Chapter 3 Review & Exercises
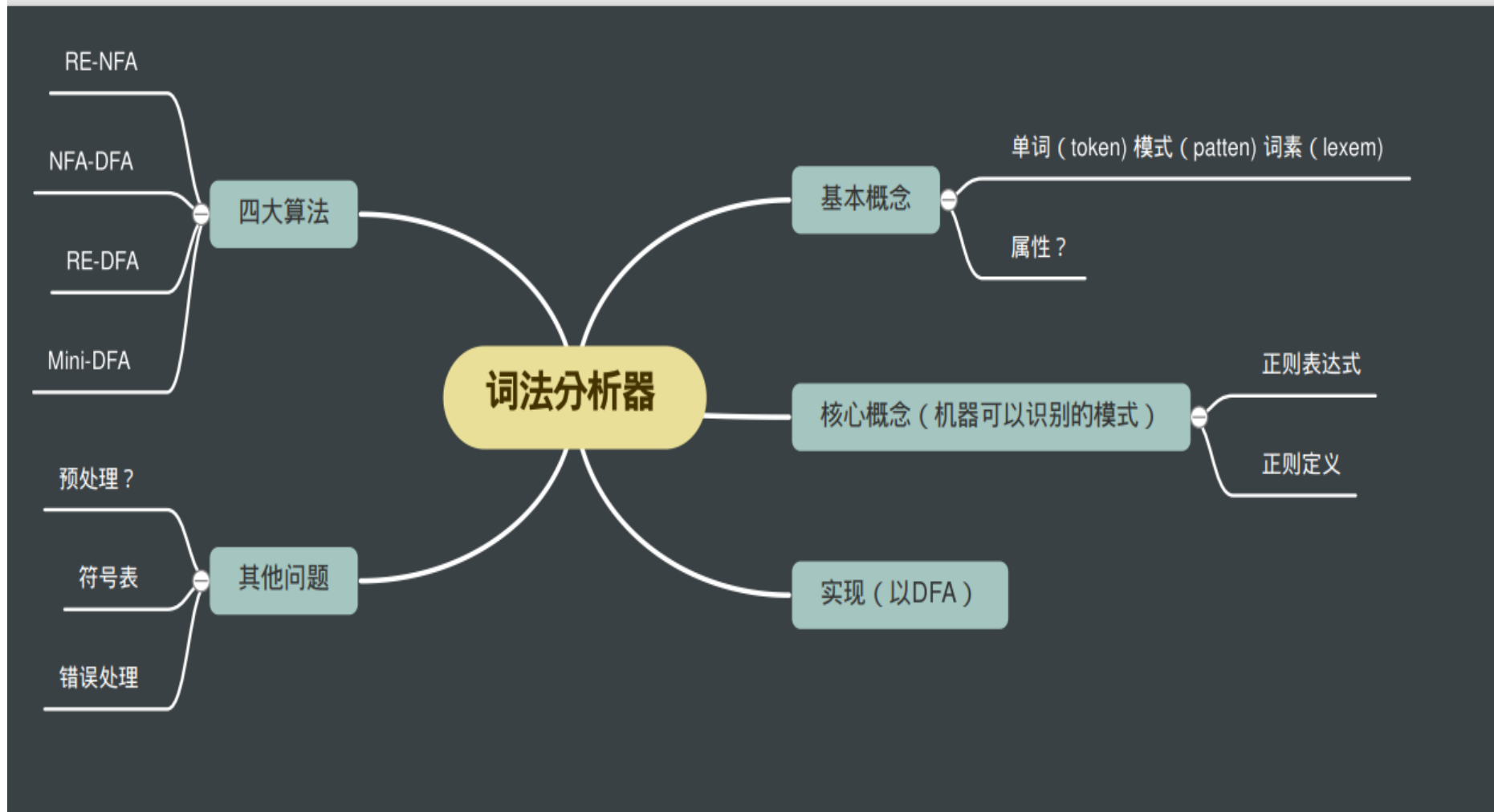
Review

Exercise in class

1. Given:∑={ a,b,c}, describing the set of all strings that contain at most one b with regular expression.

2. Example 3.5:

     Given: letter → A | B | ... | Z | a | b | ... | z

     digit → 0 | 1 | ... | 9

  Describing identifiers and numbers in Pascal or C with regular expression.

Exercise in class (cont.)

◆ EX: ∑={ a,b,c}

the set of all strings that contain at most one b.

(a|c)*(b|ε)(a|c)*   **(a|c)*|(a|c)*b(a|c)***

the same language may be generated by many different regular expressions.

◆ Ex: Identifiers in Pascal or C

letter → A | B | ... | Z | a | b | ... | z

digit → 0 | 1 | ... | 9

id → letter (letter | digit) *

If we try to write the regular expression representing identifiers without using regular definitions, that regular expression will be complex.

(A|...|Z|a|...|z) **(** (A|...|Z|a|...|z) | (0|...|9) **)** *

Exercise in class (cont.)

Ex: Unsigned numbers in Pascal or C

digit → 0 | 1 | ... | 9

digits → digit $^+$

opt-fraction → (. digits) ?

opt-exponent → (E (+|-) ? digits) ?

unsigned-num → digits opt-fraction opt-exponent

Exercise in class

Using Thompson's construction to construct NFA from regular expression.

$$a(a|b)^+(b|c)(c|\in)$$
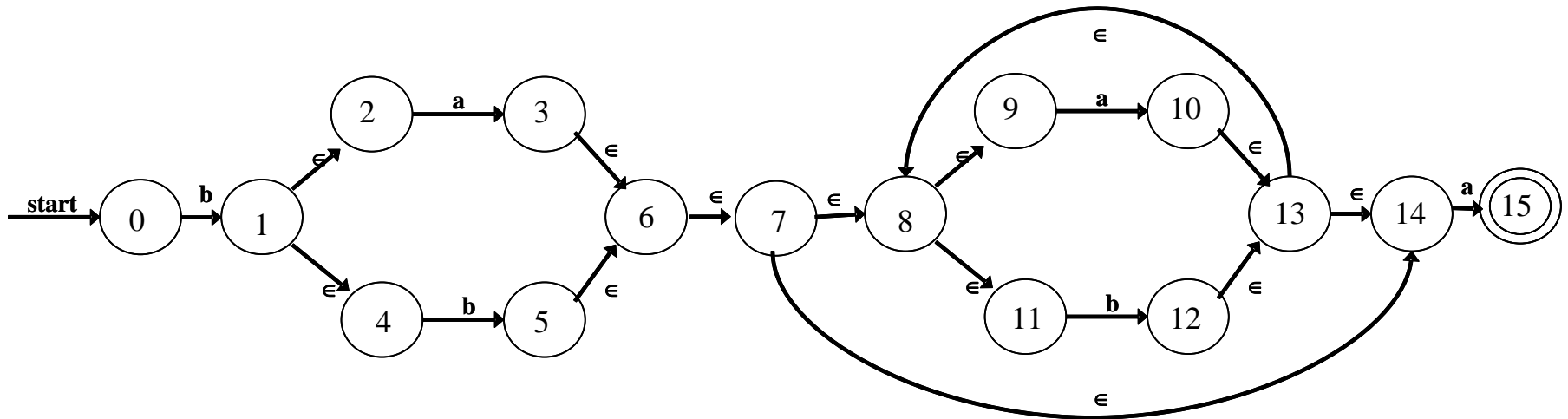
Another exercise

◆Using subset construction to construct DFA from NFA.

b(a|b)⁺a

Exercises in class

◆Construct **DFA** for the following regular expression in two ways:

$$1. \quad a(a|b)^*(b|\epsilon)$$

$$2. \quad a(a|b)^*a(a|b)a$$

➢The first way is to construct **NFA** by using Thompson's algorithm , then to construct **DFA** from **NFA** by using subset construction algorithm .

➢The second way is to construct **DFA** from regular expression directly.

➢Minimizing the states of **DFA**.