Regular Grammars

5DV037 — Fundamentals of Computer Science

Umeå University
Department of Computing Science

Stephen J. Hegner
hegner@cs.umu.se
http://www.cs.umu.se/~hegner

A Review of the Notion of a Grammar

Definition: A *(phrase-structure) grammar* is a four-tuple

$$G = (V, \Sigma, S, P)$$

in which

- V is a finite alphabet, called the *variables* or *nonterminal symbols*;
- Σ is a finite alphabet, called the set of *terminal symbols*;
- $S \in V$ is the *start symbol*;
- P is a finite subset of $(V \cup \Sigma)^+ \times (V \cup \Sigma)^*$ called the set of *productions* or *rewrite rules*;
- $V \cap \Sigma = \emptyset$;

- The production $(w_1, w_2) \in P$ is typically written $w_1 \underset{G}{\rightarrow} w_2$, or just $w_1 \rightarrow w_2$ if the context G is clear.
- The meaning of $w_1 \rightarrow w_2$ is that w_1 may be replaced by w_2 in a string.
- Usually, for $w_1 \rightarrow w_2$, w_1 will contain at least one variable, although this is not strictly necessary.

The Derivation of Words from a Grammar

Context: $G = (V, \Sigma, S, P)$

- Let $w_1 \underset{G}{\rightarrow} w_2$, and let $w \in (V \cup \Sigma)^+$ be a string which contains w_1; *i.e.*, $w = \alpha_1 w_1 \alpha_2$ for some $\alpha_1, \alpha_2 \in (V \cup \Sigma)^*$.
- A possible *single-step derivation* on w replaces w_1 with w_2.
- Write $\alpha_1 w_1 \alpha_2 \underset{G}{\Rightarrow} \alpha_1 w_2 \alpha_2$ (or just $\alpha_1 w_1 \alpha_2 \Rightarrow \alpha_1 w_2 \alpha_2$).
- Note that many derivation steps may be possible on a given string, and that applying one may preclude the application of another.
- This process is thus inherently nondeterministic.
- Write $w \overset{*}{\underset{G}{\Rightarrow}} u$ (or just $w \overset{*}{\Rightarrow} u$) if $w = u$ or else there is a sequence

$$w = \alpha_0 \overset{*}{\underset{G}{\Rightarrow}} \alpha_1 \overset{*}{\underset{G}{\Rightarrow}} \alpha_2 \ldots \overset{*}{\underset{G}{\Rightarrow}} \alpha_k = u$$

  called a *derivation* of u from w (for G).
- The *language of G* is $\mathcal{L}(G) = \{w \in \Sigma^* \mid S \overset{*}{\underset{G}{\Rightarrow}} w\}$.
- The grammars G_1 and G_2 are *equivalent* if $\mathcal{L}(G_1) = \mathcal{L}(G_2)$.

The Basic Idea of Regular Grammars

- A grammar $G = (V, \Sigma, S, P)$ is *right linear* if every production is of one of the following forms:
  - $A \to \alpha B$ for $A, B \in V$, $\alpha \in \Sigma^*$.
  - $A \to \alpha$ for $A \in V$, $\alpha \in \Sigma^*$.
- In a left-linear grammar, the variable occurs on the left:
- A grammar $G = (V, \Sigma, S, P)$ is *left linear* if every production is of one of the following forms:
  - $A \to B\alpha$ for $A, B \in V$, $\alpha \in \Sigma^*$.
  - $A \to \alpha$ for $A \in V$, $\alpha \in \Sigma^*$.
- A grammar is *regular* if it is either right linear or else left linear.

Example: The following grammar is right linear, with $\mathcal{L}(G) = (\mathfrak{Hallo})^*\mathfrak{Welt}$.

- $V = \{S\}$
- $\Sigma = \{\mathfrak{A},\mathfrak{B},\ldots,\mathfrak{Z},\mathfrak{a},\mathfrak{b},\ldots,\mathfrak{z}\}$
- $S \to \mathfrak{Hallo} \cdot S \mid S_1, \qquad S_1 \to \mathfrak{Welt}$.

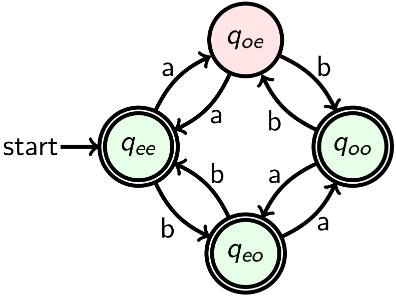
Simple Regular Grammars

- A right-linear grammar $G = (V, \Sigma, S, P)$ is *simple* if every production is of one of the following forms:
  - $A \to aB$ for $A, B \in V$, $a \in \Sigma$.
  - $A \to B$ for $A, B \in V$.
  - $A \to \lambda$ for $A \in V$.
- A simple right-linear grammar is a way of encoding an NFA as a grammar, and conversely, so that they accept the same language.

Automaton $M = (Q, \Sigma, \delta, q_0, F)$	Grammar
Q	variables
q_0	start symbol
$q' \in \delta(q, a)$	$q \to aq'$
$q' \in \delta(q, \lambda)$	$q \to q'$
$q \in F$	$q \to \lambda$

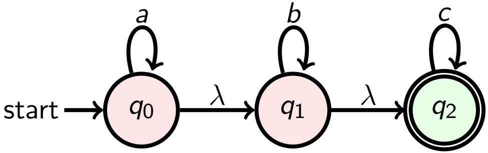
Grammar $G = (V, \Sigma, S, P)$	Automaton
V	states
S	initial state
$A \to aB$	$B \in \delta(A, a)$
$A \to B$	$B \in \delta(A, \lambda)$
$A \to \lambda$	$A \in F$

Examples of Simple Regular Grammars and Conversion



- Simple right-linear grammar:

  variables $= \{q_{ee}, q_{eo}, q_{oe}, q_{oo}\}$
  start symbol $= q_{ee}$
  $q_{ee} \to aq_{oe} \mid bq_{eo} \mid \lambda$
  $q_{oe} \to aq_{ee} \mid bq_{oo}$
  $q_{eo} \to aq_{oo} \mid bq_{ee} \mid \lambda$
  $q_{oo} \to aq_{eo} \mid bq_{oe} \mid \lambda$



- Simple right-linear grammar:

  variables $= \{q_0, q_1, q_2\}$
  start symbol $= q_0$
  $q_0 \to aq_0 \mid q_1$
  $q_1 \to bq_1 \mid q_2$
  $q_2 \to cq_2 \mid \lambda$

General Right Linear Grammars to Simple

- The constructions apply only to simple right-linear grammars.
- However, a conversion to simple is very easy.
- Let $G = (V, \Sigma, S, P)$ be a right-linear grammar.
- Construct an equivalent simple right-linear grammar $G = (V', \Sigma, S, P')$ as follows.
  - For each $A \to a_1 a_2 \ldots a_n B \in P$, (with $a_1 a_2 \ldots a_n \in \Sigma^+$ and $B \in V$), create $n - 1$ new nonterminal symbols $\{A_1, A_2, \ldots, A_{n-1}\}$ and n new productions: $A \to a_1 A_1$, $A_1 \to a_2 A_2$, ..., $A_{n-2} \to a_{n-1} A_{n-1}$, $A_{n-1} \to a_n B$.
  - For each $A \to a_1 a_2 \ldots a_n \in P$, (with $a_1 a_2 \ldots a_n \in \Sigma^+$), create n new nonterminal symbols $\{A_1, A_2, \ldots, A_n\}$ and $n + 1$ new productions: $A \to a_1 A_1$, $A_1 \to a_2 A_2$, ..., $A_{n-2} \to a_{n-1} A_{n-1}$, $A_{n-1} \to a_n A_n$, $A_n \to \lambda$.
  - The new nonterminals must be distinct for each construction for a production in P.

Example of the Conversion

Example: The following grammar $G = (V, \Sigma, S, P)$ is right linear but not simple, with $\mathcal{L}(G) = (\mathfrak{Hallo})^*\mathfrak{Welt}$.

- $V = \{S\}$
- $\Sigma = \{\mathfrak{A},\mathfrak{B},\ldots,\mathfrak{Z},\mathfrak{a},\mathfrak{b},\ldots,\mathfrak{z}\}$
- $S \to \mathfrak{Hallo} \cdot S \mid S_1, \qquad S_1 \to \mathfrak{Welt}$.

- Following the construction, an equivalent simple right-linear grammar $G' = (V', \Sigma, S, P')$ is given by:
  - $V = \{S\}$
  - $\Sigma = \{\mathfrak{A},\mathfrak{B},\ldots,\mathfrak{Z},\mathfrak{a},\mathfrak{b},\ldots,\mathfrak{z}\}$
  - $S \to \mathfrak{H}A_1 \quad A_1 \to \mathfrak{a}A_2 \quad A_2 \to \mathfrak{l}A_3 \quad A_3 \to \mathfrak{l}A_4 \quad A_4 \to \mathfrak{o}S$

    $S \to S_1$

    $S_1 \to \mathfrak{W}B_1 \quad B_1 \to \mathfrak{e}B_2 \quad B_2 \to \mathfrak{l}B_3 \quad B_3 \to \mathfrak{t}B_4 \quad B_4 \to \lambda$

- A derivation of $\mathfrak{HalloWelt}$ using G':
  $S \Rightarrow \mathfrak{H}A_1 \Rightarrow \mathfrak{Ha}A_2 \Rightarrow \mathfrak{Hal}A_3 \Rightarrow \mathfrak{Hall}A_4 \Rightarrow \mathfrak{Hallo}S \Rightarrow \mathfrak{Hallo}S_1 \Rightarrow$
  $\mathfrak{HalloW}B_1 \Rightarrow \mathfrak{HalloWe}B_2 \Rightarrow \mathfrak{HalloWel}B_3 \Rightarrow \mathfrak{HalloWelt}B_4 \Rightarrow \mathfrak{HalloWelt}$.

The Main Theorem for Regular Constructions

Theorem: For any finite alphabet Σ and any language $L \subseteq \Sigma$, the following conditions are equivalent.

- There is a DFA M with $\mathcal{L}(M) = L$.
- There is an NFA M with $\mathcal{L}(M) = L$.
- There is an RE r with $\mathcal{L}(r) = L$.
- There is a simple right-linear grammar G with $\mathcal{L}(G) = L$.
- There is a right-linear grammar G with $\mathcal{L}(G) = L$.

Furthermore, there are algorithms to translate between these representations. $\square$

Definition: Let RegLang(Σ) denote the class of languages over Σ which satisfy any of the above equivalent conditions.

- Clearly, based upon the earlier definition of regular language based upon acceptance by an NFA, RegLang(Σ) is just the set of all regular languages over Σ.

Left-Linear Grammars

- Recall that a *regular grammar* is defined to be one which is either right linear or else left linear.
- A grammar $G = (V, \Sigma, S, P)$ is *left linear* if every production is of one of the following forms:
  - $A \to B\alpha$ for $A, B \in V$, $\alpha \in \Sigma^*$.
  - $A \to \alpha$ for $A \in V$, $\alpha \in \Sigma^*$.
- It is easy to see that the constructions for right-linear grammars may be applied to their left-linear counterparts with only minor changes.
- Therefore, the theorem may be generalized:

Theorem: For any finite alphabet Σ and any language $L \subseteq \Sigma$, the following conditions are equivalent.

- There is a DFA M with $\mathcal{L}(M) = L$.
- There is an NFA M with $\mathcal{L}(M) = L$.
- There is an RE r with $\mathcal{L}(r) = L$.
- There is a regular grammar G with $\mathcal{L}(G) = L$. $\Box$

A Warning about Combining Left and Right Linearity

- The following grammar $G = (V, \Sigma, S, P)$ contains one right-linear production and one left-linear production:
  - $V = \{S, S'\}$
  - $\Sigma = \{a, b\}$
  - $P = \{S \rightarrow aS' \mid \lambda, \;\; S' \rightarrow Sb\}$.
- $\mathcal{L}(G) = \{a^k b^k \mid k \in \mathbb{N}\}$, which is known not to be a regular language.
- Therefore, if left-linear and right-linear productions are combined in the same grammar, the result may not be a regular language.
- In a regular grammar, either all productions are left linear or else all productions are right linear.