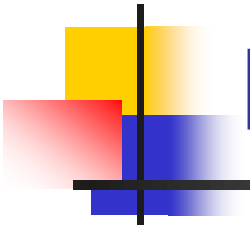
CSC 2204 Finite Automata Theory and Formal Languages

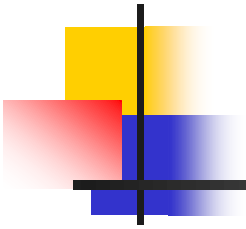
Department of Computer Science
SZABIST (Islamabad Campus)

Week 5 (Lecture 1)



FA and REs

While the regular-expression approach to describing languages is fundamentally different from the finite-automaton approach, these two notations turn out to represent exactly the same set of languages, which we have termed the "regular languages."



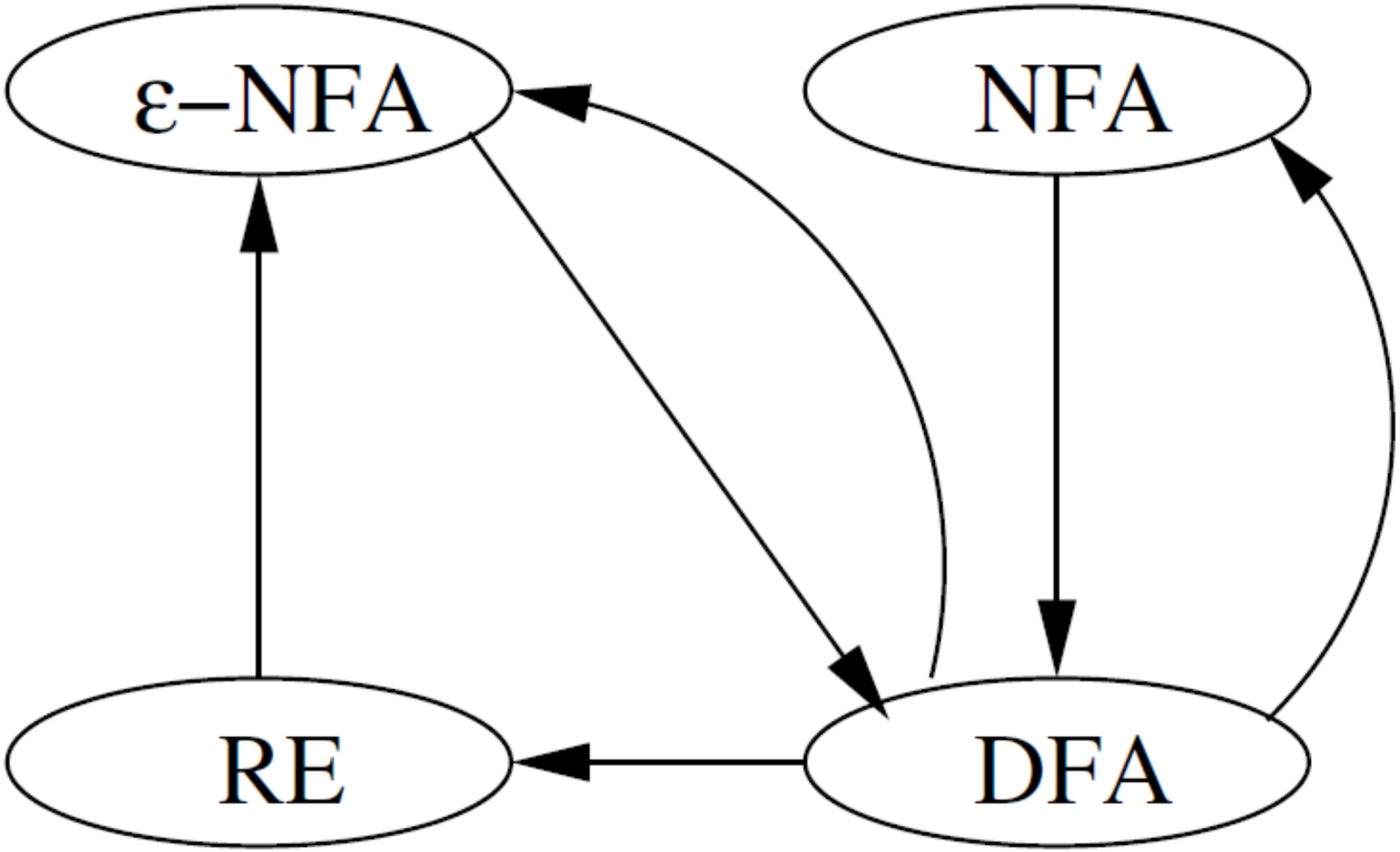
FA and REs

In order to show that the regular expressions define the same class, we must show that:

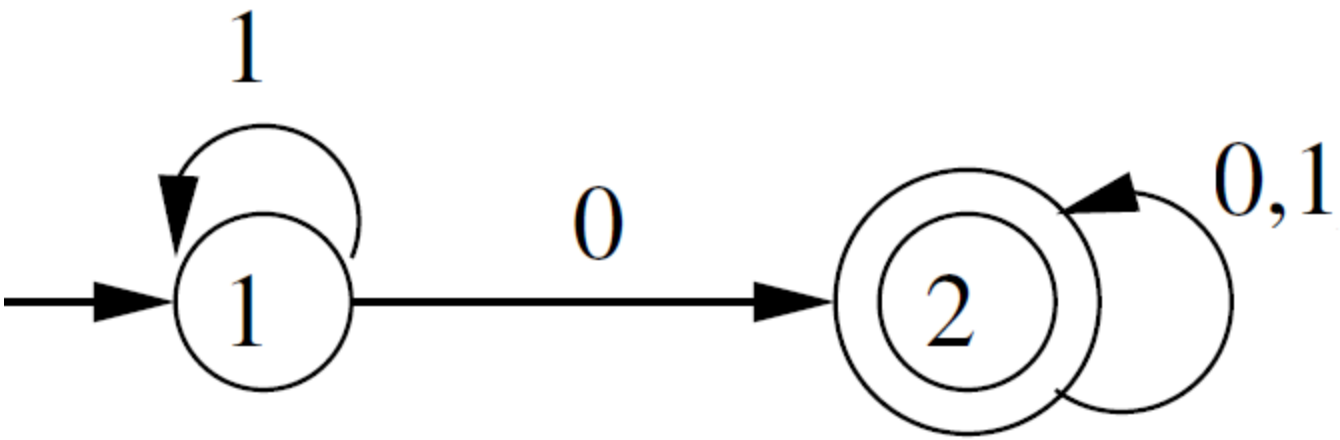
1. Every language defined by one of these automata is also defined by a regular expression. For this proof, we can assume the language is accepted by some DFA.

2. Every language defined by a regular expression is defined by one of these automata. For this part of the proof, the easiest is to show that there is an NFA with ϵ-transitions accepting the same language.

Equivalence

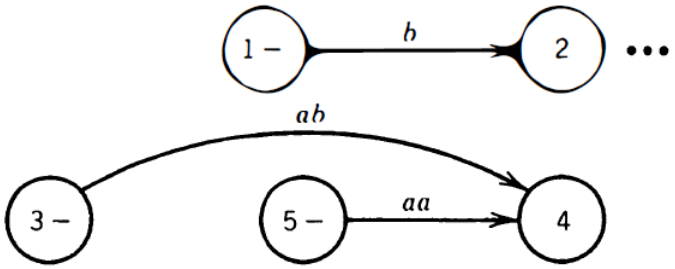


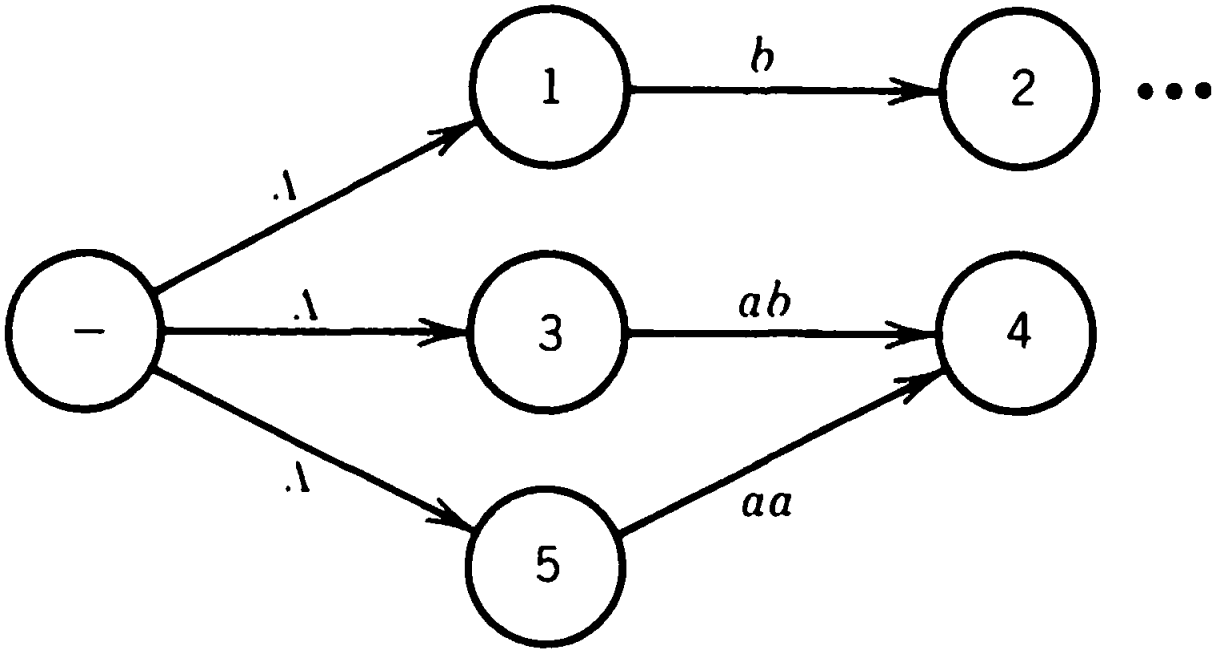
DFA to RE (Method 1)



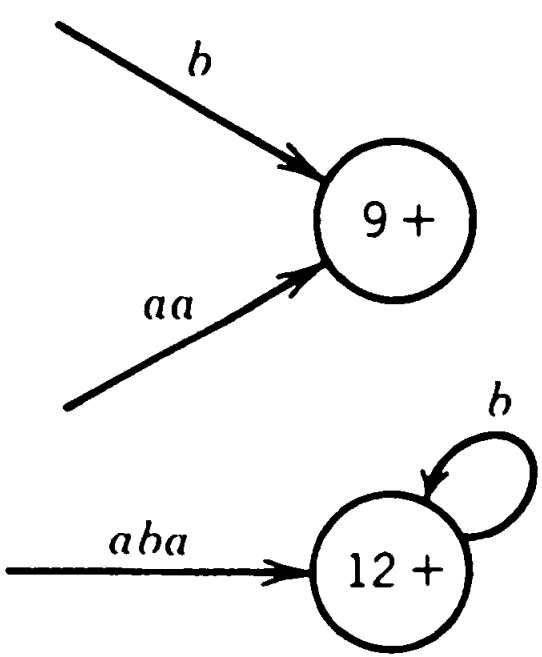
1*0(0+1)*

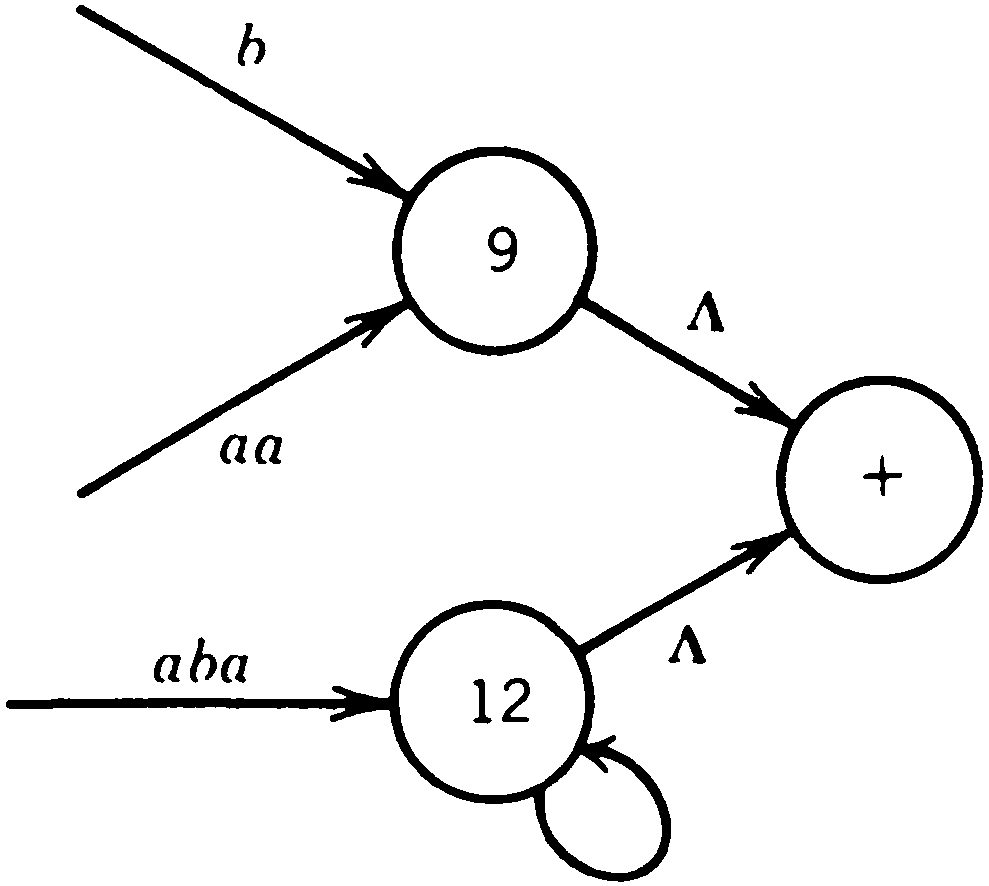
DFA to RE (Method 2)



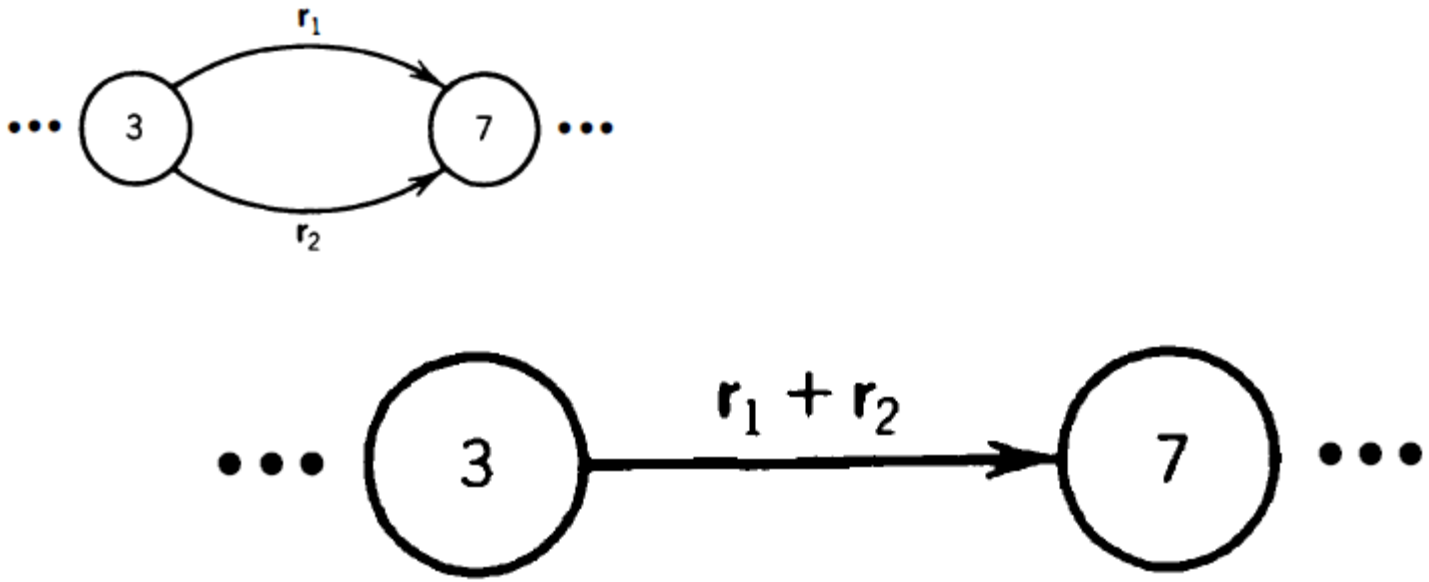


DFA to RE (Method 2)

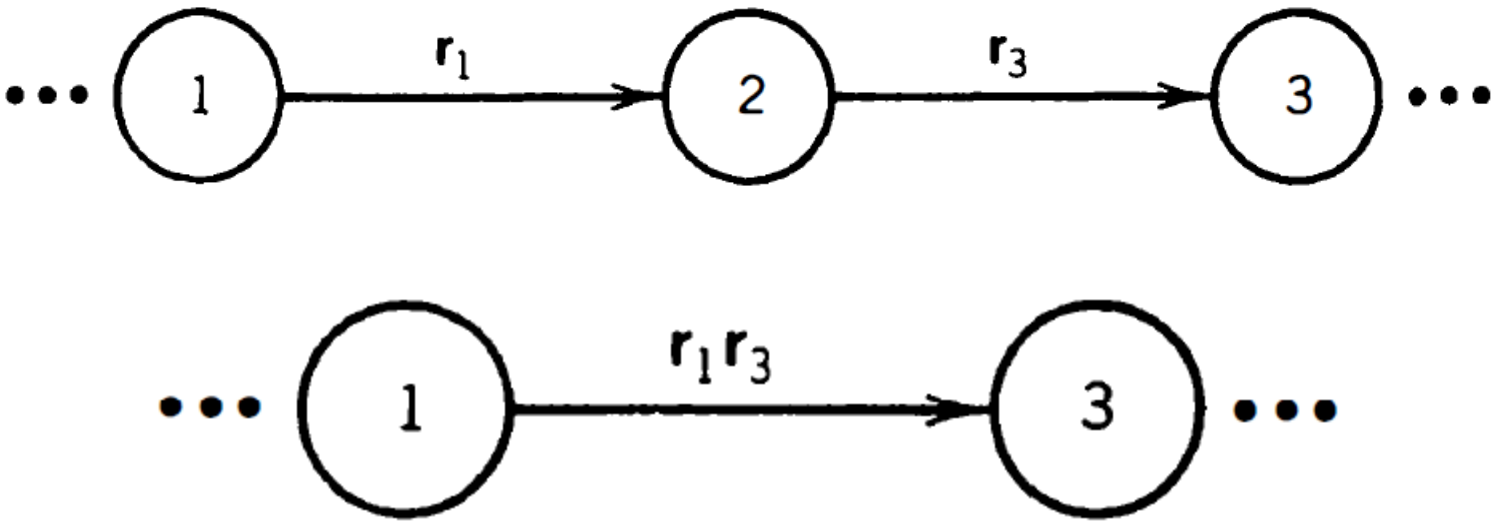




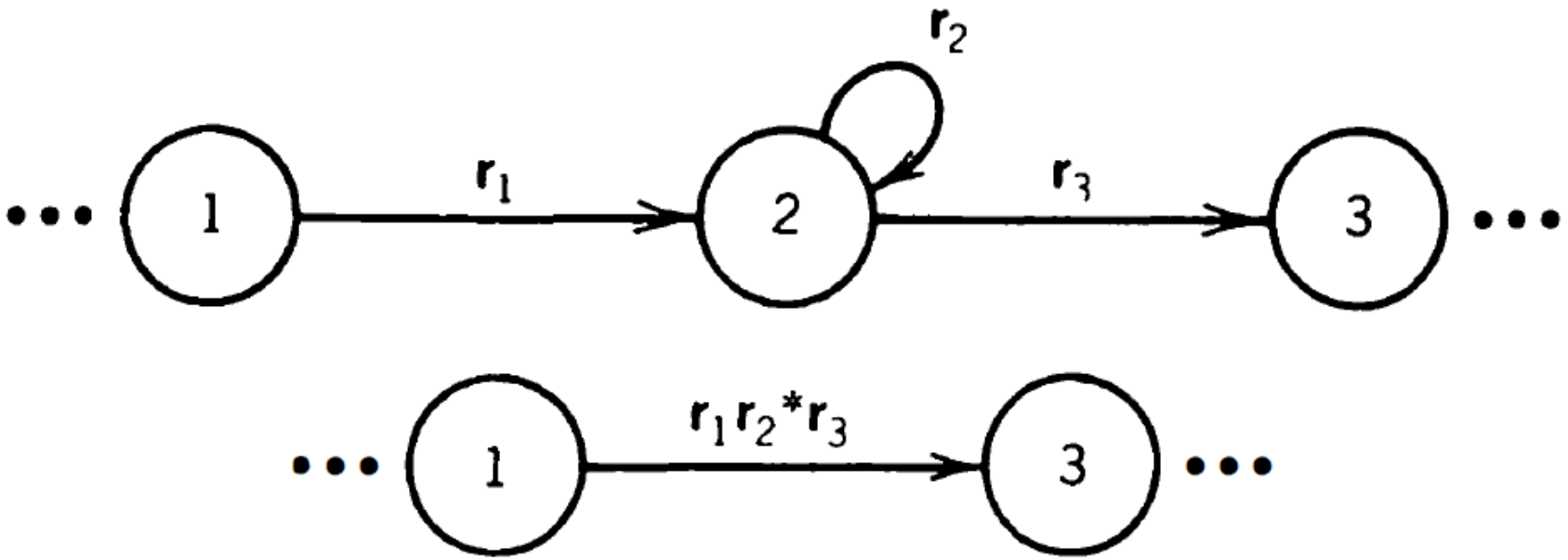
DFA to RE (Method 2)



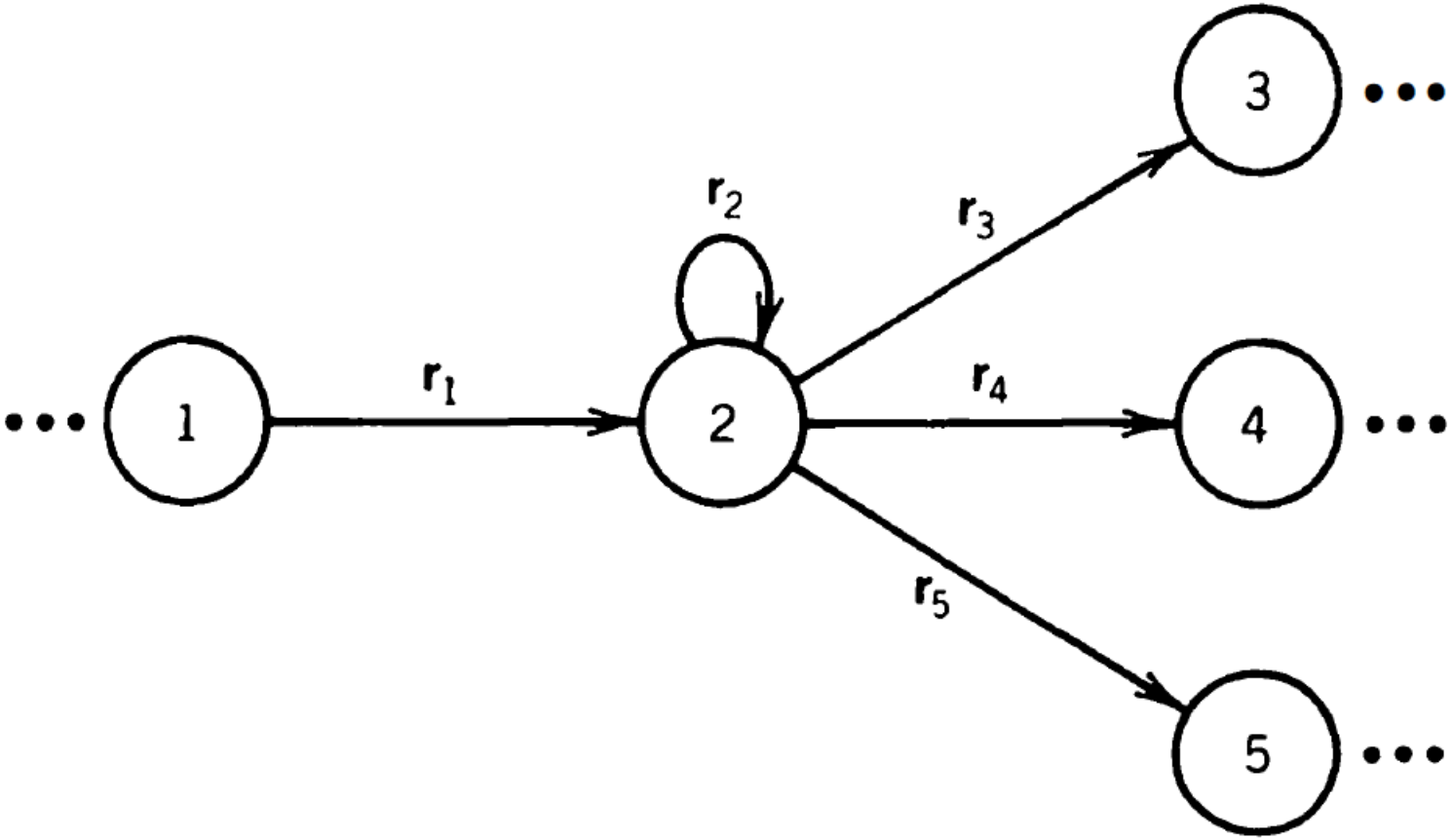
DFA to RE (Method 2)



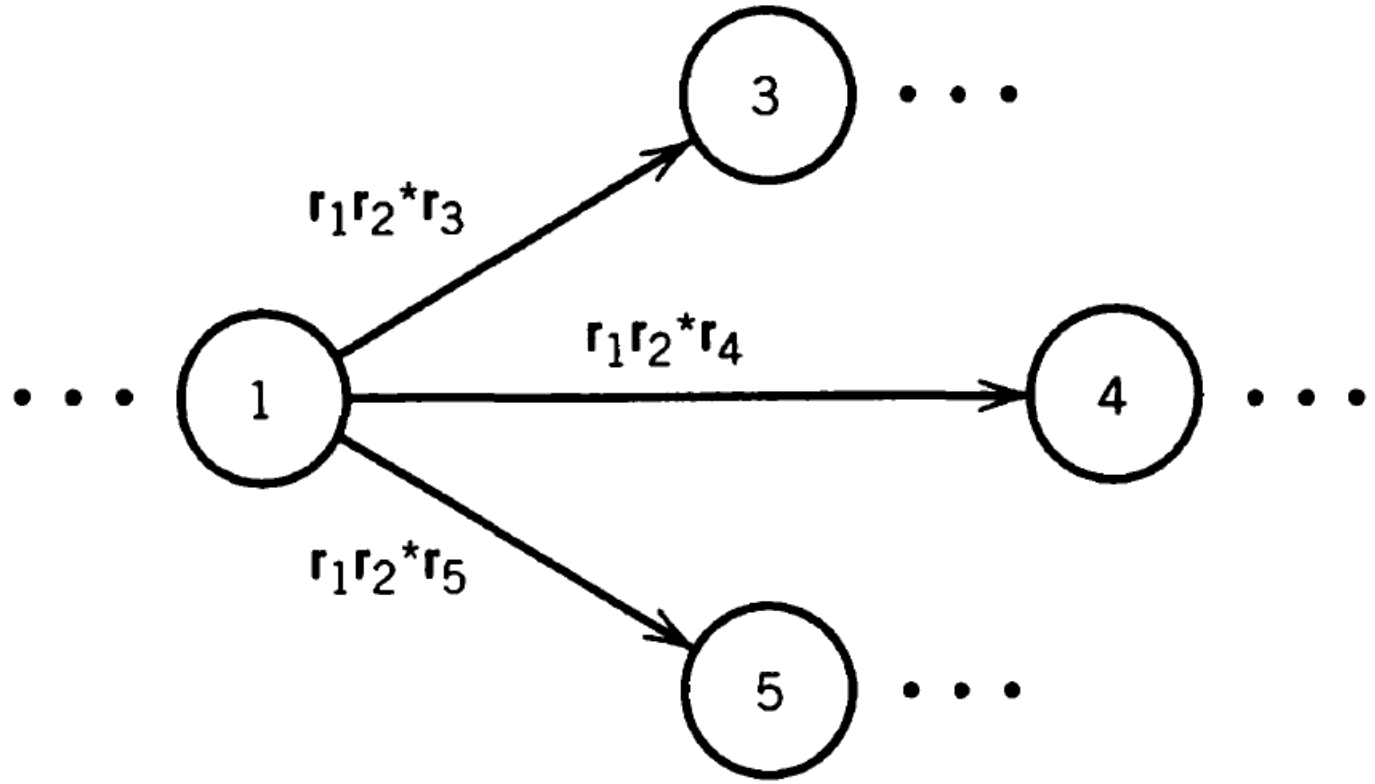
DFA to RE (Method 2)



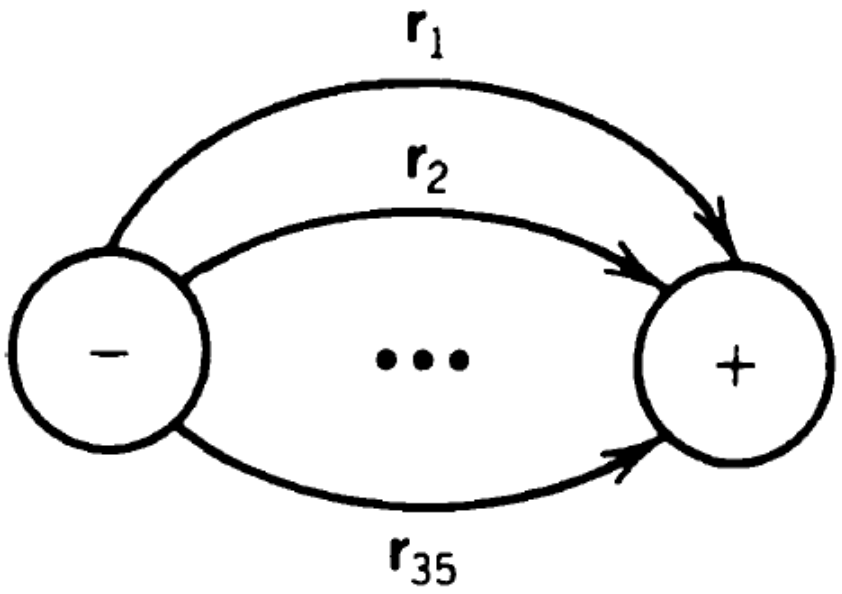
DFA to RE (Method 2)

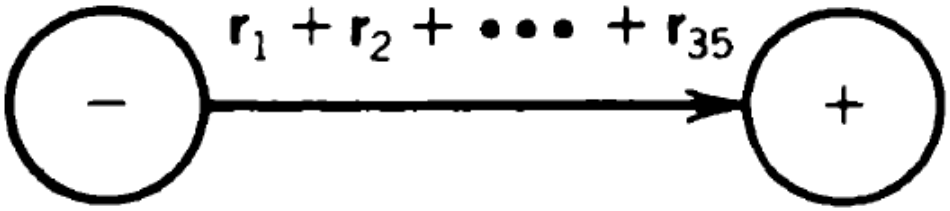


DFA to RE (Method 2)

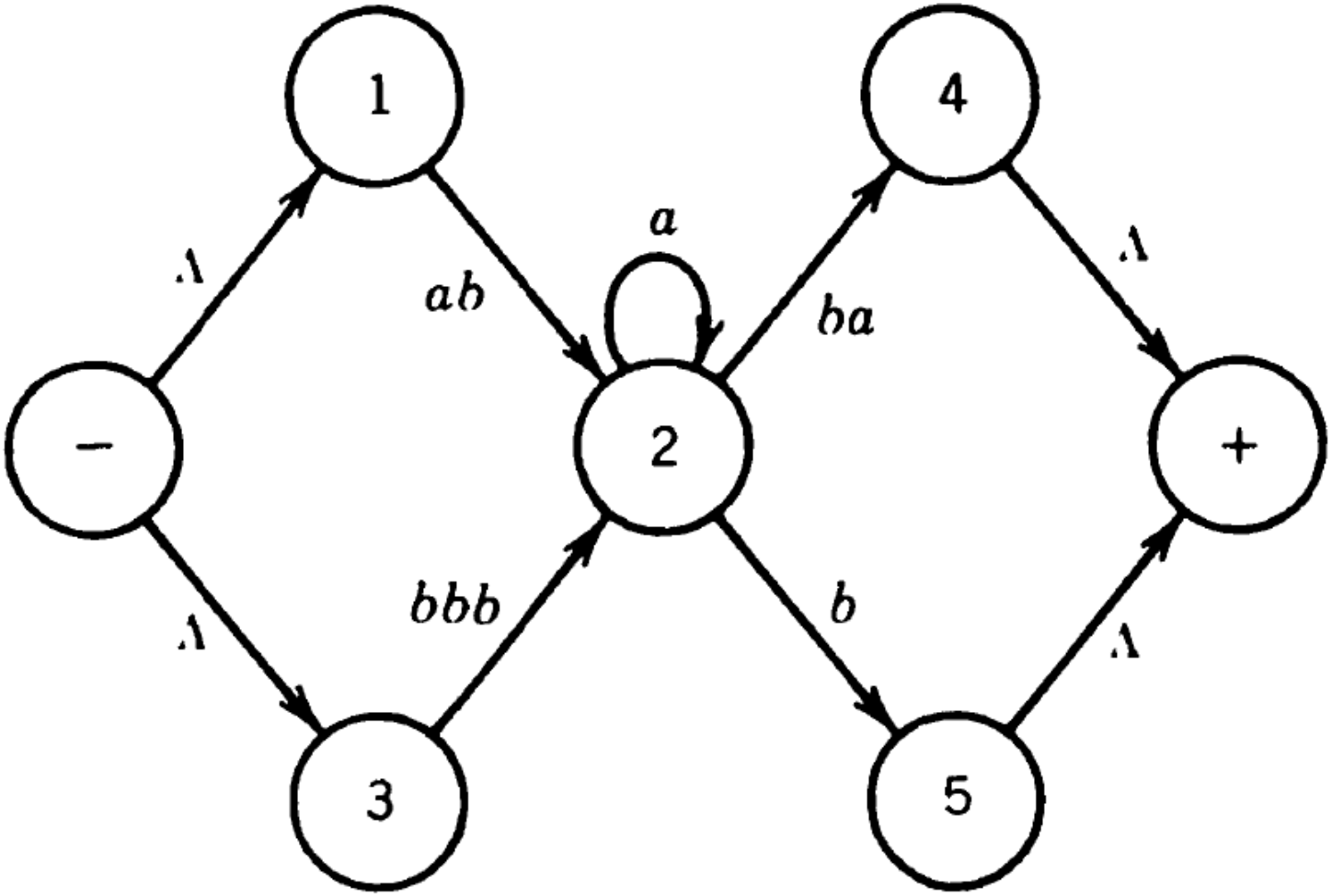


DFA to RE (Method 2)

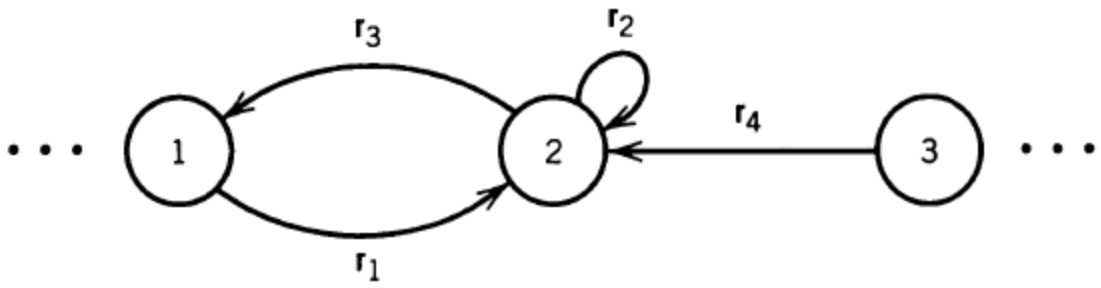




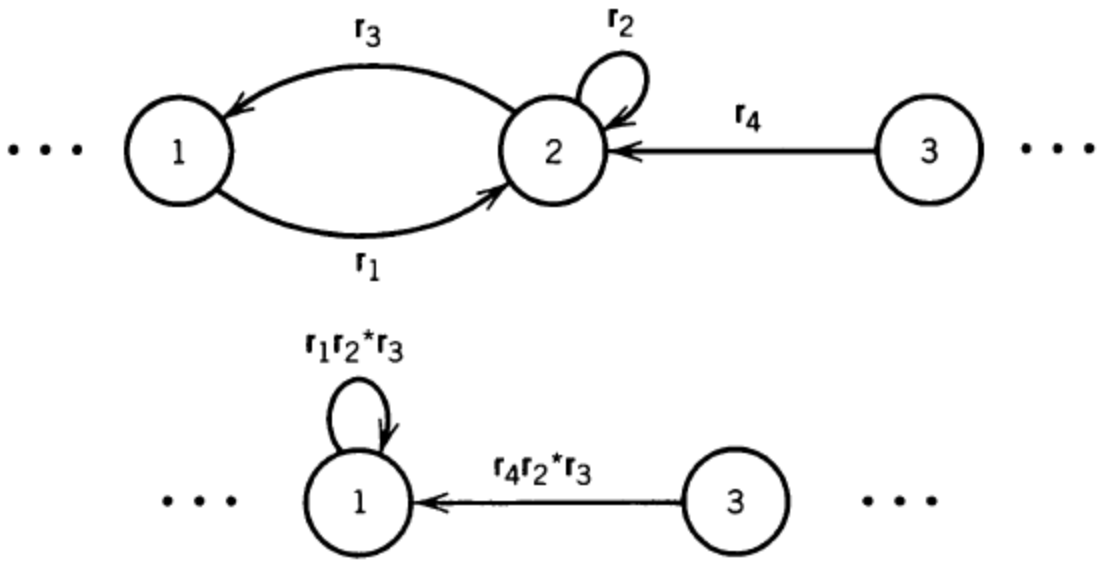
DFA to RE (Method 2)



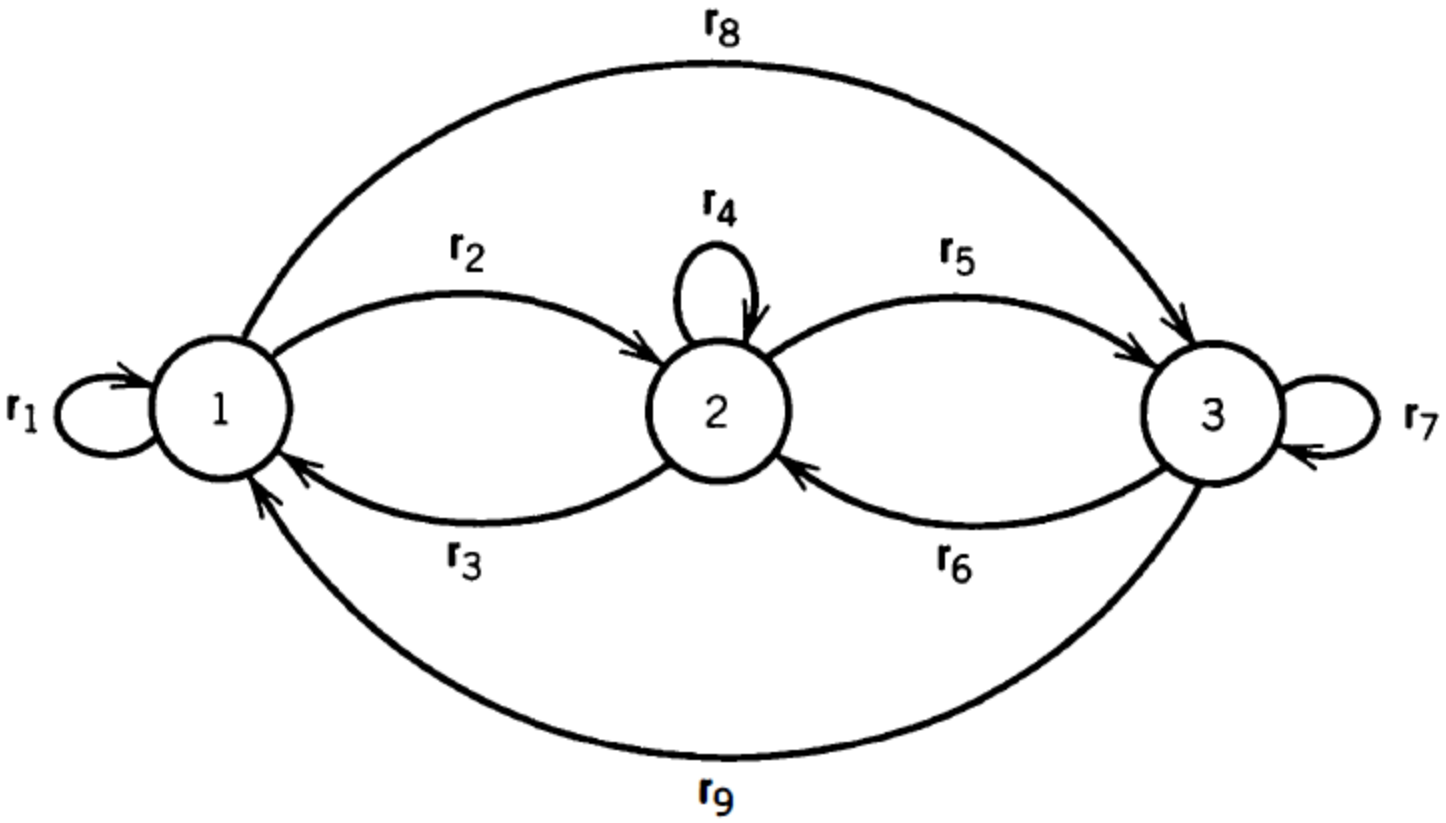
DFA to RE (Method 2)



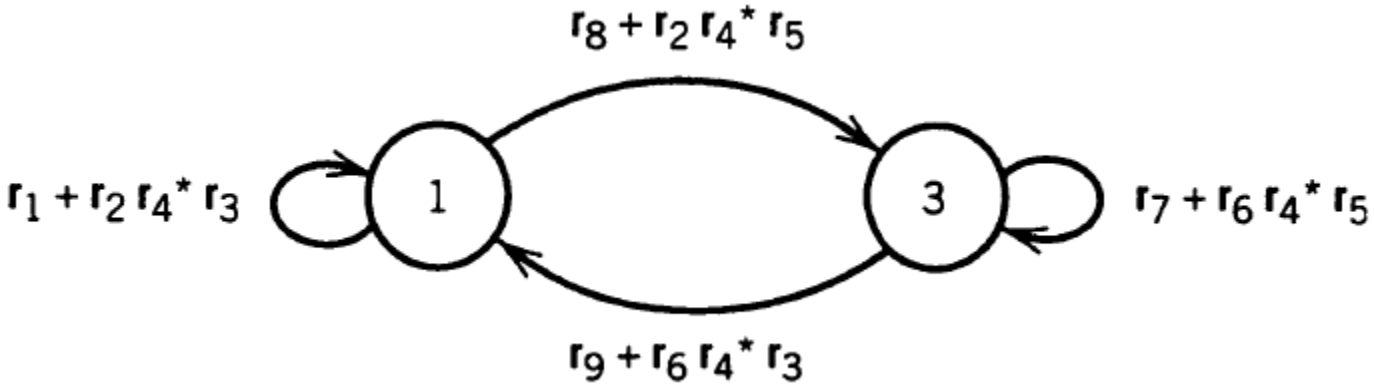
DFA to RE (Method 2)

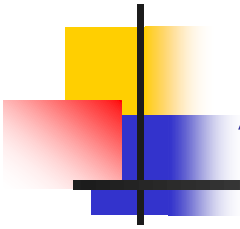


DFA to RE (Method 2)



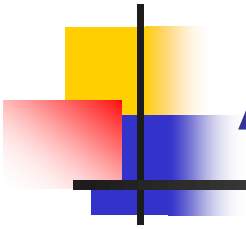
DFA to RE (Method 2)





Applications of REs

- Unix allows to write character classes.
  - Rules:
    - The symbol . (dot) stands for “any character”
    - [abc] means the RE a+b+c
    - Ranges: [A-Z][A-Za-z0-9]
  - Special notations for most common classes:
    - [:digit:] means [0-9]
    - [:alpha:] means [A-Za-z]
    - [:alnum:] means [A-Za-z0-9]
  - Several operators such as | ? + etc.



Applications of REs

- Lexical Analyzers
- Finding patterns in text