Regular Grammars

CS 712
Dept. of Computer Science
Univ. of New Hampshire

A grammar is <u>regular</u> if each production takes one of the following forms, where uppercase letters are nonterminals and w is a nonempty string of terminals:

$$N \to \lambda$$

$$N \to w$$

$$N \to T$$

$$N \to wT$$

i.e. only one nonterminal can appear on the right side of a production, and it must appear at the right end of the right side.

Theorem Regular grammars generate exactly the set of regular languages.

Proof by construction

1) regular grammar to NFA

2) NFA to regular grammar

Key idea nonterminals $\longleftrightarrow$ states

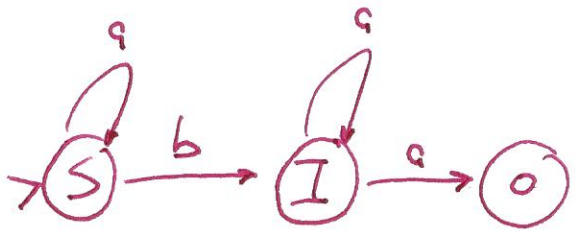
example regular grammar → NFA

$S \to aS$

$S \to bI$

$I \to a$

$I \to aI$



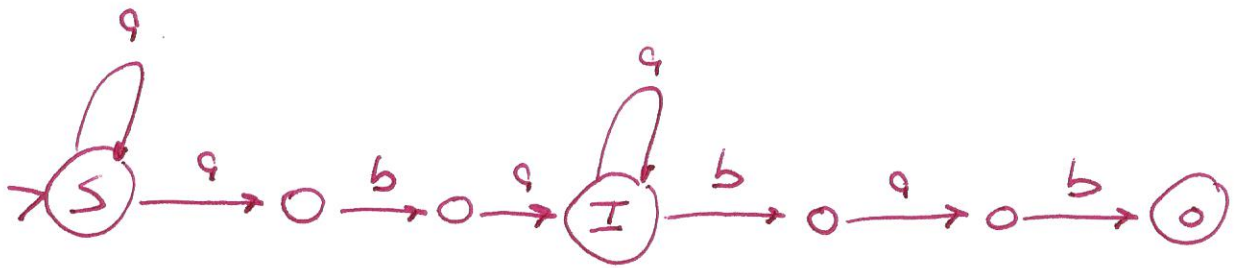
example regular grammar → NFA

$S \to a\,S$

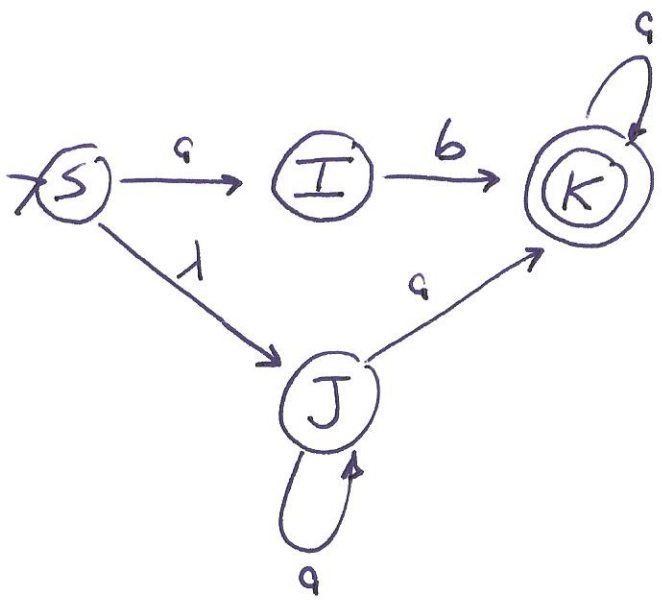
$S \to a\,b\,a\,I$

$I \to b\,a\,b$

$I \to a\,I$



example NFA → regular grammar



$S \to aI$

$S \to J$

$I \to bK$

$J \to aJ$

$J \to aK$

$K \to \lambda$

$K \to aK$

Key model final state with null rule

$S \to aI \to abK \to abaK \to aba$