

Ling 566
Nov 22, 2019
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 ____.

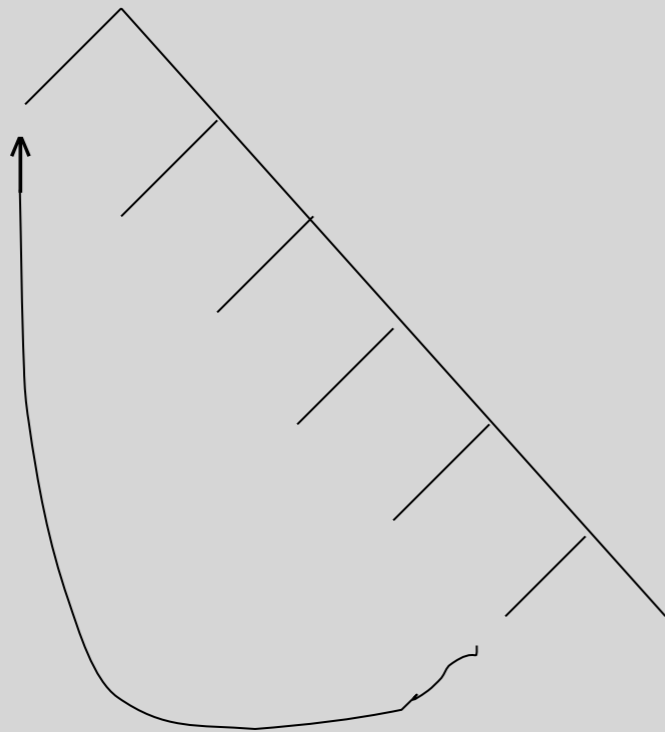
**To Pat, you can rely ____.*

LDDs in TG

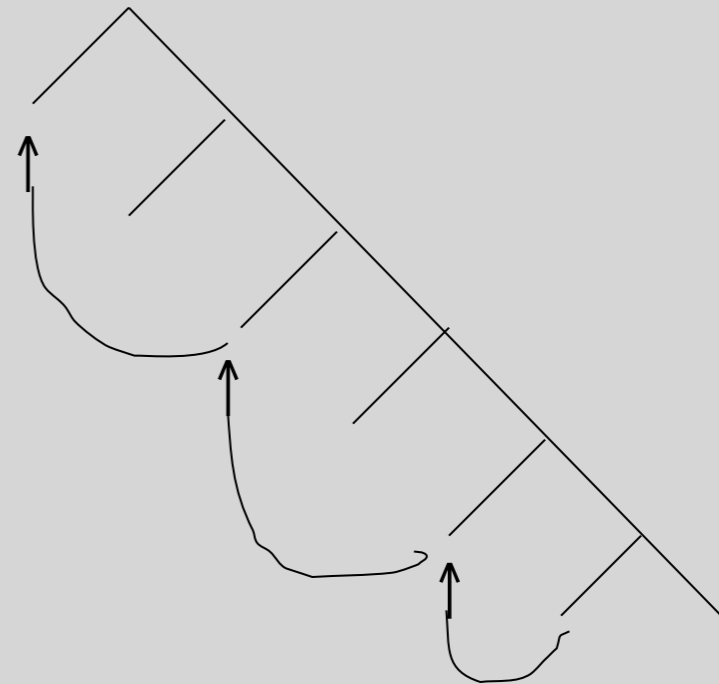
- These were long thought to constitute the strongest evidence for transformations.
- They were handled in TG by moving the filler from the gap position.
- Case, agreement, preposition selection could apply before movement.

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

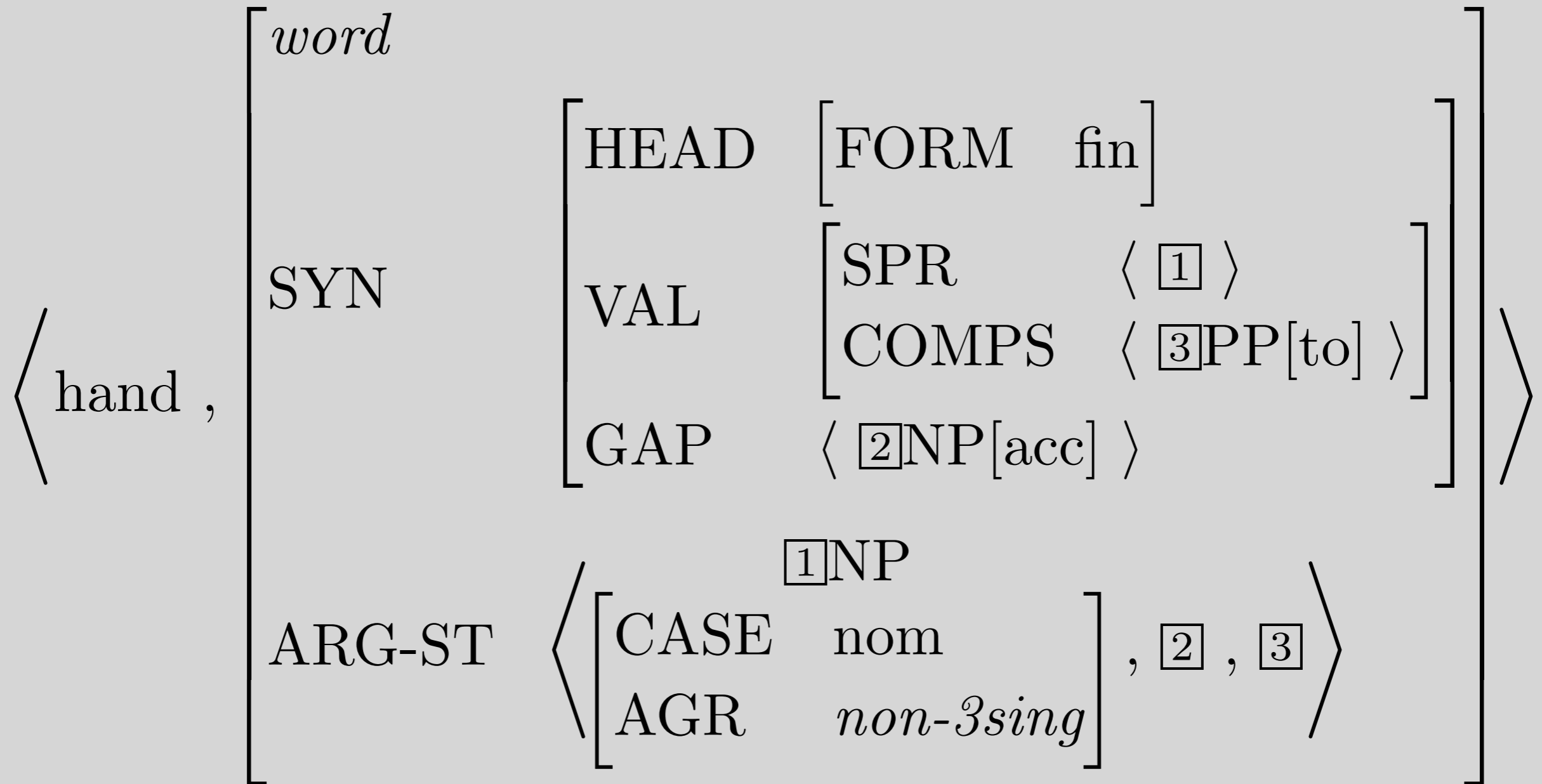
- Like valence features and ARG-ST, GAP's value is a list of feature structures (often empty).
- Subject gaps are introduced by a lexical rule.
- Non-subject gaps are introduced by revising the Argument Realization Principle.

The Revised ARP

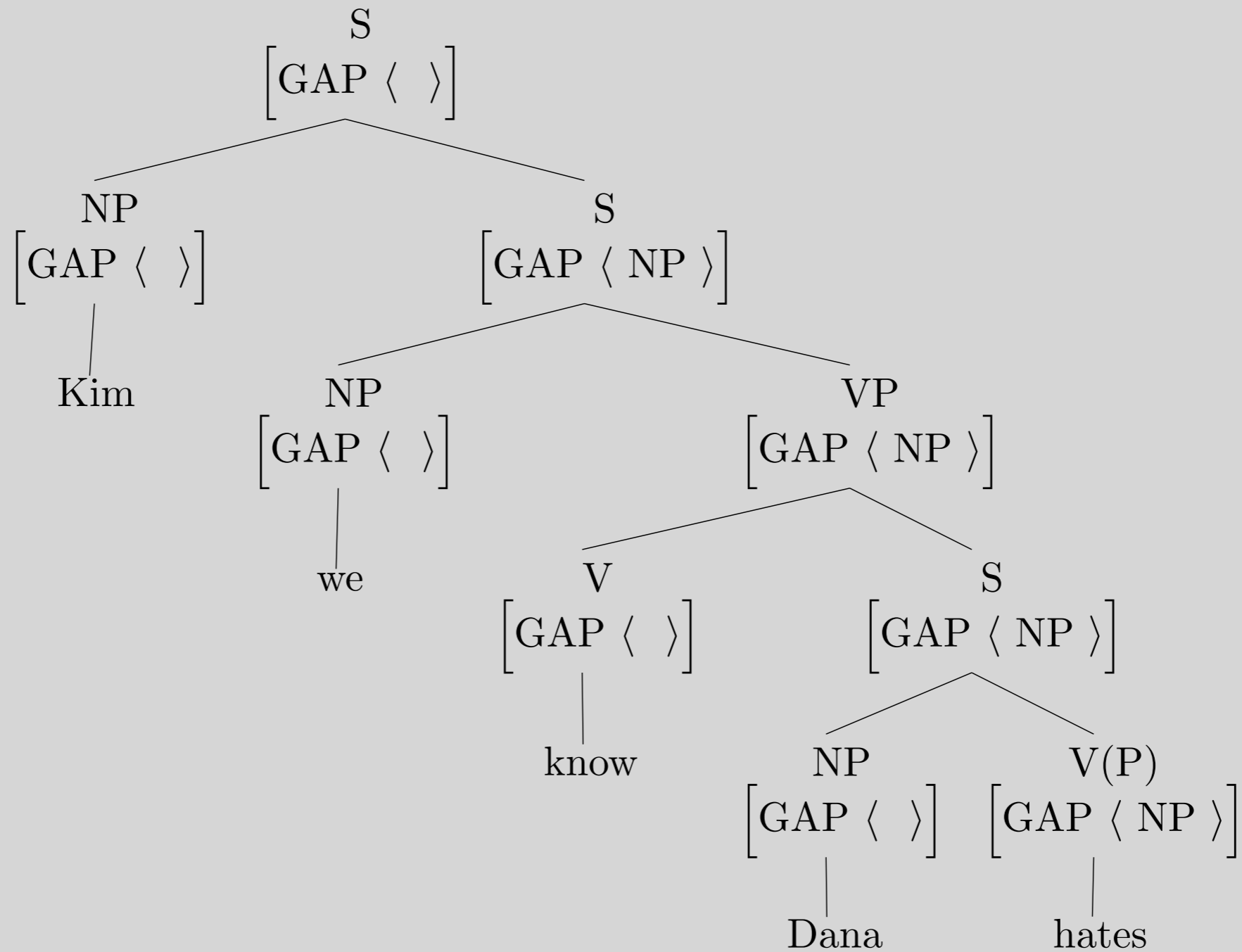
$$\text{word:} \left[\begin{array}{l} \text{SYN} \\ \text{ARG-ST} \end{array} \left[\begin{array}{l} \text{VAL} \\ \text{GAP} \end{array} \left[\begin{array}{l} \text{SPR} \\ \text{COMPS} \end{array} \left[\begin{array}{l} \boxed{A} \\ \boxed{B} \ominus \boxed{C} \end{array} \right] \right] \right] \right]$$

- \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

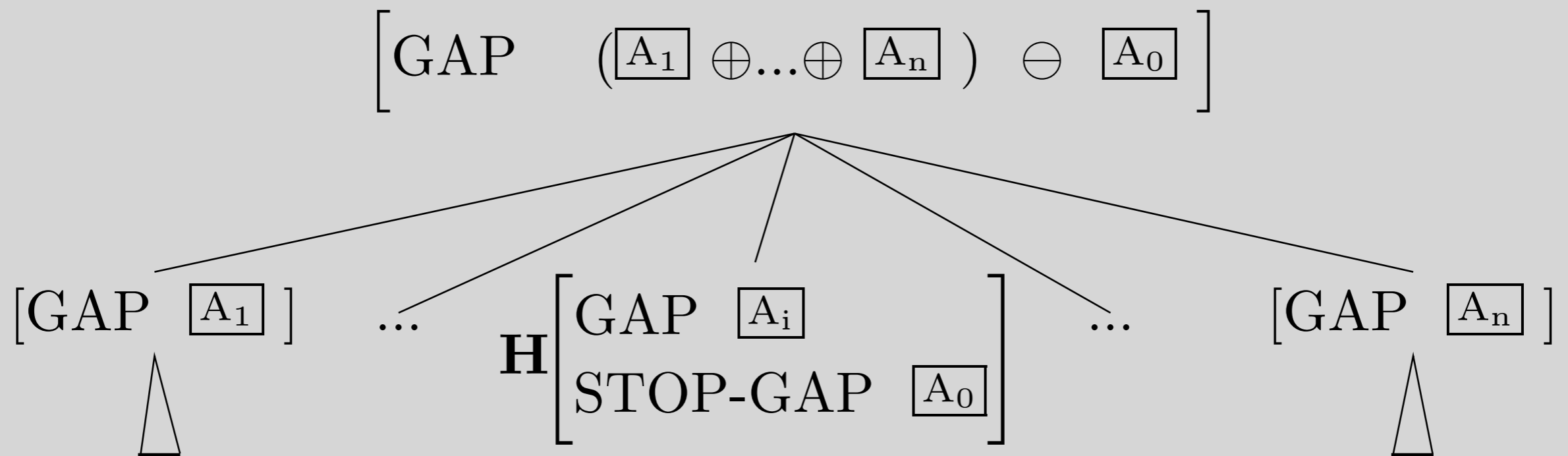
- Pass any GAP values from daughters up to their mothers,
- **except** when the filler is found.
- For topicalization, we can write the exception into the grammar rule, but
- For *easy*-adjectives, the NP that corresponds to the gap is the subject, which is introduced by the Head-Specifier Rule.
- Since 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?

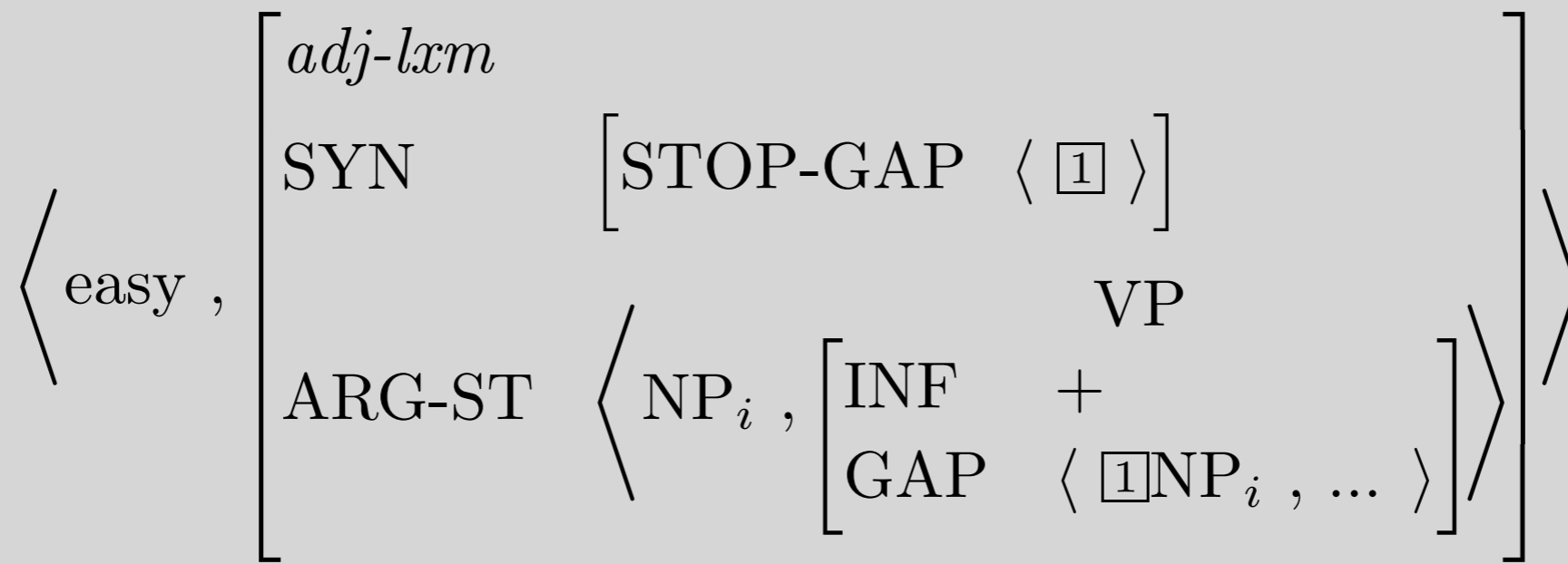
- STOP-GAP is empty almost everywhere
- When a gap is filled, STOP-GAP is nonempty, and its value is the same as the gap being filled.
- This blocks propagation of that GAP value, so gaps are only filled once.
- The nonempty STOP-GAP values come from two sources:
 - a stipulation in the Head-Filler Rule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

The Head-Filler Rule

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

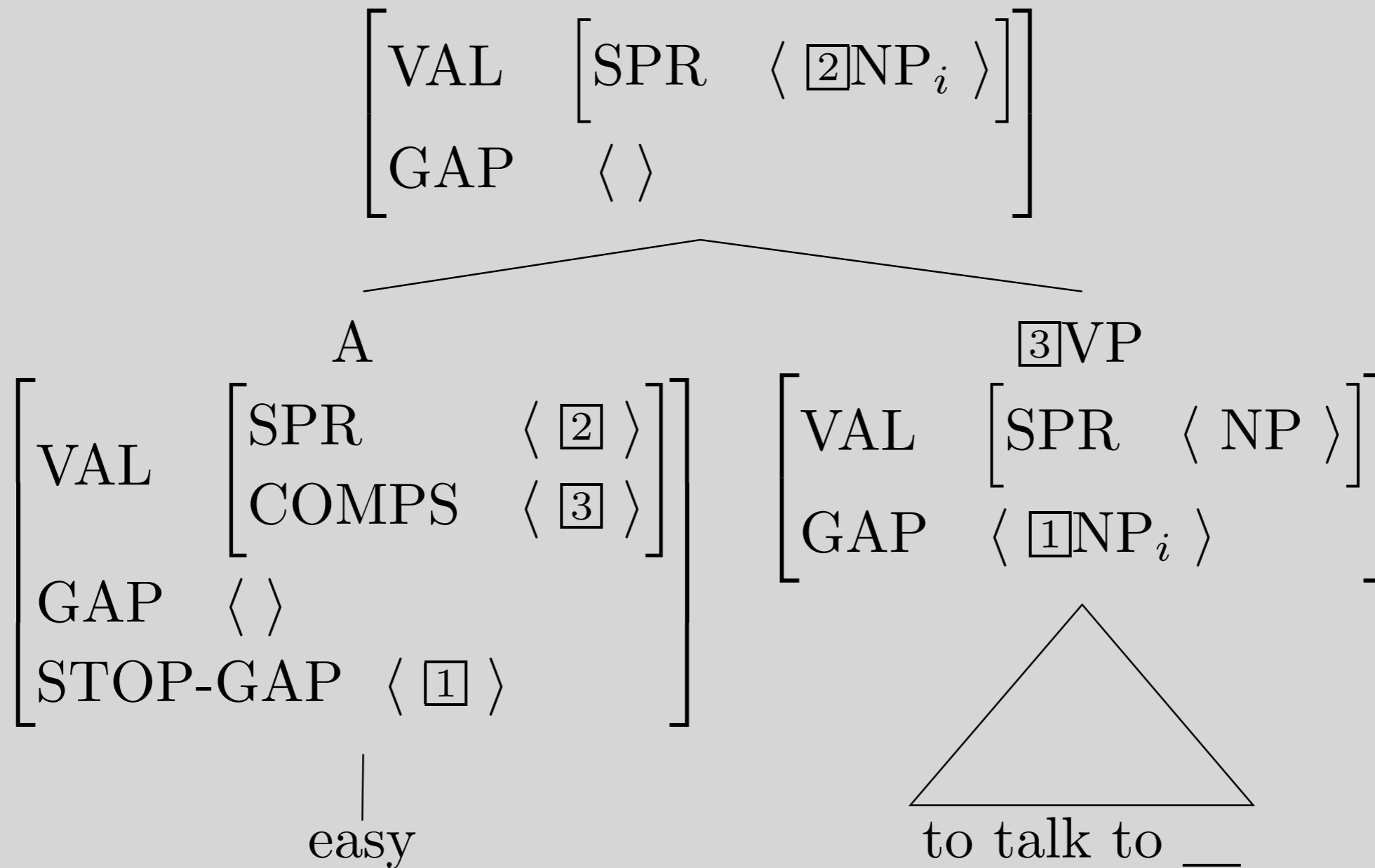
- This only covers gap filling in finite Ss
- The filler has to be identical to the GAP value
- The STOP-GAP value is also identical
- The GAP Principle ensures that the mother's GAP value is the empty list

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

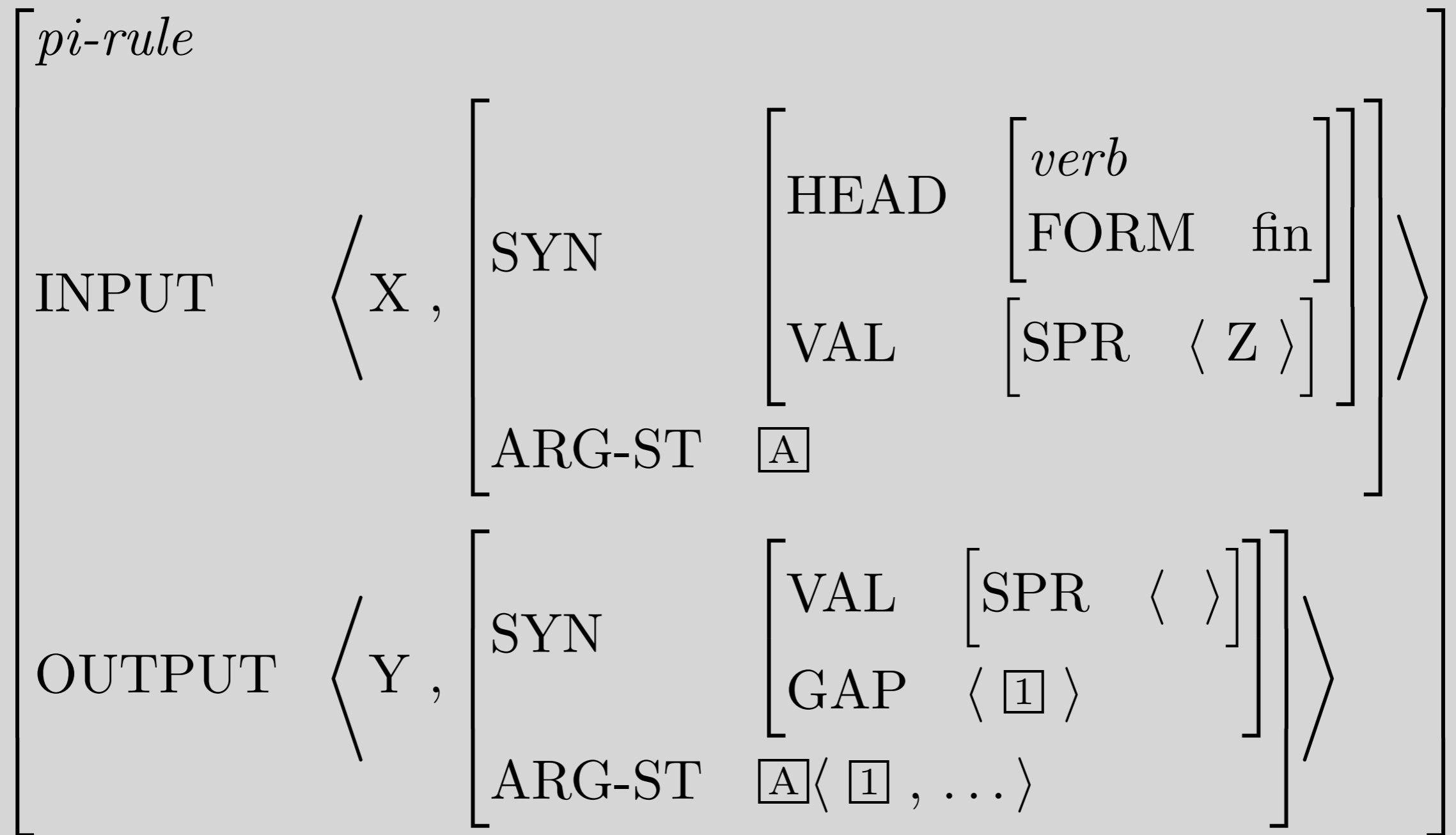
On to New Material....

- Sentences with subject gaps
- Gaps in coordinate constructions

Subject Gaps

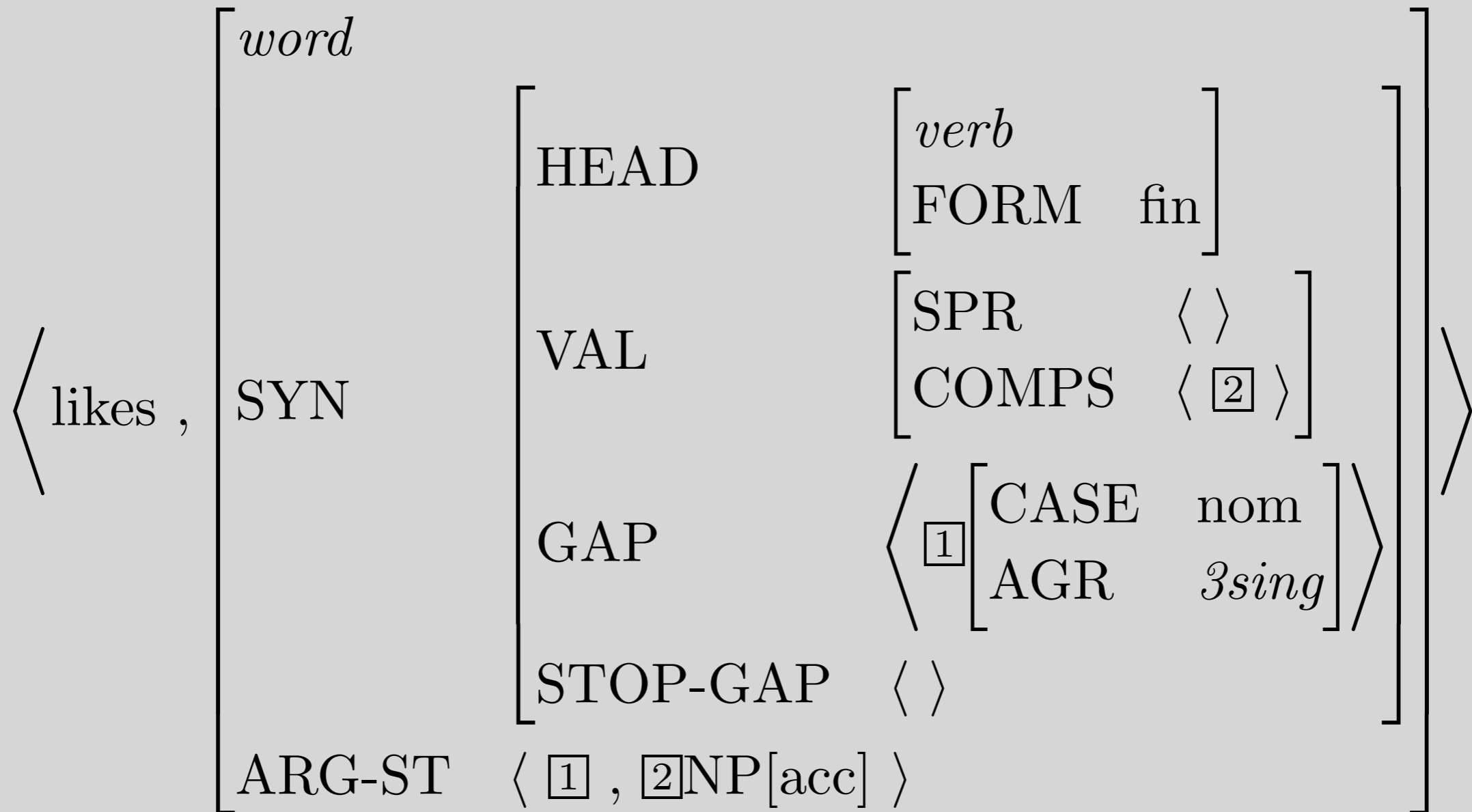
- The ARP revision only allowed missing complements.
- But gaps occur in subject position, too:
This problem, everyone thought ____ was too easy.
- We handle these via a lexical rule that, in effect, moves the contents of the SPR list into the GAP list

The Subject Extraction Lexical Rule



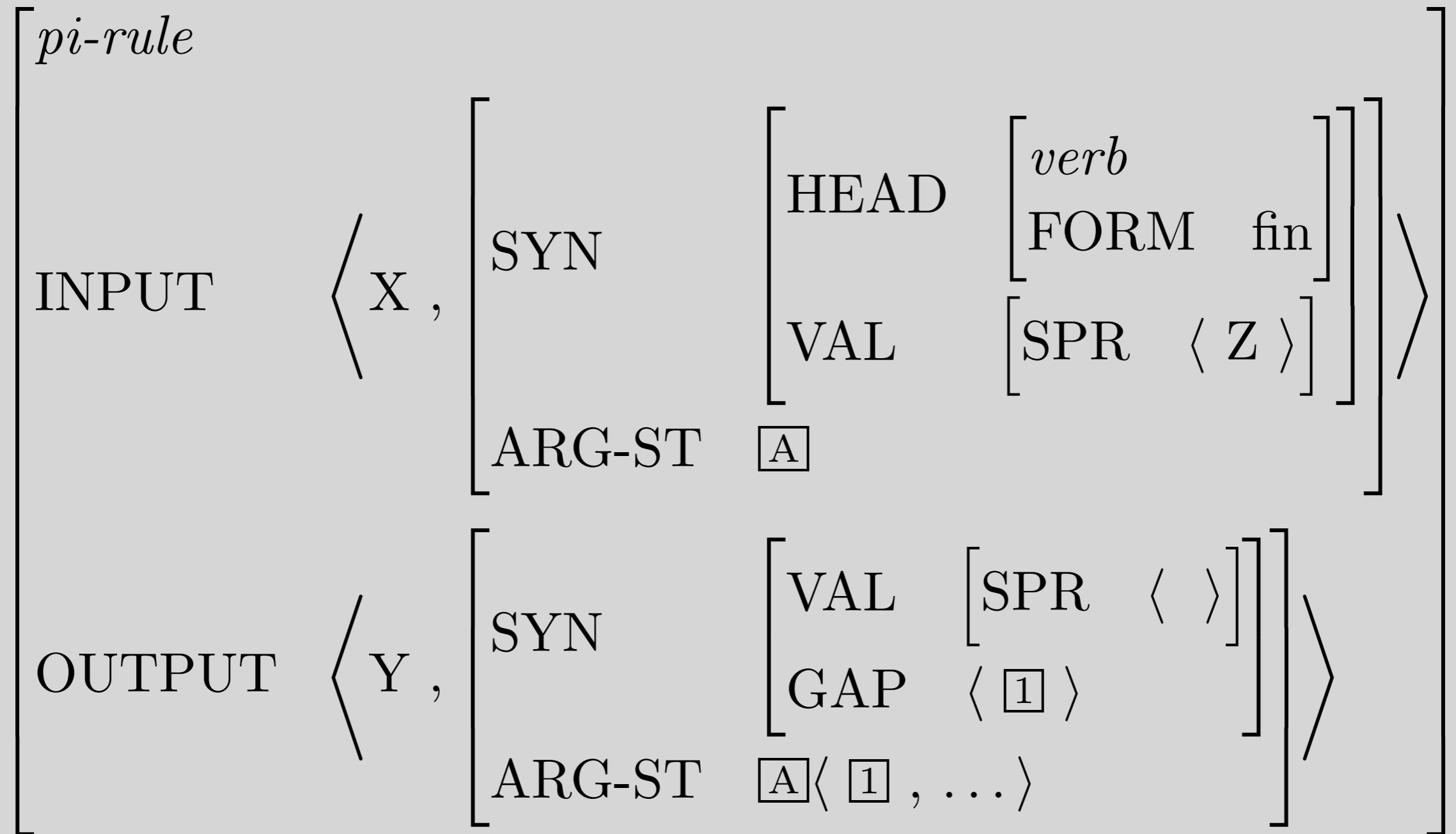
- NB: This says nothing about the phonology, because the default for *pi-rules* is to leave the phonology unchanged.

A Lexical Sequence This Licenses



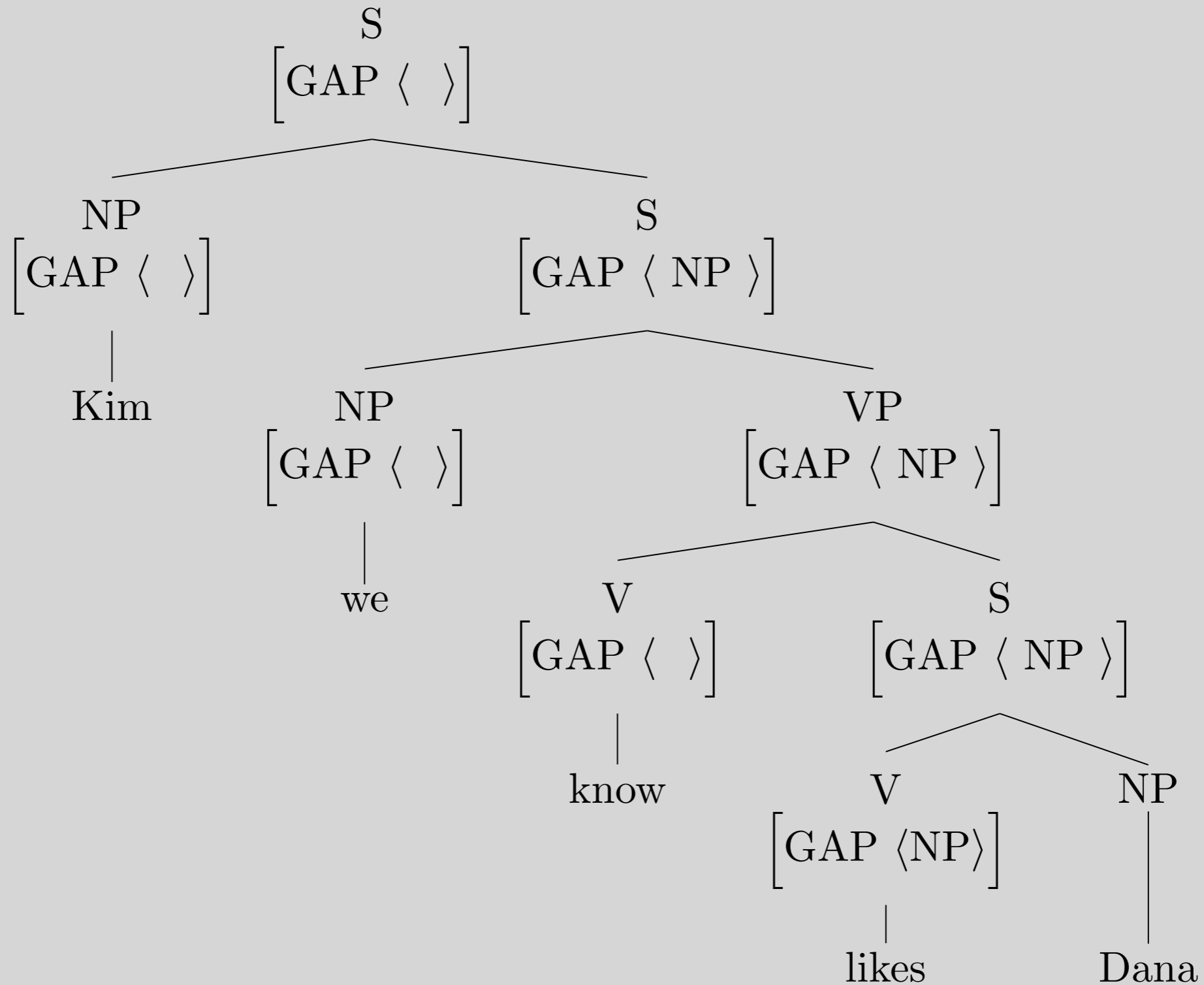
- Note that the ARP is satisfied

The Subject Extraction Lexical Rule



- RQ: Isn't Z actually [1]? Why doesn't the rule say so?
- RQ: Why isn't the HEAD value of the OUTPUT constrained?

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

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

- 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

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

- 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

- As per my understanding, the long distance dependency is constrained within one sentence itself right? For example, in the following set of sentences: *John likes to drive. He likes to take his car on long drives.* Can the dependency between *John* and *He* be classified as an LDD?

Reading Questions

- I don't understand how we end up with the multiple lexical sequences for "hand". Are those just sequences that we create "manually" when defining the lexicon, with knowledge of what gaps are allowed? Or are they generated by something.

Reading Questions

- p.431: "we could accomplish this (dealing with "missing element") by means of a lexical rule, but a more general solution is to modify the Argument Realization Principle" but I don't understand why we need to modify the Argument Realization Principle. Is the modification of Argument Realization Principle really the key to accomplish LDD?

Reading Questions

- It seems like the point of the revised ARP is to take the GAP value out from the ARG-ST list, but then in the examples on p. 433 the GAP is back in ARG-ST. What am I missing here?
- The update to the Argument Realization Principle 'guarantees that any argument that can appear on a word's COMPS list can appear on its GAP list instead'. So why do we need the GAP list when things could just as well appear on the COMPS list?

Reading Questions

- Since sometimes there are more than one results for equations like $A \ominus B$, does the revised Argument Realization Principle license multiple possibilities? I'm confused about "we will interpret an equation like $A \ominus B = C$ to mean that there is some value for $A \ominus B$ that is identical to C " in the textbook.
- The GAP feature is supposed to remove certain POS out which otherwise is required for the grammar. Since COMPS is now COMPS - GAP, shouldn't that also be reflected the ARG-ST? Why is the ARG-ST still the same?

Reading Questions

- Why are we introducing a GAP feature instead of just referencing the constituent via indexing in the head's RESTR list? I don't see why this is necessary.
- "because that GAP element is identified with the GAP element of the V 'likes' (and therefore also with an element of its ARG-ST list), any requirements that 'likes' places on its complement (that it be [CASE acc] NP, that its INDEX be identified with the LIKED in the relation) must be satisfied by the filler Kim." I understand that the filler must comply with any constraints that the V puts on its complements - but I'm not really following how that is realized on the filler's lexical entry. Could you walk through how 'Kim' ends up with CASE acc and the INDEX is identified with LIKED?

Reading Questions

- It seems like the GAP value does not really disappear but rather moves to a new "location" in the sentence, it can not really be implicit or optional but instead it has to be topicalized or be associated with a filler. Would treating this phenomenon as movement as opposed to GAP/STOP-GAP values work with our grammar?

Reading Questions

- It is mentioned on page 430 that since the gap is introduced at the smallest subtree of an LDD, "(m)any theories handle the bottom by positing an empty element in the tree." I'm curious about the reasoning behind why our grammar chose to avoid positing an empty element? It somehow feels like a very straightforward approach.

Reading Questions

- The idea of a gap feels very similar to a trace, and the filler feels like a constituent that was moved from the gap, except that the gap is not a node. So, what makes this grammar's account of LDD fundamentally different from an analysis involve movement? Do we have an example that can be correctly captured using feature passing, but would be wrong when analyzed transformationally (e.g. a deep structure does not exist)?

Reading Questions

- Is it only when the filler and the phrase that contains non-empty GAP list appear on the same level (see (35)) can the feature STOP-GAP appear as non-empty?
- How do we deal with ordering of the STOP-GAP list? Do we use the natural order as the fillers appear or do we manually order them so the \ominus operation is defined?
- We state that the lexical entries for easy and hard will specify non-empty values for STOP-GAP. I assume this will not affect the behavior of these words when they appear in contexts that are not GAP-related (e.g. this class is hard), but I'm not sure how to articulate that fact.

Reading Questions

- When we say that easy and hard are gap stoppers - does this mean that when we introduce one of these adjectives to a clause containing a gap (e.g. GAP <NP>), it changes the clause so that it no longer has a gap (GAP <>)? How is the adjective able to fill this role, since A \neq NP?

Reading Questions

- "[STOP-GAP] signals what gap is to be filled in the local subtree where it appears. Most nodes will be [STOP-GAP <>], but where a gap is associated with its filler, the feature has a non-empty list as its value." (page 437) I'm having trouble understanding what "where a gap is associated with its filler" means. In (35) "Kim we know Dana hates", STOP-GAP appears as the phrase "we know Dana hates" but why not earlier? Is there something else that says "this is where we define a STOP-GAP to appear in a tree"?

Reading Questions

- In the final version of the GAP Principle ((33) on page 437), why does the value of STOP-GAP have to be removed from the GAP value of the mother node, as opposed to inheriting up the STOP-GAP value from the head daughter? Is this just a more succinct way of doing it?

Reading Questions

- Are there other adjectives besides those mentioned in the book that perform gap-filling functions or are those two the only exceptions in English?

```
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 wondering if the GAP solution breaks in examples like 2:

Q: How and why did you rob the bank?

1: I robbed the bank (with a sharp spoon) and (because I wanted money).

2: I robbed the bank (with a sharp spoon) (because I wanted wanted money).

- Seems like if I take the bolded conjunction out, it wouldn't work. Is 2 a valid counter-example? I know it's ambiguous, but in the context in which the question is asked first, it makes sense to me, and I think that by the time we get up to the last part of the tree before we start plucking things off of the GAP list, we'd have 2 things on the list, but we're only able to take 1 thing off, because we have 'How and why' as one constituent.

Reading Questions

- It seemed like we didn't place many restrictions on GAPs as to what types of feature structures they can be (like NPs or PPs), but I didn't notice any verby GAPs in the chapter. Is it possible to have VP gaps? I can't think of any, and it seems like it would be weird, but I'm curious since on the surface it seems like this chapter leaves us open to that.

Reading Questions

- I'm just curious: do the presence of topicalized sentences make 'English as an SOV language' less valid? It sounds like yoda speech, and I've thought they were more of casual exceptions yet somehow ungrammatical.

Reading Questions

- Something I kept wondering during this chapter was how the concept of LDDs interacts with the notion that language is processed in real time and that a good model of grammar is surface-oriented. LDDs seem to go against this in a way because a piece of information is provided that almost is "held onto" with no clear role until the "gap" pops up. Though one thing I did notice is anything to do with wh- questions was easier for me to process than the topicalized sentences like *That toy, they handed to the baby*. My initial guess is wh- words almost inherently signal that they will be filling some sort of gap so I am already prepared for it, where as starting with *That toy* then following it with *they* feels like a momentary mini garden path where I'm thrown off because it was unexpected that an NP would follow. However as another example, *The baby they handed the toy to was cute*, is very natural to me.

Reading Questions

- pp 430 (15) a. states *Problems this involved, my friends on the East Coast are hard to talk to _ about _*. And then, on the next page, it says "In (15a), for example, the filler for the first gap is my friends on the East Coast, and the filler for the second one is problems this involved".
- The GAP feature and STOP-GAP features are lists, so I assume our grammar fragment can handle having multiple elements in their lists (like the example above).
- If a GAP feature contains two elements and both could satisfy the constraints imposed by an element on the STOP-GAP list, would this give rise to multiple interpretations of the sentence?
- On a related note, the formulation of the Head-Filler Rule (pp438 (34)) seems to be formulated such that the GAP and STOP-GAP lists contain exactly one element. Does having multiple elements on these lists affect the formulation of this rule?