

3.1

(d)

$q_1 000000$	$\sqcup q_5 xoxox \sqcup$
$\sqcup q_2 00000$	$q_5 \sqcup xoxox \sqcup$
$\sqcup x q_3 0000$	$\sqcup q_2 xoxox \sqcup$
$\sqcup x o q_4 000$	$\sqcup x q_2 oxox \sqcup$
$\sqcup x ox q_3 00$	$\sqcup x x q_3 xox \sqcup$
$\sqcup x oxox q_4 0$	$\sqcup x x x q_3 ox \sqcup$
$\sqcup xoxox q_3 \sqcup$	$\sqcup x x x o q_4 x \sqcup$
$\sqcup xoxox q_5 x \sqcup$	$\sqcup x x x ox q_4 \sqcup$
$\sqcup xox q_5 ox \sqcup$	$\sqcup x x x ox \sqcup q_{reject}$
$\sqcup x o q_5 xox \sqcup$	
$\sqcup x q_5 oxox \sqcup$	

3.2

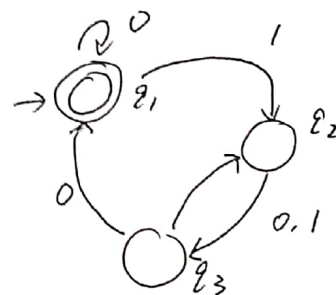
(d) $q_1 10 \# 11$	$xx q_2 \# x  $
$x q_3 0 \# 11$	$xx \# q_4 x  $
$x o q_3 \# 11$	$xx \# x q_4  $
$x o \# q_5 11$	$xx \# x   q_{reject}$
$x o q_5 \# x  $	
$x q_5 o \# x  $	
$q_7 x o \# x  $	
$x q_1 o \# x  $	

3.8

(b)  $M =$  " 对于输入  $w$  :

- 1, 若第一个字符为  $\sqcup$ , 接受; 否则将其改为  $\begin{cases} 1 \rightarrow 1' \\ 0 \rightarrow 0' \end{cases}$ , 作为开头的标志, 读马头不动
- 2, 读马头向右, 将第一个  $1$  或  $1'$  改为  $\begin{cases} 1 \rightarrow x \\ 1' \rightarrow x' \end{cases}$ , 若无法找到, 拒绝 (带上还有  $0$  或  $0'$ )
- 3, 读马头右移到第一个  $\sqcup$  处
- 4, 读马头不断左移, 寻找两个  $0$  或  $0'$ , 将其改为  $x$ ; 若无法找到, 拒绝; 当读到  $1$  或  $0'$  或  $x'$  时, 转到第 2 步 "  $\begin{cases} 0 \rightarrow x \\ 0' \rightarrow x' \end{cases}$

4.1



- 对。经  $q_1, q_1, q_2, q_3, q_1$ , 接受。
- 错。经  $q_1, q_1, q_2, q_3$ , 拒绝。
- 错。  $\langle M \rangle$  不是合法的编码。
- 错。  $M$  不是合法的正则表达式，故  $\langle M, 0100 \rangle \notin AREX$ 。
- 错。由 a.  $M$  接受  $0100$ ，故  $L(M)$  非空。
- 对。因为  $M$  和  $M$  接受同样的语言。

4.2

$$EQ_{DFA, REX} = \{ \langle A, R \rangle \mid A \text{ 为 DFA, } R \text{ 为正则表达式, } L(A) = L(R) \}$$

构造图灵机  $M$  判定  $EQ_{DFA, REX}$  $M =$  “对于合法的输入  $\langle A, R \rangle$  :

- 用引理 1.29 的方法，构造 DFA  $NFA$   $B$ ，使  $L(B) = L(R)$
- 用定理 1.19 的方法，构造 DFA  $D$ ，使  $L(D) = L(B)$
- 根据定理 4.5， $EQ_{DFA}$  是可判定的，设  $F$  判定  $EQ_{DFA}$ ，在输入  $\langle A, D \rangle$  上模拟  $F$ ，若  $F$  接受，则接受；若  $F$  拒绝，则拒绝。”

4.3 构造图灵机  $M$  判定  $ALL_{DFA}$  $M =$  “对于输入  $\langle A \rangle$ ， $A$  为 DFA :

- 构造  $\bar{A}$ ， $\bar{A}$  的接受状态为  $A$  的拒绝状态， $\bar{A}$  的拒绝状态为  $A$  的接受状态。
- 由定理，有图灵机  $F$  判定  $EQ_{DFA}$ ，在输入  $\langle \bar{A} \rangle$  上模拟  $F$ ，若  $F$  接受，则接受。否则拒绝。”

4.4

(1) 构造图灵机  $M$  判定  $A_{CFG}$ : $M =$  "对于输入  $\langle G \rangle$ ,  $G$  为 CFG根据定理 4.6, 有 TM  $F$  判定  $A_{CFG}$ 在输入  $\langle G, \varepsilon \rangle$  上模拟  $F$ , 若  $F$  接受, 则接受,否则拒绝。" 由于  $F$  判定  $A_{CFG}$ ,  $F$  必停机,  $M$  也必停机。(2) 若题为 CFG  $G$  只域  $\varepsilon$ :  $M =$  "将  $G$  转成乔姆斯基范式, 设起始变元为  $S$ , 判断  $H$  是否只含一条规则  $S \rightarrow \varepsilon$ , 若是, 接受, 否则拒绝。"5.1 假设 TM  $M$  判定  $EQ_{CFG}$ 则构造 TM  $F =$  "对于输入  $\langle G \rangle$ ,  $G$  为 CFG1, 构造 CFG  $H$ ,  $L(H) = \Sigma^*$  (规则:  $S \rightarrow aS \mid \varepsilon$ , 其中  $a$  为  $\Sigma$  中每一个字母,  $S$  为起始变元)2. 在输入  $\langle G, H \rangle$  上模拟  $M$ 若  $M$  接受, 则接受, 若  $M$  拒绝, 则拒绝。"则  $F$  判定  $A_{CFG}$ , 但  $A_{CFG}$  为不可判定的, 矛盾。故  $EQ_{CFG}$  不可判定5.2 构造 TM  $M =$  "对于输入  $\langle G, H \rangle$ ,  $G, H$  为 CFG1. 枚举地生成字串  $x \in \Sigma^*$ 2. 分别在  $\langle G, x \rangle$  和  $\langle H, x \rangle$  上模拟 TM  $F$ , 其中  $F$  判定  $A_{CFG}$   
若两次模拟的结果 ~~状态~~ 结果 ~~都~~ 都接受或者都拒绝, 继续  
下一步的枚举; 若一个接受, 一个拒绝, 则接受。"由此,  $M$  识别  $EQ_{CFG}$ 

5.3

$$\left[ \frac{aa}{a} \right] \left[ \frac{b}{a} \right] \left[ \frac{aba}{b} \right] \left[ \frac{ab}{abab} \right]$$
$$\frac{ab}{abab} \quad \frac{ab}{abab} \quad \frac{aba}{b} \quad \frac{b}{a} \quad \frac{b}{a} \quad \frac{aa}{a} \quad \frac{aa}{a}$$

5.4 不对。

如  $L(B) = \{1^n \mid n \geq 0\}$   $f(w) = \begin{cases} 1^n & w = 0^n 1^n, n \geq 0 \\ 0 & w \neq 0^n 1^n, n \geq 0 \end{cases}$  $L(A) = \{0^n 1^n \mid n \geq 0\}$ 则  $w \in A \Leftrightarrow f(w) \in B$ ,  $A \leq_m B$ ,  $B$  是正则的但  $A$  不是正则的



6.2

~~证明与定理6.1基本相同:~~假设  $S$  是  $MIN_{TM}$  的无限子集且图灵可识别则存在  $TM$   $E$  枚举  $S$ 构造  $TM$   $M = "$  对于输入  $w$ :

1. 由递归定理得到自己的一个描述  $\langle M \rangle$
2. 运行  $E$ , 直到一个比  $\langle M \rangle$  描述更长的机器  $D$  出现
3. 在输入  $w$  上模拟  $D$ ."

由于  $S$  是无限的, 故第二步必然在某时终止, 出现  $D$ ,  $D$  比  $M$  描述更长, 但  $M$  与  $D$  等价,  $D$  不是最小图灵机, 矛盾.

6.1

c语言代码如下:

```
#include <stdio.h>
char s[1000]={59,10,32,32,32,32,105,110,116,32,109,97,105,110,40,41,123,10,32,32,32,32,112,114,105,110,116,
,102,40,34,35,105,110,99,108,117,100,101,32,60,115,116,100,105,111,46,104,62,92,110,99,104,97,114,32,115,
91,49,48,48,48,93,61,123,34,41,59,10,32,32,32,32,102,111,114,40,105,110,116,32,105,61,48,59,115,91,105,93,
33,61,48,59,105,43,43,41,32,112,114,105,110,116,102,40,34,37,100,37,99,34,44,115,91,105,93,44,115,91,105,4
3,49,93,33,61,48,63,39,44,39,58,39,125,39,41,59,10,32,32,32,32,102,111,114,40,105,110,116,32,105,61,48,59,1
15,91,105,93,33,61,48,59,105,43,43,41,32,112,114,105,110,116,102,40,34,37,99,34,44,115,91,105,93,41,59,10,
32,32,32,32,114,101,116,117,114,110,32,48,59,10,125,10};
int main(){
    printf("#include <stdio.h>\nchar s[1000]={");
    for(int i=0;s[i]!=0;i++) printf("%d%c",s[i],s[i+1]!=0?',';');
    for(int i=0;s[i]!=0;i++) printf("%c",s[i]);
    return 0;
}
```

整个 `char s[] {...}` 数组相当于构造中的A部分; 其它相当于B部分, 它可以通过指针 `s` 从A获得B的信息。那一长串数组内容就是 `main` 函数对应的ASCII码 (前面再加上一个分号和换行符)。