

Theory of Computation

Course note based on *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science*, 2nd edition, authored by Martin Davis, Ron Sigal, and Elaine J. Weyuker.

course note prepared by

Tyng-Ruey Chuang

Institute of Information Science, Academia Sinica

Department of Information Management, National Taiwan University

Week 13, Spring 2008

About This Course Note

- ▶ It is prepared for the course *Theory of Computation* taught at the National Taiwan University in Spring 2008.
- ▶ It follows very closely the book *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science*, 2nd edition, by Martin Davis, Ron Sigal, and Elaine J. Weyuker. Morgan Kaufmann Publishers. ISBN: 0-12-206382-1.
- ▶ It is available from Tyng-Ruey Chuang's web site:

<http://www.iis.sinica.edu.tw/~trc/>

and released under a Creative Commons
"Attribution-ShareAlike 2.5 Taiwan" license:

<http://creativecommons.org/licenses/by-sa/2.5/tw/>

Regular Grammars

Definition. A context-free grammar is called *regular* if each of its productions has one of the two forms

$$U \rightarrow aV \quad \text{or} \quad U \rightarrow a$$

where U, V are variables and a is a terminal. □

Theorem 2.1. If L is a regular language, then there is a regular grammar Γ such that either $L = L(\Gamma)$ or $L = L(\Gamma) \cup \{0\}$. □

A Regular Grammar for Every Regular Language

Proof of Theorem 2.1. Let $L = (M)$, where M is a dfa with states q_1, \dots, q_m , alphabet $\{s_1, \dots, s_n\}$, transition function δ , and the set of accepting states F . We construct a grammar Γ with variables q_1, \dots, q_m , terminals s_1, \dots, s_n , and start symbol q_1 . The productions are

1. $q_i \rightarrow s_r q_j$ whenever $\delta(q_i, s_r) = q_j$, and
2. $q_i \rightarrow s_r$ whenever $\delta(q_i, s_r) \in F$.

Clearly the grammar Γ is regular. To show that $L(\Gamma) = L - \{0\}$ we suppose $u \in L$, $u = s_{i_1} s_{i_2} \dots s_{i_l} s_{i_{l+1}} \neq 0$. Thus, $\delta^*(q_1, u) \in F$, so that we have

$$\delta(q_1, s_{i_1}) = q_{j_1}, \quad \delta(q_{j_1}, s_{i_2}) = q_{j_2}, \quad \dots, \quad \delta(q_{j_l}, s_{i_{l+1}}) = q_{j_{l+1}} \in F.$$

A Regular Grammar for Every Regular Language, Continued

Proof of Theorem 2.1. (Continued) By construction, grammar Γ contains the productions

$$q_1 \rightarrow s_{i_1} q_{j_1}, \quad q_{j_1} \rightarrow s_{i_2} q_{j_2}, \quad \dots, \quad q_{j_{l-1}} \rightarrow s_{i_l} q_{j_l}, \quad q_{j_l} \rightarrow s_{i_{l+1}}.$$

Thus, we have in Γ

$$q_1 \Rightarrow s_{i_1} q_{j_1} \Rightarrow s_{i_1} s_{i_2} q_{j_2} \Rightarrow \dots \Rightarrow s_{i_1} s_{i_2} \dots s_{i_l} q_{j_l} \Rightarrow s_{i_1} s_{i_2} \dots s_{i_l} s_{i_{l+1}} = u$$

so that $u \in L(\Gamma)$.

Conversely, suppose that $u \in L(\Gamma)$, $u = s_{i_1} s_{i_2} \dots s_{i_l} s_{i_{l+1}}$. Then there is a derivation of u from q_1 in Γ . By construction, Γ has all the necessary productions to simulate the transition $\delta^*(q_1, u) \in F$ in the dfa \mathcal{M} . □

A Regular Language for Every Regular Grammar

Theorem 2.2. Let Γ be a regular grammar. Then $L(\Gamma)$ is a regular language.

Proof. Let Γ have the variables V_1, V_2, \dots, V_K , where $S = V_1$ is the start symbol, and terminals s_1, s_2, \dots, s_n . Since Γ is regular, its productions are of the form $V_i \rightarrow s_r V_j$ and $V_i \rightarrow s_r$. We now construct the following ndfa \mathcal{M} which accepts precisely $L(\Gamma)$.

- ▶ The states are V_1, V_2, \dots, V_K and an additional state W . V_1 is the initial state and W is the only accepting state.
- ▶ For transition functions, let

$$\begin{aligned} \delta_1(V_i, s_r) &= \{V_j \mid V_i \rightarrow s_r V_j \text{ is a production of } \Gamma\}, \\ \delta_2(V_i, s_r) &= \begin{cases} \{W\} & \text{if } V_i \rightarrow s_r \text{ is a production of } \Gamma \\ \emptyset & \text{otherwise.} \end{cases} \end{aligned}$$

Then define the transition function δ as

$$\delta(V_i, s_r) = \delta_1(V_i, s_r) \cup \delta_2(V_i, s_r).$$

A Regular Language for Every Regular Grammar

Proof of Theorem 2.2. (Continued) Now let

$u = s_{i_1}s_{i_2} \dots s_{i_l}s_{i_{l+1}} \in L(\Gamma)$. Thus we have

$$V_1 \Rightarrow s_{i_1} V_{j_1} \Rightarrow s_{i_1}s_{i_2} V_{j_2} \Rightarrow^* s_{i_1}s_{i_2} \dots s_{i_l} V_{j_l} \Rightarrow s_{i_1}s_{i_2} \dots s_{i_l}s_{i_{l+1}}$$

where Γ contains the productions

$$V_1 \rightarrow s_{i_1} V_{j_1}, \quad V_{j_1} \rightarrow s_{i_2} V_{j_2}, \quad \dots, \quad V_{j_{l-1}} \rightarrow s_{i_l} V_{j_l}, \quad V_{j_l} \rightarrow s_{i_{l+1}}$$

Thus,

$$V_{j_1} \in \delta(V_1, s_{i_1}), \quad V_{j_2} \in \delta(V_{j_1}, s_{i_2}), \quad \dots, \quad W \in \delta(V_{j_l}, s_{i_{l+1}}).$$

Thus $W \in \delta^*(V_1, u)$ and $u \in L(\mathcal{M})$.

Conversely, if $u = s_{i_1}s_{i_2} \dots s_{i_l}s_{i_{l+1}}$ is accepted by \mathcal{M} , then there must be a sequence of transitions of the form above. Hence, the productions listed above must all belong to Γ , so that there is a derivation of u from V_1 .

Every Regular Language Is Context-free

Theorem 2.3. A language L is regular if and only if there is a regular grammar Γ such that either $L = L(\Gamma)$ or $L = L(\Gamma) \cup \{0\}$. \square

Every Regular Language Is Context-free

Theorem 2.3. A language L is regular if and only if there is a regular grammar Γ such that either $L = L(\Gamma)$ or $L = L(\Gamma) \cup \{0\}$. \square

Corollary 2.4. Every regular language is context-free. \square

Right-linear Grammars

Definition. A context-free grammar is called *right-linear* if each of its productions has one of the two forms

$$U \rightarrow xV \quad \text{or} \quad U \rightarrow x,$$

where U, V are variables and $x \neq \epsilon$ is a word consisting entirely of terminals. \square

Right-linear Grammars

Definition. A context-free grammar is called *right-linear* if each of its productions has one of the two forms

$$U \rightarrow xV \quad \text{or} \quad U \rightarrow x,$$

where U, V are variables and $x \neq \epsilon$ is a word consisting entirely of terminals. □

Thus, a regular grammar is just a right-linear grammar in which $|x| = 1$.

Right-linear Grammars, Continued

Theorem 2.5. Let Γ be a right-linear grammar. Then $L(\Gamma)$ is regular.

Proof. We replace each production of Γ of the form

$$U \rightarrow a_1 a_2 \dots a_n V, \quad n > 1$$

by the productions

$$U \rightarrow a_1 Z_1, \quad Z_1 \rightarrow a_2 Z_2, \quad Z_{n-2} \rightarrow a_{n-1} Z_{n-1}, \quad Z_{n-1} \rightarrow a_n V,$$

where Z_1, \dots, Z_{n-1} are new variables. Do similar replacement for production

$$U \rightarrow a_1 a_2 \dots a_n, \quad n > 1$$



Chomsky Normal Form

Definition. A context-free grammar Γ with variables \mathcal{V} and terminals T is in *Chomsky normal form* if each of its productions has one of the forms

$$X \rightarrow YZ \quad \text{or} \quad X \rightarrow a,$$

where $X, Y, Z \in \mathcal{V}$ and $a \in T$. □

Theorem 3.1. There is an algorithm that transforms a given positive context-free grammar Γ into a Chomsky normal form grammar Δ such that $L(\Gamma) = L(\Delta)$. □

Chomsky Normal Form, Continued

Proof of Theorem 3.1. Using Theorem 1.5, we begin with a branching context-free grammar Γ with variable \mathcal{V} and terminals T . We then perform the following two steps:

1. a new variable X_a is introduced for each $a \in T$, and for each production $X \rightarrow x \in \Gamma, |x| > 1$, we replace it with $X \rightarrow x'$ where x' is obtained from x by replacing each terminal a by the corresponding new variable X_a ;
2. For productions of the form $X \rightarrow X_1 X_2 \dots X_k, k > 2$, we introduce new variables Z_1, Z_2, \dots, Z_{k-2} and replace the production with the following

$$\begin{array}{l}
 X \rightarrow X_1 Z_1 \\
 \quad \vdots \\
 Z_{k-3} \rightarrow X_{k-2} Z_{k-2} \\
 Z_{k-2} \rightarrow X_{k-1} X_k
 \end{array}$$

Chomsky Normal Form, Examples

Consider the following branching context-free grammar

$$S \rightarrow aXbY, \quad X \rightarrow aX, \quad Y \rightarrow bY, \quad X \rightarrow a, \quad Y \rightarrow b$$

The resulting grammar, respectively, from the two steps is:

1.

$$S \rightarrow X_aXX_bY, \quad X \rightarrow X_aX, \quad Y \rightarrow X_bY, \\ X \rightarrow a, \quad X_a \rightarrow a, \quad Y \rightarrow b, \quad X_b \rightarrow b$$

2. For the production $S \rightarrow X_aXX_bY$, we replace it with the following:

$$S \rightarrow X_aZ_1 \\ Z_1 \rightarrow XZ_2 \\ Z_2 \rightarrow X_bY.$$

The resulting grammar is in Chomsky normal form.

Bar-Hillel's Pumping Lemma

An application of Chomsky normal form is in the proof of the following theorem, which is an analogy for context-free languages of the pumping lemma for regular languages.

Theorem 4.1. Let Γ be a Chomsky normal form grammar with exactly n variables, and let $L = L(\Gamma)$. Then, for every $x \in L$ for which $|x| > 2^n$, we have $x = r_1 q_1 r q_2 r_2$, where

1. $|q_1 r q_2| \leq 2^n$;
2. $q_1 q_2 \neq \epsilon$;
3. for all $i \geq 0$, $r_1 q_1^{[i]} r q_2^{[i]} r_2 \in L$.



A Small Lemma

Lemma. Let $S \Rightarrow_{\Gamma}^* u$, where Γ is a Chomsky normal form grammar. Suppose that \mathcal{T} is a derivation tree for u in Γ and that no path in \mathcal{T} contains more than k nodes. Then $|u| \leq 2^{k-2}$.

Proof. First, suppose, that \mathcal{T} has just one leaf labeled by a terminal a . Then $u = a$, and \mathcal{T} just have two nodes, S and a , and one path of length $1 < k = 2$. Clearly $|u| = 1 \leq 2^{2-2}$.

Otherwise, since Γ is in Chomsky normal form, the root of \mathcal{T} is labeled by S where $S \rightarrow XY$ for variables X and Y . Let \mathcal{T}_1 and \mathcal{T}_2 be the two trees whose roots are labeled by X and Y , respectively. In each of \mathcal{T}_1 and \mathcal{T}_2 , the longest path must contain $\leq k - 1$ nodes. Proceeding inductively, we may assume that each of the $\mathcal{T}_1, \mathcal{T}_2$ have $\leq 2^{k-3}$ leaves. Hence

$$|u| \leq 2^{k-3} + 2^{k-3} = 2^{k-2}.$$

Bar-Hillel's Pumping Lemma, Proof

Proof of Theorem 4.1. Let $x \in L$, where $|x| > 2^n$, and let \mathcal{T} be a derivation tree for x in Γ . Let $\alpha_1, \alpha_2, \dots, \alpha_m$ be the longest path in \mathcal{T} . Then $m \geq n + 2$ and α_m is a leaf. This is because, if $m \leq n + 1$, by the small lemma, $|x| \leq 2^n - 1$ is a contradiction.

Note that $\alpha_1, \alpha_2, \dots, \alpha_{m-1}$ are all labeled by variables, while α_m is labeled by a terminal. Let $\gamma_1, \gamma_2, \dots, \gamma_{n+2}$ be the path consisting of the vertices $\alpha_{m-n-1}, \alpha_{m-n-2}, \dots, \alpha_{m-1}, \alpha_m$.

Since there are only n variables in the alphabet of Γ , the pigeon-hole principle guarantees that there is a variable X that labels two different vertices: $\alpha = r_i$ and $\beta = r_j$, where $i < j$. (See Fig. 4.2.)

Bar-Hillel's Pumping Lemma, Proof

(Proof of Theorem 4.1., Continued)

Hence, the operations of *pruning* and *splicing* can be applied. Let $r = \langle \mathcal{I}^\beta \rangle$. Then we have, for example,

$$\begin{aligned} \langle \mathcal{I}_p \rangle &= r_1 r r_2, \\ \langle \mathcal{I}_s \rangle &= r_1 q_1^{[2]} r q_2^{[2]} r_2, \\ \langle (\mathcal{I}_s)_s \rangle &= r_1 q_1^{[3]} r q_2^{[3]} r_2 \end{aligned}$$

That is, $r_1 q_1^i r q_2^i r_2 \in L(\Gamma), i \geq 0$. Note that the path in \mathcal{I}^α consists of $\leq n + 2$ nodes, so by the small lemma $|q_1 r q_2| = |\mathcal{I}^\alpha| \leq 2^n$. □

Bar-Hillel's Pumping Lemma, Application

Theorem 4.2. The language $L = \{a^{[n]}b^{[n]}c^{[n]} \mid n > 0\}$ is *not* context-free.

Bar-Hillel's Pumping Lemma, Application

Theorem 4.2. The language $L = \{a^{[n]}b^{[n]}c^{[n]} \mid n > 0\}$ is *not* context-free.

Proof. Suppose that L is context-free with $L = L(\Gamma)$, where Γ is a Chomsky normal form grammar with n variables. Choose k so large that $|a^{[k]}b^{[k]}c^{[k]}| > 2^n$. Then $a^{[k]}b^{[k]}c^{[k]} = r_1 q_1^{[i]} r q_2^{[j]} r_2$, where

$$x_i = r_1 q_1^{[i]} r q_2^{[j]} r_2 \in L$$

for all $i \geq 0$. As $x_2 = r_1 q_1 q_1 r q_2 q_2 r_2 \in L$, we know that q_1 and q_2 must each contain only one of the letters a, b, c . That is, one letter is missing in both q_1 and q_2 .

But as $i = 2, 3, 4, \dots$ contains more and more copies of q_1 and q_2 and since $q_1 q_2 \neq \emptyset$, it is impossible for x_i to have the same number of occurrences of a, b , and c . This contradiction shows that L is not context-free.

$$L_1 \cup L_2$$

Theorem 5.1. If L_1, L_2 are context-free languages, then so is $L_1 \cup L_2$.

$L_1 \cup L_2$

Theorem 5.1. If L_1, L_2 are context-free languages, then so is $L_1 \cup L_2$.

Proof. Let $L_1 = L(\Gamma_1), L_2 = L(\Gamma_2)$, where Γ_1, Γ_2 are context-free grammars with disjoint sets of variables \mathcal{V}_1 and \mathcal{V}_2 , and start symbols S_1, S_2 , respectively.

Let Γ be the context-free grammar with variables $\mathcal{V}_1 \cup \mathcal{V}_2 \cup \{S\}$ and start symbol S . The productions of Γ are those of Γ_1 and Γ_2 , together with the two additional productions $S \rightarrow S_1$ and $S \rightarrow S_2$. Obviously $L(\Gamma) = L(\Gamma_1) \cup L(\Gamma_2)$. \square

$$L_1 \cap L_2$$

Theorem 5.2. There are context-free languages L_1 and L_2 such that $L_1 \cap L_2$ is not context-free.

$L_1 \cap L_2$

Theorem 5.2. There are context-free languages L_1 and L_2 such that $L_1 \cap L_2$ is not context-free.

Proof. The following two languages L_1 and L_2 are context free.

$$L_1 = \{a^{[n]}b^{[n]}c^{[m]} \mid n, m > 0\}$$

$$L_2 = \{a^{[m]}b^{[n]}c^{[n]} \mid n, m > 0\}$$

However, as shown by Theorem 4.2, their intersection

$$L_1 \cap L_2 = \{a^{[n]}b^{[n]}c^{[n]} \mid n > 0\}$$

is not context-free. □

$A^* - L$

Corollary 5.3. There is a context-free language $L \subseteq A^*$ such that $A^* - L$ is not context-free.

$A^* - L$

Corollary 5.3. There is a context-free language $L \subseteq A^*$ such that $A^* - L$ is not context-free.

Proof. Suppose otherwise, that is, for every context-free language $L \subseteq A^*$, $A^* - L$ is context-free. Then the De Morgan identity

$$L_1 \cap L_2 = A^* - ((A^* - L_1) \cup (A^* - L_2))$$

together with Theorem 5.1 would contradict Theorem 5.2. \square