

Department of Computer Science  
CSC 247/447 and BCS 235/535  
Fall 2006

### Assignment 3 (non-Lisp portion)

Due: 3:15pm, Tuesday, Oct. 24

#### Problem 1 - Long-distance dependencies

Draw a phrase structure tree for each of the following sentences, showing verb SUBCAT and VFORM features wherever they appear. In (a) and (b), also show AGR features. (*Use the concise notation that lists only feature values, not names like AGR, SUBCAT, etc.*) For (f) assume a coordination rule

$X \rightarrow X \text{ COORD } X$ ,

where  $X$  is any category and both  $X$ 's on the RHS are head daughters. (COORD expands as *and* or *or*.)

- (a) Who arrived?
- (b) What did you do?
- (c) How fast is your turtle?
- (d) How does the program she wrote work?
- (e) Where she goes, he goes.  
(Treat *where she goes* as a topicalized PP (or ADVP), realized as a “free relative clause”).
- (f) He bought more caviar than he could afford or consume.

#### Problem 2 - Other gap phenomena (grads only)

Some “it cleft” constructions involve relative clauses, as in the following examples (with relative clauses bracketed):

It was Mary (who spoke up)

It was Alice (that Bill tried to notify)

These are accounted for by the rules

$S \rightarrow \text{NP}[\text{it}] \text{VP}[\text{it}]$

$\text{VP}[\text{it}] \rightarrow \text{V}[\text{be}] \text{NP R}$

$\text{NP}[\text{it}] \rightarrow \text{it}$

(along with rules for NP's and R's, of course). Some it-clefts of a slightly different form are the following:

It was with Mary that John danced

It was to see Mary that John travelled to Denver

It was rich John wanted most of all to be

Devise a rule or rules to account for these cases. (Take “*that*” to be a complementizer in these 3 sentences, forming an S[that], not a relative clause.) Do the two sets of rules cover

some of the same sentences? Give examples of how your rules work (phrase structure trees).

### **Problem 3 - Parsing with gaps**

- (a) Carefully explain how a chart parser using Earley-style top-down prediction would generate an item of general form

S[/NP] → ◦ NP VP[/NP]

in processing the material following *than* in (f). I.e., what are the steps in the creation of a dotted rule (active arc) with this particular configuration of gap features? Similarly explain how a dotted rule of form

VP[/NP] → ◦ VP[/NP] COORD VP[/NP]

would be generated (with this particular configuration of gap features). Finally, explain how the gapped VP *afford* is processed from this point on, based on dotted rules deriving from the above rule.

- (b) If instead of Earley-style top-down predictive parsing we used simple left-corner parsing, how would the dotted rules in (a) be formed? [You can assume either of the two methods sketched in class, i.e., the one that immediately forms “slashed” variants of active arcs upon instantiating a rule (under certain conditions), or the one that forms XP/XP traces under certain conditions, and forms larger active arcs with slash features only when it can extend an active arc with a constituent containing such a trace.]

### **Problem 4 – Attachment Preferences**

What factors might plausibly determine attachment choices in the following sentences? (Include comments on how these factors help to explain intuitive reactions in reading these sentences.)

- (a) She chided the woman using her cell phone.
- (b) He stopped talking constantly.
- (c) He met the woman he adored in a church.
- (d) She describes men who have worked on a farm as cowboys.

### **Problem 5 – Shift-reduce parsing**

- (a) Exercise 4(a) (p.184), ch.6 of NLU. Represent the oracle by drawing the transition graph (as in Fig. 6.5) and labelling *reduce*-states in the graph with the phrase structure rule to be used for the reduction. (There is no need to represent the oracle as a table, as in Fig. 6.6.) In the construction of the graph, treat the *to* in the INF rule just as if it were a lexical category.<sup>1</sup>

---

<sup>1</sup>There’s a slight inconsistency in the book concerning how words in the input stack are processed: according to Figs. 6.5 and 6.6, the parser never expects to see words, only lexical and phrasal symbols. Fig. 6.7 seems to allow for direct processing of words, but in the illustration starting on p.166, words are tacitly converted to lexical categories when they are shifted from the input stack to the parse stack. Assume that such a conversion is done here.

- (b) Do enough of part (b) of Exercise 4 to show that guesswork would be required in the conversion of words to lexical categories in a shift-reduce parse of the sentences shown, using the oracle in (a).
- (c) Explain how the oracle can be modified to deal with this ambiguity deterministically.
- (d) (**grads only**) At a glance, transition graphs for shift-reduce parsers seem rather similar to RTNs. State some differences between them.