Grammars & Parse Trees

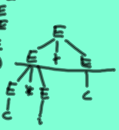
Fix a grammar $G = \langle N, T, P, S \rangle$

N: nonterminals; T: terminals; P: productions (rules); S: start symbol

Def: A parse tree (over G) is a rooted, ordered tree, where each node is labelled by a grammar symbol or ε, and each label of an internal node is a nonterminal and looks like

where $A \to X_1 \cdots X_k$ is a production of G

$E \to E+E$
$E \to E-E$
$E \to E*E$
$E \to E/E$
$E \to (E)$
$E \to c$
$E \to i$



c*i+c ⇐

Def: Let T be a parse tree. The yield of T is the concatenation of its leaves (labels thereof) in left-to-right order (in-order traversal)

Feb 9-10:57 AM

Def: A parse tree T is complete if
1) The root label is the start symbol, and
2) Each leaf label is either a terminal or ε.

[only time ε is allowed:
$A \to \varepsilon$] empty production
is a production and ε is the only child of A

Ex:
$S \to aS$
$S \to Sb$
$S \to \varepsilon$

shorthand: $S \to aS \mid Sb \mid \varepsilon$

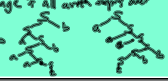
yield = ab

Def: The language of a grammar G (notation: L(G)) is the set of all strings of terminals that yielded by complete parse trees of G.

$E \to E+E \mid E-E \mid \ldots \mid i \mid c$

Language is all arith exprs over i, c.

aadbb



Feb 9-11:11 AM

Def: A grammar is ambiguous if there exists some string in the language that has more than one parse tree

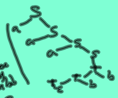
Equivalent (same language) unambiguous grammar:

$S \to aSb \mid \varepsilon$ (no)

$S \to SS \mid a \mid b \mid \varepsilon$ (no)

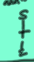
$S \to aS \mid T$
$T \to Tb \mid \varepsilon$

aaabb



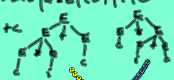
unique parse tree yielding aaabb

ε

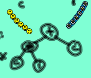


$E \to E+E \mid E*E \mid (E) \mid i \mid c$

c*i+c



goal of front end: produce a syntax tree:



Disambiguating expr grammar

Feb 9-11:25 AM

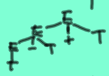
$\underbrace{c*c}-\underbrace{c*i*c}+\underbrace{i}$

+, − take lower precedence than *, /

equal precedence, left-associative (application order is left to right)

$\underbrace{3*4}_{T}-\underbrace{5*6}_{T}+\underbrace{7} = -11$

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



term: $\underbrace{c*i/c*i/i}_{F}$

$T \to T*F \mid T/F \mid F$

$F \to i \mid c \mid (E)$

c*(c+c)

unique parse tree

This grammar is unambiguous & natural

Feb 9-11:39 AM

adding:
Exponentiation (^)

Questions: precedence? associativity?

$3*x^2$

$3*x$^2

^ precedes over *, /

2^3^4

$= 2^{3^4} = 2$^(3^4)

^ associates right-to-left (right-associative)

$E \to E+T \mid E-T \mid T$ (same)
$T \to T*F \mid T/F \mid F$
$F \to B$^$F \mid B$
$B \to c \mid i \mid (E)$

c^i^(i+c):

Alternate Notations for grammars:

BNF (Backhaus-Naur Form)

Feb 9-11:54 AM

BNF:

nonterminals look like <name>

terminals are everything else: symbols in quotes or names (no < >)

→ turns into ::=

```
<expr> ::= <expr> '+' <term>
         | <expr> '-' <term>
         | <term>
<term> ::= <term> '*' <factor>
         | <term> '/' <factor>
         | <factor>
<factor> ::= const
           | id
           | '(' <expr> ')'
```

Extended BNF (EBNF)

Production bodies can be regular expressions

```
<expr> ::= (<term> ('+'|'-'))* <term>
```

Feb 9-12:08 PM