

Ling 566
Dec 1, 2020
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.

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 ____.

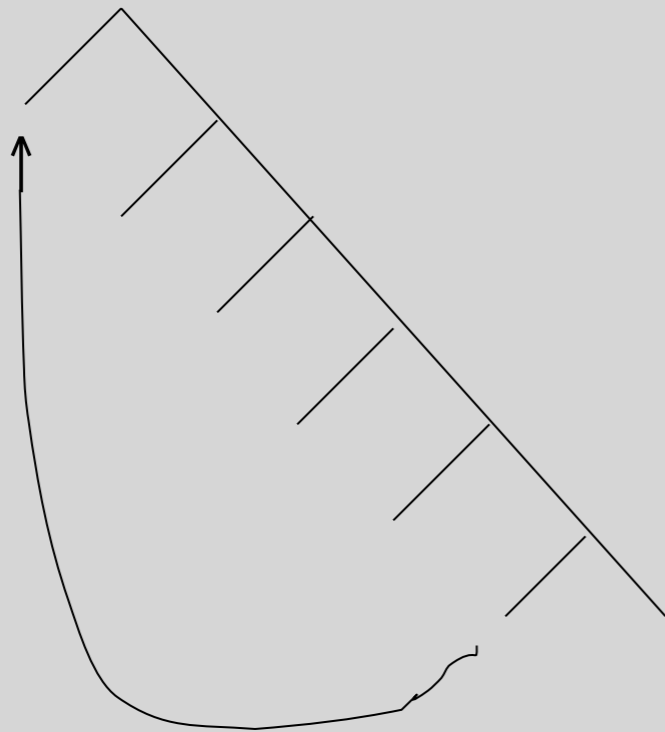
**To Pat, you can rely ____.*

LDDs in TG

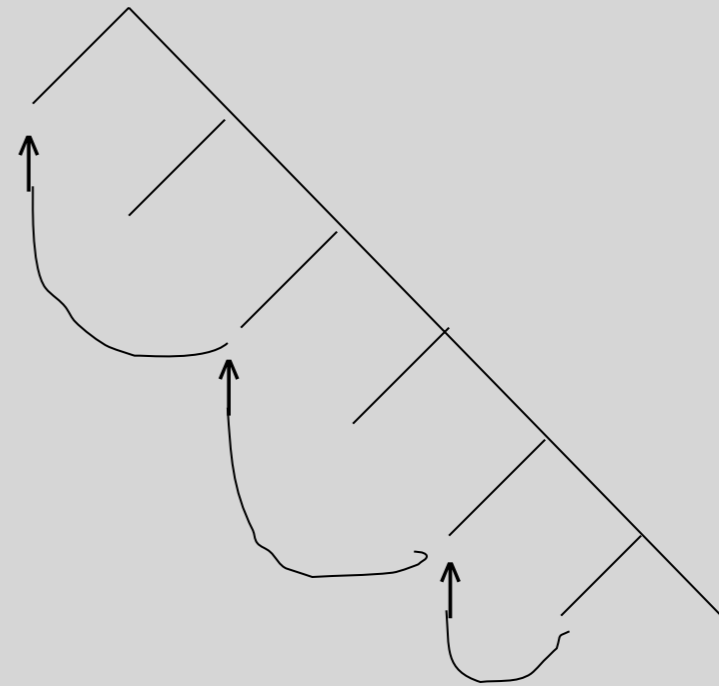
- These were long thought to constitute the strongest evidence for transformations.
- They were handled in TG by moving the filler from the gap position.
- Case, agreement, preposition selection could apply before movement.

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

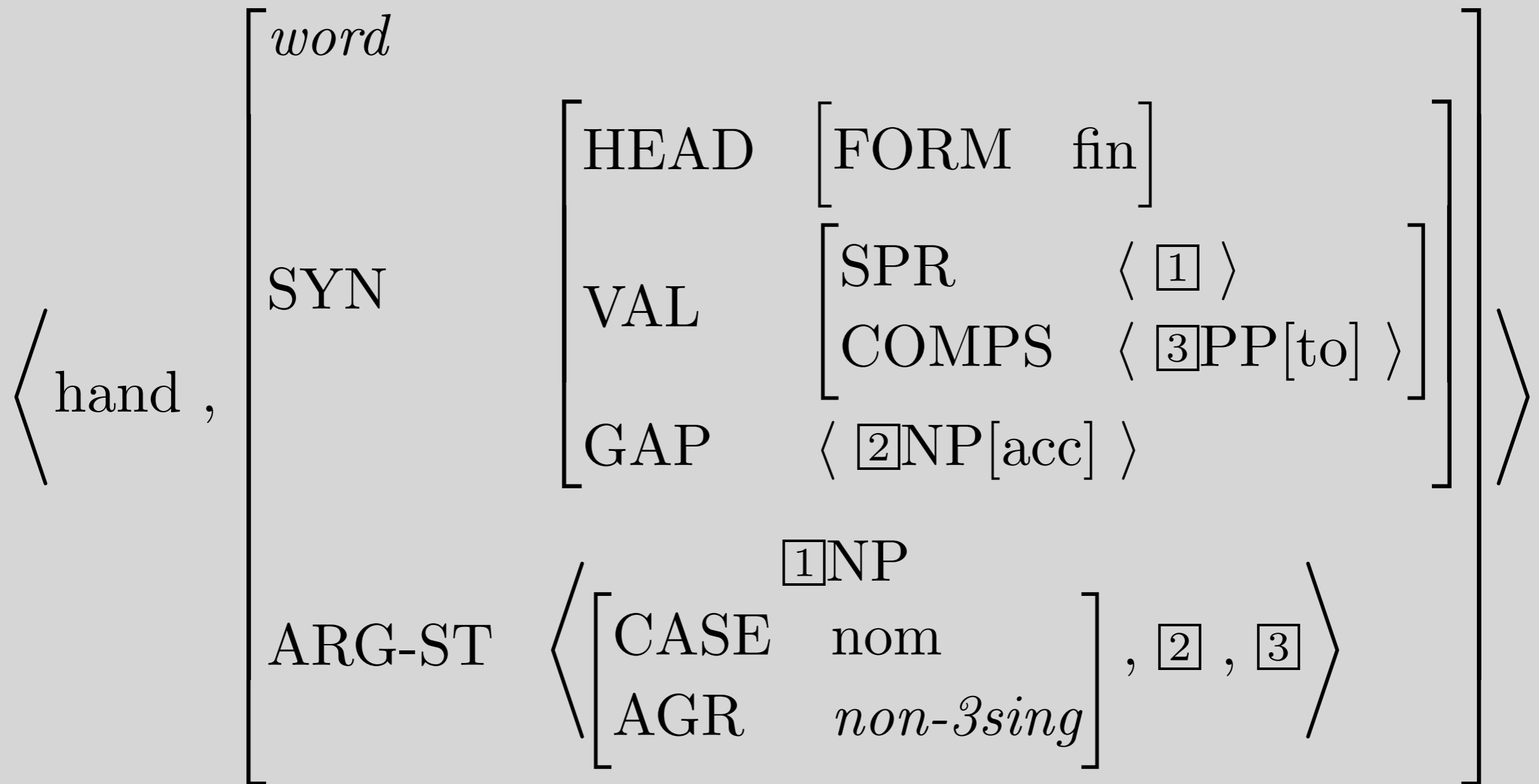
- Like valence features and ARG-ST, GAP's value is a list of feature structures (often empty).
- Subject gaps are introduced by a lexical rule.
- Non-subject gaps are introduced by revising the Argument Realization Principle.

The Revised ARP

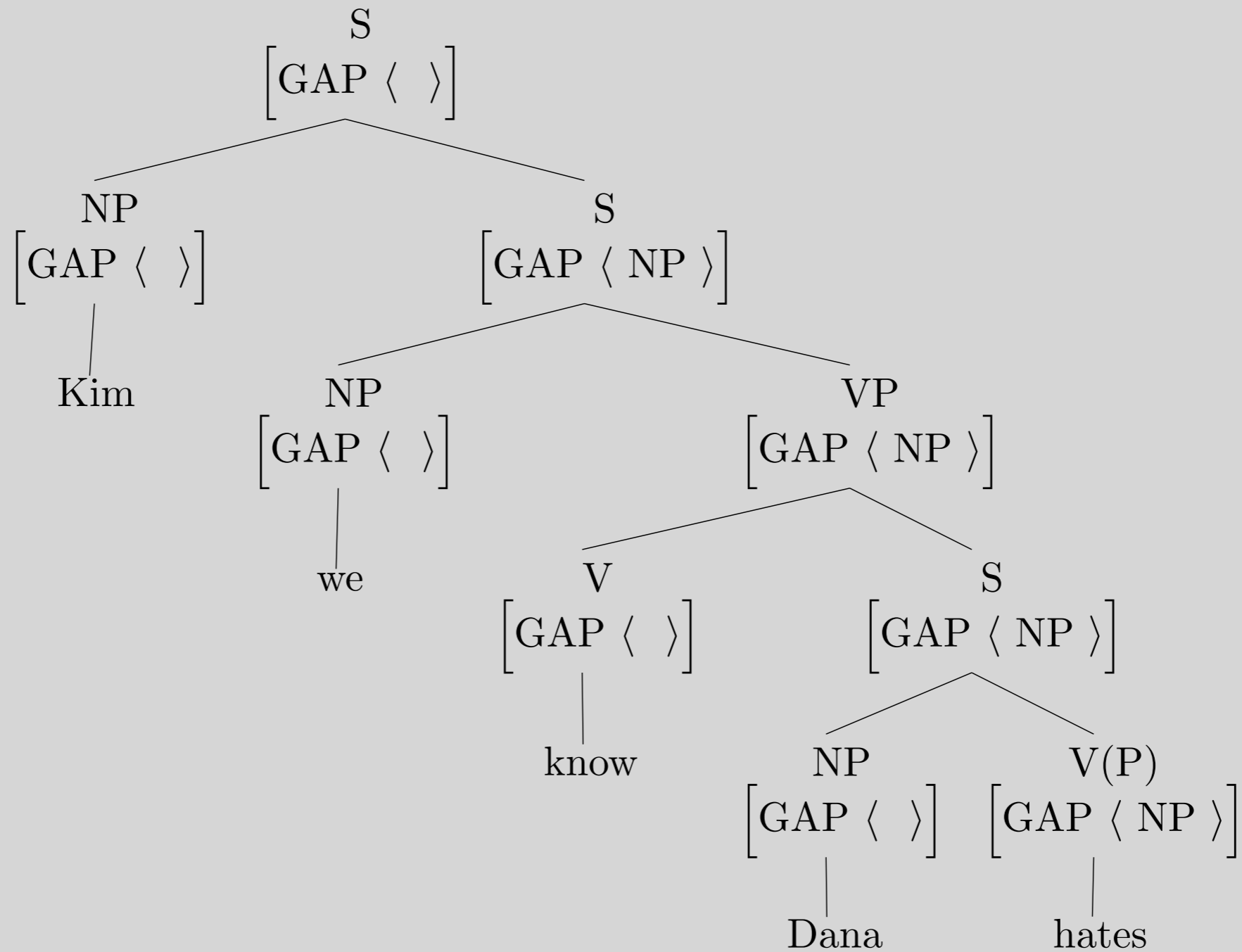
$$\text{word:} \left[\begin{array}{l} \text{SYN} \\ \text{ARG-ST} \end{array} \left[\begin{array}{l} \text{VAL} \\ \text{GAP} \end{array} \left[\begin{array}{l} \text{SPR} \\ \text{COMPS} \end{array} \left[\begin{array}{l} \boxed{A} \\ \boxed{B} \end{array} \ominus \boxed{C} \right] \right] \oplus \boxed{B}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

- \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

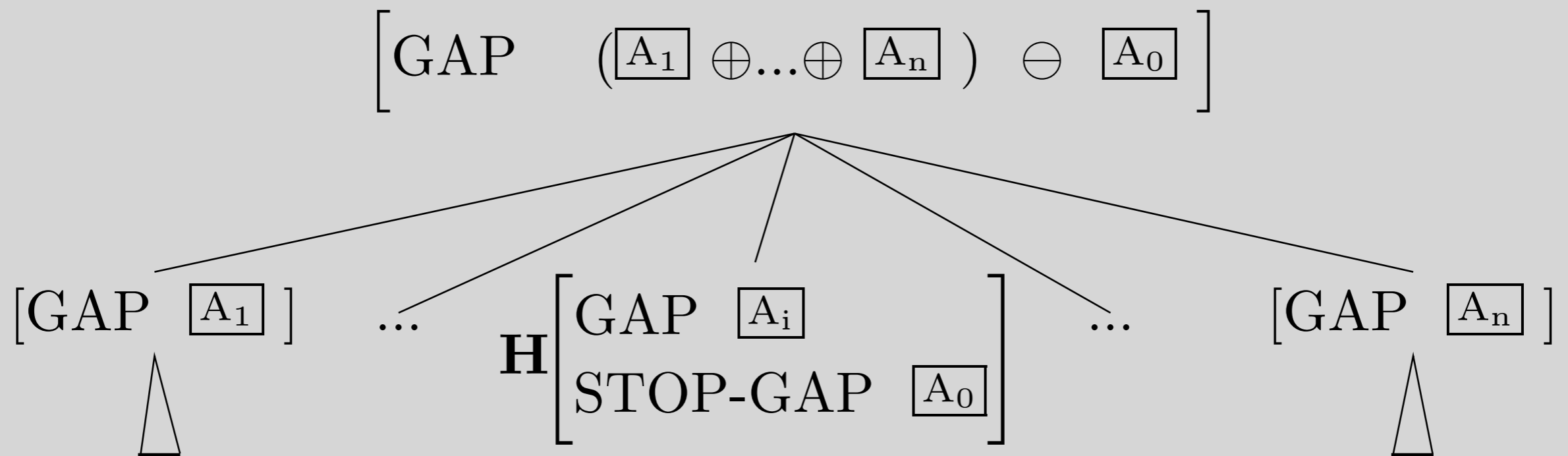
- Pass any GAP values from daughters up to their mothers,
- **except** when the filler is found.
- For topicalization, we can write the exception into the grammar rule, but
- For *easy*-adjectives, the NP that corresponds to the gap is the subject, which is introduced by the Head-Specifier Rule.
- Since 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 Φ satisfies the GAP Principle with respect to a headed rule ρ if and only if Φ satisfies:



How does STOP-GAP work?

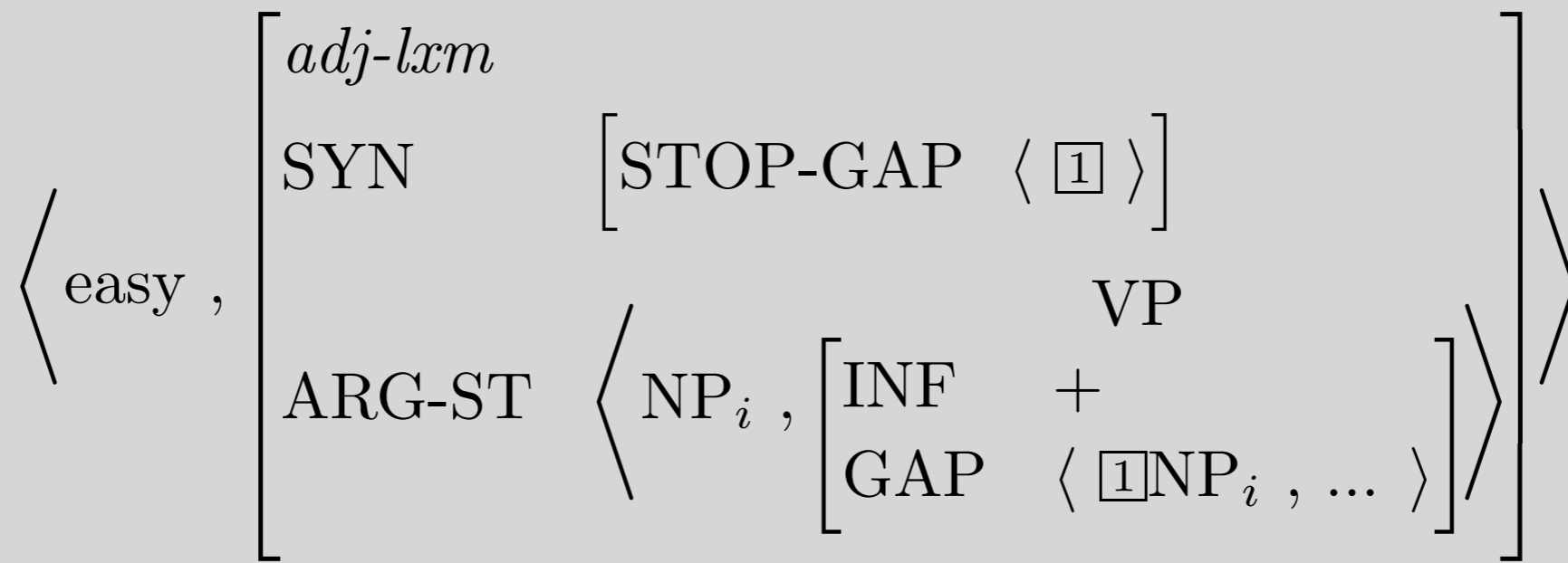
- STOP-GAP is empty almost everywhere
- When a gap is filled, STOP-GAP is nonempty, and its value is the same as the gap being filled.
- This blocks propagation of that GAP value, so gaps are only filled once.
- The nonempty STOP-GAP values come from two sources:
 - a stipulation in the Head-Filler Rule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

The Head-Filler Rule

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

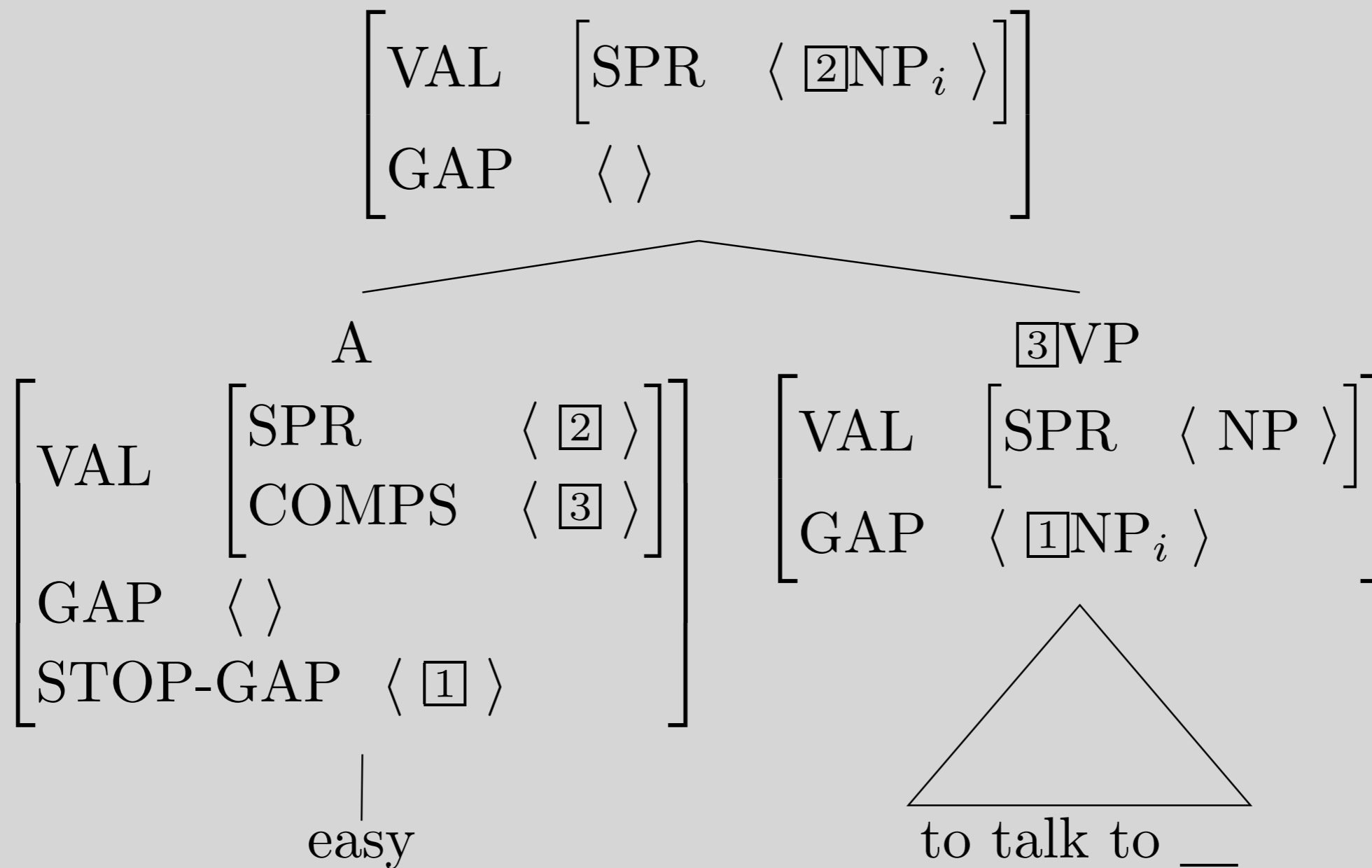
- This only covers gap filling in finite Ss
- The filler has to be identical to the GAP value
- The STOP-GAP value is also identical
- The GAP Principle ensures that the mother's GAP value is the empty list

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*_____



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

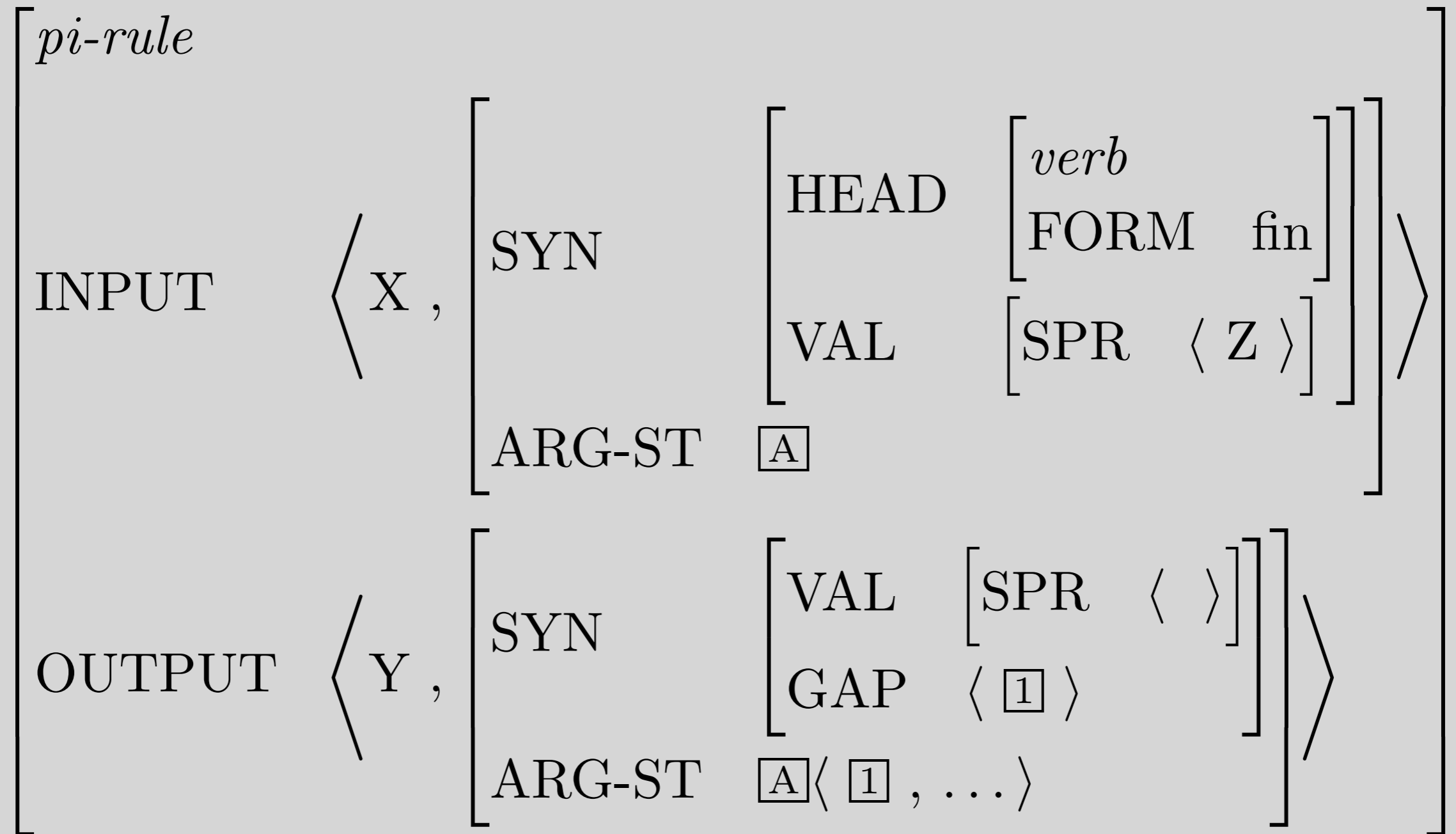
On to New Material....

- Sentences with subject gaps
- Gaps in coordinate constructions

Subject Gaps

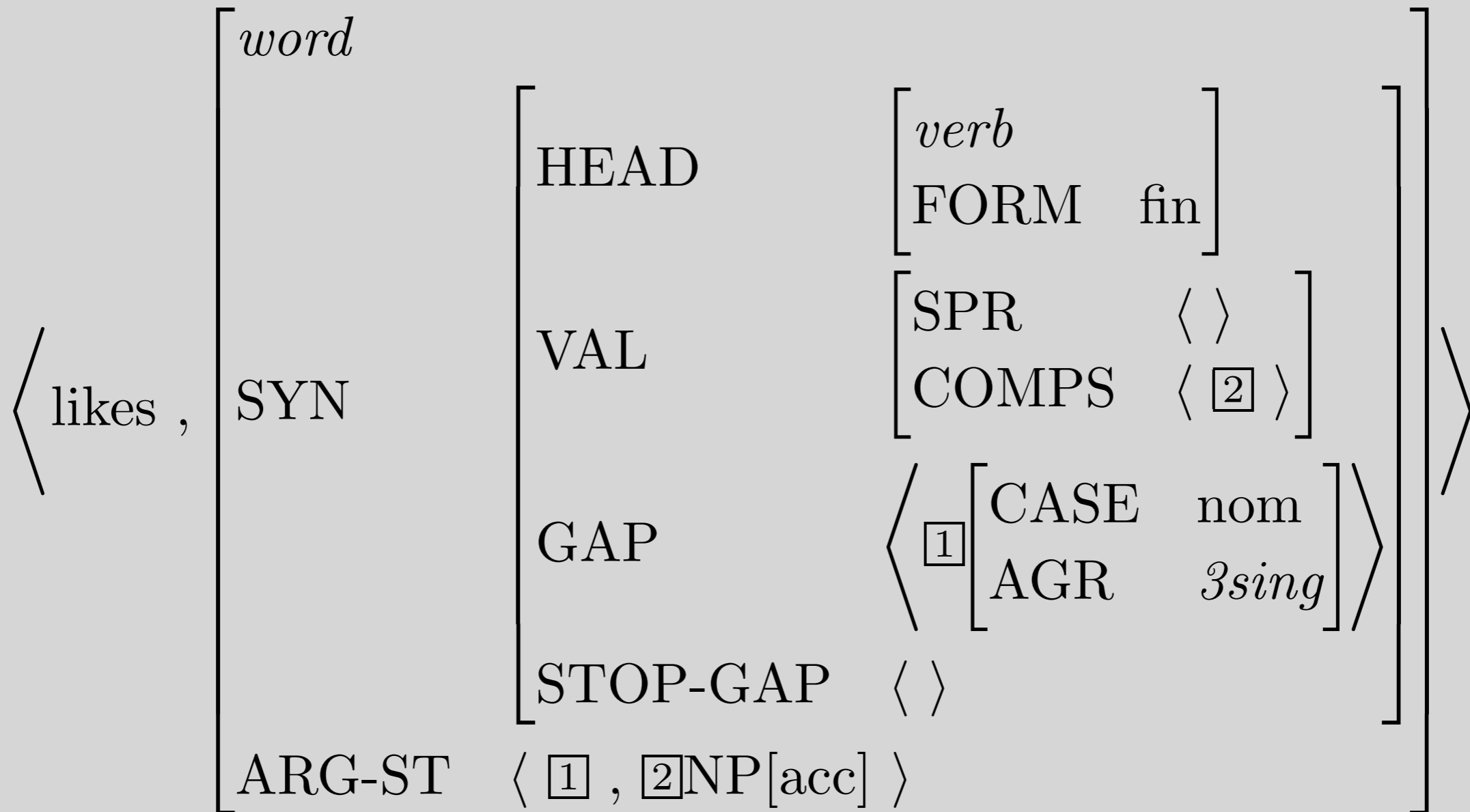
- The ARP revision only allowed missing complements.
- But gaps occur in subject position, too:
This problem, everyone thought ____ was too easy.
- We handle these via a lexical rule that, in effect, moves the contents of the SPR list into the GAP list

The Subject Extraction Lexical Rule



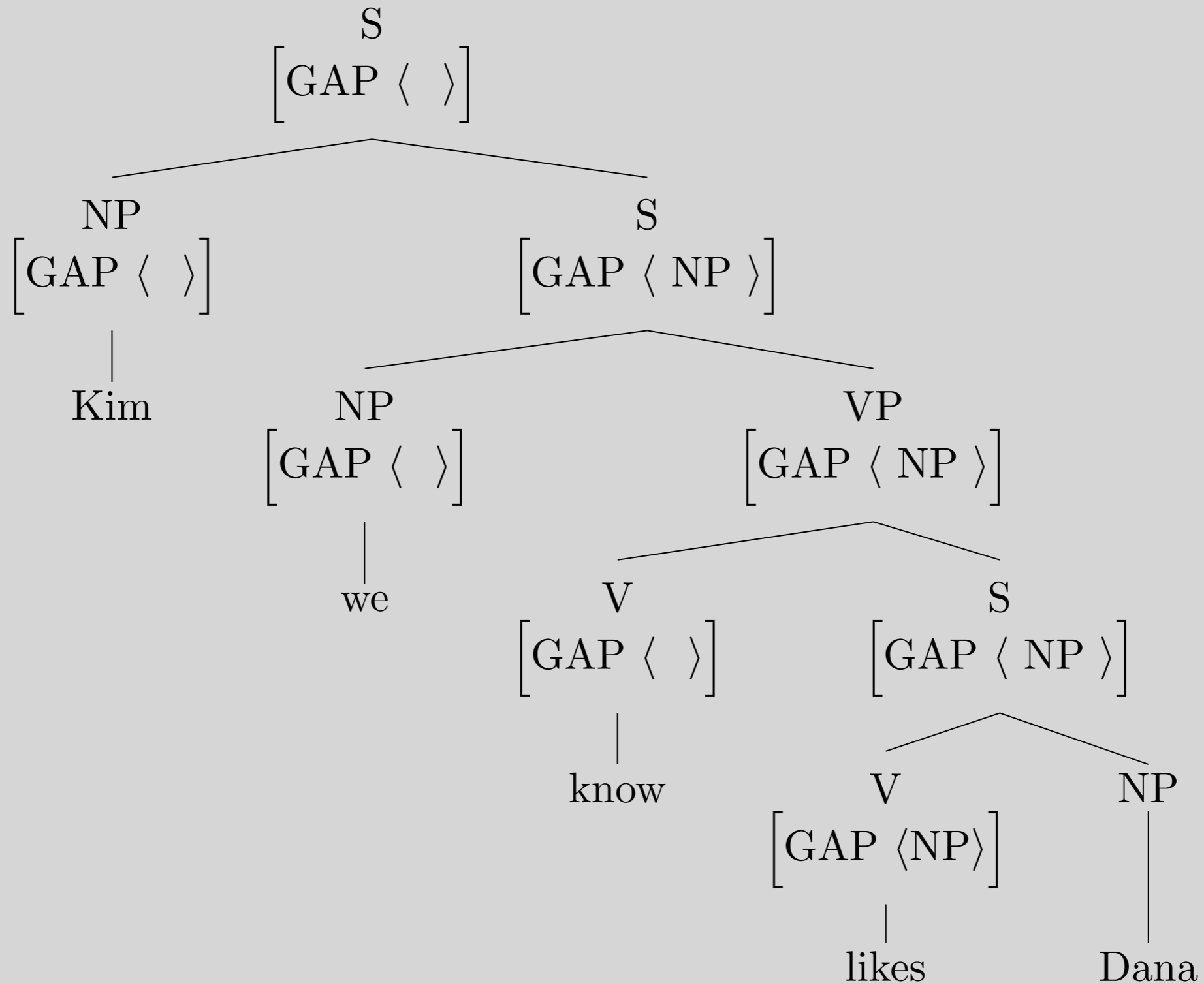
- NB: This says nothing about the phonology, because the default for *pi-rules* is to leave the phonology unchanged.

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

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

- 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

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

- 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

- Is the concept of gaps in LDD related to the concept of gaps in ellipsis? If so, how?
- The abbreviation for *hates* in the example *Kim, we know Dana hates* is V(P). Is this just to point out that it has an empty COMPS list, but has a non-empty GAP list? Will this be used formally when a verb has an empty COMPS list and non-empty GAP list?
- Why is GAP a feature on syn-cat and placed where it is? Why can it not be inside the VAL feature, along with SPR and COMPS? Similarly, why can it not be outside SYN, along with ARG-ST?

Reading Questions

- On learning what \ominus means ($A \ominus B = A$ without the elements of B), I now wonder what the difference is between $\langle [A] \rangle \oplus \langle [B] \rangle$ and $\langle [A], [B] \rangle$ when defining the ARG-ST of a phrase or sentence.
- When there are multiple possible values for $A \ominus B$, does that introduce ambiguity? Is that ambiguity desired? Thinking of like... "My son, I gave the world" & "the world, I gave my son"? These could both have the meaning "I gave the world to my son" and "I gave my son to the world" and so it's ambiguous which of the NP complements is in GAP?

Reading Questions

- In the new ARP we only have A+B on the ARG-ST list, but then all of our lexical sequence examples include the GAP element (C) as well. Why is our ARG-ST not A+B+C in the new ARP rule?

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.

Reading Questions

- My first thought is on the ambiguity of GAP subtracting from COMPS, on p. 434 we have an example of $A = \langle \text{NP}, \text{PP}, \text{NP} \rangle$ and $B = \langle \text{NP} \rangle$ so when $A - B = C$, C is ambiguously either $\langle \text{NP}, \text{PP} \rangle$ or $\langle \text{PP}, \text{NP} \rangle$. Will we be introduced to anything that would constrain the subtraction operation to either the first NP (first on A), or the second NP (third on A). Also would this be a valid subtraction $\langle \text{NP}, \text{PP}, \text{NP} \rangle - \langle \text{NP}, \text{NP} \rangle = \langle \text{PP} \rangle$ (despite the fact NPs aren't next to each other on the first list but are on the second)?

Reading Questions

- Why does the GAP Principle only apply to headed rules?
- Is STOP-GAP generally only found on a head daughter node?
- STOP-GAP is a little confusing to me. It seems like whenever we need to empty out a GAP, we just have a STOP-GAP. How do we know when to do this? Is it when we're on the node that is a sibling of the GAP value(s)?

Reading Questions

- It's interesting to me that the Head-Filler Rule specifies that the head daughter (an S) has a non-empty STOP-GAP value with one element, which matches the one element on its GAP list. Why is this the case? The only other alternative I can think of would be a lexical rule that modifies the STOP-GAP value of the element of the filler. That seems unintuitive, but would it work?
- Besides the adjective exceptions, I am still unsure of the general use of the feature STOP-GAP and its purpose. I thought GAP already signaled what gap there is to be filled?
- Further, are there any examples in which STOP-GAP != GAP?

Reading Questions

- Could we go over the motivation for STOP-GAP again? I'm confused about what the difference between the gappy (32a) and not gappy (32b) APs are (and more generally how we can identify what is gappy or not).
- I'm confused about the notation used in the GAP principle. How can the subtraction in the GAP of the parent node be defined if A_0 is not in the range $A_1 \dots A_n$?

Reading Questions

- What is the commonality of adjectives such as 'easy' and 'hard'? Why do they have non empty STOP-GAP?
- I'm curious what are some other "easy"-like adjectives, and what they have in common. How large is this set?
- Is there a particular way to identify gap-filling adjectives?

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

- I'm so intrigued by the "easy" class of adjectives. They seem like such a strange exception. I see how they're handled in the grammar, but I'm wondering what about them allows them to have this unique property.
- Is the main reason we need STOP-GAPs-rather than just having empty GAP lists once its element has been found in a phrase combination-because of words such as easy/hard?
- Is this kind of adjective common cross-linguistically?

Reading Questions

- Is there an alternate explanation for the “easy” adjectives?
- From what I've gathered, STOP-GAP's purpose is to 'fill' GAPS that are in our sentence. I'm confused however by our decision to define in the lexical entry. Does that imply that uses of easy that don't have a gap are lexically different? And in general does the STOP-GAP absolutely always have to 'fill' in the gaps?

Reading Questions

- Maybe I missed something, but how does the Head-Filler Rule handle multiple gaps? It seems very specific to a single gap being filled to form an empty GAP list.

Reading Questions

- For (43) on pg 441, why is GAP specified as A and not $\langle \rangle$? The text mentions that the mother of the imperative rule must satisfy the initial symbol's GAP $\langle \rangle$ restriction, so A is always $\langle \rangle$. So why are we using A to label the GAP? Is there some scenario where A wouldn't be $\langle \rangle$?

Reading Questions

- Why is the Subject Extraction LR a pi-rule?
- Why can we not modify our existing lexemes and the ARP to account for (47), instead of the Subject Extraction Lexical Rule?

Reading Questions

- A clarifying question on islands, are they the part of the phrase encompassed by the outer brackets?
- Does the grammar account for other island constraints as well or just the Coordinate Structure Constraint?
- Has there been further findings/explanations of syntactic islands, since the publishing of this book?

Reading Questions

- I'm curious about how LDDs are handled in relative clauses. I'm guessing they're not licensed by the Head-Filler Rule since combining a relative clause with its antecedent gives an NP, which isn't of HEAD verb. If that's the case, how should we fill the gap? Are we going to postulate a new rule that is similar to the HFR?

Reading Questions

- How well does this handling of LDDs generalize cross-linguistically? It seems to me that it would generalize quite well, as it really only relies on headedness to work, but would generalize over other differences in language structures.
- On page 441, the book says GAPS are never filled in head-specifier and head modifier phrases. Why is that? Is this constraint specific to English only, or is it also common in other languages?
- It (easy adjectives) also makes me wonder how many examples of such exceptions to broader constraints in HPSG pop up in other languages (not just specifically about LDD).

Reading Questions

- Will our grammar take punctuation in to a count at all or are there any variants of HSPG that do so?
- I am wondering though if places of ambiguity in the ARP w.r.t. the operation "COMPS (-)", can cause practical problems when parsing trees?
- Maybe we could review some of the motivations for not positing empty categories, like traces? I feel like sentence processing might be among them, but are there others?