Formal Languages and Automata

Assignment 3

Assignment 3

Reference to Textbook 6th Edition

Ch6
- 6.2: 4, 5

Ch7
- 7.1: 6(b)

Ch8
- 8.1: 6

Ch9
- 9.1: 9

Ch10
- 10.2: 1(d)
- 10.3: 2(a)

- Submitted in A4 size paper (紙本型式)
- Due 6/3 12:00PM (星期二期末考下課前)
  - Delay penalty: -10% per day

6.2

In general, though, neither the conversion of a given grammar to Greibach normal form nor the proof that this can always be done is a simple matter. We introduce Greibach normal form here because it will simplify the technical discussion of an important result in the next chapter. However, from a conceptual viewpoint, Greibach normal form plays no further role in our discussion, so we only quote the following general result without proof.

THEOREM 6.7

For every context-free grammar G with $\lambda \notin L(G)$, there exists an equivalent grammar $\widehat{G}$ in Greibach normal form.

EXERCISES

1. Convert the grammar $S \to aSS|a|b$ into Chomsky normal form.
2. Convert the grammar $S \to aSb|Sab|ab$ into Chomsky normal form.
3. Transform the grammar
$$S \to aSaaA|A, A \to abA|bb$$
into Chomsky normal form.
4. Transform the grammar with productions
$$\begin{aligned} S &\to baAB, \\ A &\to bAB|\lambda, \\ B &\to BAa\,|A|\,\lambda \end{aligned}$$
into Chomsky normal form.
5. Convert the grammar
$$\begin{aligned} S &\to AB|aB, \\ A &\to abb|\lambda, \\ B &\to bbA \end{aligned}$$
into Chomsky normal form.
6. Let $G = (V, T, S, P)$ be any context-free grammar without any λ-productions or unit-productions. Let k be the maximum number of symbols on the right of any production in P. Show that there is an equivalent grammar in Chomsky normal form with no more than $(k-1)\,|P| + |T|$ production rules.
7. Draw the dependency graph for the grammar in Exercise 5.

7,1

The nondeterministic alternative for locating the middle of the string is taken at the third move. At that stage, the pda has the instantaneous descriptions (q_0, ba, baz) and has two choices for its next move. One is to use $\delta(q_0, b, b) = \{(q_0, bb)\}$ and make the move

$$(q_0, ba, baz) \vdash (q_0, a, bbaz),$$

the second is the one used above, namely $\delta(q_0, \lambda, b) = \{(q_1, b)\}$. Only the latter leads to acceptance of the input.

EXERCISES

1. Find a pda that accepts the language $L = \{a^n b^{2n} : n \geq 0\}$.
2. Show the sequence of instantaneous descriptions for the acceptance of $aabbbb$ by the pda in Exercise 1.
3. Construct npda's that accept the following regular languages:
   (a) $L_1 = L(aaa^*bab)$.
   (b) $L_2 = L(aab^*aba^*)$.
   (c) The union of L_1 and L_2.
   (d) $L_1 - L_2$.
   (e) The intersection of L_1 and L_2.
4. Find a pda with fewer than four states that accepts the same language as the pda in Example 7.2.
5. Prove that the pda in Example 7.5 does not accept any string not in $\{ww^R\}$.
6. Construct npda's that accept the following languages on $\Sigma = \{a, b, c\}$:
   (a) $L = \{a^n b^{3n} : n \geq 0\}$.
   ✓(b) $L = \{wcw^R : w \in \{a, b\}^*\}$.
   (c) $L = \{a^n b^m c^{n+m} : n \geq 0, m \geq 0\}$.
   (d) $L = \{a^n b^{n+m} c^m : n \geq 0, m \geq 1\}$.
   (e) $L = \{a^3 b^n c^n : n \geq 0\}$.
   (f) $L = \{a^n b^m : n \leq m \leq 3n\}$.
   (g) $L = \{w : n_a(w) = n_b(w) + 1\}$.
   (h) $L = \{w : n_a(w) = 2n_b(w)\}$.
   (i) $L = \{w : n_a(w) + n_b(w) = n_c(w)\}$.

8.1

This example answers the general question raised on the relation between the families of context-free and linear languages. The family of linear languages is a proper subset of the family of context-free languages.

EXERCISES

1. Show that the language

$$L = \{w : n_a(w) < n_b(w) < n_c(w)\}$$

is not context free.

2. Show that the language

$$L = \{w \in \{a, b, c\}^* : n_a(w) = n_b(w) \geq n_c(w)\}$$

is not context free.

3. Determine whether or not the language $L = \{a^n b^i c^j d^k, n + k \leq i + j\}$ is context free.

4. Show that the language $L = \{a^n : n \text{ is a prime number}\}$ is not context free.

5. Is the language $L = \{a^n b^m : m = 2^n\}$ context free?

6. Show that the language $L = \left\{a^{n^2} : n \geq 0\right\}$ is not context free.

7. Show that the following languages on $\Sigma = \{a, b, c\}$ are not context free:

(a) $L = \{a^n b^j : n \leq j^2\}$.

(b) $L = \{a^n b^j : n \geq (j-1)^3\}$.

(c) $L = \{a^n b^j c^k : k = jn\}$.

(d) $L = \{a^n b^j c^k : k > n, k > j\}$.

(e) $L = \{a^n b^j c^k : n < j, n \leq k \leq j\}$.

(f) $L = \{w : n_a(w) = n_b(w) * n_c(w)\}$.

(g) $L = \{w \in \{a, b, c\}^* : n_a(w) + n_b(w) = 2n_c(w), n_a(w) = n_b(w)\}$.

(h) $L = \{a^n b^m : n \text{ and } m \text{ are both prime}\}$.

(i) $L = \{a^n b^m : n \text{ is prime or } m \text{ is prime}\}$.

(j) $L = \{a^n b^m : n \text{ is prime and } m \text{ is not prime}\}$.

8. Determine whether or not the following languages are context free:

(a) $L = \{a^n w w^R b^n : n \geq 0, w \in \{a, b\}^*\}$.

(b) $L = \{a^n b^j a^n b^j : n \geq 0, j \geq 0\}$.

(c) $L = \{a^n b^j a^j b^n : n \geq 0, j \geq 0\}$.

9.1

(d) $L = \{a^n b^m : n \geq 2, n = m\}$.

(e) $L = \{w : n_a(w) \neq n_b(w)\}$.

(f) $L = \{a^n b^m a^{n+m} : n \geq 0, m \geq 1\}$.

(g) $L = \{a^n b^n a^n b^n : n \neq 0\}$.

(h) $L = \{a^n b^{2n} : n \geq 1\}$.

For each problem, give a transition graph; then check your answers by tracing several test examples.

9. Design a Turing machine that accepts the language

$$L = \{ww : w \in \{a, b\}^+\}.$$

10. Construct a Turing machine to compute the function

$$f(w) = w^R,$$

where $w \in \{0, 1\}^+$.

11. Design a Turing machine that computes the function

$$\begin{aligned} f(w) &= 1 \text{ if } w \text{ is even} \\ &= 0 \text{ if } w \text{ is odd.} \end{aligned}$$

12. Design a Turing machine that computes the function

$$\begin{aligned} f(x) &= x - 2 \text{ if } x > 2 \\ &= 0 \text{ if } x \leq 2. \end{aligned}$$

13. Design Turing machines to compute the following functions for x and y positive integers represented in unary:

(a) $f(x) = 2x + 1$.

(b) $f(x, y) = x + 2y$.

(c) $f(x, y) = 2x + 3y$.

(d) $f(x) = x \bmod 5$.

(e) $f(x) = \lfloor \frac{x}{2} \rfloor$, where $\lfloor \frac{x}{2} \rfloor$ denotes the largest integer less than or equal to $\frac{x}{2}$.

14. Design a Turing machine with $\Gamma = \{0, 1, \sqcup\}$, which, when started on any cell containing a blank or a 1, will halt if and only if its tape has a 0 somewhere on it.

15. Write out a complete solution for Example 9.8.

16. Show the sequence of instantaneous descriptions that the Turing machine in Example 9.10 goes through when presented with the input 111. What happens when this machine is started with 110 on its tape?

10.2

or address with the cells of the two-dimensional tape. This can be done in a number of ways, for example, in the two-dimensional fashion indicated in Figure 10.12. The two-track tape of the simulating machine will use one track to store cell contents and the other one to keep the associated address. In the scheme of Figure 10.12, the configuration in which cell $(1, 2)$ contains a and cell $(10, -3)$ contains b is shown in Figure 10.13. Note one complication: The cell address can involve arbitrarily large integers, so the address track cannot use a fixed-size field to store addresses. Instead, we must use a variable field-size arrangement, using some special symbols to delimit the fields, as shown in the picture.

Let us assume that, at the start of the simulation of each move, the read-write head of the two-dimensional machine M and the read-write head of the simulating machine $\widehat{M}$ are always on corresponding cells. To simulate a move, the simulating machine $\widehat{M}$ first computes the address of the cell to which M is to move. Using the two-dimensional address scheme, this is a simple computation. Once the address is computed, $\widehat{M}$ finds the cell with this address on track 2 and then changes the cell contents to account for the move of M. Again, given M, there is a straightforward construction for $\widehat{M}$.

EXERCISES

1. Give a formal definition of a two-tape Turing machine; then write programs that accept the languages below. Assume that $\Sigma = \{a, b, c\}$ and that the input is initially all on tape 1.

   (a) $L = \{a^n b^n c^n\}, n \geq 1$.

   (b) $L = \{a^n b^n c^m, m > n\}$.

   (c) $L = \{ww : w \in \{a, b\}\}$.

   ✓(d) $L = \{ww^R w : w \in \{a, b\}\}$.

   (e) $L = \{n_a(w) = n_b(w) = n_c(w)\}$.

   (f) $L = \{n_a(w) = n_b(w) \geq n_c(w)\}$.

2. Define what one might call a multitape off-line Turing machine and describe how it can be simulated by a standard Turing machine.

3. A multihead Turing machine can be visualized as a Turing machine with a single tape and a single control unit but with multiple, independent read-write heads. Give a formal definition of a multihead Turing machine, and then show how such a machine can be simulated with a standard Turing machine.

4. Give a formal definition of a multihead-multitape Turing machine. Then show how such a machine can be simulated by a standard Turing machine.

10.3

EXERCISES

1. Discuss the simulation of a nondeterministic Turing machine by a deterministic one. Indicate explicitly how new machines are created, how active machines are identified, and how machines that halt are removed from further consideration.

2. Write programs for nondeterministic Turing machines that accept the languages below. In each case, explain if and how the nondeterminism simplifies the task.

   (a) $L = \{ww : w \in \{a,b\}^+\}$.

   (b) $L = \{ww^R w : w \in \{a,b\}^+\}$.

   (c) $L = \{xww^R y : x, y, w \in \{a,b\}^+, |x| \geq |y|\}$.

   (d) $L = \{w : n_a(w) = n_B(w) = n_c(w)\}$.

   (e) $L = \{a^n : n \text{ is not a prime number}\}$.

3. A two-stack automaton is a nondeterministic pushdown automaton with two independent stacks. To define such an automaton, we modify Definition 7.1 so that

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \times \Gamma \to \text{ finite subsets of } Q \times \Gamma^* \times \Gamma^*.$$

   A move depends on the tops of the two stacks and results in new values being pushed on these two stacks. Show that the class of two-stack automata is equivalent to the class of Turing machines.

10.4 A UNIVERSAL TURING MACHINE

Consider the following argument against Turing's thesis: "A Turing machine as presented in Definition 9.1 is a special-purpose computer. Once δ is defined, the machine is restricted to carrying out one particular type of computation. Digital computers, on the other hand, are general-purpose machines that can be programmed to do different jobs at different times. Consequently, Turing machines cannot be considered equivalent to general-purpose digital computers."

This objection can be overcome by designing a reprogrammable Turing machine, called a **universal Turing machine**. A universal Turing machine M_u is an automaton that, given as input the description of any Turing machine M and a string w, can simulate the computation of M on w. To construct such an M_u, we first choose a standard way of describing Turing machines. We may, without loss of generality, assume that

$$Q = \{q_1, q_2, ..., q_n\},$$

6.2

4. $S \to baAB \mid baB \mid baA \mid ba$

$A \to bAB \mid bB \mid bA \mid b$

$B \to BAa \mid Aa \mid Ba \mid a \mid \cancel{A}$

$bAB \mid bB \mid bA \mid b$

$S \to S_1S_2 \mid S_1B \mid S_1A \mid T_bT_a$

$S_1 \to T_bT_a$

$S_2 \to AB$

$A \to T_bS_2 \mid T_bB \mid T_bA \mid b$

$B \to S_3T_a \mid AT_a \mid BT_a \mid a \mid T_bS_2 \mid T_bB \mid T_bA \mid b$

$S_3 \to BA$

$T_a \to a$

$T_b \to b$ #

5. $S \to AB \mid T_aB \mid B$

$A \to T_aT_bT_b$

$B \to bbA \mid bb$

$S \to AB \mid T_aB \mid S_1A \mid T_bT_b$

$S_1 \to T_bT_b$

$T_a \to a$

$T_b \to b$

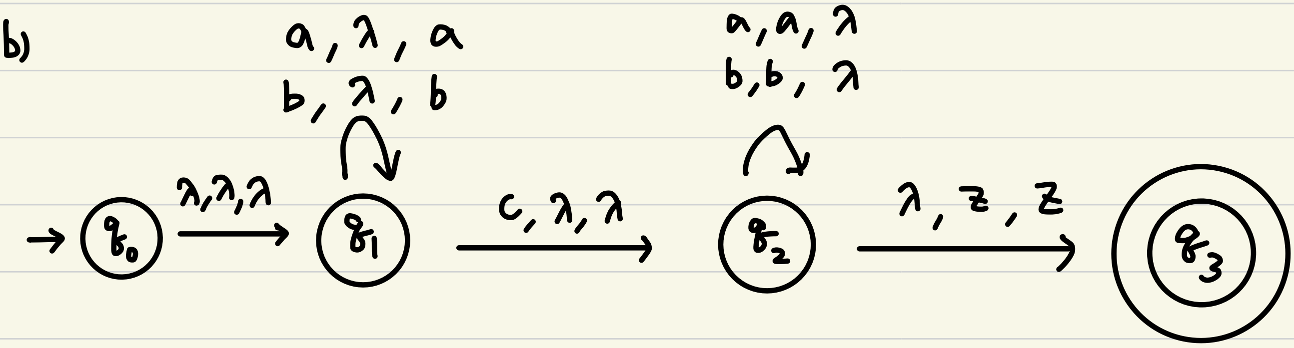
$A \to T_aS_1$

$B \to S_1A \mid T_bT_b$

#

7.1

6 (b)



8.1

6. Assume it's context free and it's infinite
So we can apply pumping lemma
pumping lemma gives a magic number m
s.t. pick any string $w \in L$ with length $|w| \geq m$
we pick $a^{m^2} = uvxyz$
$|vxy| \leq m$, $|vy| \geq 1$

$(m+1)^2 = m^2 + 2m + 1$
$(m+2)^2 = m^2 + 4m + 4$

$uv^2xy^2z = a^{m^2+k}$
$uv^3xy^3z = a^{m^2+2k}$
assume that $m^2 + k = (m+1)^2$ $\therefore k = 2m+1$
but $m^2 + 2k = m^2 + 4m + 2 \neq (m+2)^2$

$\therefore$ if $uv^2xy^2z \in L$
then $uv^3xy^3z \notin L$

$\therefore$ we can know this language isn't context free.

9.1

9.