

Ch. 10: Parsing

WHAT IS PARSING?

Input = a sentence and a grammar

Output = a tree for the sentence derived from the grammar

TWO BASIC KINDS OF PARSING:

Top-down parsing = start from the start symbol (e.g., S), then systematically try all the possibilities until we find a suitable tree or prove that there isn't one.

See figures 10.7 and 10.8 for example of a top-down, depth-first, left-to-right parse.

Would this be a good idea in a language with a different word order?

Bottom-up parsing: Start with the words of the input, then systematically try to build up trees from the input until you find one starting with the start symbol (i.e., S), or prove that there isn't one.

See figure 10.4 for a partial example.

Advantages of top-down parsing:

- It never wastes time on subtrees that cannot have an S at the root. Bottom-up parsing does this.

Disadvantages of top-down parsing:

- Wastes time on trees that don't match the input (compare the first word of the input with the leftmost branch of the tree). Bottom-up parsing doesn't do this.

IMPLEMENTING TOP-DOWN PARSING:

- Depth-first strategy.
- Breadth-first strategy.
- Advantages and disadvantages of depth-first strategy.
- Advantages and disadvantages of breadth-first strategy.

Combining top-down and bottom-up strategies:

One way is to use top-down parsing, but use the bottom-up concept to *filter* the emerging parse trees, i.e. drop those that don't match the first word of the input.

Here is the left-corner table for the grammar in figure 10.2:

S: det, properNoun, aux, verb

NP: det, properNoun

Nominal: noun

VP: verb

PROBLEMS WITH TOP-DOWN PARSING

- Left recursion (see figure 10.10)

Two ways to fix the left-recursion problem:

-- Rewrite the grammar

Example:

Input: $A \rightarrow AX \mid Y$. (where X is a string of one or more symbols)

Output: (a right-recursive grammar)

$A \rightarrow YA'$

$A' \rightarrow XA' \mid \epsilon$

Why rewriting the grammar really doesn't help:

a) Because the rewritten grammar is hard for people to understand.

b) Because it will be difficult to use trees generated from the rewritten grammar to obtain the meaning of the sentence.

These are really the same thing: a) refers to people's direct use of the grammar; b) refers to people's indirect use of the grammar, i.e. using the grammar to write a semantic analyzer.

-- Controlling the depth of search.

Theoretically ugly, and might lose a possible parse.

- Structural ambiguity

a) Attachment ambiguity

1) PP-attachment

One morning I shot an elephant in my pajamas. How he got into my pajamas I don't know. (Groucho Marx)

See figure 10.11.

This is PP-attachment ambiguity.

Here is another example:

John hit a boy with a broken leg.

Cf. John hit a boy with a hammer.

2) Gerundive attachment

There are many other kinds of attachment ambiguity. Here is one involving gerundive attachment.

We saw the Eiffel Tower flying to Paris.

3) NP attachment

Can you book TWA flights?

See figure 10.12 for the two parses.

b) Coordination ambiguity

[old [men and women]]

[old men] and [women]

President Kennedy today pushed aside other White House business to devote all his time and attention to working on the Berlin crisis address he will deliver tomorrow night to the American people over nationwide TV and radio.

[nationwide [TV and radio]]

[nationwide TV] and [radio]

What is the object of “pushed aside”?

other White House business?

[other White House business to devote all his time and attention to working]

cf. Kennedy denied [his attention to propose a new budget to address the deficit].

c) Noun-phrase bracketing ambiguity

Show me the meal on flight UA 386 from San Francisco to Denver.

Does “from San Francisco” mean that that’s where the showing is happening.

See figure 10.13.

Syntactic vs. semantic ambiguity:

All of the above are examples of syntactic ambiguity.

Cf. semantic ambiguity:

John wants to marry a Swede.

- Repeated parsing of subtrees.

See figure 10.13.

See figure 10.14.