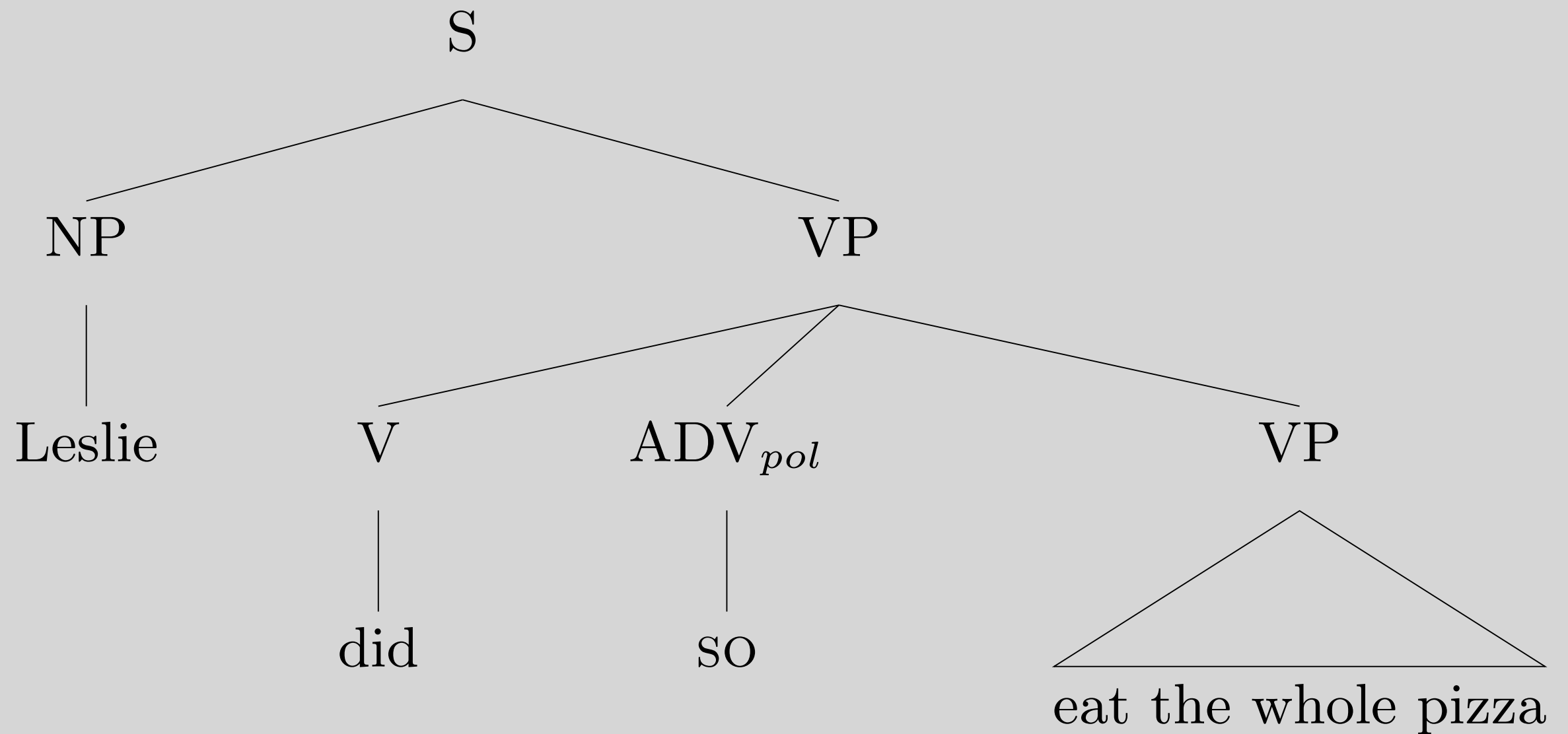
Kleptomaniacs cannot not steal.



Kleptomaniacs cannot not steal.



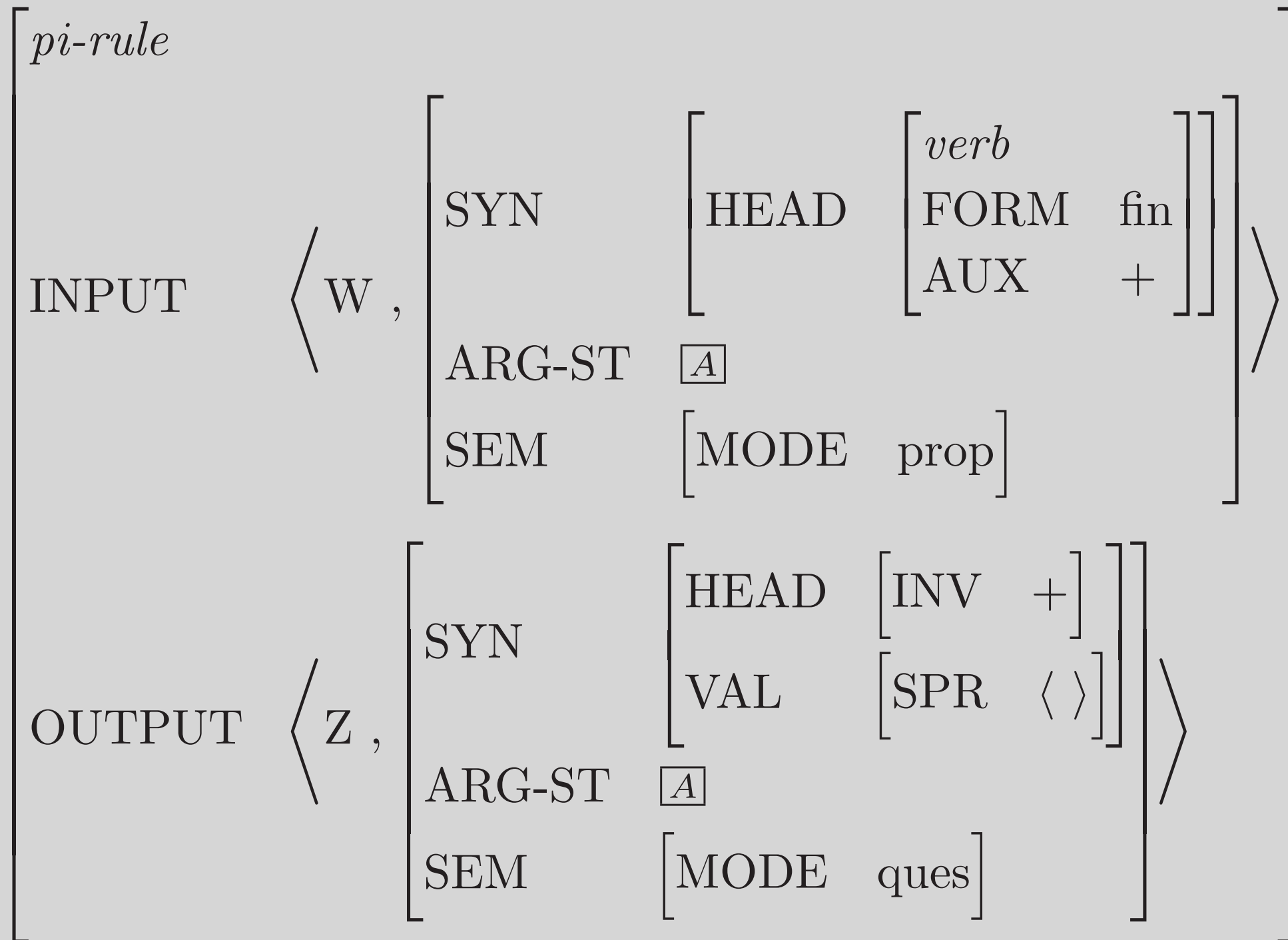
Negation and Reaffirmation: A Sample Tree



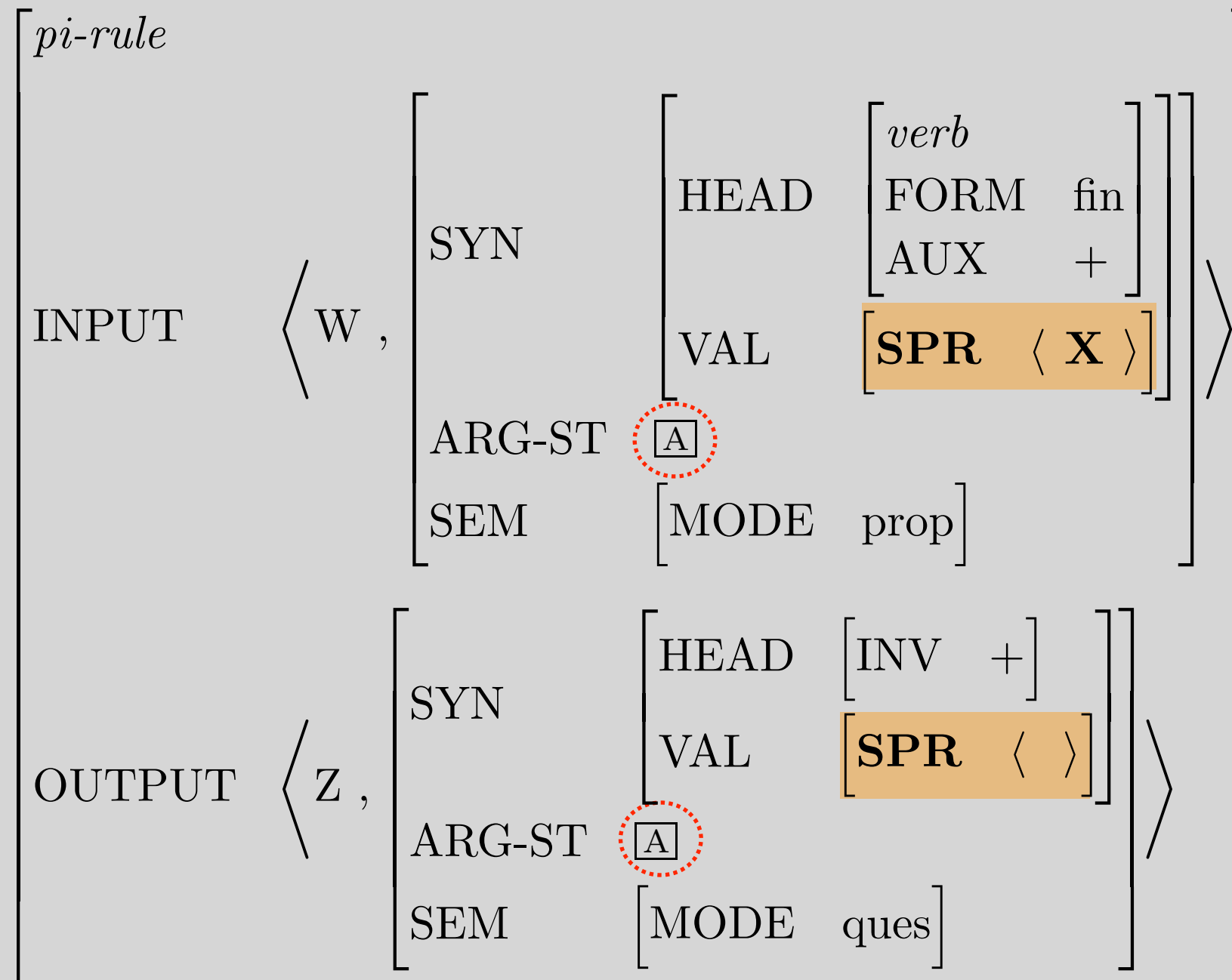
Inversion

- Yes-no questions begin with an auxiliary:
Will Robin win?
- The NP after the auxiliary has all the properties of a subject
 - Agreement: *Have they left?* vs. **Has they left?*
 - Case: **Have them left?*
 - Raising: *Will there continue to be food at the meetings?*
- What happens if you make a question out of a sentence without an auxiliary?
Robin won
Did Robin win?

The Inversion Lexical Rule



How the Rule Yields Inverted Order



...plus the ARP

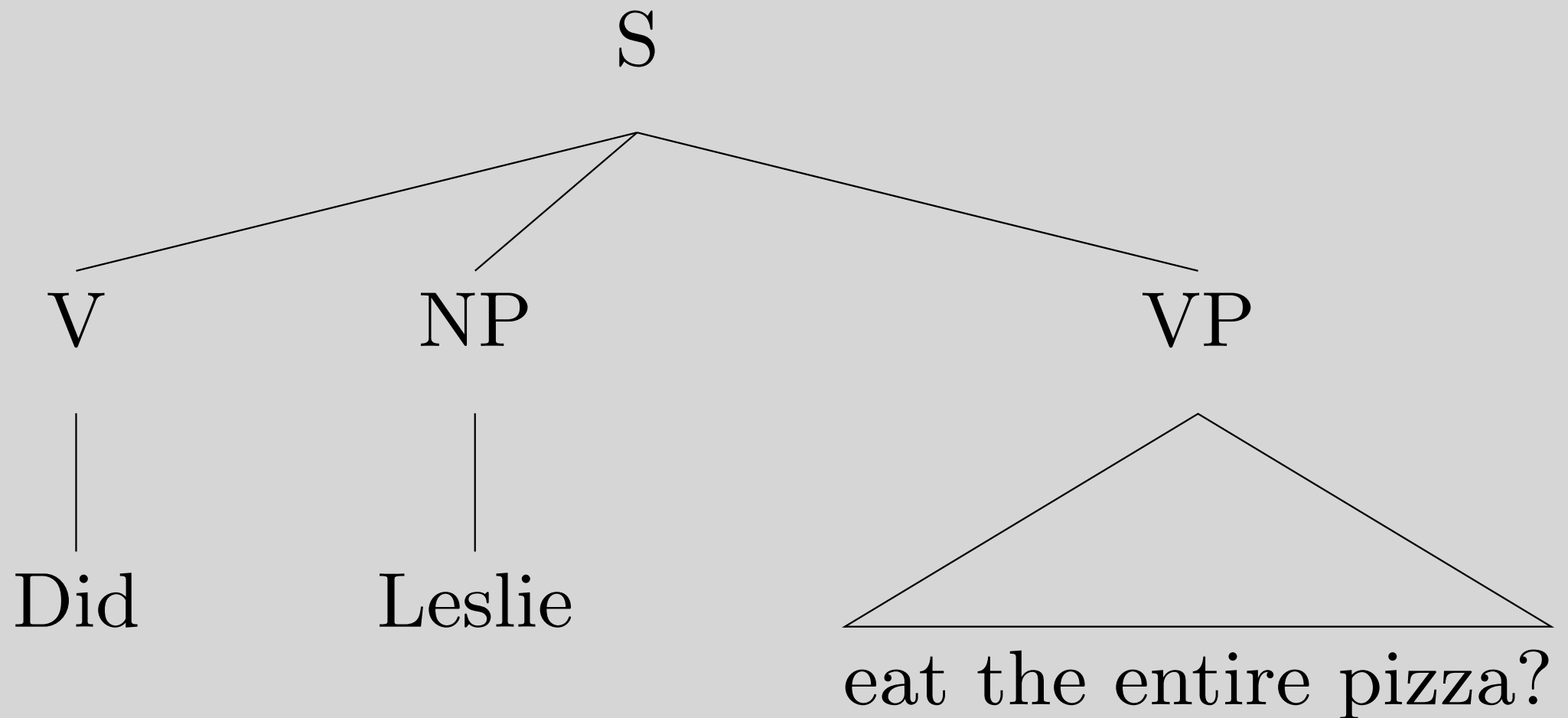
The Feature INV

- What is the INV value of inputs to the Inversion LR?
 - Perhaps surprisingly, the input is [INV +]
 - Word-to-word rules (*pi-rules*) have default identity of HEAD features, and no INV value is given on the input
- Then what work is the feature doing?
 - It's used to mark auxiliaries that can't or must be inverted
You better watch out vs. **Better you watch out*
I shall go (*shall* ~ 'will') vs. *Shall I go?* (*shall* ~ 'should')

Other Cases of Inversion

- Inversion is not limited to questions
- Preposed negatives: *Never have I been so upset!*
- Conditionals: *Had we known, we would have left.*
- Exclamations: *May your teeth fall out!*
- Does our rule account for these?
- No. Our rule's output says [MODE ques]. And each construction has slightly different idiosyncrasies.
- How might we extend our analysis to cover them?
- Define a type of inversion lexical rules, sharing certain properties, but with some differences.

Inversion: A Sample Tree



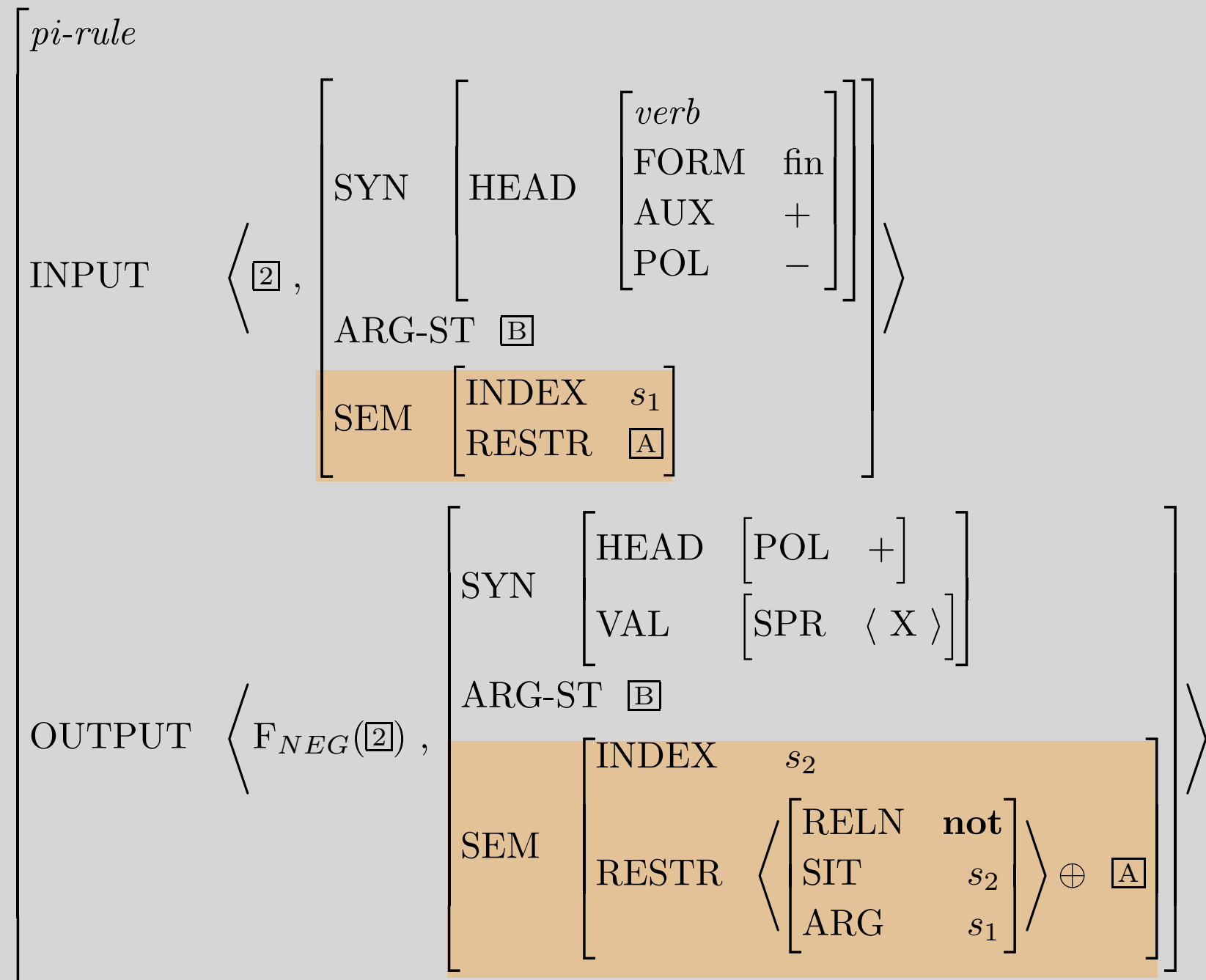
Contraction

- There are several types of contraction in English, but we're only talking about words ending in *n't*
- It may seem like just *not* said fast, but there's more to it
 - Only finite verbs can take *n't*:
**Terry must haven't seen us*
 - There are morphological irregularities:
won't, not **willn't* *%shan't*, not **shalln't*
mustn't pronounced *mussn't*
don't pronounced *doen't*, not *dewn't*
**amn't*

The Contraction Lexical Rule

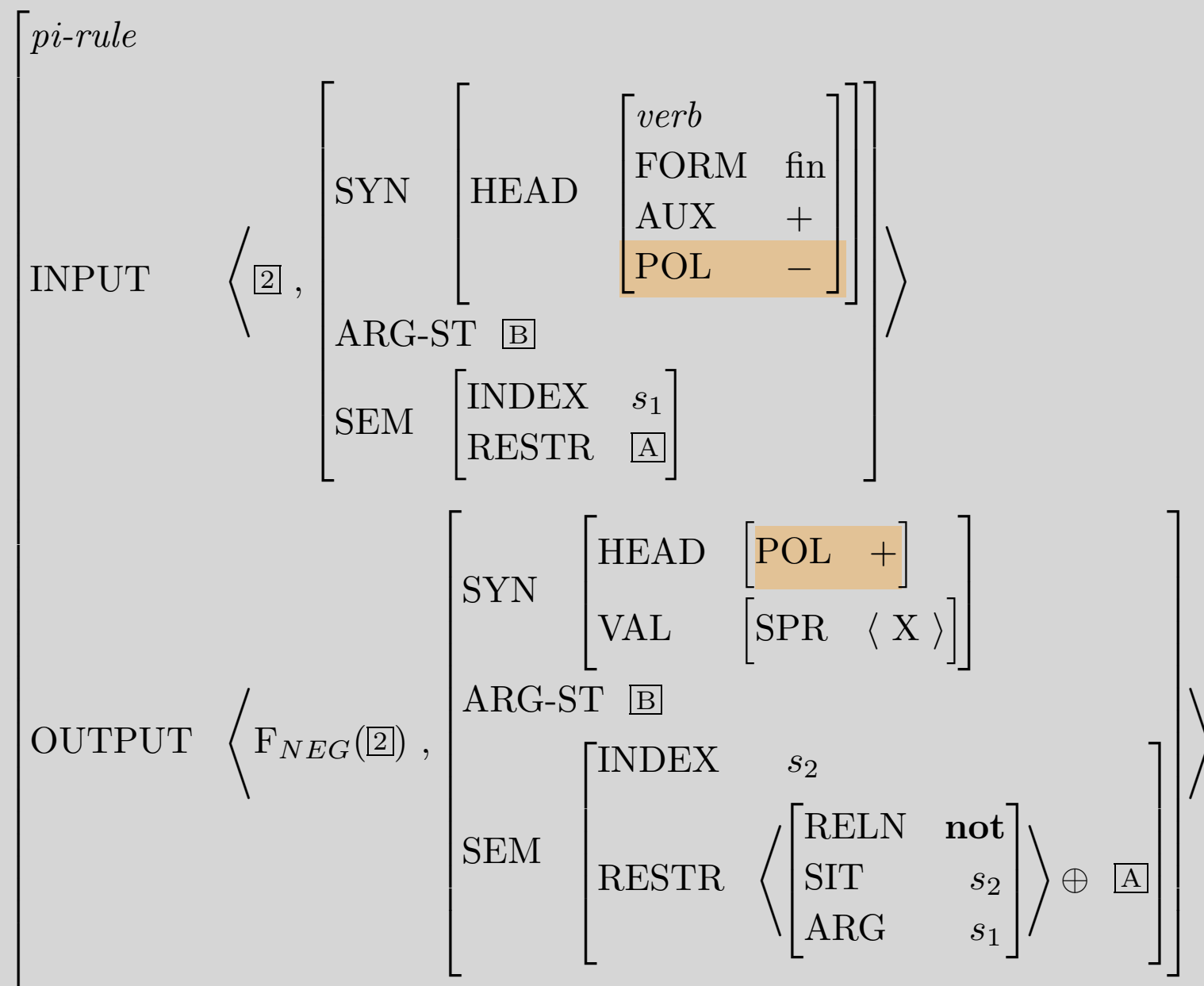
$$\begin{array}{l}
 \text{INPUT} \left\langle \boxed{2}, \left[\begin{array}{l} \text{SYN} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{verb} \\ \text{FORM} \quad \text{fin} \\ \text{AUX} \quad + \\ \text{POL} \quad - \end{array} \right] \\ \text{ARG-ST} \quad \boxed{B} \\ \text{SEM} \left[\begin{array}{l} \text{INDEX} \quad s_1 \\ \text{RESTR} \quad \boxed{A} \end{array} \right] \end{array} \right] \right\} \\
 \text{OUTPUT} \left\langle F_{NEG}(\boxed{2}), \left[\begin{array}{l} \text{SYN} \left[\begin{array}{l} \text{HEAD} \left[\begin{array}{l} \text{POL} \quad + \\ \text{VAL} \left[\begin{array}{l} \text{SPR} \quad \langle X \rangle \end{array} \right] \end{array} \right] \\ \text{ARG-ST} \quad \boxed{B} \\ \text{SEM} \left[\begin{array}{l} \text{INDEX} \quad s_2 \\ \text{RESTR} \left\langle \begin{array}{l} \text{RELN} \quad \text{not} \\ \text{SIT} \quad s_2 \\ \text{ARG} \quad s_1 \end{array} \right\rangle \oplus \boxed{A} \end{array} \right] \end{array} \right] \right\}
 \end{array}
 \right.
 \end{array}$$

Most of the work is in the semantics



Why?

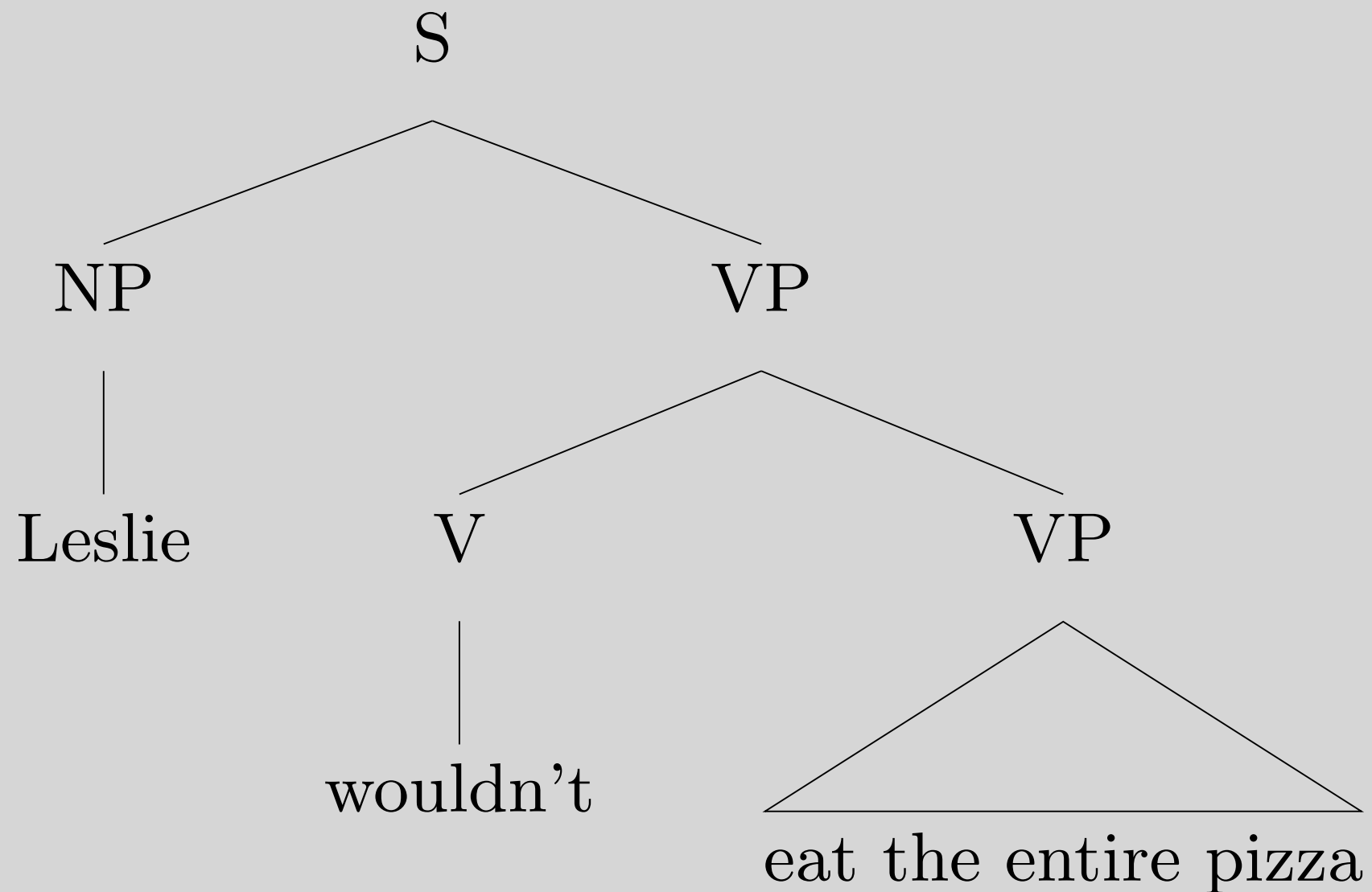
What does POL do?



**We can'tn't stop*

**They won't TOO mind*

Contraction: Sample Tree





How do you feel about very simple trees in our NICE analyses?



Satisfyingly elegant/simple

0

Seems to hide a lot of complexity

0

That's it?!

0

Other

0

Ellipsis

- Ellipsis allows VPs to be omitted, so long as they would have been preceded by an auxiliary

* *Pat couldn't have been watching us, but
Chris*

- Unlike the other NICE properties, this holds of all auxiliaries, not just finite ones.
- What is the elliptical counterpart to a sentence with no auxiliary?

Whenever Pat watches TV, Chris watches TV
Whenever Pat watches TV, Chris does

The Ellipsis Lexical Rule

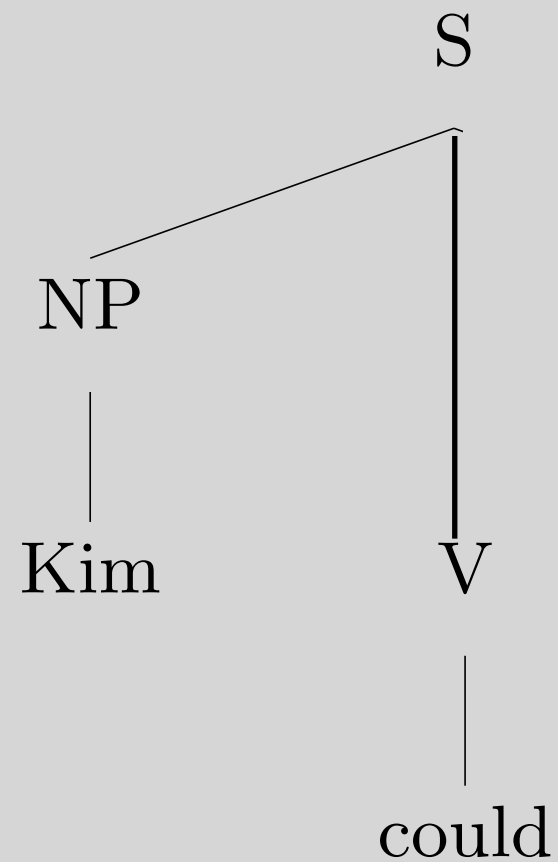
$$\left[\begin{array}{l} d\text{-rule} \\ \text{INPUT} \quad \left\langle \boxed{1}, \left[\begin{array}{l} auxv\text{-}lxm \\ \text{ARG-ST} \quad \langle \boxed{2} \rangle \oplus \boxed{A} \end{array} \right] \right\rangle \\ \text{OUTPUT} \quad \left\langle \boxed{1}, \left[\begin{array}{l} derivv\text{-}lxm \\ \text{ARG-ST} \quad \langle \boxed{2} \rangle \end{array} \right] \right\rangle \end{array} \right]$$

- Note that this is a derivational LR (*d-rule*) -- that is, lexeme-to-lexeme
- This means that SYN and SEM are unchanged, by default

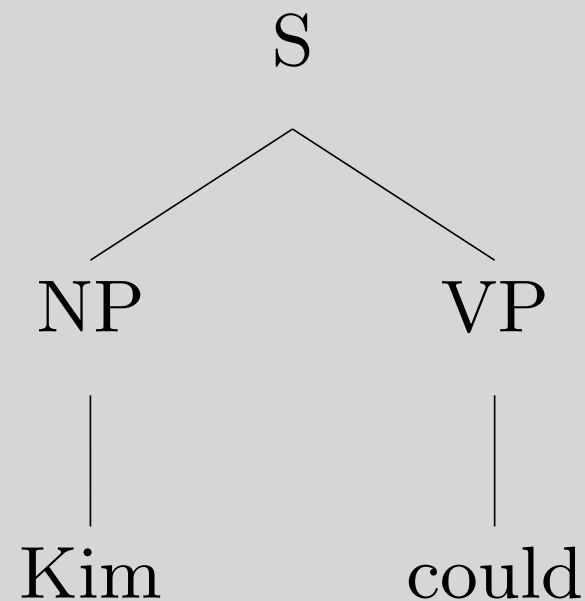
Ellipsis: A Sample Output

$\langle$ could , $\rangle$		<i>dervv-lxm</i>	
		SYN	$\left[\begin{array}{l} \text{HEAD} \left[\begin{array}{ll} \text{FORM} & \text{fin} \\ \text{AUX} & + \\ \text{POL} & - \\ \text{AGR} & \boxed{1} \end{array} \right] \\ \text{VAL} \left[\text{SPR} \quad \langle [\text{AGR } \boxed{1}] \rangle \right] \end{array} \right]$
		ARG-ST	$\langle \text{NP} \rangle$
		SEM	$\left[\begin{array}{ll} \text{MODE} & \text{prop} \\ \text{INDEX} & s_1 \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \text{could} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2 \end{array} \right] \right\rangle \end{array} \right]$

Ellipsis: A Sample Tree



Semantics of Ellipsis



What is the SEM value of the S node of this tree?

$$\left[\begin{array}{ll} \text{INDEX} & s_1 \\ \text{MODE} & \text{prop} \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \text{name} \\ \text{NAME} & \text{Kim} \\ \text{NAMED} & i \end{array} \right], \left[\begin{array}{ll} \text{RELN} & \text{could} \\ \text{SIT} & s_1 \\ \text{ARG} & s_2 \end{array} \right] \right\rangle \end{array} \right]$$

Note: s_2 has to be filled in by context.

Infinitival *to* Revisited

- VP Ellipsis can occur after *to*:

We didn't find the solution, but we tried to.

- This is covered by our Ellipsis LR if we say *to* is [AUX +].
- Since AUX is declared on type *verb*, it follows that *to* is a verb.

do Revisited

- Chomsky's old analysis: in sentences w/o auxiliaries...
 - Tense can get separated from the verb in various ways
 - Negation/Reaffirmation inserts something between Tense and the following verb
 - Inversion moves Tense to the left of the subject NP
 - Ellipsis deletes what follows Tense
 - When this happens, *do* is inserted to support Tense
- Our counterpart:
 - NICE properties hold only of auxiliaries
 - *do* is a semantically empty auxiliary, so negated, reaffirmed, inverted, and elliptical sentences that are the semantic counterparts to sentences w/o auxiliaries are ones with *do*.

Summary

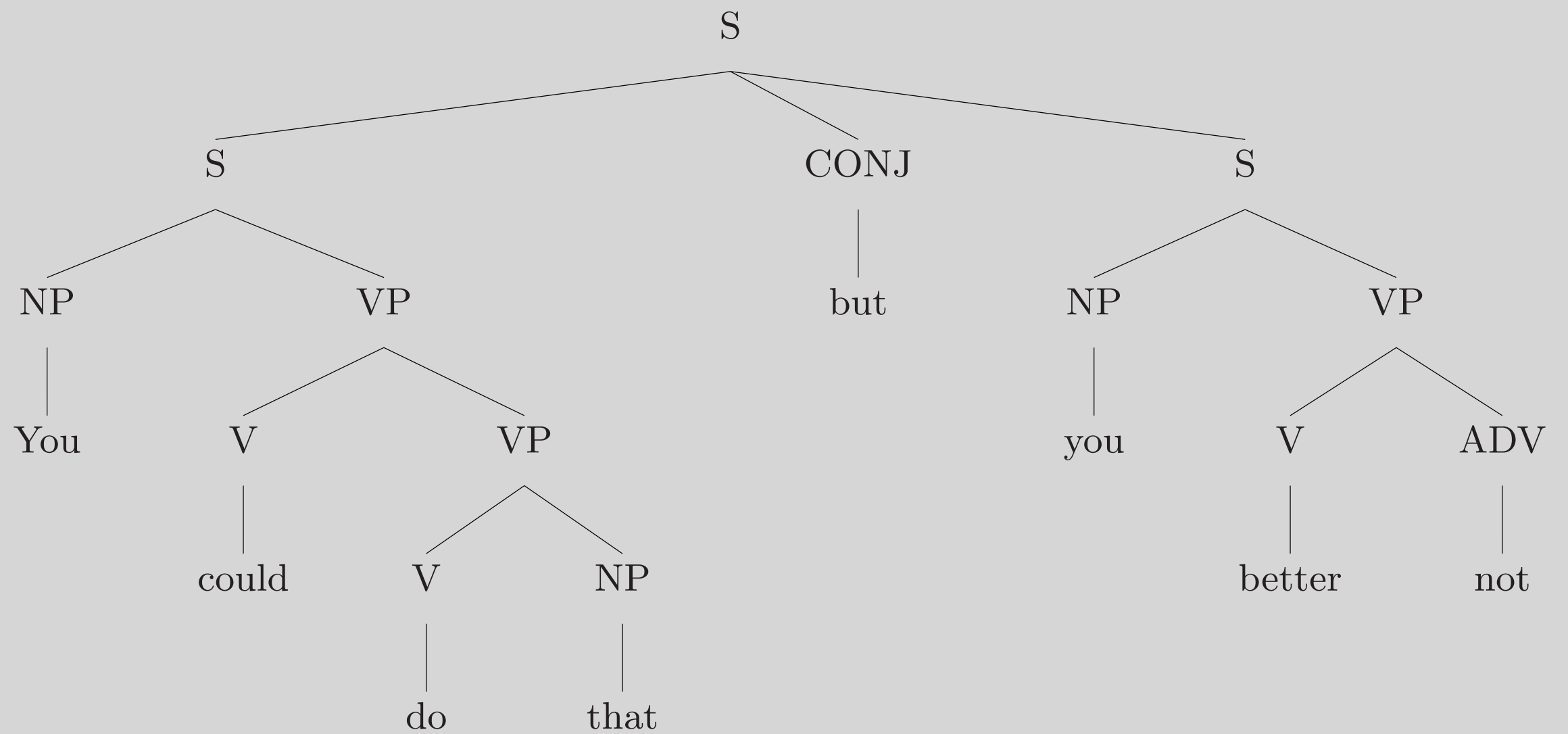
- Our analysis employs straightforward mechanisms
 - Lexical entries for auxiliaries
 - 3 new features (AUX, POL, INV)
 - 4 lexical rules
- We handle a complex array of facts
 - co-occurrence restrictions (ordering & iteration)
 - the NICE properties
 - auxiliary *do*
 - combinations of NICE constructions

Overview

- NICE properties of auxiliaries
- The auxiliary *do*
- NICE properties (lexical rules)
- Reading questions

Practice tree

- You could do that, but you better not



(52) ADV_{pol} -Addition Lexical Rule (with inherited constraints)

$$\left[\begin{array}{c} \textit{pi-rule} \\ \text{INPUT} \left\langle \boxed{0}, \right. \\ \left. \begin{array}{c} \textit{word} \\ \text{SYN} \left[\begin{array}{c} \text{HEAD} \left[\begin{array}{c} \textit{verb} \\ \text{FORM} \quad \text{fin} \\ \text{POL} \quad - \\ \text{AUX} \quad + \end{array} \right] \\ \text{VAL} \left[\begin{array}{c} \text{MOD} \quad \boxed{B} \end{array} \right] \end{array} \right] \\ \text{ARG-ST} \quad \langle \boxed{1} \rangle \oplus \boxed{A} \\ \text{SEM} \left[\begin{array}{c} \text{INDEX} \quad s_1 \\ \text{MODE} \quad \boxed{2} \\ \text{RESTR} \quad \boxed{C} \end{array} \right] \end{array} \right] \\ \text{OUTPUT} \left\langle \boxed{0}, \right. \\ \left. \begin{array}{c} \textit{word} \\ \text{SYN} \left[\begin{array}{c} \text{HEAD} \left[\begin{array}{c} \textit{verb} \\ \text{FORM} \quad \text{fin} \\ \text{POL} \quad + \\ \text{AUX} \quad + \end{array} \right] \\ \text{VAL} \left[\begin{array}{c} \text{SPR} \quad \langle \boxed{1} \rangle \\ \text{COMPS} \quad \langle \boxed{3} \rangle \oplus \boxed{A} \\ \text{MOD} \quad \boxed{B} \end{array} \right] \end{array} \right] \\ \text{ARG-ST} \quad \langle \boxed{1} \rangle \oplus \left\langle \begin{array}{c} \boxed{3}ADV_{pol} \\ \left[\begin{array}{c} \text{INDEX} \quad s_2 \\ \text{RESTR} \quad \langle \boxed{ARG} \quad s_1 \rangle \end{array} \right] \end{array} \right\rangle \oplus \boxed{A} \\ \text{SEM} \left[\begin{array}{c} \text{INDEX} \quad s_2 \\ \text{MODE} \quad \boxed{2} \\ \text{RESTR} \quad \boxed{C} \end{array} \right] \end{array} \right] \end{array} \right]$$

Here it is important to see that the INPUT and OUTPUT specify conflicting values for POL and INDEX.¹⁴ As a result, the inherited defeasible identity constraints ‘push down’ to identify the values of all other features within HEAD and SEM whose values are not specified as incompatible.

- What does ‘push down’ mean here?
- What does ‘conflicting values’ have to do with it?

RQs: Negation

- I still don't understand why we don't want ADVpol to be complements of a finite auxiliary verb, rather than as a modifier. Technically, ADVpol is optional - wouldn't this support a theory of MOD? Also, it makes more sense to have ADVpol be modifying the auxiliary.
- Why isn't POL a feature on adv-lxms?

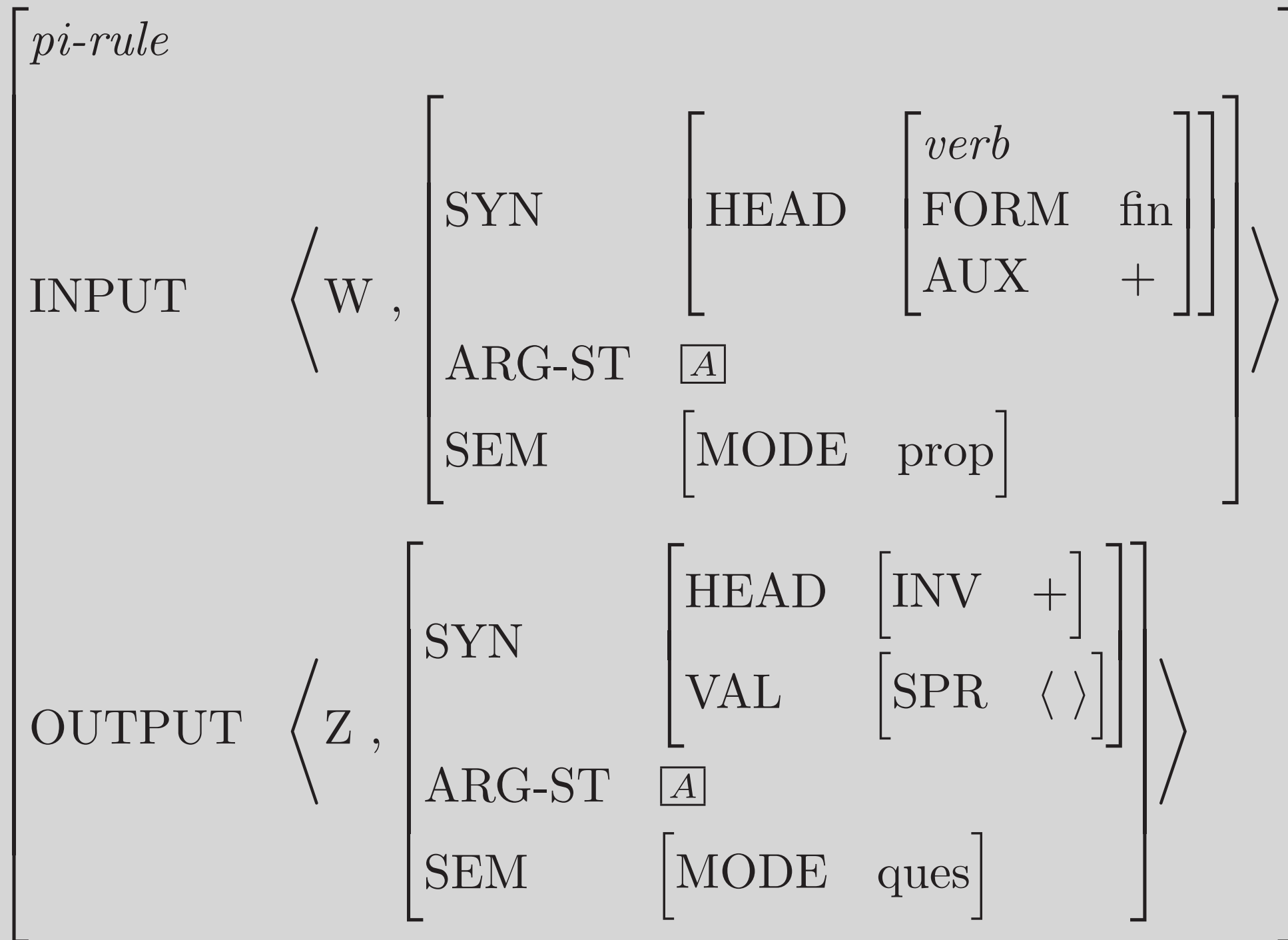
RQs: Negation

- Do other languages have similar double negation examples to the "Kleptos can't not steal" sentence having its elevated meaning beyond just "They can steal"? Is there a term for this? Or just the observation about there only being one sentential not per clause?

RQs: Inversion

- What does the feature INV actually do?
- I'm struggling with how the SHAC constraint is carried through the inversion rule to apply to the first complement of the verb in inverted sentences.

The Inversion Lexical Rule



RQs: Inversion

- How similar is this simplified rule compared to what I imagine how the ERG handles inversion? Its always interesting to know how much the textbook contents differ from a production system.
- English allows declarative questions with raising intonation like "You're coming?" without inversion. Some other languages also don't use inversion to mark questions. How would this grammar represent or forbid such cases -- do we need a separate construction, or are these outside the formal syntax covered here?

RQs: Contraction

- Contraction as a NICE property in 13.5.3 — is this Contraction Lexical Rule specific to American English? I don't usually use contractions in this way, but I know that some other speakers of English would have a preference for saying something like *I've not seen the movie* or *I'll not answer that* as opposed to *I haven't seen the movie* or *I won't answer that*. Or is there some other rule that would cover this? (since technically, will and have are still being used in contractions, they're just not being used specifically in the *-n't* contractions)

RQs: Ellipsis

- Ellipsis seems to have parallels to the imperative rule in the sense that we are removing parts of the sentence that are normally required. Why don't we handle these similarly?
- *We asked them to open the window, and they tried to.* How do we license the semantics of *to*, where it is referencing the situation of opening the window?

RQs: Ellipsis

- If ellipsis is handled purely by a lexical rule that strips off all complements from an [AUX +] element, how would we formally encode the extra discourse requirement that there be an appropriate antecedent VP in context?

RQs: Cross-linguistic

- How are auxiliary verbs identified in other languages? The NICE properties seem very English specific, and I'm curious what features from our aux verbs can be extrapolated to higher levels. For example, does the POL feature exist elsewhere?
- How English specific is this exactly? like are we just building a (admittedly very impressive) grammar of English or is this immediately transferable? or somewhere in between?

RQs: *Better*

- How do we come to the conclusion that "better" is an auxiliary verb, rather than an adverb?