

Ling 566  
Nov 29, 2022  
Long Distance Dependencies

# Overview

- Some examples of the phenomenon
- What is new and different about it
- Brief sketch of the TG approach
- Broad outlines of our approach
- Details of our approach
- Subject extraction
- Coordinate Structure Constraint
- Reading questions

# Examples

- *wh*-questions:

*What did you find?*

*Tell me who you talked to*

- relative clauses:

*the item that I found*

*the guy who(m) I talked to*

- topicalization:

*The manual, I can't find*

*Chris, you should talk to.*

- *easy*-adjectives:

*My house is easy to find.*

*Pat is hard to talk to.*



# W Did you catch a topicalization example in the wild?

No

Maybe?

Yes and it sounded weird

Yes and it sounded fine

Total Results: 0

# What these have in common

- There is a ‘gap’: nothing following *find* and *to*, even though both normally require objects.
- Something that fills the role of the element missing from the gap occurs at the beginning of the clause.
- We use topicalization and *easy*-adjectives to illustrate:

*The manual*, *I can't find* \_\_\_\_\_

*Chris* *is easy to talk to* \_\_\_\_\_

# Gaps and their fillers can be far apart:

- *The solution to this problem, Pat said that someone claimed you thought I would never find\_\_\_\_\_.*
  - *Chris is easy to consider it impossible for anyone but a genius to try to talk to\_\_\_\_\_.*
- ☞ That's why we call them "long distance dependencies"

# Fillers often have syntactic properties associated with their gaps

*Him, I haven't met \_\_\_\_.*

*\*He, I haven't met \_\_\_\_.*

*The scissors, Pat told us \_\_\_\_\_ were missing.*

*\*The scissors, Pat told us \_\_\_\_\_ was missing.*

*On Pat, you can rely \_\_\_\_.*

*\*To Pat, you can rely \_\_\_\_.*

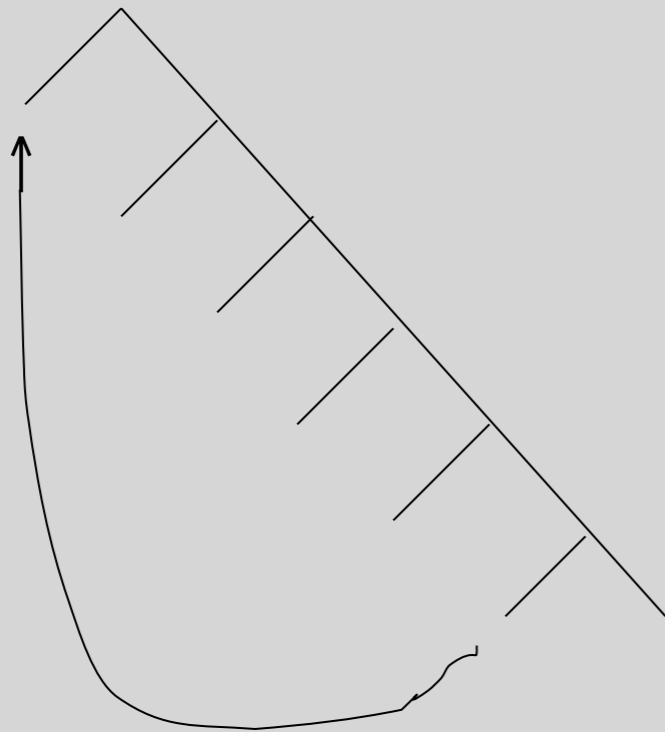
# LDDs in TG

- These were long thought to constitute the strongest evidence for transformations.
- They were handled in TG by moving the filler from the gap position.
- Case, agreement, preposition selection could apply before movement.

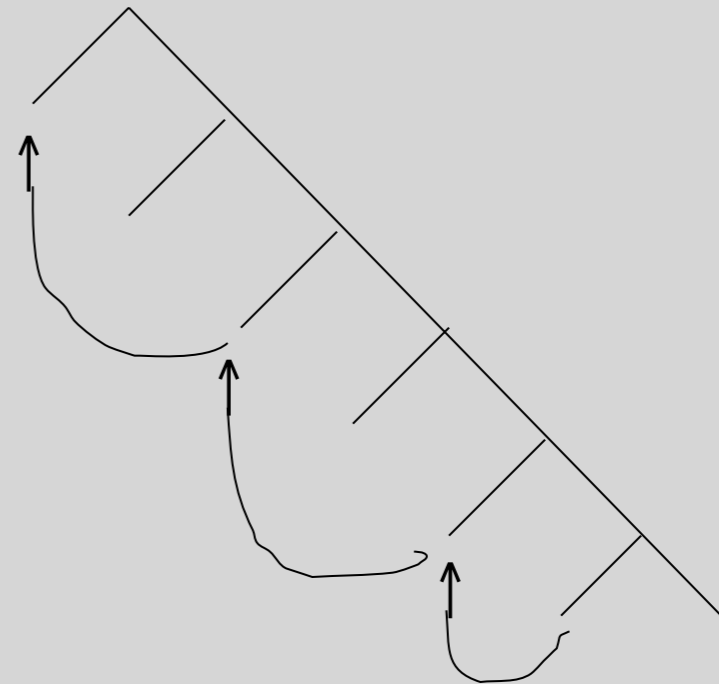


# A big debate about LDDs in TG

- Does long-distance movement take place in one fell swoop or in lots of little steps?



Swooping



Looping

# Looping is now generally accepted in TG

- Various languages show morphological marking on the verbs or complementizers of clauses between the filler and the gap.
- Psycholinguistic evidence indicates increased processing load in the region between filler and gap.
- This opens the door to non-transformational analyses, in which the filler-gap dependency is mediated by local information passing.

# Very Rough Sketch of Our Approach

- A feature GAP records information about a missing constituent.
- The GAP value is passed up the tree by a new principle.
- A new grammar rule expands S as a filler followed by another S whose GAP value matches the filler.
- Caveat: Making the details of this general idea work involves several complications.

# The Feature GAP

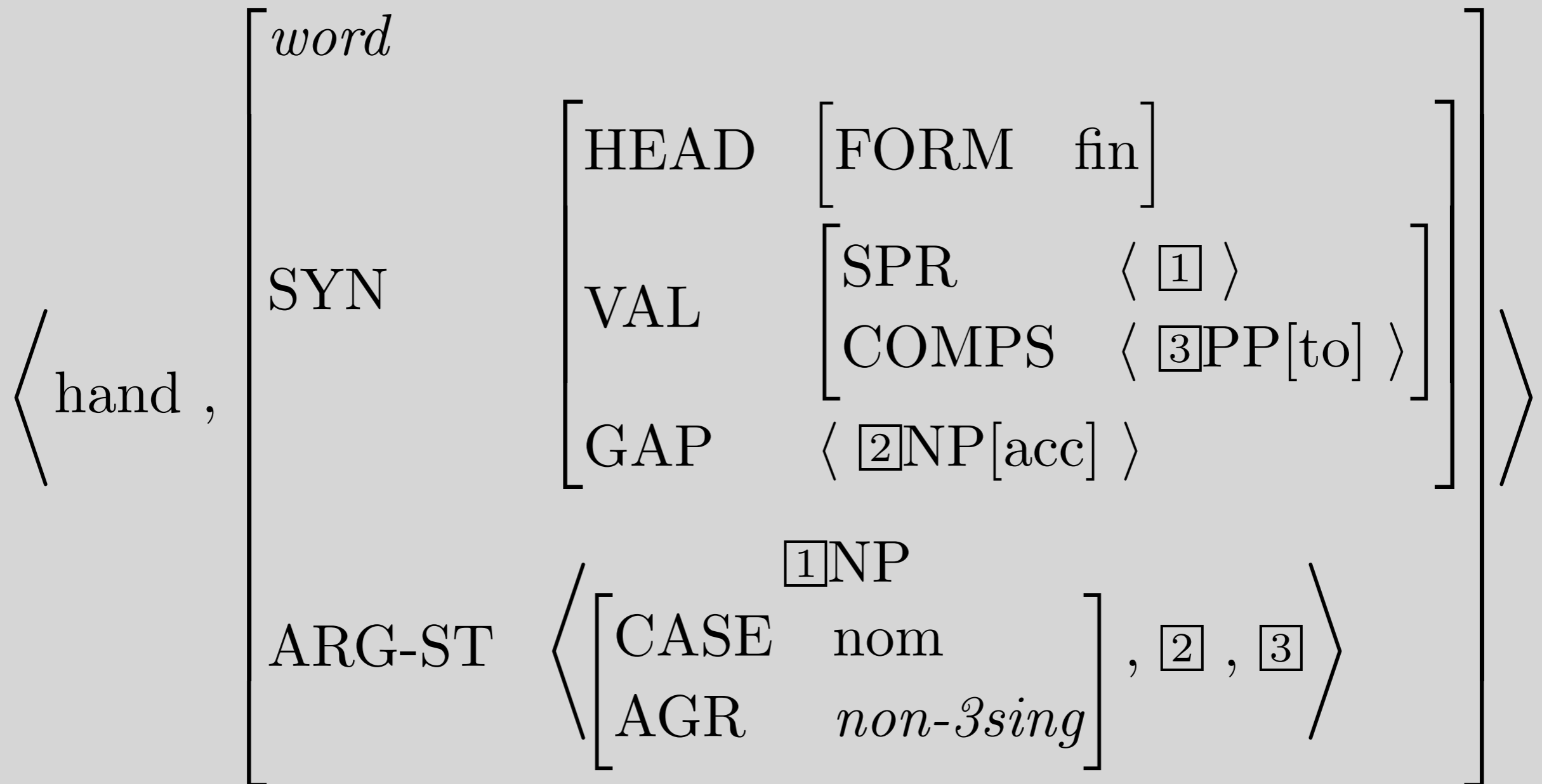
- Like valence features and ARG-ST, GAP's value is a list of feature structures (often empty).
- Subject gaps are introduced by a lexical rule.
- Non-subject gaps are introduced by revising the Argument Realization Principle.

# The Revised ARP

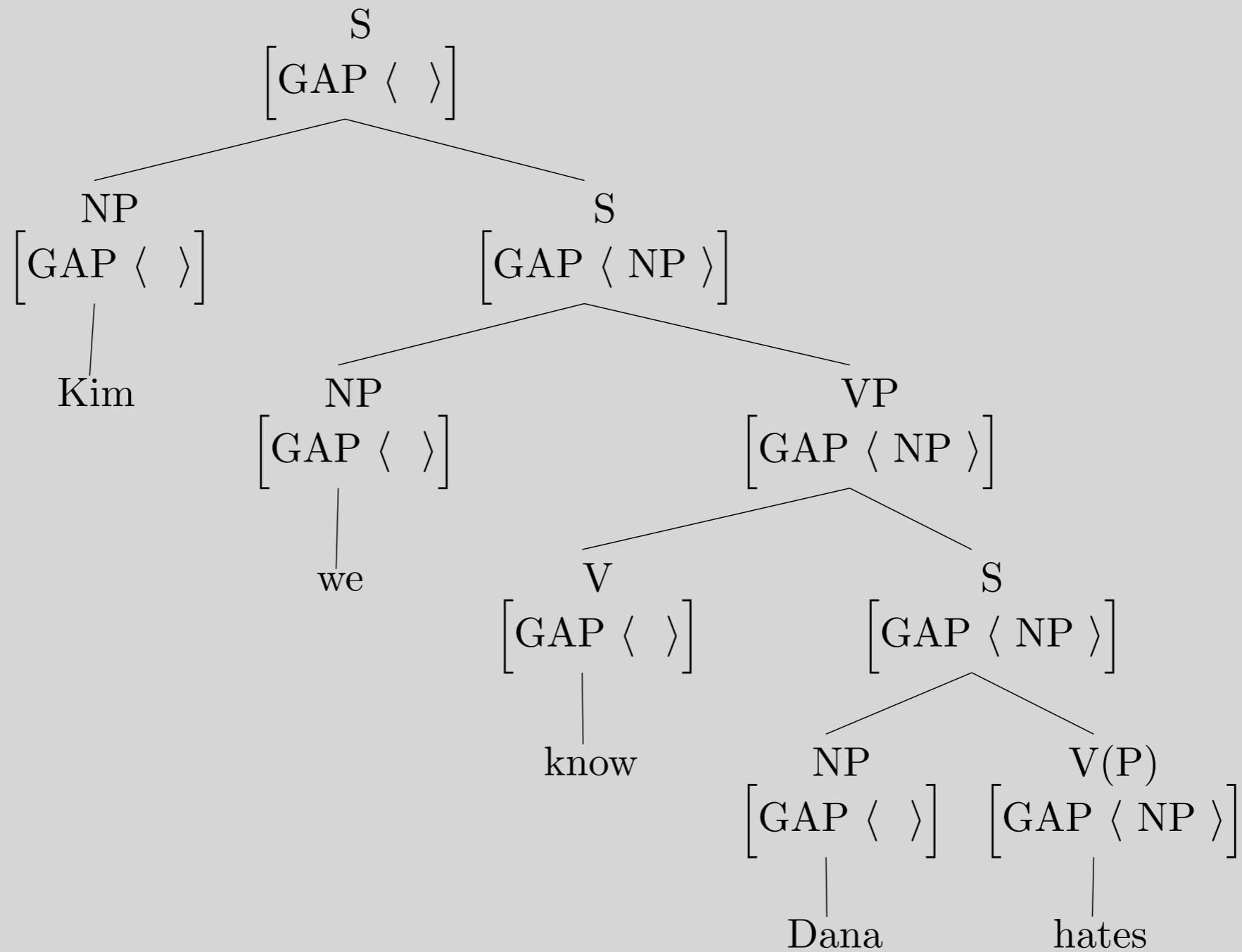
$$\text{word:} \left[ \begin{array}{l} \text{SYN} \\ \text{ARG-ST} \end{array} \left[ \begin{array}{l} \text{VAL} \\ \text{GAP} \end{array} \left[ \begin{array}{l} \text{SPR} \\ \text{COMPS} \end{array} \left[ \begin{array}{l} \boxed{A} \\ \boxed{B} \end{array} \ominus \boxed{C} \right] \right] \oplus \boxed{B} \right] \right]$$

- $\ominus$  is a kind of list subtraction, but:
  - it's not always defined, and
  - when defined, it's not always unique
- The ARP now says the non-SPR arguments are distributed between COMPS and GAP.

# A Word with a Non-Empty GAP Value



# How We Want GAP to Propagate



# What We Want the GAP Propagation Mechanism to Do

- Pass any GAP values from daughters up to their mothers,
- **except** when the filler is found.
- For topicalization, we can write the exception into the grammar rule, but
- For *easy*-adjectives, the NP that corresponds to the gap is the subject, which is introduced by the Head-Specifier Rule.
- Since specifiers are not generally gap fillers, we can't write the gap-filling into the HSR.

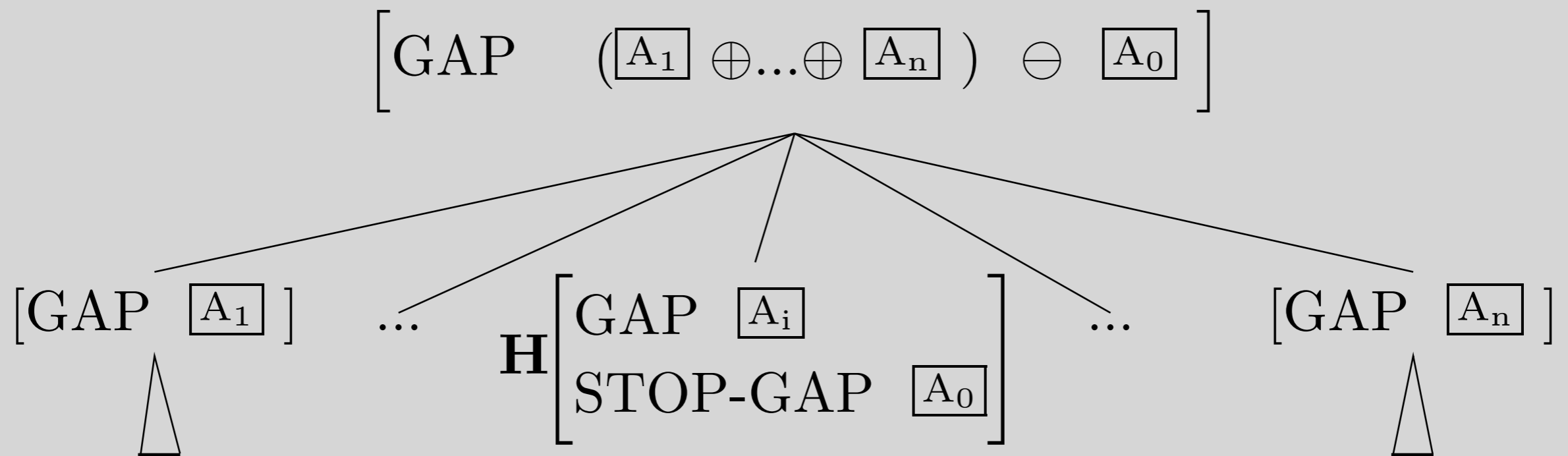


# Our Solution to this Problem

- For *easy*-adjectives, we treat the adjective formally as the filler, marking its SPR value as coindexed with its GAP value.
- We use a feature STOP-GAP to trigger the emptying of the GAP list.
  - STOP-GAP stops gap propagation
  - *easy*-adjectives mark STOP-GAP lexically
  - a new grammar rule, the Head-Filler Rule mentions STOP-GAP

# The GAP Principle

A local subtree  $\Phi$  satisfies the GAP Principle with respect to a headed rule  $\rho$  if and only if  $\Phi$  satisfies:



# How does STOP-GAP work?

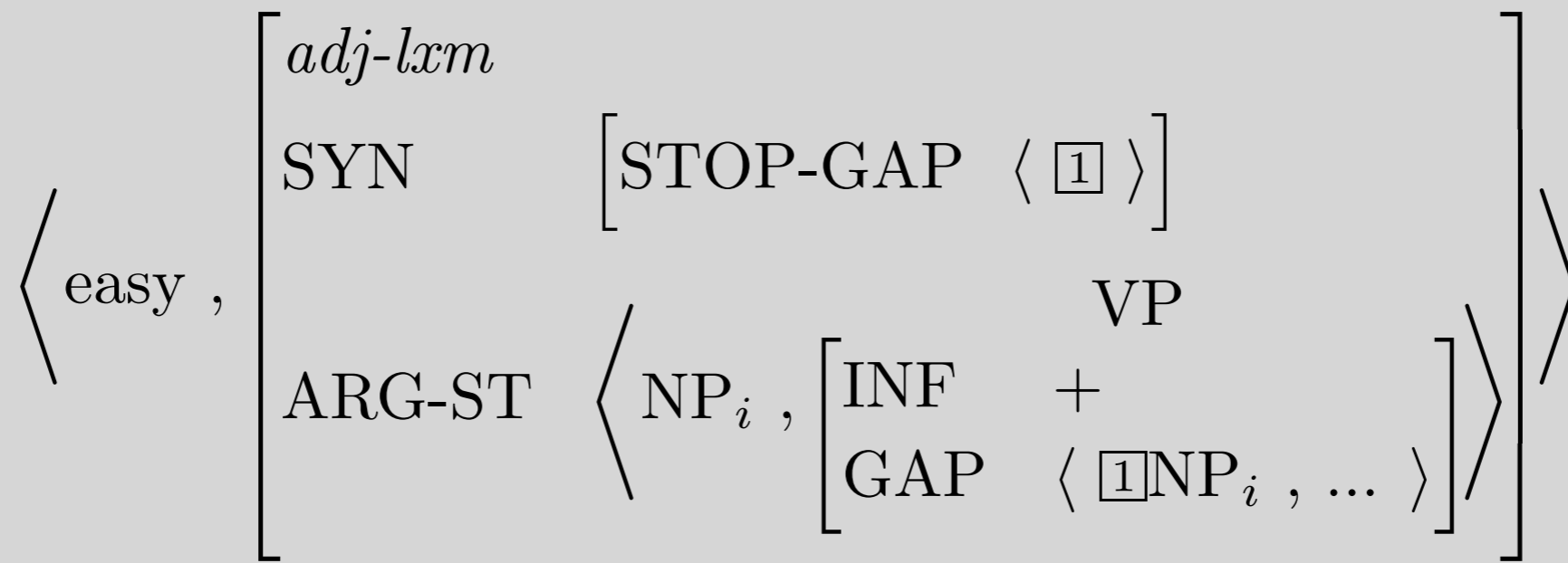
- STOP-GAP is empty almost everywhere
- When a gap is filled, STOP-GAP is nonempty, and its value is the same as the gap being filled.
- This blocks propagation of that GAP value, so gaps are only filled once.
- The nonempty STOP-GAP values come from two sources:
  - a stipulation in the Head-Filler Rule
  - lexical entries for *easy*-adjectives
- No principle propagates STOP-GAP

# The Head-Filler Rule

$$[phrase] \rightarrow \boxed{1} \left[ \text{GAP} \quad \langle \rangle \right] \mathbf{H} \left[ \begin{array}{l} \text{HEAD} \quad \left[ \begin{array}{l} \textit{verb} \\ \text{FORM} \quad \textit{fin} \end{array} \right] \\ \text{VAL} \quad \left[ \begin{array}{l} \text{SPR} \quad \langle \rangle \\ \text{COMPS} \quad \langle \rangle \end{array} \right] \\ \text{STOP-GAP} \quad \langle \boxed{1} \rangle \\ \text{GAP} \quad \langle \boxed{1} \rangle \end{array} \right]$$

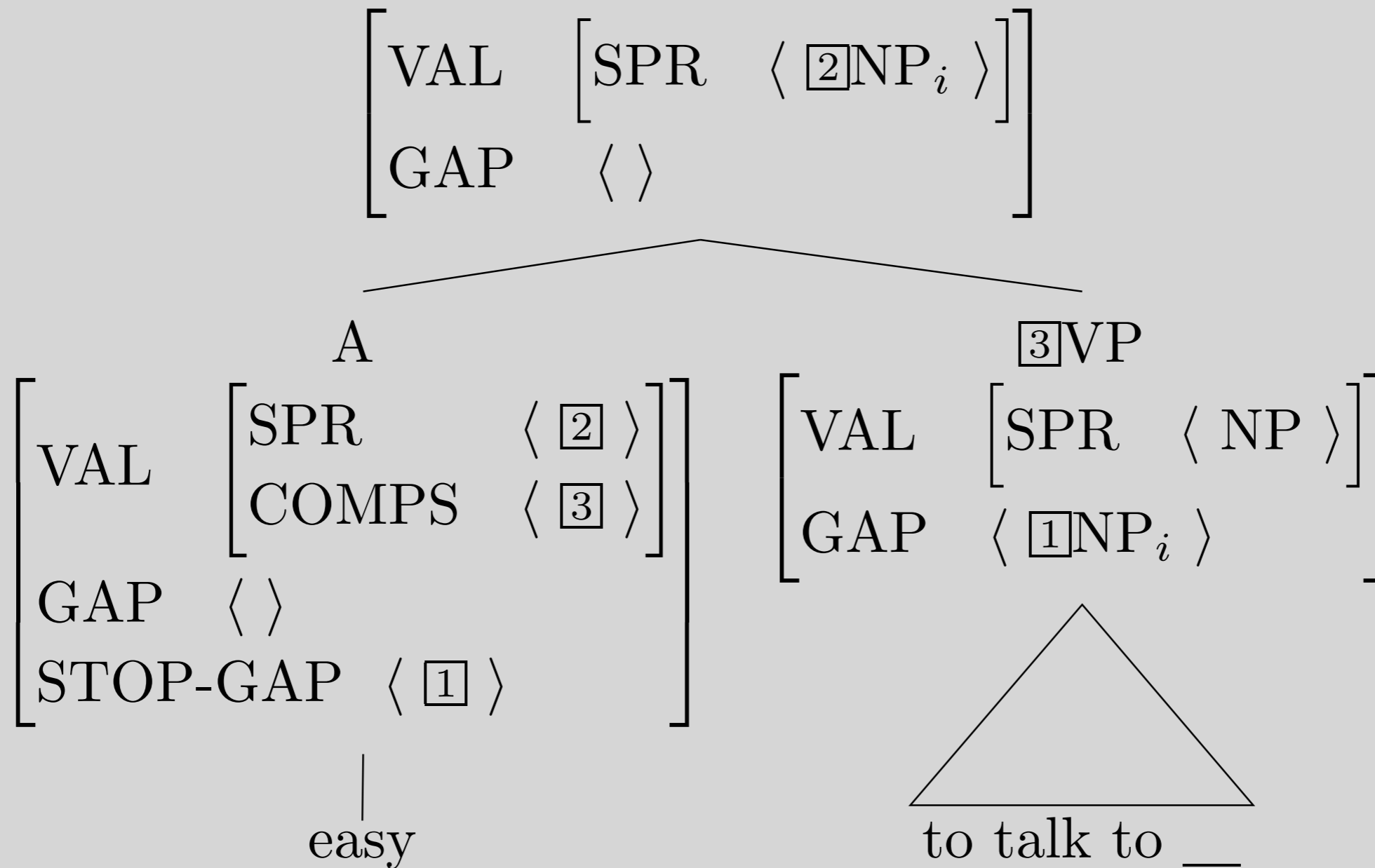
- This only covers gap filling in finite Ss
- The filler has to be identical to the GAP value
- The STOP-GAP value is also identical
- The GAP Principle ensures that the mother's GAP value is the empty list

# Gap Filling with *easy*-Adjectives



- Because STOP-GAP and GAP have the same value, that value will be subtracted from the mother's GAP value.
- The first argument is coindexed with the GAP value, accounting for the interpretation of the subject as the filler.

# A Tree for *easy to talk to* \_\_\_\_\_





# W SPR of complement of easy not-identified with anything

Hang on, that is a  
SPR requirement!

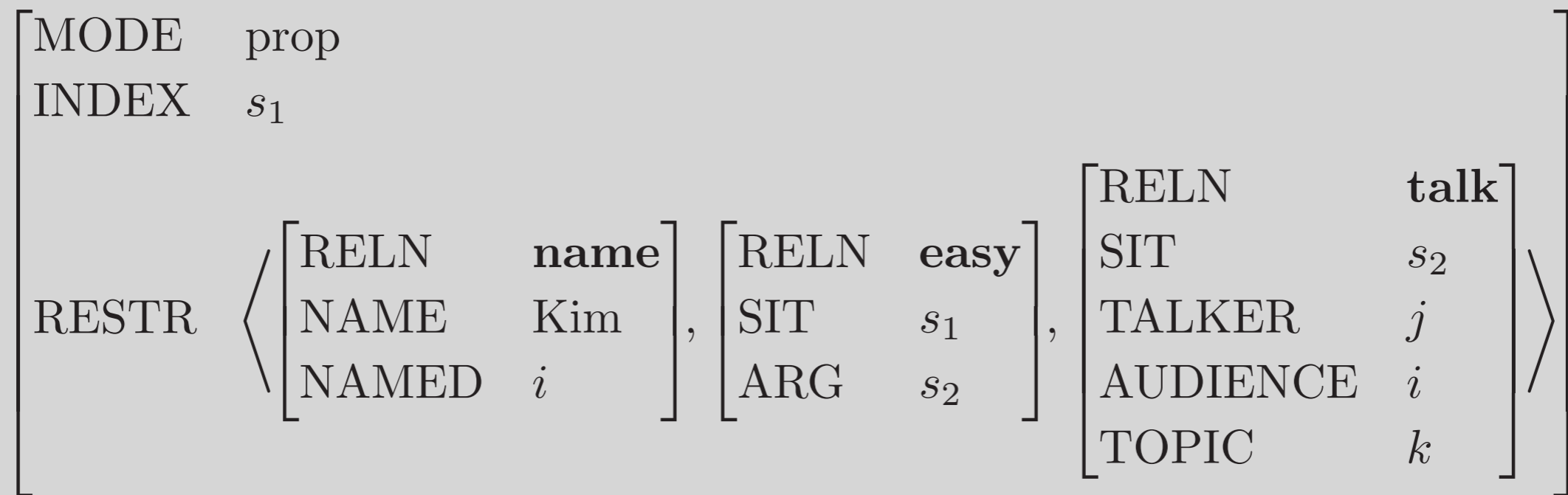
Seems fishy

That's cool, no  
worries

Ohh subtle ... nice!

Total Results: 0

# Semantics for *Kim is easy to talk to*





# STOP-GAP Housekeeping

- Lexical entries with nonempty STOP-GAP values (like *easy*) are rare, so STOP-GAP is by default empty in the lexicon.
- Head-Specifier and Head-Modifier rules need to say [STOP-GAP < >]
- Lexical rules preserve STOP-GAP values.

# GAP Housekeeping

- The initial symbol must say [GAP < >]. Why?
  - To block *\*Pat found* and *\*Chris talked to* as stand-alone sentences.
- The Imperative Rule must propagate GAP values. Why?
  - It's not a headed rule, so the effect of the GAP Principle must be replicated
  - Imperatives can have gaps:  
*This book, put on the top shelf!*

# Sentences with Multiple Gaps

- Famous examples:

*This violin, sonatas are easy to play\_\_\_ on\_\_\_.*

*\*Sonatas, this violin is easy to play\_\_\_ on\_\_\_.*

- Our analysis gets this:

- The subject of *easy* is coindexed with the **first** element of the GAP list.
- The Head-Filler rule only allows one GAP remaining.
- There are languages that allow multiple gaps more generally.

# Where We Are

- filler-gap structures:

*The solution to this problem, nobody understood\_\_\_\_\_*

*That problem is easy to understand\_\_\_\_\_*

- The feature GAP encodes information about missing constituents
- Modified ARP allows arguments that should be on the COMPS list to show up in the GAP list
- GAP values are passed up the tree by the GAP Principle

## Where We Are (continued)

- The feature STOP-GAP signals where GAP passing should stop
- The Head-Filler Rule matches a filler to a GAP and (via STOP-GAP) empties GAP
- Lexical entries for *easy*-adjectives require a gap in the complement, coindex the subject with the gap, and (via STOP-GAP) empty GAP on the mother

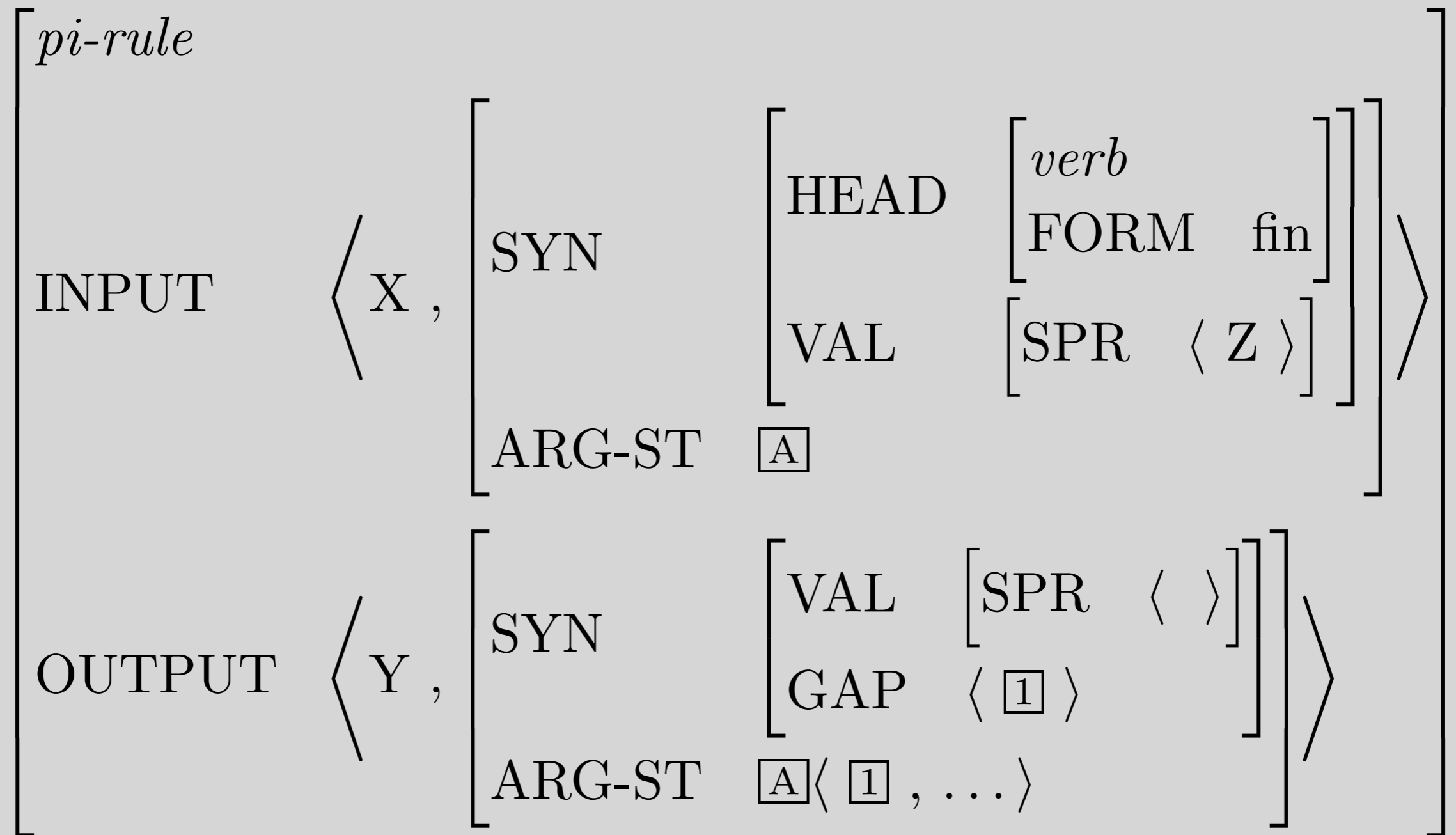
# On to New Material....

- Sentences with subject gaps
- Gaps in coordinate constructions

# Subject Gaps

- The ARP revision only allowed missing complements.
- But gaps occur in subject position, too:  
*This problem, everyone thought \_\_\_\_ was too easy.*
- We handle these via a lexical rule that, in effect, moves the contents of the SPR list into the GAP list

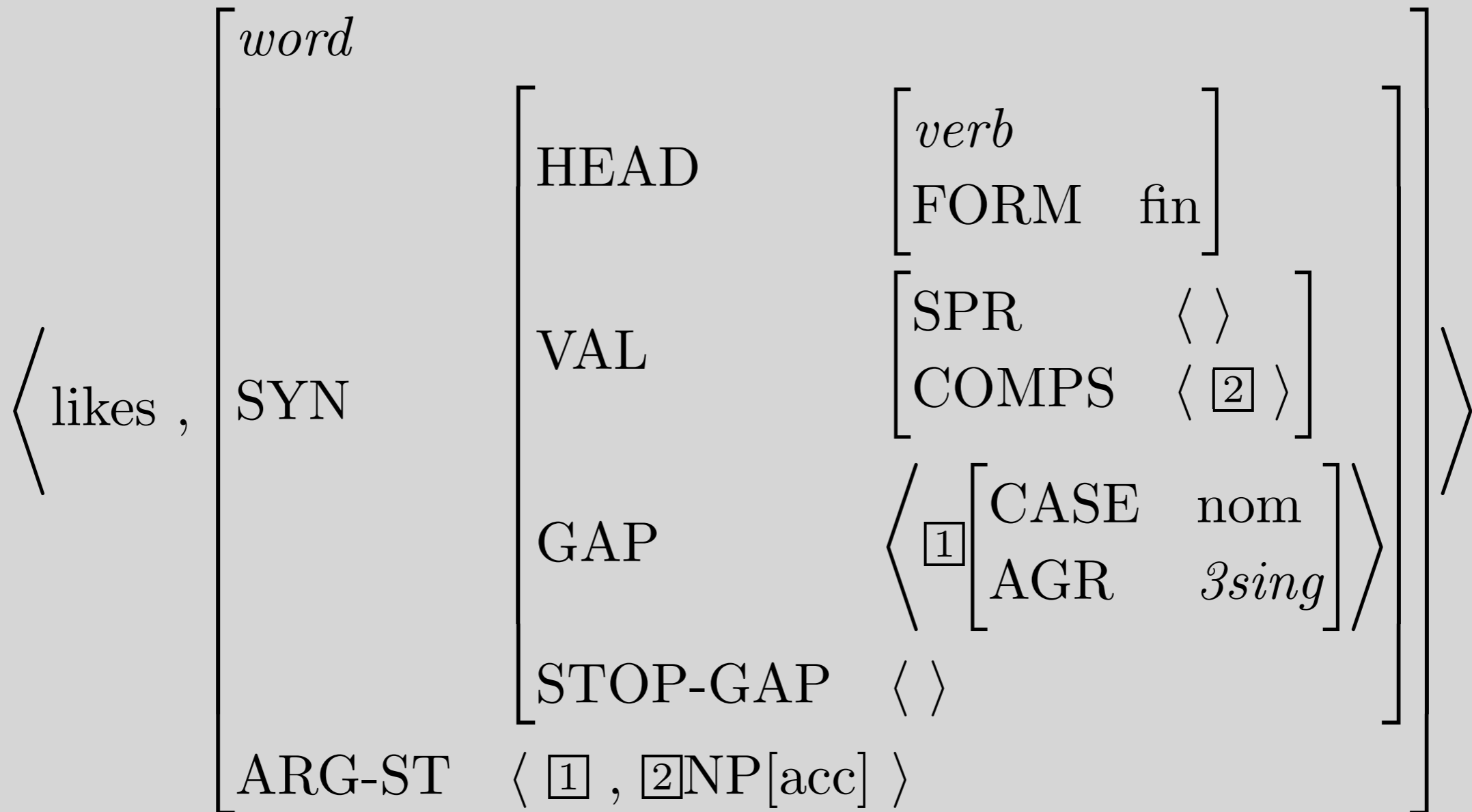
# The Subject Extraction Lexical Rule



- NB: This says nothing about the phonology, because the default for *pi-rules* is to leave the phonology unchanged.

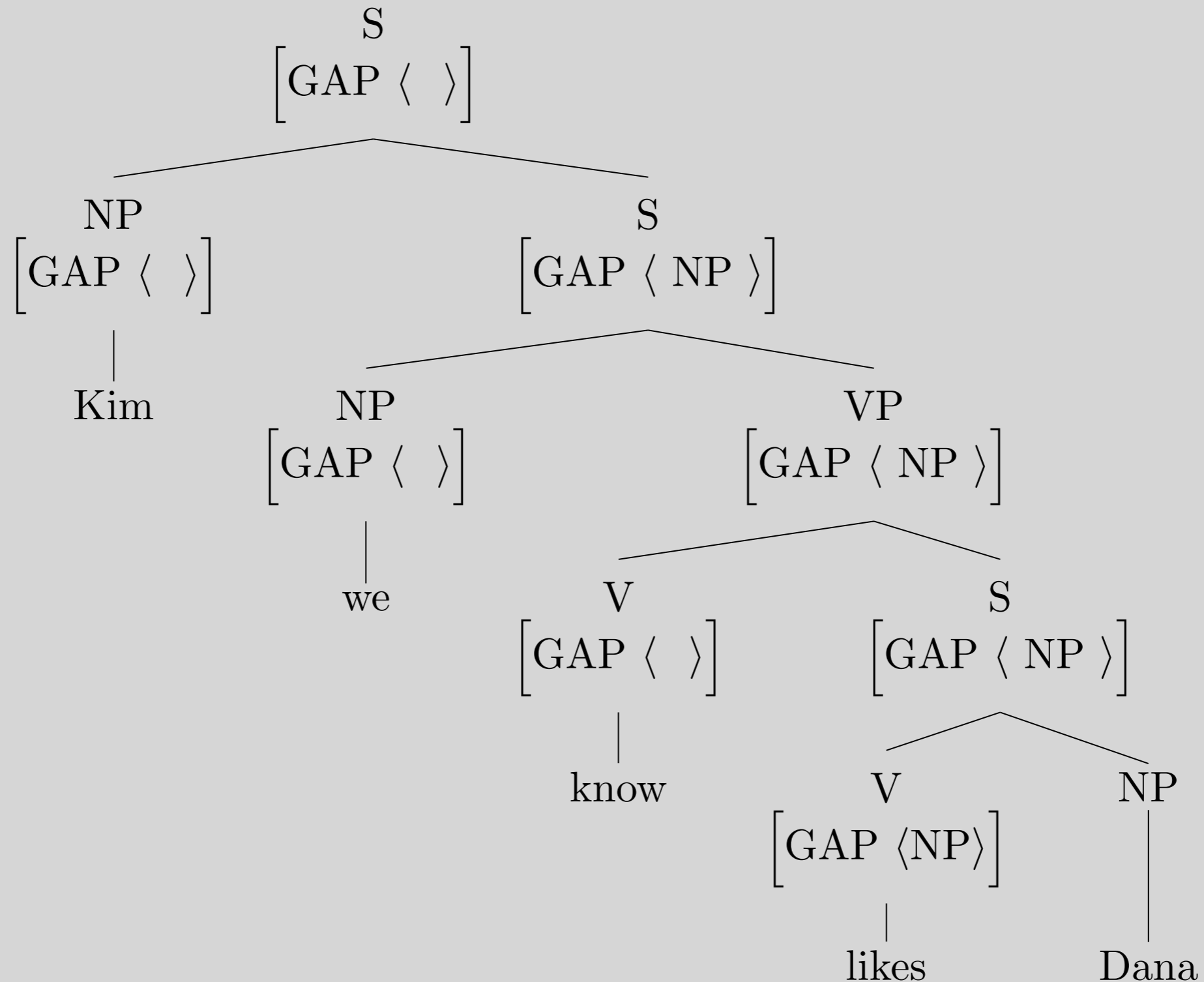


# A Lexical Sequence This Licenses



- Note that the ARP is satisfied

# A Tree with a Subject Gap



# Island Constraints

- There are configurations that block filler-gap dependencies, sometimes called “islands”
- Trying to explain them has been a central topic of syntactic research since the mid 1960s
- We’ll look at just one, Ross’s so-called “Coordinate Structure Constraint”
- Loose statement of the constraint: a constituent outside a coordinate structure cannot be the filler for a gap inside the coordinate structure.

# Coordinate Structure Constraint Examples

- \*This problem, nobody finished the extra credit and \_\_\_\_\_
- \*This problem, nobody finished \_\_\_\_\_ and the extra credit.
- \*This problem, nobody finished \_\_\_\_\_ and started the extra credit.
- \*This problem, nobody started the extra credit and finished \_\_\_\_\_

- But notice:

This problem, everybody started \_\_\_\_\_ and nobody finished \_\_\_\_\_

# The Coordinate Structure Constraint

- In a coordinate structure,
  - no conjunct can be a gap (conjunct constraint),  
and
  - no gap can be contained in a conjunct if its filler is outside of that conjunct (element constraint)
- .....unless each conjunct has a gap that is paired with the same filler (across-the-board exception)

# These observations cry out for explanation

- In our analysis, the conjunct constraint is an immediate consequence: individual conjuncts are not on the ARG-ST list of any word, so they can't be put on the GAP list
- The element constraint and ATB exception suggest that GAP is one of those features (along with VAL and FORM) that must agree across conjuncts.
- Note: There is no ATB exception to the conjunct constraint.  
*\*This problem, you can compare only \_\_\_\_\_ and \_\_\_\_\_.*

# Our Coordination Rule, so far

$$\begin{array}{l} \left[ \begin{array}{ll} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_0 \end{array} \right] \end{array} \rightarrow \begin{array}{l} \left[ \begin{array}{ll} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_1 \end{array} \right] \dots \left[ \begin{array}{ll} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_{n-1} \end{array} \right] \left[ \begin{array}{ll} \text{HEAD} & conj \\ \text{IND} & s_0 \\ \text{RESTR} & \langle \left[ \text{ARGS} \langle s_1 \dots s_n \rangle \right] \rangle \end{array} \right] \left[ \begin{array}{ll} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_n \end{array} \right] \end{array}$$

- Recall that we have tinkered with what must agree across conjuncts at various times.
- Now we'll add GAP to the things that conjuncts must share

# Our Final Coordination Rule

$$\begin{array}{l}
 \left[ \begin{array}{ll} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_0 \end{array} \right] \rightarrow \\
 \\
 \left[ \begin{array}{ll} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_1 \end{array} \right] \dots \left[ \begin{array}{ll} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_{n-1} \end{array} \right] \left[ \begin{array}{ll} \text{HEAD} & conj \\ \text{IND} & s_0 \\ \text{RESTR} & \langle \left[ \text{ARGS} \langle s_1 \dots s_n \rangle \right] \rangle \end{array} \right] \left[ \begin{array}{ll} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{A} \\ \text{IND} & s_n \end{array} \right]
 \end{array}$$

- We've just added GAP to all the conjuncts and the mother.
- This makes the conjuncts all have the same gap (if any)
- Why do we need it on the mother?



# Closing Remarks on LDDs

- This is a huge topic; we've only scratched the surface
  - There are many more kinds of LDDs, which would require additional grammar rules
  - There are also more island constraints, which also need to be explained
- Our account of the coordinate structure constraint (based on ideas of Gazdar) is a step in the right direction, but it would be nice to explain why certain features must agree across conjuncts.

# Overview

- Some examples of the phenomenon
- What is new and different about it
- Brief sketch of the TG approach
- Broad outlines of our approach
- Details of our approach
- Subject extraction
- Coordinate Structure Constraint

# Reading Questions

```
erb@aditi:~$ cd tmp/erg/
erb@aditi:~/tmp/erg$ grep aj_pp-vp_i-tgh_le lexicon.tdl
all_right_a3 := aj_pp-vp_i-tgh_le &
available_a3 := aj_pp-vp_i-tgh_le &
dangerous_a2 := aj_pp-vp_i-tgh_le &
difficult_a3 := aj_pp-vp_i-tgh_le &
easier_a3 := aj_pp-vp_i-tgh_le &
easiest_a3 := aj_pp-vp_i-tgh_le &
easy_a3 := aj_pp-vp_i-tgh_le &
enjoyable_a2 := aj_pp-vp_i-tgh_le &
entertaining_a2 := aj_pp-vp_i-tgh_le &
feasible_a3 := aj_pp-vp_i-tgh_le &
fine_a2 := aj_pp-vp_i-tgh_le &
good_a3 := aj_pp-vp_i-tgh_le &
handy_a2 := aj_pp-vp_i-tgh_le &
hard_a3 := aj_pp-vp_i-tgh_le &
harder_a3 := aj_pp-vp_i-tgh_le &
hazardous_a2 := aj_pp-vp_i-tgh_le &
important_a4 := aj_pp-vp_i-tgh_le &
impossible_a4 := aj_pp-vp_i-tgh_le &
interesting_a4 := aj_pp-vp_i-tgh_le &
left_a3 := aj_pp-vp_i-tgh_le &
liberating_a3 := aj_pp-vp_i-tgh_le &
[safe_a3 := aj_pp-vp_i-tgh_le &
[sensible_a3 := aj_pp-vp_i-tgh_le &
[simple_a3 := aj_pp-vp_i-tgh_le &
[tedious_a3 := aj_pp-vp_i-tgh_le &
[tough_a3 := aj_pp-vp_i-tgh_le &
[wonderful_a4 := aj_pp-vp_i-tgh_le &
```

# Reading Questions

- Does the grammar fragment license Yoda speak?
  - You can rely on this.
  - This, you can rely on.
  - On this, you can rely.
  - ?Rely on this, you can. (@ Yoda)
  - \*Can rely on this, you. (Lol ignore this)

# Reading Questions

- Why does the ARP have to be defined in terms of list subtraction, rather than something like [ ARG-ST A (+) B (+) C ]?

# The Revised ARP

$$\text{word:} \left[ \begin{array}{l} \text{SYN} \\ \text{ARG-ST} \end{array} \left[ \begin{array}{l} \text{VAL} \\ \text{GAP} \end{array} \left[ \begin{array}{l} \text{SPR} \\ \text{COMPS} \end{array} \left[ \begin{array}{l} \boxed{A} \\ \boxed{B} \ominus \boxed{C} \end{array} \right] \right] \right] \right] \oplus \left[ \begin{array}{l} \boxed{A} \\ \boxed{B} \end{array} \right]$$

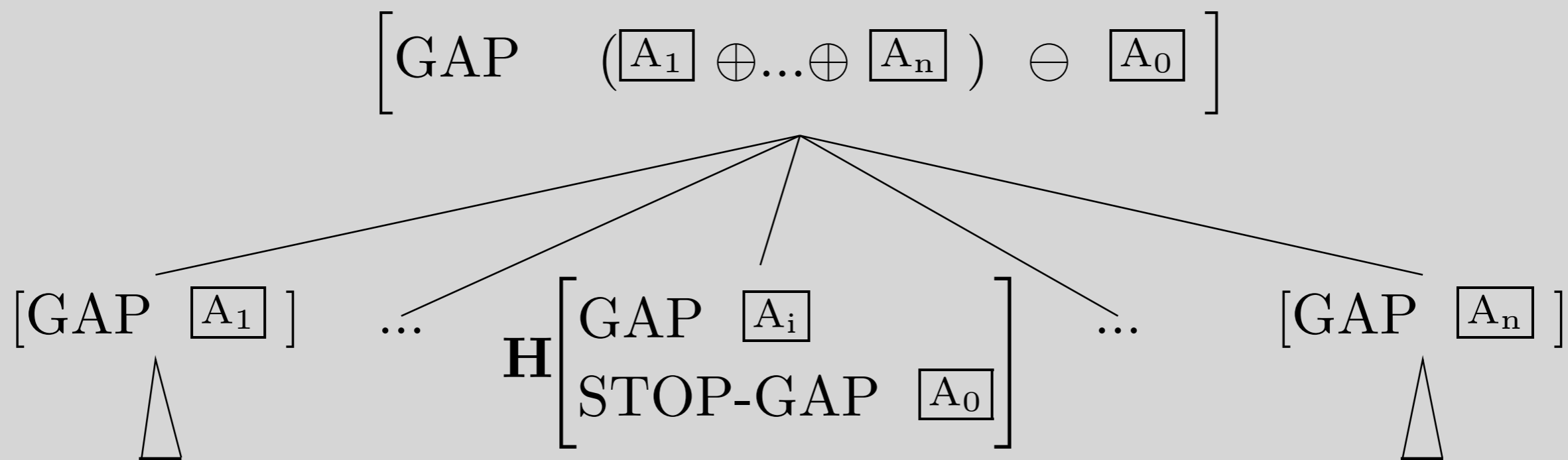
- $\ominus$  is a kind of list subtraction, but:
  - it's not always defined, and
  - when defined, it's not always unique
- The ARP now says the non-SPR arguments are distributed between COMPS and GAP.

# Reading Questions

- I'm confused about why the GAP and STOP-GAP would have the same values within the same structure
- I don't really understand why the second daughter in the Head Filler Rule would need both GAP and STOP-GAP on it. Wouldn't they always be identical? Wouldn't the GAP list imply what the STOP-GAP list is supposed to be?

# The GAP Principle

A local subtree  $\Phi$  satisfies the GAP Principle with respect to a headed rule  $\rho$  if and only if  $\Phi$  satisfies:



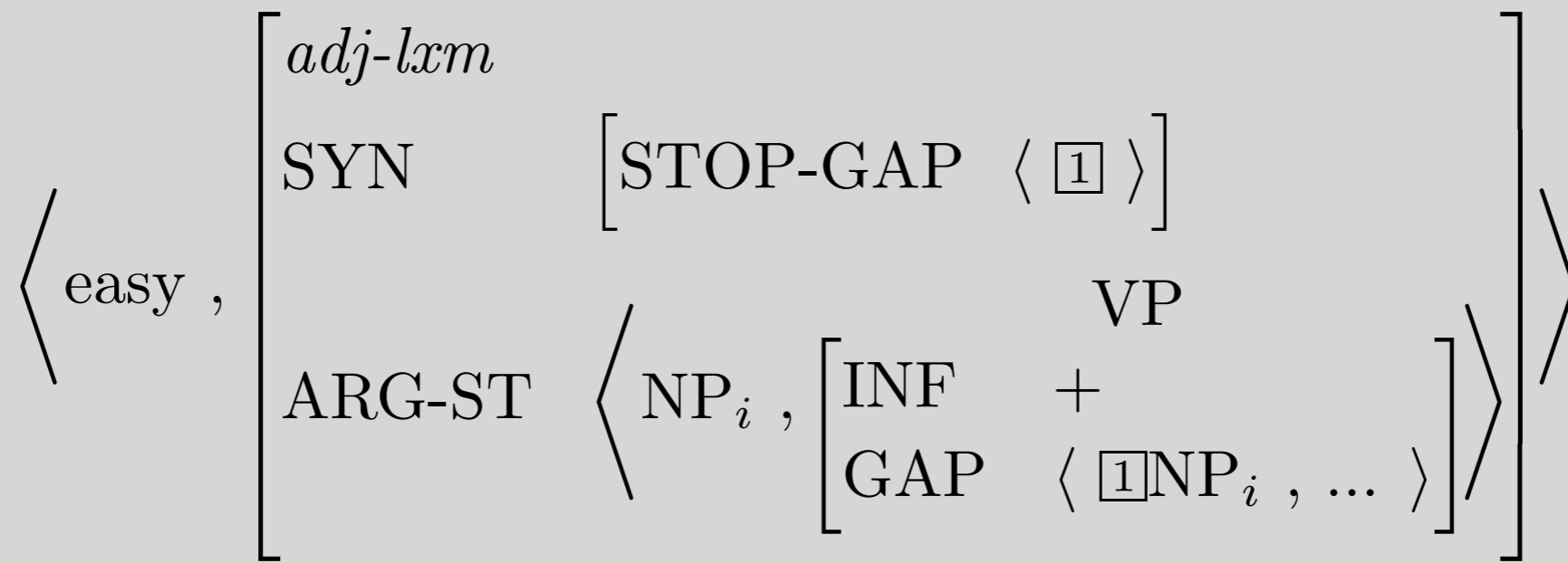


# The Head-Filler Rule

$$[phrase] \rightarrow \boxed{1} \left[ \text{GAP} \quad \langle \rangle \right] \mathbf{H} \left[ \begin{array}{l} \text{HEAD} \quad \left[ \begin{array}{l} \textit{verb} \\ \text{FORM} \quad \textit{fin} \end{array} \right] \\ \text{VAL} \quad \left[ \begin{array}{l} \text{SPR} \quad \langle \rangle \\ \text{COMPS} \quad \langle \rangle \end{array} \right] \\ \text{STOP-GAP} \quad \langle \boxed{1} \rangle \\ \text{GAP} \quad \langle \boxed{1} \rangle \end{array} \right]$$

- This only covers gap filling in finite Ss
- The filler has to be identical to the GAP value
- The STOP-GAP value is also identical
- The GAP Principle ensures that the mother's GAP value is the empty list

# Gap Filling with *easy*-Adjectives



- Because STOP-GAP and GAP have the same value, that value will be subtracted from the mother's GAP value.
- The first argument is coindexed with the GAP value, accounting for the interpretation of the subject as the filler.

# Reading Questions

- It's a bit unclear to me how to determine which node to place a non-empty STOP-GAP on. In 35, a non-empty STOP-GAP is on S and on the same node as the GAP it fills. With adjectives like easy, the non-empty STOP-GAP is on the lexical entry for easy, and in 38 the word structure for easy has an empty GAP. The text says, "where a gap is associated with its filler, the feature [STOP-GAP] has a non-empty list as its value", but why is GAP empty then for easy? How is the association between gap and filler occurring?
- In which cases is the non-empty STOP-GAP on the same node as the gap it fills and in which cases is it not? And why?

# Reading Questions

- Is it possible to have ambiguities regarding which element on the GAP list is filled? And if so, how do we resolve those ambiguities? For example, if we have two NP's on the GAP list and a STOP-GAP list with NP on it, is there anything in our formalism that indicates which NP on the GAP list is filled?
- I want to ask whether STOP-GAP value purely comes from lexical entry. Do adjectives like easy and hard always have non-empty STOP-GAP values?

# Reading Questions

- Other island constraints are mentioned in passing near the end of the chapter. Could you give us some examples of constraints not examined in the text?
- What are some other examples of long-distance dependencies beyond the four mentioned in the chapter?
- Can a ‘gap’ ever precede a ‘filler’, or must it always appear somewhere later in an utterance?

# Reading Questions

- From (55) a, it seems acceptable to also say "*his is the dancer that we bought a portrait and two photos of*. omitting the first preposition *of*. How does the grammar deal with this construction? Could we say *portrait* has the same GAP value as *of*?"

# Reading Questions

- In general, when is it appropriate to modify a preexisting principle or constraint (such as the ARP), and when is it preferable to formalize a new lexical rule (such as the Subject Extraction LR)?

# Reading Questions

- Are GAP and STOP-GAP features necessary in languages with markers that have more flexible word ordering? E.g. Japanese -wa or Korean -eun/-neun?
- Page 431 mentions that there are languages (for example several Slavic and Scandinavian languages) that allow multiple gaps more freely. What does this look like? Do gaps serve different purposes in different languages or are they similar to English?