

CSSE 304 Day 07 Summary

1. After class: Look over the solutions for matrix-transpose, minimize-interval-list, pascal-triangle, and largest-in-lists. Write down questions that you have about them and ask someone (possibly me) sometime.

2. Write `(all-positive? lon)` using `map`. Why is this hard? How do we get around this?

3. What do we mean by the *syntax* of a programming language?

What do we mean by the *semantics* of a programming language?

The left-hand side of a context-free (BNF) grammar production always contains a single _____ symbol.

A string that is in the language of a grammar consists of (circle one):

terminals only nonterminals only both terminals and nonterminals

4. A **BNF Grammar** (a.k.a. Context-free grammar) example:

- a. **Nonterminals:** `<exp>` `<term>` `<factor>` `<number>` `<digit>`
- b. **Terminals:** `+` `*` `)` `(` `0` `1` `2` `3` `4` `5` `6` `7` `8` `9`
- c. **Start Symbol:** `<exp>`
- d. **Productions:**
 - i. `<exp>` ::= `<exp> + <term>` | `<term>`
 - ii. `<term>` ::= `<term> * <factor>` | `<factor>`
 - iii. `<factor>` ::= `(<exp>)` | `<number>`
 - iv. `<number>` ::= `<number> <digit>` | `<digit>`
 - v. `<digit>` ::= `0` | `1` | `2` | `3` | `4` | `5` | `6` | `7` | `8` | `9`

5. Show a derivation and a derivation tree for `1 * (2 + 34)` from `<exp>` [you will need to write small]

What is the meaning of Kleene * ?

Kleene + ?

6. The slides contain (all in one place, the **collection of grammars** used in chapter 1 of EoPL.

- e. $\langle \text{list-of-numbers} \rangle ::= (\{ \langle \text{number} \rangle \}^*)$
- f. $\langle \text{s-list} \rangle ::= (\{ \langle \text{symbol-exp} \rangle \}^*)$
 $\langle \text{symbol-exp} \rangle ::= \langle \text{symbol} \rangle \mid \langle \text{s-list} \rangle$
- g. $\langle \text{bintree} \rangle ::= \langle \text{number} \rangle \mid (\langle \text{symbol} \rangle \langle \text{bintree} \rangle \langle \text{bintree} \rangle)$
- h. $\langle \text{BST} \rangle ::= () \mid (\langle \text{number} \rangle \langle \text{BST} \rangle \langle \text{BST} \rangle)$
- i. $\langle \text{datum} \rangle ::= \langle \text{number} \rangle \mid \langle \text{symbol} \rangle \mid \langle \text{string} \rangle \mid \langle \text{boolean} \rangle \mid \langle \text{dotted-datum} \rangle \mid \langle \text{list} \rangle \mid \langle \text{vector} \rangle$
 $\langle \text{list} \rangle ::= (\{ \langle \text{datum} \rangle \}^*)$
 $\langle \text{dotted-datum} \rangle ::= (\{ \langle \text{datum} \rangle \}^+ . \langle \text{datum} \rangle)$
 $\langle \text{vector} \rangle ::= \# \langle \text{list} \rangle$
- j. $\langle \text{LcExp} \rangle ::= \text{Identifier} \mid$; variable reference
 $(\text{lambda } (\text{Identifier}) \langle \text{LcExp} \rangle) \mid$; abstraction
 $(\langle \text{LcExp} \rangle \langle \text{LcExp} \rangle) \mid$; application

7. The s-list grammar can also be written as $\langle \text{s-list} \rangle ::= () \mid (\langle \text{s-exp} \rangle . \langle \text{s-list} \rangle)$
 $\langle \text{s-exp} \rangle ::= \langle \text{symbol} \rangle \mid \langle \text{s-list} \rangle$

Examples of s-lists:

8. (probably tomorrow) we will write and test some of these procedures:

- a. contains?
- b. count-occurrences
- c. notate-depth
- d. flatten
- e. subst