CS 5000: Lecture 8

Vladimir Kulyukin
Department of Computer Science
Utah State University

Outline

- Regular Grammars

A Grammar: Formal Definition

A grammar G is a 4-tuple $G = (V, \Sigma, S, P)$, where

V is the nonterminal alphabet;

Σ is the terminal alphabet;

$S \in V$ is the start symbol;

P is a finite set of productions; each production is of the form $x \rightarrow y$, where x and y are strings over $\Sigma \cup V$ and $x \neq \varepsilon$.

$\Sigma \cap V = \emptyset$, i.e. Σ and V are disjoint.

Example

Consider the language defined by $a^* b^*$.

$$S \rightarrow aS \mid X$$
$$X \rightarrow bX \mid \varepsilon.$$

$$G = (\{S, X\}, \{a, b\}, S, \{S \rightarrow aS, S \rightarrow X, X \rightarrow bX, X \rightarrow \varepsilon\}).$$

Derivation

$w \Rightarrow z$ (w derives z) if and only if

1) there exist strings $u, x, y, v \in (\Sigma \cup V)^*, x \neq \varepsilon$,

2) $w = uxv, z = uyv$, and $(x \rightarrow y) \in P$.

$w \Rightarrow^* z$ if and only if there is a derivation of zero or more steps that starts with w and ends with z.

The Language Generated by a Grammar

$$L(G) = \{x \in \Sigma^* \mid S \Rightarrow^* x\}$$

Example

Consider $G =$

$\left(\{S, X\}, \{a, b\}, S, \{S \to aS, S \to X, X \to bX, X \to \varepsilon\}\right).$

Let us write the rules out for ease of reference:

1) $S \to aS$; 2) $S \to X$; 3) $X \to bX$; 4) $X \to \varepsilon$

Is $aaab \in L(G)$? By definition, $aaab \in L(G)$ iff $S \Rightarrow^* aaab$.

Let us try to find a derivation.

$S \Rightarrow aS$ (Rule 1)

$\Rightarrow aaS$ (Rule 1)

$\Rightarrow aaaS$ (Rule 1)

$\Rightarrow aaaX$ (Rule 2)

$\Rightarrow aaabX$ (Rule 3)

$\Rightarrow aaab$ (Rule 4)

.

Regular Grammars

- Two types of regular grammars
  - Right linear grammars
  - Left linear grammars
- They are called regular, because they generate regular languages

A Right Linear Grammar

A grammar $G = (V, \Sigma, S, P)$ is right linear iff every production in P has one of the two forms:

1. $X \to zY$;
2. $X \to z$;

where $X \in V$, $Y \in V$, and $z \in \Sigma^*$.

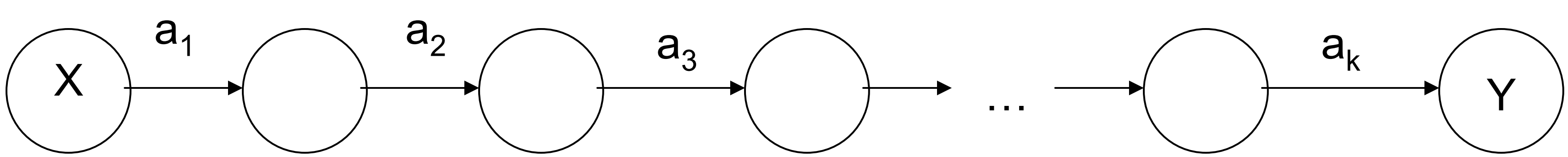
Right Linear Grammars & Regular Languages

Theorem: If ***G*** is a right linear grammar, ***L(G)*** is regular.

Proof Insight: Construct an NFA ***M*** such that ***L(M) = L(G)***.

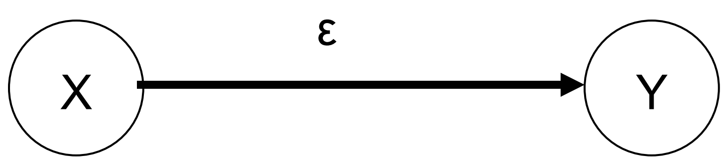
Proof

For every production of the form
$\boldsymbol{X} \rightarrow a_1a_2a_3a_k\boldsymbol{Y}$, add the following transitions to the NFA:



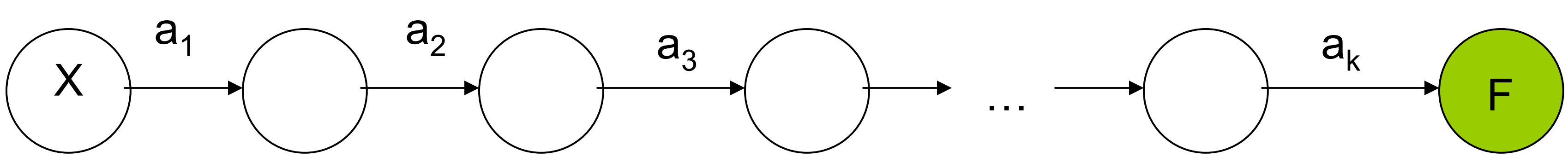
Proof

Note that if ***X*** ➔ ***Y***, then we add the following production:



Proof

For every production of the form
$X \rightarrow a_1a_2a_3a_k$, add the following transitions to the NFA. ***F*** is the symbol that we use to designate the final state.



Left Linear Grammars & Regular Languages

Theorem: If ***G*** is a left linear grammar, then ***L(G)*** is regular.

Proof Insight: Convert ***G*** to a right linear grammar ***G'*** such that $L(G) = [L(G')]^R$.

Left Linear Grammars & Regular Languages

If G has a production of the form $X \rightarrow Yz$, then G' has a production $X' \rightarrow z^R Y$. If G has a production $X \rightarrow z$, then G' has a production $X' \rightarrow z^R$. Then if $x \in L(G)$, then $x^R \in L(G')$. Thus, $x = \left(x^R\right)^R \in [L(G')]^R$.

Regular Grammars & Regular Languages

$$\{\text{Regular Grammars}\} \subseteq \{\text{Regular Languages}\}$$

Recommended Reading

- Section 10.2