Meeting 08: Syntax

Today

Lab 1 Comments
Your Questions
Concrete + Abstract Syntax

Lab 2 Autotester Available

Lab 1 Survey
Hours Spent:
6.58 avg, 6 median, 3.41 std dev
Difficulty:
3.55 avg, 4.1 median, 1.06 std dev
What was hard?
- pattern matching
- recursive "reconstruction"
- remembering search trees
- delete
- Scala is complex

Concrete syntax is surface structure of a language (= strings)

Abstract syntax is a deeper structure of a language (= trees)

Parser converts concrete syntax to abstract syntax
Why is there a gap?

precedence
associativity
are ways to
deal with
ambiguity

\[ e ::= n \mid e_1 + e_2 \]

\[ 1 + (2 + 3) \quad (1 + 2) + 3 \]

grammar
ambiguous
because 2 parse
trees for
the
same
string

\[ e \]
\[ e \]
\[ \frac{e}{e} \]
\[ \frac{e}{e} \]
\[ \frac{e}{e} \]

\[ n(1) + n(2) + n(3) \]

\[ n(1) + n(2) + n(3) \]
\[ e ::= n \mid e - e \]

1 - 2 - 3

- right associative

\[ e ::= n \mid e - n \]

\[ e \]

\[ e \]

\[ e \]

\[ n(1) - n(2) - n(3) \]

\[ e \]

- left associative

\[ e \]

\[ e \]

\[ e \]

\[ e \]

\[ n(1) - n(2) - n(3) \]

left recursion for left associativity
\[ e ::= n | e + n | e * n \]

\[ [2+3]+4 \quad [2+3]*5 \quad \text{not in the language} \]

\[ [([6 * 2]+3)*5 \quad \times \text{has higher precedence than} \quad + \]

\[ e ::= n \mid e + e \mid e * e \]