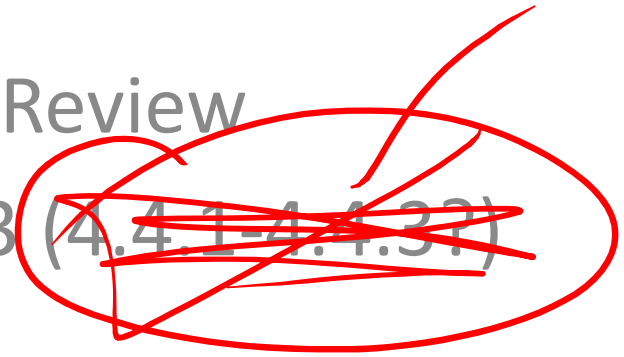
CS480
Translators

Test Review

Chap. 1 - 4.3 (4.4.1-4.4.3?)



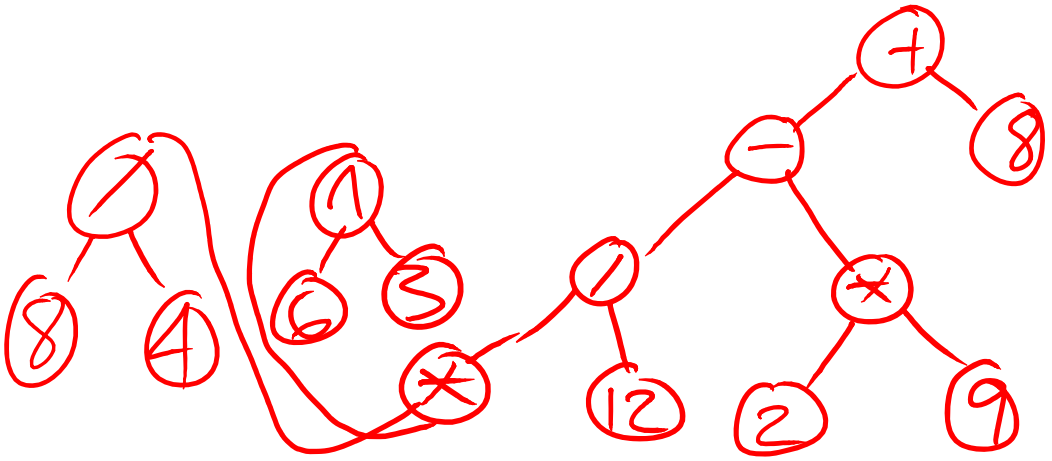
Test 1 Review

- Characteristics common to all translators
- Construct an expression tree
- Polish and Reverse polish notation
- Construct a DFA from a regular expression
- Change a NFA to a DFA w/ subset construction
- Syntax vs. Semantics
- Why would a lexical analyzer peek at a character
- Purpose of Symbol Table(s)
- The purpose of a token vs. a lexeme
- What is the purpose of Lex
- Transforming a Grammar
- Ambiguity and Derivations

Expression Tree Example

- Provide the expression tree, and provide the polish and reverse polish notation.

8 / 4 * 6 ^ 3 / 12 – 2 * 9 + 8



Example Question for DFAs

- Suppose comments are defined by a two character sequence, a slash followed by a star, as in the C convention of /* whatever */. Comments can contain any text, including the star symbol and slash, just not the two character sequence that ends the comment. So /***/ is a legal comment, but /*/ is not. Write a DFA recognizing legal comments.

Example Question for NFA to DFA

- Using subset construction, create a DFA from this NFA

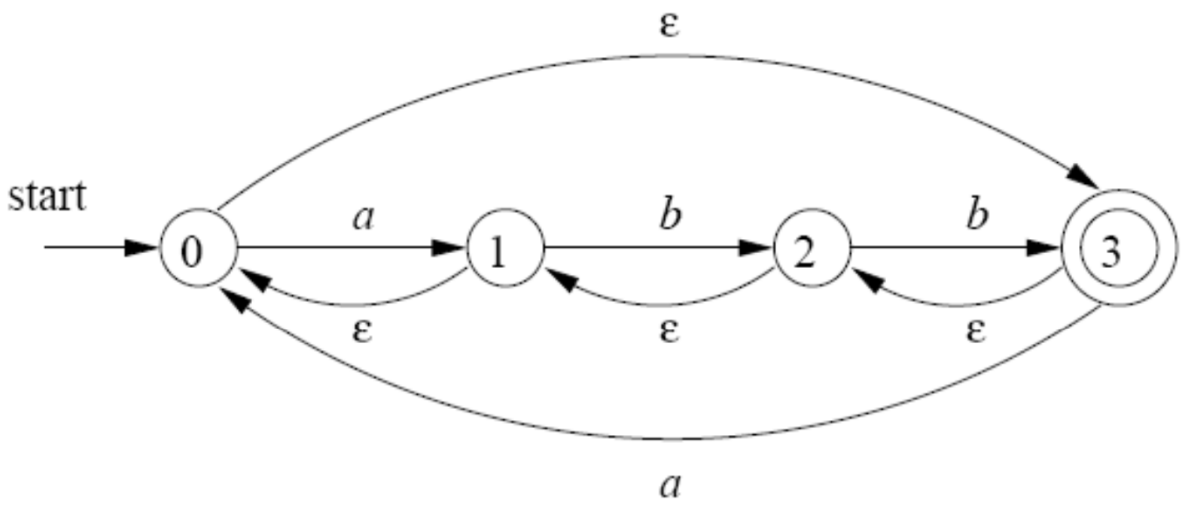


Figure 3.30: NFA for Exercise 3.6.4

Quiz #4

- Consider *S* -> *SS*+ | *SS** | **a** and string aa+a*
  - Give a leftmost derivation
  - Give a rightmost derivation
  - Is the grammar ambiguous or unambiguous, why?
- For each grammar below, left factor and/or remove left recursion first.
  - (a) *S* -> *SS*+ | *SS** | **a**
  - (b) *S* -> 0 *S* 1 | 0 1
  - (c) *S* -> *S* (*S*) *S* | ε