Parsing, Grammars, & syntax-directed translation

Context-Free Grammar (CFG or grammar)

terminals (tokens or lexemes)
nonterminals (variables)
productions (rules)

Example
Grammar for balanced parentheses:

$S \to (S) S$
$S \to \varepsilon$ (empty string (not a token))

Abstract notation:
nonterminals are capital letters
terminals are anything else except ε

S represents "any string of balanced parentheses"

A derivation from some nonterminal A is a sequence of the form

$A \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_n$

where α_i is a string of terminals and nonterminals ("sentential form")
and each α_{i+1} results from α_i by replacing some occurrence of a nonterminal V with the right-hand side of a production whose left-hand side is V.

Production: (in general)

$\underbrace{V}_{\text{head}} \to \underbrace{X_1 X_2 X_3 \cdots X_k}_{\text{body}}$

where V is a nonterminal and $X_1, \ldots, X_k$ are either terminals or nonterminals ($k \geq 0$)

A context-free grammar is a list of productions.

$S \to \varepsilon$
$S \to (S) S$

$S \Rightarrow \varepsilon$
$S \Rightarrow (S)S \Rightarrow ()S$
$\Rightarrow ()(S)S$,
$\Rightarrow ()((S)S)S$
$\Rightarrow ()(()S)S$
$\Rightarrow ()(())S$
$\Rightarrow ()(())$

$\Rightarrow^*$ means zero or more steps of a derivation

One nonterminal in a grammar is designated as the start symbol.
(By default, the start symbol will be the head of the first production in the list.)
start symbol is S in this case

A derivation is complete
iff (it's finite and)
starts with the start symbol (only)
and ends with a string consisting only of terminals.

Let G be a grammar:
N = set of nonterminals
T = set of terminals
$[N \cap T = \emptyset]$
$S \in N$ is start symbol,
P is the set of productions.

A string α of terminals is generated by G (derivable from S) if there is a complete derivation ending in α.

Feb 2-11:00 AM

Grammar G.

L(G) is the set of all strings of terminals derivable from G.

Example: If

$$G = \begin{cases} S \to \varepsilon \\ S \to (S)\,S \end{cases}$$

then L(G) is the set of all strings of balanced parentheses.

[may not be obvious]

ε S

(----) S

Arithmetic expressions:

atomic expressions: constants or variables (numerical)

operators +, −, *, /, % binary infix

~~possibly allow +, − (unary prefix)~~

parentheses for grouping

Tokens (terminals):

{ c, v, +, −, *, /, %, (,) }

Start symbol represents the entire construct:

E (E for "expression")

productions:

$E \to c$

$E \to v$

$E \to E + E$

$E \to E - E$

$E \to E * E$

$E \to E / E$

$E \to E \% E$

$E \to (E)$

More compactly:

$E \to c$

$E \to v$

$E \to E\,O\,E$

$E \to (E)$

$O \to +$

$O \to -$

⋮

$O \to \%$

Shorthand:

$A \to \alpha_1 \mid \alpha_2 \mid \cdots \mid \alpha_k$

short for

$A \to \alpha_1$
$A \to \alpha_2$
⋮
$A \to \alpha_k$

$E \to c \mid v \mid E\,O\,E \mid (E)$

$O \to + \mid - \mid * \mid / \mid \%$

c*c + c ← derive this

$E \Rightarrow E\,O\,E \Rightarrow E\,O\,E\,O\,E$

$\Rightarrow c\,O\,E\,O\,E \Rightarrow c * E\,O\,E$

$\Rightarrow c * c\,O\,E \Rightarrow c * c + E$

$\Rightarrow c * c + c$

Leftmost derivation: always replace the leftmost nonterminal occurrence in a sentential form.

$E \Rightarrow E\,O\,E \Rightarrow c\,O\,E$

$\Rightarrow c * E \Rightarrow c * E\,O\,E$

$\Rightarrow c * c\,O\,E \Rightarrow c * c + E$

$\Rightarrow c * c + c$

c + c * c

A grammar is ambiguous if there exists more than one complete, left-most derivation of some string.

Ambiguous grammars should be avoided, if possible for use in compilers.

E expression: can be

$T \pm T \pm T \pm T \pm \cdots \pm T$

where each T is a term

$E \to T$

$E \to E + T \mid E - T$

$E \Rightarrow E + T \Rightarrow E - T + T$

$\Rightarrow T - T + T$

Feb 2-11:32 AM

term (T) is a string of one or more factors (F) separated by *, /, %
(grouped from left to right)

$$T \rightarrow F \mid T * F \mid T / F \mid T \% F$$

$$F \rightarrow c \mid v \mid (E)$$

This is the standard unambiguous grammar for arithmetic expressions.

Feb 2-12:10 PM