B62005Y-02 理论计算机科学基础	2017 年 4 月 22 日

Solutions to Homework # 3

主讲教师：杨光 张远航 2015K8009929045

Sipser 3.1 (d) $q_1$000000, $\sqcup q_2$00000, $\sqcup$x$q_3$0000$\sqcup$, $\sqcup$x0$q_4$000$\sqcup$, $\sqcup$x0x$q_3$00, $\sqcup$x0x0$q_4$0, $\sqcup$x0x0x$q_3\sqcup$, $\sqcup$x0x0q_5x$\sqcup$, $\sqcup$x0x$q_5$0x$\sqcup$, $\sqcup$x0q_5x0x$\sqcup$, $\sqcup$x$q_5$0x0x$\sqcup$, $\sqcup q_5$x0x0x$\sqcup$, $q_5\sqcup$x0x0x$\sqcup$, $\sqcup q_2$x0x0x$\sqcup$, $\sqcup$x$q_2$0x0x$\sqcup$, $\sqcup$xxq_3x0x$\sqcup$, $\sqcup$xxx$q_3$0x$\sqcup$, $\sqcup$xxx0q_4x$\sqcup$, $\sqcup$xxx0x$q_4\sqcup$, $\sqcup$xxx0x $\sqcup$ q_{reject}.

Sipser 3.2 (d) $q_1$10#11, x$q_3$0#11, x0q_3#11, x0#$q_5$11, x0q_6#x1, x$q_7$0#x1, q_7x0#x1, x$q_1$0#x1, xxq_2#x1, xx#q_4x1, xx#x$q_4$1, xx#x1q_{reject}.

Sipser 3.8 (b) The following machine M works:

M = "On input string w:

1. Scan the tape and mark the first 1 which has not been marked. If no unmarked 1 is found, go to step 5. Otherwise, move the head back to the front of the tape.
2. Scan the tape and mark the first 0 which has not been marked. If no unmarked 0 is found, reject.
3. Move on to mark the next unmarked 0. If no such unmarked 0 is found, reject.
4. Move the head back to the front of the tape and go to step 1.
5. Move the head back to the front of the tape. Scan the tape to see if there is any unmarked 0 found. If yes, reject. Otherwise, accept."

Sipser 4.1 **(a)** Yes, because M on input 0100 ends in an accept state.

(b) No, because M on input 011 ends in a non-accept state.

(c) No, because the input is not in correct form: the second component of the input is missing.

(d) No, because the input is not in correct form: the first component should be a regular expression but not a DFA.

(e) No, because M accepts ε and hence, $L(M) \neq \emptyset$.

(f) Yes, because $L(M) = L(M)$.

Sipser 4.2 The problem of testing whether a DFA and a regular expression are equivalent can be expressed by the following language:

$$EQ_{\mathsf{DFA-REX}} = \{\langle M, r\rangle \mid M \text{ is a } \mathsf{DFA} \text{ and } r \text{ is a regular expression}\}$$

and $L(M) = L(r)$. We can prove that the language $EQ_{\mathsf{DFA-REX}}$ is decidable by constructing a TM P that decides it as follows:

$P =$ "On input $\langle M, r\rangle$:

1. Convert the regular expression r into a DFA M_r by using the procedure described in Theorem 1.28.

2. Apply the algorithm given in Theorem 4.5 to decide whether $\langle M, M_r\rangle \in EQ_{\mathsf{DFA}}$.

3. If $\langle M, M_r\rangle \in EQ_{\mathsf{DFA}}$ then accept, else reject."

Sipser 4.3 Let M_{Σ^*} be a DFA that accepts Σ^*; then for every DFA A,

$$A \in ALL_{\mathsf{DFA}} \Leftrightarrow \langle A, M_{\Sigma^*}\rangle \in EQ_{\mathsf{DFA}}.$$

Therefore, to decide whether $A \in ALL_{\mathsf{DFA}}$, we just need to decide whether $\langle A, M_{\Sigma^*}\rangle \in EQ_{\mathsf{DFA}}$. The latter can be done by applying the proof in Theorem 4.5. Thus ALL_{DFA} is decidable.

Sipser 4.4 Since $A_{\varepsilon\mathsf{CFG}}$ is just a special case of A_{CFG}, it is possible to adapt TM S for $A_{\varepsilon\mathsf{CFG}}$ as follows.

$S =$ "On input (G, ε), where G is a CFG and ε is an empty string:

1. Convert G to an equivalent grammar in Chomsky normal form.

2. If "$S \rightarrow \varepsilon$" is a production rule in Chomsky normal form, accept; if not, reject."