Chapter 15

Long-Distance Dependencies

15.1 Lecture notes

Chapter 15

I. **Historical Perspective**

- LDDs seemed like the clearest evidence for transformations: dependencies of valence, semantics, case, and agreement can all hold between non-adjacent elements in a sentence. This could be handled by assuming that what's far apart on the surface starts out close together and gets moved.
- The unbounded character of these dependencies seemed to pose major obstacle to other approaches.
- Early analyses of LDDs assumed swooping, i.e. movement from the gap position to the filler position in one fell swoop. **[Slides:1–2]**
- The discovery of islands—that is, the fact that there are constraints on the relative positions of gaps and their fillers—created one of the most extensive and fruitful areas of research in generative grammar. The seminal work was Ross's (1967) dissertation.
- Chomsky's (1973) 'Conditions on Transformations' proposals to account for islands analyzed LDDs in terms of a succession of short movements ('looping'). It also proposed that movement rules leave behind a 'trace'—what amounts to a silent pronominal copy of the moved element.

- Subsequent research provided evidence for looping over swooping, including *inter alia* the facts about Irish complementizers in our Problem 4. A number of languages have some overt marking of material in the region between the filler and the gap.
- This evidence that LDDs involve a chain of local dependencies provides one kind of motivation for non-transformational analyses in which the information about fillers and gaps is propagated through trees.
- Early non-transformational analyses employed traces in the gap positions, but traces proved eliminable from this treatment (though controversy has remained about their reality).

II. **Crucial elements of our analysis**

- The transmission of information between fillers and the gap location has three parts:
  1. Bottom: information about the gap must be available down near the level of the selecting lexical item
  2. Middle: information about the gap must be propagated through the tree
  3. Top: the filler must be matched to the gap information
- The devices we introduce, by function, are: **[Slides:3]**
  1. Bottom: a feature GAP, a revision of the ARP, the Subject Extraction Lexical Rule
  2. Middle: the GAP Principle
  3. Top: the Head-Filler (phrase structure) Rule

? GAP is a feature of *gram-cats*. Why?

- Assuming (as we have) that the Coordination Rule identifies SYN values of the conjuncts and the mother, GAP must be somewhere within SYN; otherwise the GAP information of the coordinate mother and its daughters won't be properly identified.
- But GAP can't be within HEAD because phrases don't necessarily share GAP values with their head daughters, e.g.: **[Slides:4–5]** (The lower S and the VP both get their non-empty GAP values from non-head daughters.)

(9) What did you put it into?

- The revised Argument Realization Principle allows arguments to end up on the GAP list, instead of the COMPS list, because it says that the COMPS list is everything after the SPR on the ARG-ST, except what shows up in the GAP list.

? The technical device used to ensure this is list subtraction, which doesn't necessarily give a unique output. Why do we want that? (So that a single lexical entry for, e.g. a ditransitive verb can license multiple word structures with different arguments on the GAP list.) **[Slides:6–7]**

? Subject gaps are introduced via a lexical rule **[Slides:8]**. Why? (Because all verbs require a singleton SPR list; subtracting from that list would conflict with this constraint.)

? The output of this rule has an empty SPR value and the first element of the ARG-ST on the GAP list. Why doesn't the ARP cause the first ARG-ST element to end up on the COMPS list? (Since the SPR list is empty, the ARP will be satisfied if the ARG-ST value equals the COMPS list minus the GAP list. The GAP list has to be the first element of the ARG-ST, as specified by the output of the lexical rule, so this element can't be on the COMPS list.)

III. Some applications

- Recall from Chapter 14 that AAVE doesn't allow copula absence in contexts like (10): **[Slides:9]**

(10) a. I wonder where he is.
b.*I wonder where he.

? How is this predicted by the analyses of the zero copula and of long distance dependencies? (In these sentences, *where* is the filler. In the grammatical sentence, there is a GAP originating from the word structure over *is*. In the ungrammatical sentence, there is no bottom for the long-distance dependency—in zero copula sentences like *He over there*, *over there* is not selected by any other element so there is nothing for the ARP to make a gap out of.)

- This analysis of LDDs, together with the analysis of coordination, has Ross's Coordinate Structure Constraint (and its codicils) as an automatic consequence:
- No conjunct can be a gap: **[Slides:10]**

  (11)*Here is the student that the principal suspended __ and Sandy.

? How does our account rule this out? (The Coordination Rule provides no way of building *and Sandy* without another conjunct before the *and*.)

- A gap cannot be contained in a conjunct if its filler is outside of that conjunct: **[Slides:11]**

  (12)*Apple bagels, I can assure you that Leslie likes __ and Sandy hates cream cheese.

? How does our account rule this out? (The Coordination Rule requires that the SYN value of all of the conjuncts be the same, but GAP is a SYN feature so unless the GAP values are the same the SYN values won't unify.)

- ... unless each conjunct has a gap paired with the same filler: **[Slides:12–13]**

  (13) Apple bagels, (I can assure you that) Leslie likes __ and Sandy hates __ .

? Why does our account allow these? (The GAP values are the same, so the SYN values do unify. Note that the semantic index of the filler, *apple bagels*, gets transmitted through the GAP dependency to both of the verbs, *likes* and *hates*.)

- The Across-the-Board exception doesn't apply if the GAP is the whole conjunct: **[Slides:14]**

  (14)*Which rock legend would it be ridiculous to compare __ and __ ?

? How does our account rule this out? (Again, the Coordination Schema requires actual conjuncts, not just *and*.)