Parsing

Parsing

- Parsing is the process of constructing a parse tree for a sentence generated by a given grammar.

- Subsets of context-free languages typically require O(n) time
  - Top-down parsing method - Predictive parsing using LL(1) grammars.
  - Bottom-up parsing method - Shift-Reduce parsing using LR(1) grammars ()

Approaches to Parsing

- *Top-down parsing*:

  Attempts to figure out the derivation for the input string, starting from the start symbol.

- *Bottom-up parsing*:

  Starting with the input string, attempts to “derive in reverse” and end up with the start symbol;

  forms the basis for parsers obtained from parser-generator tools such as yacc, bison.

Top-down Parsing

"top-down:" starting with the start symbol of the grammar, try to derive the input string.

Parsing process:

Use the current state of the parser, and the next input token, to guide the derivation process.

Implementation:

Use a finite state automaton augmented with a runtime stack ("*pushdown automaton*").

Bottom-up Parsing

"bottom-up:"

work backwards from the input string to obtain a derivation for it.

Parsing process:

use the parser state to keep track of:

what has been seen so far, and

given this, what the rest of the input might look like.

Implementation:

use a finite state automaton augmented with a runtime stack ("*pushdown automaton*").

Parsing: Top-down vs. Bottom-up