Regular Grammars

What is a regular grammar

The Regular Grammars are either left of right:

Right Regular Grammars:

Rules of the forms

$A \rightarrow \varepsilon$

$A \rightarrow a$

$A \rightarrow aB$

A,B: variables and
a: terminal

Left Regular Grammars:

Rules of the forms

$A \rightarrow \varepsilon$

$A \rightarrow a$

$A \rightarrow Ba$

A,B: variables and
A: terminal

Example

$S \rightarrow aS \mid bA$

$A \rightarrow cA \mid \varepsilon$

This grammar produces the language produced by the regular expression a^*bc^*

$S \rightarrow aS \rightarrow aaS \rightarrow \ldots \rightarrow a\ldots aS \rightarrow a\ldots abA \rightarrow a\ldots abcA$
$\rightarrow a\ldots abccA \rightarrow \ldots \rightarrow a\ldots abc\ldots c$

The Right Regular Grammars are producing the Regular Languages

Proof: We will show that Right Regular Grammars are equivalent to NFAε

Two directions:

1. Given a Right Regular grammar construct an NFAε that recognizes the same language with the Right Regular grammar.
2. Given an NFAε construct a Right Regular grammar that describes the same language with the NFAε.

1. Right Reg Grammar → NFAε

Suppose that I have a right regular grammar (V, Σ, R, S). I construct an NFAε (Q, Σ, δ, S, {f}).

- The set of states Q will be the set V∪{f}, where f is a new symbol denoting the final state
- Productions in R have three possible forms:
  - A → ε : add the transition δ(A,ε) = f
  - A → a : add the transition δ(A,a) = f
  - A → aB : add the transition δ(A,a) = B

2. NFAε → Right Regular Grammar

Suppose that I have an NFAε (Q, Σ, δ, q_0, F,). I construct a right regular grammar (Q, Σ, R, q_0).

- For each transition δ(q_i ,a) = q_j, I construct the rule $q_i \rightarrow aq_j$ in R.
- Furthermore, for every state q_i in F I add the rule $q_i \rightarrow \varepsilon$ in R.

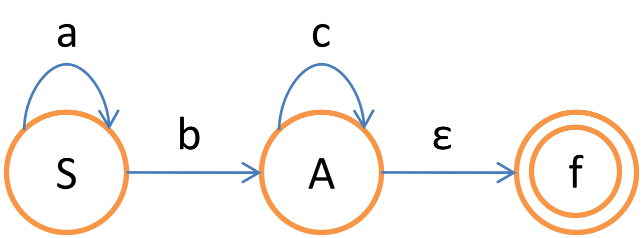
Examples

1) Transform the following Right Regular grammar in an equivalent NFAε.

$S \rightarrow aS \mid bA$

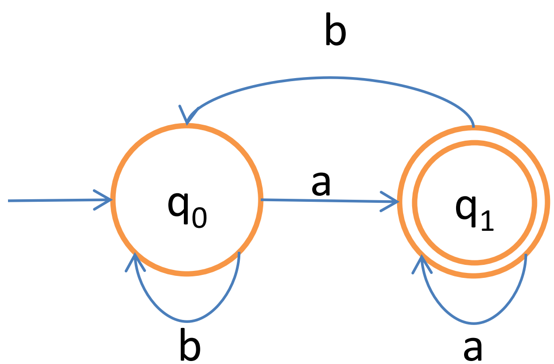
$A \rightarrow cA \mid \varepsilon$

Solution:



Examples

2) Transform the following DFA to a right regular grammar



Solution:

$Q_0 \rightarrow aQ_1 \mid bQ_0$

$Q_1 \rightarrow aQ_1 \mid bQ_0 \mid \varepsilon$

Left Regular Grammars

- It can be proved that Left Regular Grammars also produce the Regular Languages but this is not so straightforward.
- Actually, a Left Regular grammar produces the reverse of the language produced by the Right Regular grammar in which we reversed the rules $A \rightarrow Ba$ to $A \rightarrow aB$.
- But the set of the reverse languages of all the Regular Languages is exactly the set of the Regular Languages. So the Left Regular Grammars produce the Regular Languages.

Example

C → Bc

B → Ab

A → a

The derivation of abc is:

C → Bc → Abc → abc, or

abc ← Abc ← Bc ← C

So I should start creating the string abc from right to left. But this is equivalent with creating the reverse of cba.

C → cB → cbA → cba and then take the reverse.

Example (continue)

The Right Regular grammar with the rules of the form A → Ba reversed is

C → cB

B → bA

A → a

and it produces the reverse language.

So, just create the NFAε for the language produced by the Right Regular grammar and then compute the reverse (change start with final state and reverse the arrows). This is an NFAε for the Left Regular grammar.