

**Concepts of Programming Languages, CSCI 305, Fall 2021**  
**Translation of Regular Expressions to a DFA, Sept. 8**  
 Section 2.2.1 Generating a Finite Automaton, 56-61

Example from text, starts on page 54.

Regular expressions for a simple calculator language

- assign  $\rightarrow :=$
- plus  $\rightarrow +$
- minus  $\rightarrow -$
- times  $\rightarrow *$
- div  $\rightarrow /$
- lparen  $\rightarrow ($
- rpren  $\rightarrow )$
- id  $\rightarrow \text{letter}(\text{letter} \mid \text{digit})^*$
- number  $\rightarrow \text{digit} \text{digit}^* \mid \text{digit}^* ( \text{digit} \mid \text{digit} . ) \text{digit}^*$

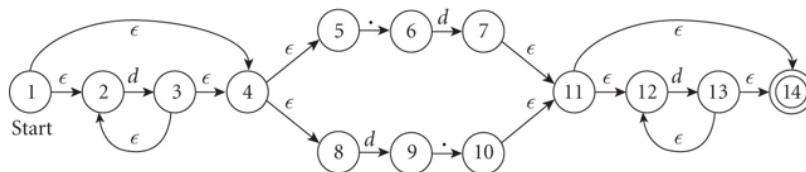
Sample “programs” in this language:

$x1 := x1 * (29.3+x2)$

Consider a portion of the definition for number:

number  $\rightarrow \text{digit}^* ( \text{digit} \mid \text{digit} . ) \text{digit}^*$

1. Convert this regular expression into an NFA



2. Convert the NFA to a DFA

