Ling 566

Nov 21, 2013

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

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:

- *<u>The solution to this problem</u>, Pat said that someone claimed you thought I would never find____.*

- *<u>Chris</u> 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

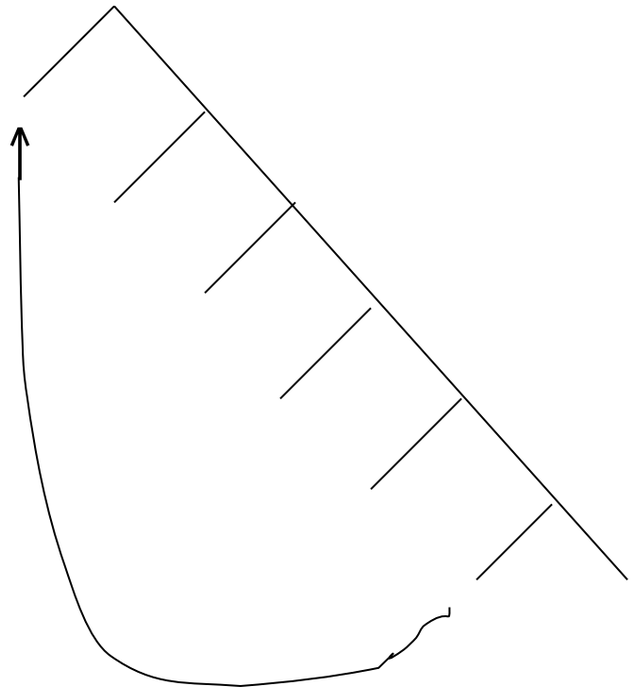
- These were long thought to constitute the strongest evidence for transformations.

- They were handled in TG by moving the filler from the gap position.

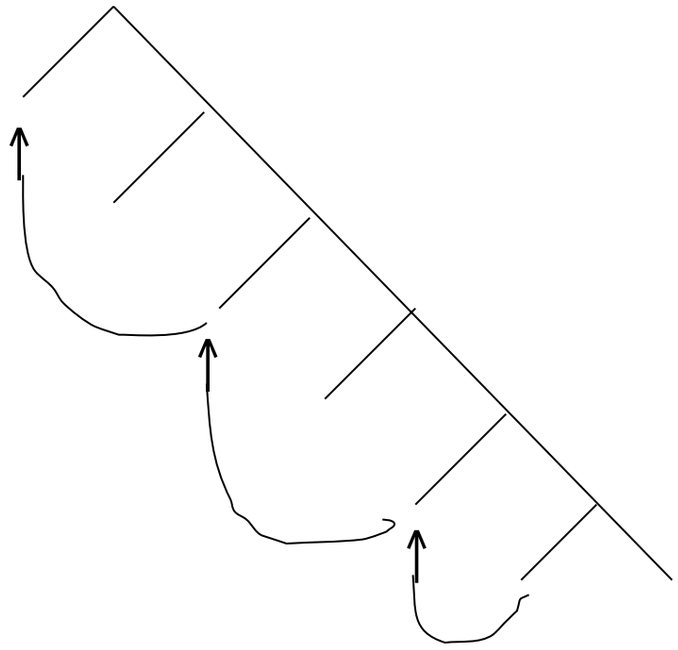
- Case, agreement, preposition selection could apply before movement.

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

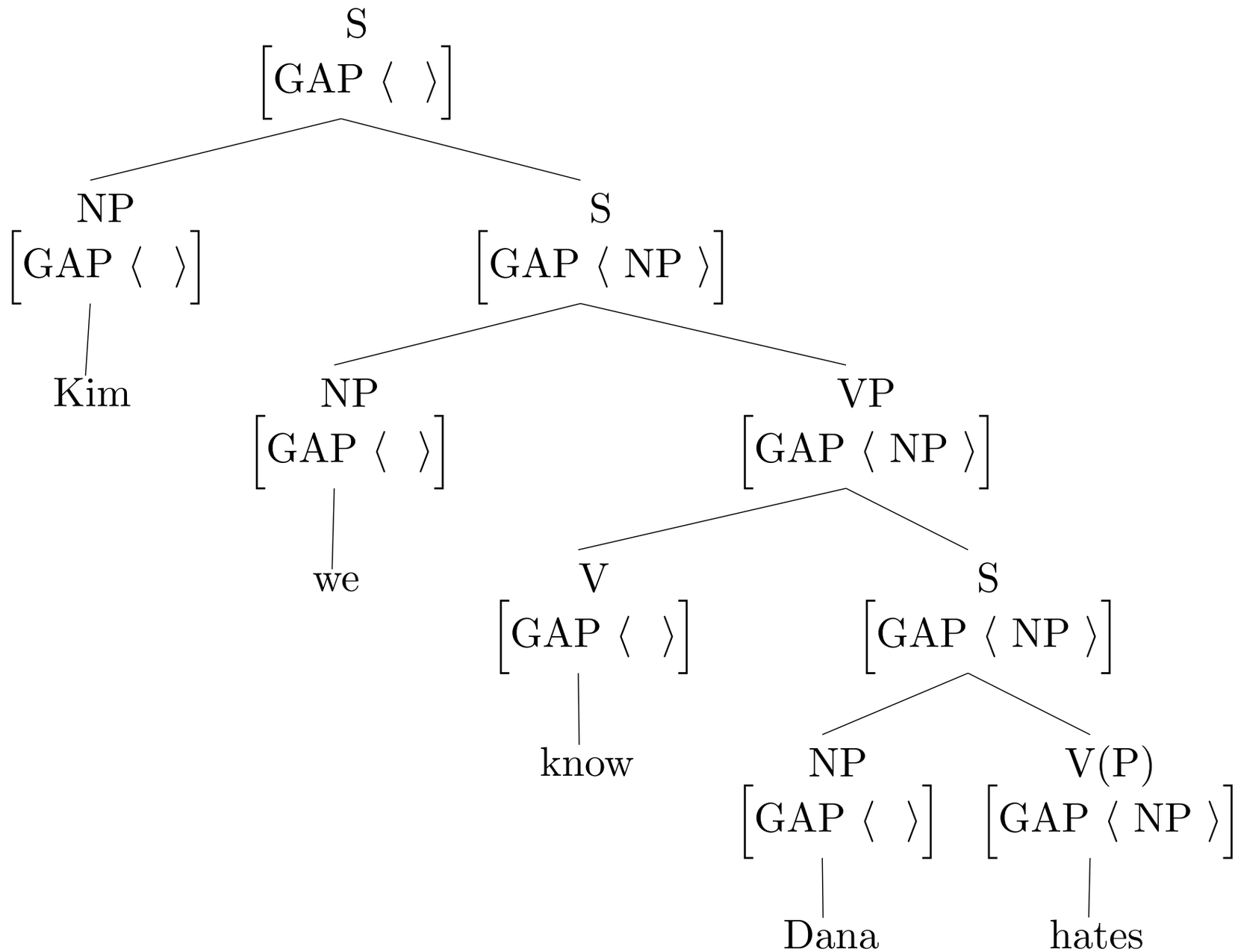
$$word\text{:}\quad \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{VAL} & \left[\begin{array}{lll} \text{SPR} & \boxed{A} & \\ \text{COMPS} & \boxed{B} & \ominus\ \boxed{C} \end{array}\right] \\ \text{GAP} & \boxed{C} \end{array}\right] \\ \text{ARG-ST} & \boxed{A} \oplus \boxed{B}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

- $\ominus$ 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.

A Word with a Non-Empty GAP Value

$$\left\langle \text{hand} , \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\text{FORM} \quad \text{fin}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\, \boxed{1} \,\rangle \\ \text{COMPS} & \langle\, \boxed{3}\text{PP[to]} \,\rangle \end{array}\right] \\ \text{GAP} & \langle\, \boxed{2}\text{NP[acc]} \,\rangle \end{array}\right] \\ \text{ARG-ST} & \left\langle \left[\begin{array}{ll} & \boxed{1}\text{NP} \\ \text{CASE} & \text{nom} \\ \text{AGR} & \textit{non-3sing} \end{array}\right] , \boxed{2} , \boxed{3} \right\rangle \end{array}\right] \right\rangle$$

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.

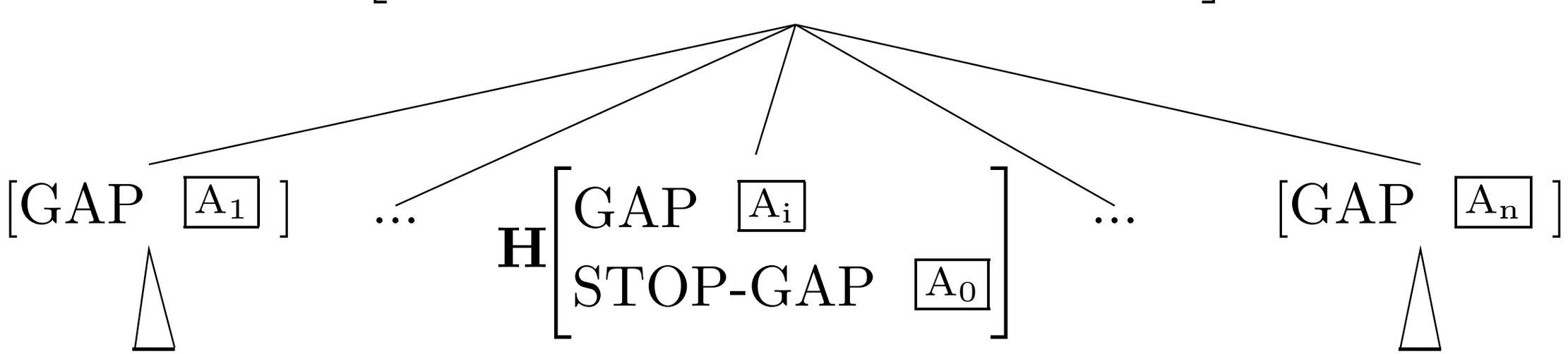
Our Solution to this Problem

- 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:

$$\left[\text{GAP} \quad (\boxed{A_1} \oplus \ldots \oplus \boxed{A_n}) \ \ominus \ \boxed{A_0}\right]$$



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

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

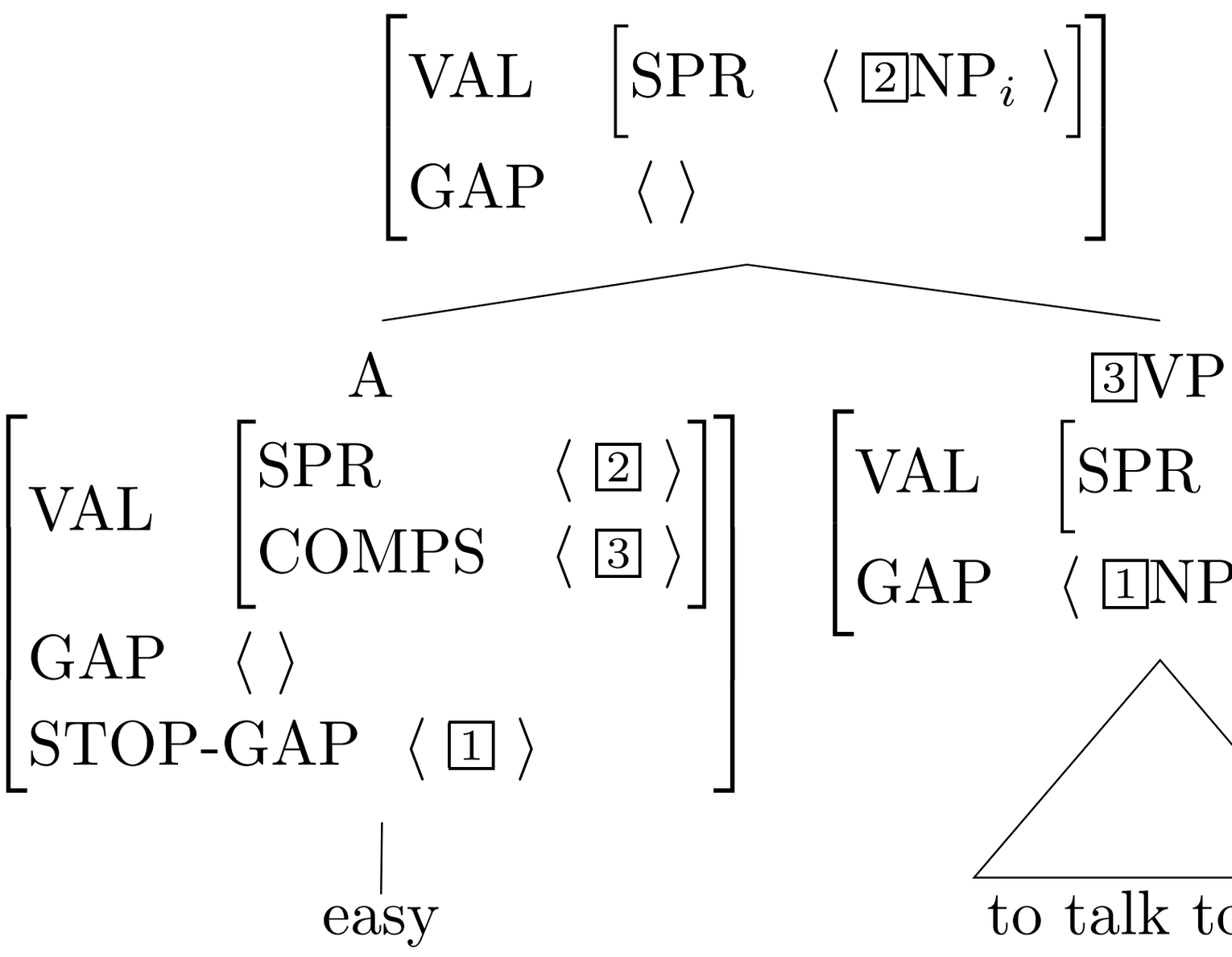
- This only covers gap filling in finite Ss
- The filler has to be identical to the GAP value
- The STOP-GAP value is also identical
- The GAP Principle ensures that the mother's GAP value is the empty list

Gap Filling with *easy*-Adjectives

$$\left\langle \text{easy} , \left[\begin{array}{ll} \textit{adj-lxm} & \\ \text{SYN} & \left[\text{STOP-GAP} \ \left\langle \boxed{1} \right\rangle\right] \\ \text{ARG-ST} & \left\langle \text{NP}_i , \left[\begin{array}{ll} \text{INF} & + \\ \text{GAP} & \left\langle \boxed{1}\text{NP}_i , \ldots \right\rangle \end{array} \right]^{\text{VP}} \right\rangle \end{array} \right] \right\rangle$$

- 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

$$\left[\begin{array}{l} \textit{pi-rule} \\ \\ \text{INPUT} \quad \left\langle \text{X} \,, \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{FORM} & \text{fin} \end{array}\right] \\ \text{VAL} & \left[\text{SPR} \quad \langle\, \text{Z} \,\rangle\right] \end{array}\right] \\ \text{ARG-ST} & \boxed{\text{A}} \end{array}\right] \right\rangle \\ \\ \text{OUTPUT} \quad \left\langle \text{Y} \,, \left[\begin{array}{ll} \text{SYN} & \left[\begin{array}{ll} \text{VAL} & \left[\text{SPR} \quad \langle\ \rangle\right] \\ \text{GAP} & \langle\, \boxed{1} \,\rangle \end{array}\right] \\ \text{ARG-ST} & \boxed{\text{A}} \langle\, \boxed{1} \,, \ldots \rangle \end{array}\right] \right\rangle \end{array}\right]$$

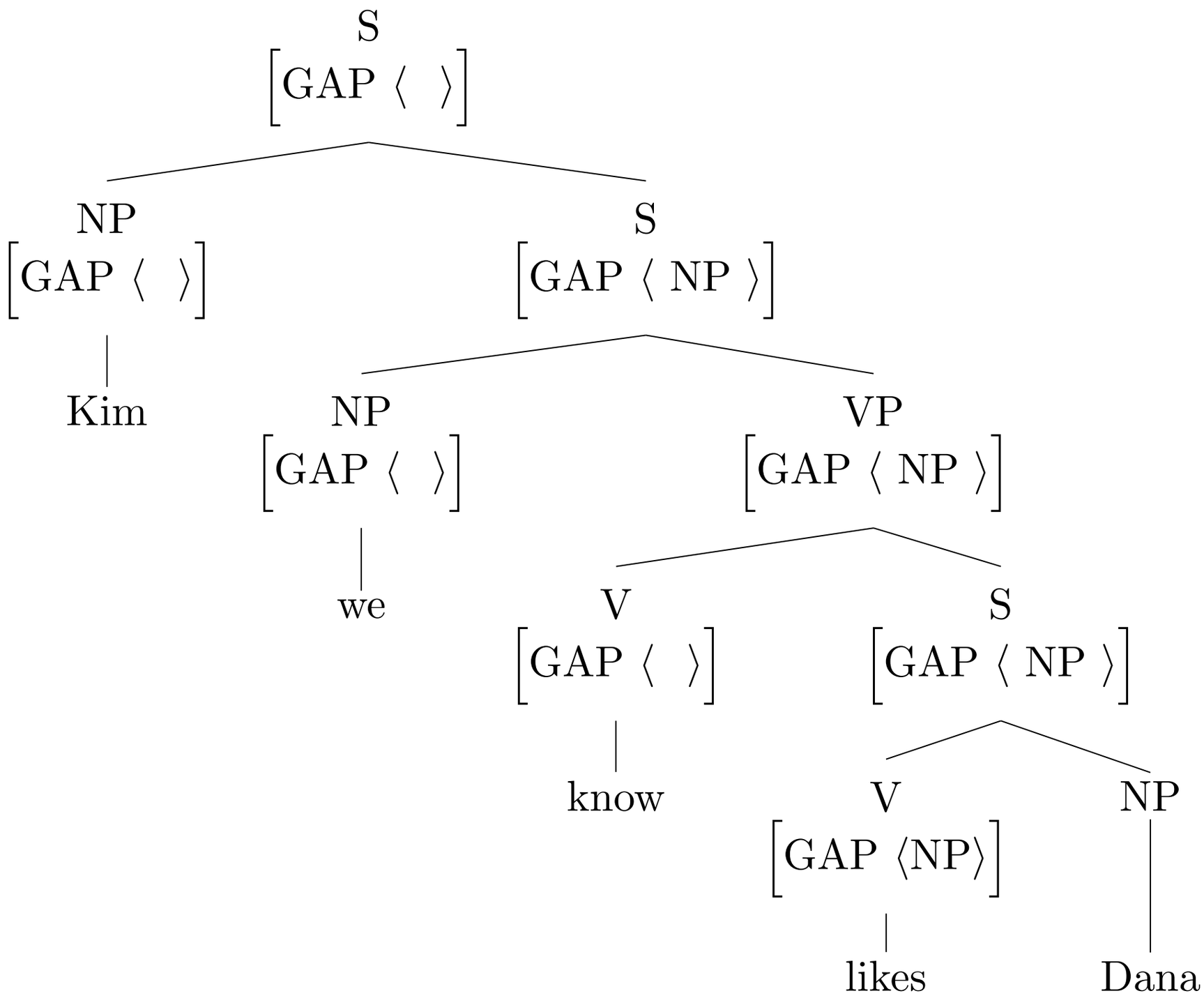
- NB: This says nothing about the phonology, because the default for *pi-rule*s is to leave the phonology unchanged.

A Lexical Sequence This Licenses

$$\left\langle \text{likes} \;,\; \left[\begin{array}{ll} \textit{word} & \\ \text{SYN} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{FORM} & \text{fin} \end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SPR} & \langle\,\rangle \\ \text{COMPS} & \langle\,\boxed{2}\,\rangle \end{array}\right] \\ \text{GAP} & \left\langle \boxed{1}\left[\begin{array}{ll} \text{CASE} & \text{nom} \\ \text{AGR} & \textit{3sing} \end{array}\right] \right\rangle \\ \text{STOP-GAP} & \langle\,\rangle \end{array}\right] \\ \text{ARG-ST} & \langle\,\boxed{1}\,,\,\boxed{2}\text{NP[acc]}\,\rangle \end{array}\right] \right\rangle$$

- 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

$$\begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_0 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_1 \end{bmatrix} \ldots \begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_{n-1} \end{bmatrix} \begin{bmatrix} \text{HEAD} & \textit{conj} \\ \text{IND} & s_0 \\ \text{RESTR} & \left\langle \begin{bmatrix} \text{ARGS} & \langle s_1 \ldots s_n \rangle \end{bmatrix} \right\rangle \end{bmatrix} \begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{IND} & s_n \end{bmatrix}$$

- 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

$$\begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{\text{A}} \\ \text{IND} & s_0 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{\text{A}} \\ \text{IND} & s_1 \end{bmatrix} \ldots \begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{\text{A}} \\ \text{IND} & s_{n-1} \end{bmatrix} \begin{bmatrix} \text{HEAD} & \mathit{conj} \\ \text{IND} & s_0 \\ \text{RESTR} & \langle \begin{bmatrix} \text{ARGS} & \langle s_1 \ldots s_n \rangle \end{bmatrix} \rangle \end{bmatrix} \begin{bmatrix} \text{FORM} & \boxed{1} \\ \text{VAL} & \boxed{0} \\ \text{GAP} & \boxed{\text{A}} \\ \text{IND} & s_n \end{bmatrix}$$

- 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

- How do you know, on the word level, whether an argument should be on the GAP list or the COMPS list?
- Between pages 432 and 433, we have for different lexical sequences for *hand*. Why that many? Is that supposed to be an exhaustive list of possible lexical sequences?

Reading Questions

- Can a GAP list have optional elements? For example, you can build a topicalized sentence with a verb that takes an optional complement, e.g. *eats* in *That cake, Sandy eats frequently*.

- Since the GAP list is filled out by the ARP and that the ARP is encoded in the word type, does that mean we are not allowed to specify in individual lexical entries what complements can go missing and what cannot? Is there even need this kind of per-word specialization?

Reading Questions

- Why:

  ARG-ST: A+B

  COMPS: B-C

  GAP: C

- instead of:

  ARG-ST: A+B+C

  COMPS: B

  GAP: C

- It makes sense to me conceptually, but what are the actual implications in the grammar?

Reading Questions

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

Reading Questions

- According to the ARP defined in p.432, does that imply that English not allow the gap in specifier ?
- What's the difference between a non-empty GAP list and a trace?

Reading Questions

- In (36) we see that STOP-GAP<> is declared in the lexeme type. Since this is defeasible (and the default is <>) this should ensure that when we are not dealing with LDDs we can basically ignore STOP-GAP<> in our trees. However, where is GAP<> introduced in our type hierarchy? I can't seem to figure it out- and the Summary section further confuses me since it says GAP<> and STOP-GAP<> are appropriate for type *syn-cat*?

- Is GAP<> then empty by default and defeasible like STOP-GAP<>?

Reading Questions

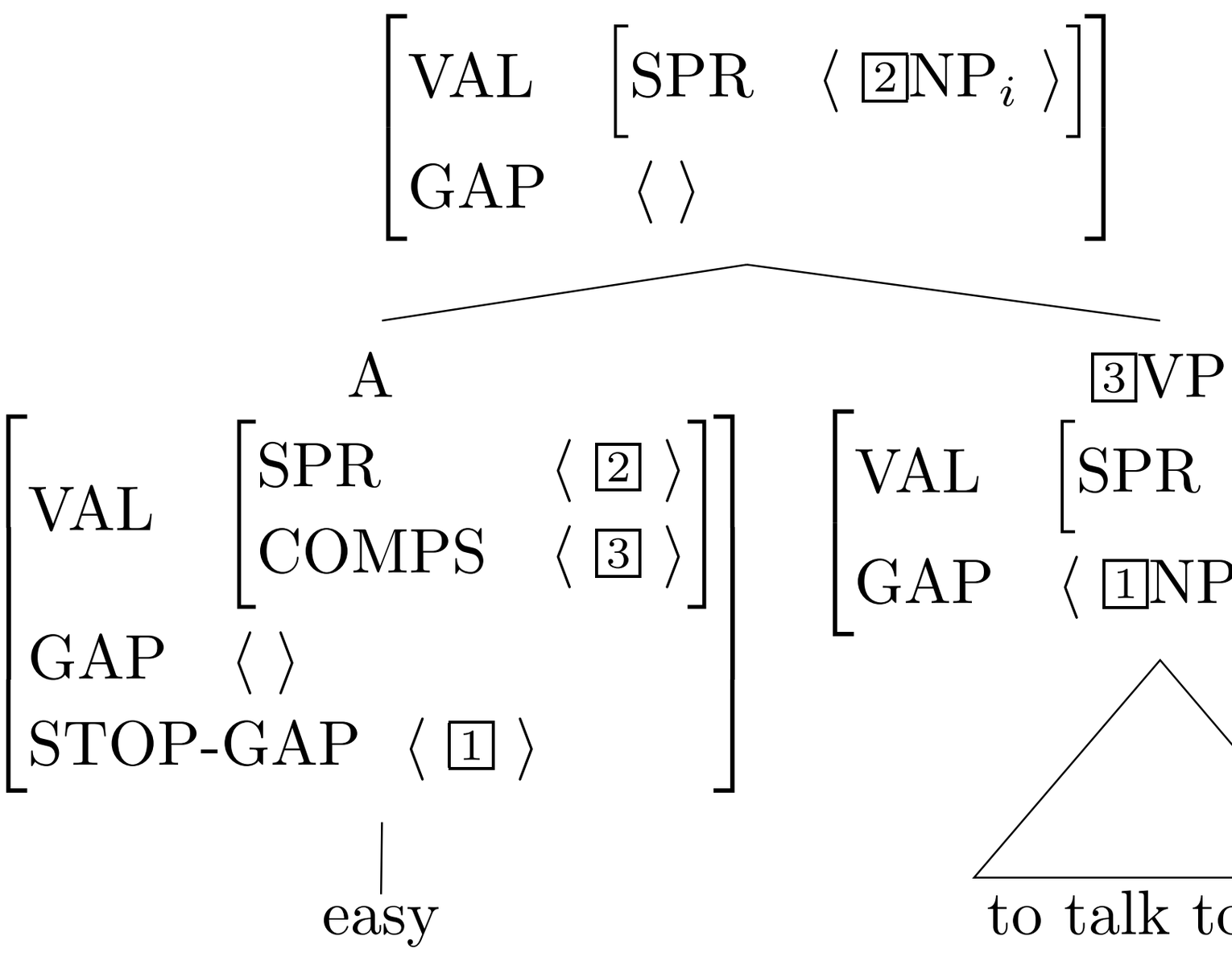
- Unless I missed something, all of our feature values so far come from somewhere - either they're specified in a lexical entry or inherited up the tree through the phrase structure rules, or introduced by a lexical rule. Here, though, we seem to be arbitrarily declaring STOP-GAP features to be non-empty in non-leaf nodes to fill gaps. Maybe that doesn't explicitly violate anything in our formalism, but doesn't that take away from the heavy lexical basis of HPSG thus far?

Reading Questions

- STOP-GAP: It's supposed to signify what sub tree includes our gapped element? Isn't that what the GAP feature says?
- How does STOP-GAP help make sure that the missing NP in (39) is fully resolved?

  (39) Pat is easy to continue to follow ___.

A Tree for *easy to talk to___*



Reading Questions

- We did just fine without STOP-SPR or STOP-COMPS. What is it about GAP that necessitates STOP-GAP? Is it because we want to limit constructions that can give rise to LDD?
- Why doesn't the Head-Filler Rule doesn't put the STOP-GAP on the preceding phrase instead of the headed phrase? It seems to be more logical.

Reading Questions

- The Head-Filler Rule allows the root S node in (35) to be Gap < > because the STOP-GAP and GAP values of its daughter match the specifier, but is there some independent reason why a non-empty STOP-GAP list appears on the head daughter S node in the first place?

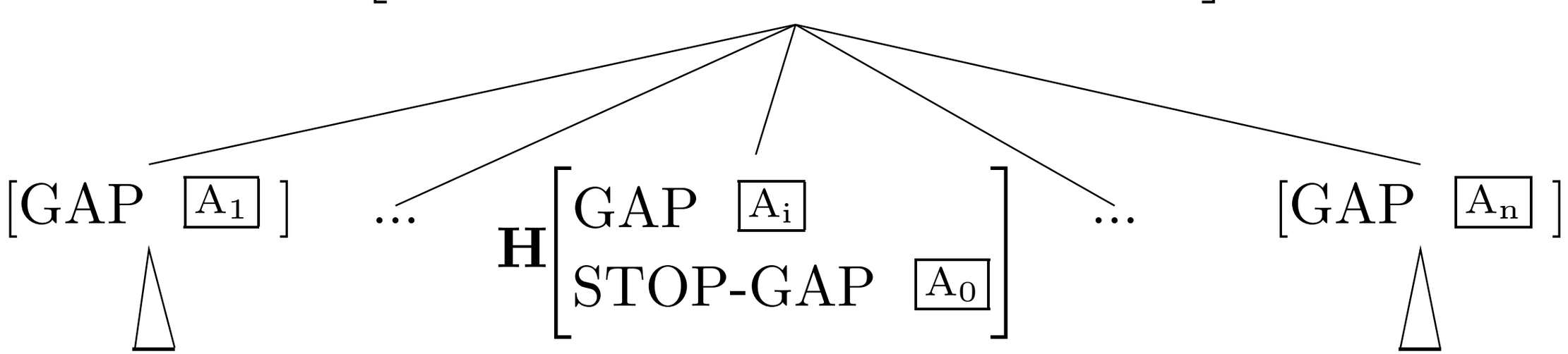
Reading Questions

- I'm wondering why the STOP-GAP middle node of the GAP principle diagram in (33) is labeled as the head. Is the STOP-GAP element always the head? I didn't think that was the case, but maybe I missed something?

The GAP Principle

A local subtree Φ satisfies the GAP Principle with respect to a headed rule ρ if and only if Φ satisfies:

$$\left[\text{GAP} \quad (\boxed{A_1} \oplus \ldots \oplus \boxed{A_n}) \ \ominus \ \boxed{A_0}\right]$$



Reading Questions

- The GAP principle as formulated in (33) on pg 437 seems to indicate that any number of daughter elements can come prior to the headed daughter that contains STOP-GAP. However the only rule that seems to license a construction like (33) is the Head-Filler rule on the next page, which only has a single GAP<> element to the left of the head. Are there any constructions where we actually need the leftmost "..." in the GAP principle?

- Also, where does the GAP principle live in our grammar? Is it a constraint on the rules themselves?

Reading Questions

- (33) seems to indicate that GAP and GAP STOP have different values in the head daughter (Ai and A0) but in (35) they are the same. What am I missing here?

Reading Questions

- Why does STOP-GAP appear in non-leaf nodes when there are no gap-stoppers? There doesn't seem to be a lexical rule licensing it.
- For cases where we are not dealing with gap stoppers like easy and hard, how does the feature STOP-GAP end up on a node like an S? The feature doesn't get passed up apparently so how does it magically appear?

Reading Questions

- The text seems to be suggesting that the easy/tough method of gap-filling is different than the kind described by the Head-Filler Rule. Is this the case, and if so, what else is going on to license the gapless AP in (38)?

Reading Questions

- Why identify the GAP value of mother and daughter in imperatives?

**Me, hand the toy!*

(?)To me, hand the toy!

This book, put on the top shelf!