Programming Languages: Syntax

Sudarshan S. Chawathe

2025-09-15

School of Computing and Information Science
& Climate Change Institute
University of Maine

Announcements and Reminders

- Sound and visuals check.
- Main online resource: Class Web site:
  - http://chaw.eip10.org/cos301/
  - also linked from my Web page, etc.
  - Includes (is) syllabus.
- Brightspace for some things only.
  - discussion forum.
  - homework and other submissions.
- Homework HW01 due soon, not trivial.
- Rules for use of Gen AI.

Plan for today

- Homework HW01 Q&A.
- (Continuing) Material from Chapter 2 of the textbook.
  - (Review) *[E]BNF*, *CFG*.
  - Theory of CFGs etc.
  - *Parse trees* and *abstract syntax trees (AST)*.
- (Review) Bigger picture question (related to homework):
  - How to implement a simple language like:
    - x = 5 + 3
    - y = 48 / (4 * 4)
    - z = x + 2 * y
    - etc.

(Review) Terminals and nonterminals

- *terminals* or *tokens*
  - $\approx$ granules of the program source that are not analyzed internally by the CFG
    - but may be analyzed internally by the *lexer*.
  - examples
    - $=$
    - ;
    - avonum
    - 6.022E23
- *nonterminals* or *syntactic categories*
  - have components that are specified and analyzed by a CFG
  - examples
    - *assignment statement*: $\text{avonum} = 6.023;$
    - *return statement*: $\text{return } 42;$
    - *predicate*: $\text{avonum} > 42$

(Review) BNF: Backus-Naur Form(at)

- BNF spec = *set* of rules
  - N.B.: above spec is in a meta meta language.
- Each rule has the form:
  - *nonterminal* ::= sequence of *terminals* and *nonterminals*
    - again a meta-meta-language spec.
    - *::=* = "is" or "is composed of" or "can be replaced by"
- Examples
  - *assignment-statement* ::= *variable-name assignment-operator rval ;* ;
    - meta-language (BNF): ::=, ;
    - language: (C-like): *;*
  - *assignment-operator* ::= = ;
  - *statements* ::= *statement statements* | ;
    - | is short-hand for multiple rules with same LHS.

(Review) BNF example

- from textbook

<primitive-type> ::= boolean | char | byte | short | int | long | float | ...
<argument-list> ::= <expression> | <argument-list> , <expression>
<selection-statement> ::= if (<expression>) <statement>
^^I^^I| if (<expression>) <statement> else <statement>
^^I^^I| switch (<expression>) <block>
^^I^^I;

- Exercises
  - For each component above: Is it in language or meta-language?
  - Describe in English as precisely as possible.
  - Provide illustrative examples (in the *language*) making reasonable assumptions.

(Review) EBNF = Extended BNF

- BNF + some convenience features
- foo? or [foo] = optional foo
  - exercise: language v. metalanguage elements above
- foo* or {foo} = a sequence of zero or more foo
- foo+ = a sequence of one or more foo
- parentheses (in metalanguage; language may have them too!)
  - (foo bar)+ = sequence of one or more instances of foo bar
  - (*(* *)*)+ = sequence of one or more *()*.
    - !!

(Review) Discussion from end of previous class

- $(\mathrm{a\ b})* \stackrel{?}{\equiv} (\mathrm{a\ b})+?$
- $(\mathrm{a\ b})+? \stackrel{?}{\equiv} (\mathrm{a\ b})?+$
- etc.
- Elements of language, metalanguage, meta-metalanguage.
- Practice proving and disproving such statements.

CFG formally

- $G = (N, T, P, S)$
  - N: a set of symbols (*nonterminals*)
  - T: another set of symbols (*terminals*)
    - $N \cap T = \emptyset$
  - P: set of *productions*
    - each of form $n \rightarrow \alpha$ where
    - where $n \in N$ and $\alpha \in \{N \cup T\}^*$
  - $S \in N$: special nonterminal called *start*

CFG for infix expressions

- example from the textbook

Derivations

- Main question: Can a given string of tokens (terminals) be generated by a given grammar?
  - If so, how? Show the steps starting with the start symbol.
  - Is the derivation unique?
  - Is the leftmost derivation unique?
- Switch to example and practice in textbook.

Parse Trees

- Informally, a tree that has S as root and the children of each node are the items on the RHS of the rule that was used to replace the corresponding nonterminal. (Leaves correspond to terminals.)
- Example for infix expressions.
- Switch to practice problems.

Abstract Syntax Trees

- closely related to, but different from, parse trees.
- abstract away unimportant details such as order in which a sequence of nonterminals is expanded.
- Two main changes:
  - Nonterminal nodes are replaced by *corresponding* parts of the input sentence.
  - Unit productions are collapsed.
- Example for infix expressions.
- Practice problems in textbook.

Summary

- (E)BNF, CFGs, derivations, parse trees, ASTs.
- Bigger picture:
  - Process source code into AST
  - then we can interpret or compile it etc.
- Class Web site: http://chaw.eip10.org/cos301/