Equivalence of Finite State Automata and Regular Grammar

- FSA accepts Regular Language
- Regular Grammar generates Regular Language
- We will show
- 1. Conversion of RG to FSA (NFA)
- 2. Conversion of FSA(DFA) to RG

Conversion of RG to FSA(NFA)

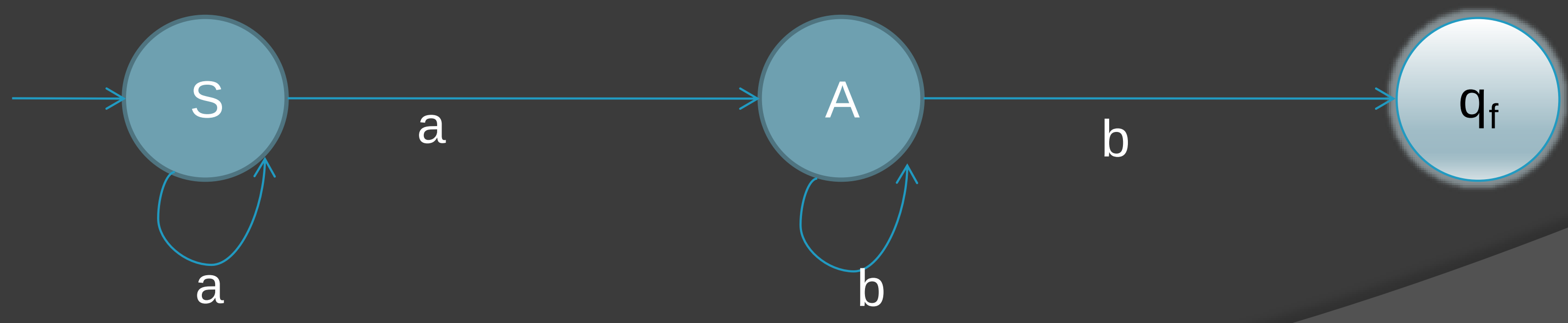
- Given RG G=(N,T,P,S) construct NFA $M=(Q,\sum,\delta,q_0,F)$
- Such that L(G)=L(M) or L(M)=L(G)
- Construction
- $Q=N \cup \{q_f\}$: q_f is new non terminal symbol for final state
- $\sum=T$
- $q_0=S$
- $F=\{q_f\}$
- Now definition of δ-
- 1. If A->aB is a rule in P the δ(A,a) contains B
- 2. If A->a is a rule in P the δ(A,a) contains q_f

Conversion of RG to FSA(NFA)

Given RG
1. S->aS
2. S->aA
3. A->bA
4. A->b

$L(G)=\{a^m b^n | m,n>=1\}$

Construct NFA M
1. $\delta(S,a)$ contains S
2. $\delta(S,a)$ contains A
3. $\delta(A,b)$ contains A
4. $\delta(A,b)$ contains q_f



$L(M)=\{a^m b^n | m,n>=1\}$