**第一章**

**练习**

1.1 图给出两台DFA M1和M2的状态图. 回答下述有关问题.

1. M1的起始状态是q1
2. M1的接受状态集是{q2}
3. M2的起始状态是q1
4. M2的接受状态集是｛q1，q4｝
5. 对输入aabb,M1经过的状态序列是q1，q2，q3，q1，q1
6. M1接受字符串aabb吗？否
7. M2接受字符串ε吗？是

1.2 给出练习2.1中画出的机器M1和M2的形式描述.

M1=(Q1,Σ,δ1,q1,F1) 其中

1. Q1={q1,q2,q3,};
2. Σ={a,b};
3. δ1为：

|  |  |
| --- | --- |
|  | a b |
| q1  q2  q3 | q2 q1  q3 q3  q2 q1 |

1. q1是起始状态
2. F1={q2}

M2=(Q2,Σ,δ2,q2,F2) 其中

1. Q2={q1,q2,q3,q4};
2. Σ={a,b};

3）δ2为：

|  |  |
| --- | --- |
|  | a b |
| q1  q2  q3  q4 | q1 q2  q3 q4  q2 q1  q3 q4 |

1. q2是起始状态
2. F2={q1,q4}

1.3 DFA M的形式描述为 ( {q1，q2，q3，q4，q5}，{u,d},δ,q3，{q3}),其中δ在表2-3中给出。试画出此机器的状态图。

q1

q5

q4

q2

q3

u

d

u

u

u

u

d

d

d

d

1.6 画出识别下述语言的DFA的状态图。

a){w | w从1开始以0结束}

0

0

1

1

1

0,1

0

b){w | w至少有3个1}

0

1

0

0

1

1

0,1

c) {w | w含有子串0101}

0,1

1

0

0

1

1

0

1

0

d) {w | w的长度不小于3，且第三个符号为0}

0,1

0

0,1

0,1

1

1

0,1

0

e) {w | w从0开始且为奇长度，或从1开始且为偶长度}

0,1

0,1

0,1

0

0,1

1

0,1

0

0,1

1

或

f) {w | w不含子串110}

0,1

0

1

0

1

1

0

g) {w | w的长度不超过5}

0,1

0,1

0,1

0,1

0,1

0,1

0,1

h){w | w是除11和111以外的任何字符}

1

1

1

0,1

0

0

0

i){w | w的奇位置均为1}

1

0

0,1

0,1

j) {w | w至少含有2个0，且至多含有1个1}

0

0

1

0

0

1

1

1

1

1

0

0

0,1

k) {ε,0}

0

0,1

0,1

1

l) {w | w含有偶数个0，或恰好两个1}

1

1

0

0

1

1

1

1

1

0

0

0

0

0

0

1

m) 空集 n) 除空串外的所有字符串

0,1

0,1

0,1

1.7 给出识别下述语言的NFA，且要求符合规定的状态数。

a. {w | w以00结束}，三个状态

0

0

0,1

b. 语言｛w | w含有子串0101，即对某个x和y，w=x0101y｝，5个状态.

0

1

0,1

0

1

0,1

c. 语言｛w | w含有偶数个0或恰好两个1｝，6个状态。

ε

0

1

1

1

0

1

0

0

0

ε

d. 语言｛0｝，2个状态。

0

e. 语言0\*1\*0\*0，3个状态。

ε

0

0

1

0

f. 语言{ε}，1个状态。

g. 语言0\*，1个状态。

0

2.11证明每一台NFA都能够转换成等价的只有一个接受状态的NFA。

证明:设NFA M={Q，Σ,δ,q0,F}，F={ri1，……，rik}.添加一个状态p后，ri1，……，rik分别向p引ε箭头，将ri1，……，rik变为非接受状态，p变为接受状态。又因为添加ε箭头不影响NFA识别语言，所以命题成立。

2.14 a 证明：设M是一台语言B的DFA，交换M的接状态与非接受状态得到一台新的DFA，则这台新的DFA是识别B 的补集，因此，正则语言类受在补运算下封闭。

b 举例说明：设M是一台识别语言B的NFA，交换M的接受状态与非接受状态得到一台新的NFA，这台新的NFA不一定识别B的补集。NFA识别的语言类在补集下封闭吗？解释你的回答。

解：

1. M是DFA, M是{Q,∑,δ,q0,F},令N={Q,∑,δ,q0,Q-F},设ω=ω1ω2…ωn是字母表上任意字符串，因为M与N均为DFA,所以必然存在Q中状态序列r0,r1,…,rn,使得：r0=q0, δ(ri, ωi+1)=ri+1, i=0,…,n-1

1)若rn∈F 则ω∈B;

2)若rn∉F,则rn∈Q-F,即N接受ω，若ω∉B,

所以N接受B的补集，即B的补集正则。

所以，正则语言类在补运算下封闭。

0

1. 设B为{0}。NFA M：

可识别它。

0

依题对它作变换，得到N：

则N识别的语言{ε}不是B的子集。所以交换M的接受状态与非接受状态得到的新的NFA不一定识别B的补集。

但是由于NFA识别的语言类与DFA识别的语言类相同，即正则语言类。由a的证明，正则语言类在补运算封闭，可知，NFA识别的语言类---正则语言类在补运算下封闭。

若NFA识别语言A，必有 等价的DFA识别A,从而由a知，可交换DFA的接受与非接受状态构造识别A的补集的DFA,则必有等价的NFA识别A的补集。只是，该NFA未必有原NFA交换接受与非接受状态构造而成。

1.15 给出一个反例，说明下述构造不能证明定理2.24，即正则语言类在星号运算下封闭。设N=（Q1,Σ,δ1,q1,F1）识别A1。如下构造N=（Q1,Σ,δ1,q1,F1）。N应该识别A1\*。

1. N的状态集是N1的状态集。
2. N的起始状态是N1的起始状态相同。
3. F=｛q1｝∪F1。F的接受状态是原来的接受状态加上它的起始状态。
4. 定义δ如下：对每一个q属于Q和每一个a属于Σ。



解：设N1识别语言A={至少含有一个1}，其中输入字母表为{0，1}，可知A\*={空串或至少含有一个1}。

1

0,1

0,1

ε

1

0,1

0,1

N1: N:

按上述规定构造N的状态图如上。可以看出L(N)={0,1}\*不等于A\*. 所以如此构造的N不一定识别A\*.

1.16 使用定理2.19中给出的构造，把下图中的两台非确定型有穷自动机转换成等价的确定型有穷自动机。

a

a,b

ε

b

a

1

2

3

a,b

a

b

1

2

a)， b)，

解：a), b)

a

a

b

1

23

b

123

∅

a

b

a,b

a,b

a

b

1

2

b

12

∅

a

a,b

2.13 给出生成练习2.4中语言的正则表达式。(注: 答案不唯一)

a. {w | w从1开始以0结束} 1Σ\*0.

b. {w | w至少有3个1} Σ\*1Σ\*1Σ\*1Σ\*.

1. {w | w含有子串0101} Σ\*0101Σ\*.
2. {w | w的长度不小于3，且第三个符号为0} ΣΣ0Σ\*.
3. {w | w从0开始且为奇长度，或从1开始且为偶长度} 0(ΣΣ)\*∪1Σ(ΣΣ)\*.
4. {w | w不含子串110} (0∪10) \*1\*.
5. {w | w的长度不超过5} ε∪Σ∪ΣΣ∪ΣΣΣ∪ΣΣΣΣ∪ΣΣΣΣΣ.
6. {w | w是除11和111以外的任何字符} 0Σ\*∪10Σ\*∪110Σ\*∪111ΣΣ\*.
7. {w | w的奇位置均为1} (1Σ)\*( ε∪1).
8. {w | w至少含有2个0，且至多含有1个1} 0\*(00∪010∪001∪100) 0\*.
9. {ε,0}. ε∪0.
10. {w | w含有偶数个0，或恰好两个1} (1\*01\*0)\*1\*∪0\*10\*10\*.
11. 空集. ∅.
12. 除空串外的所有字符串ΣΣ\*.

1.19对下述每一个语言，给出4个字符串，2个是这个语言的成员，2个不是这个语言的成员。这里假设字母表Σ={a,b}.

a. a\*b\*  成员：ab，aab 非成员：aba，ba

b. a(ba)\*  成员：ab，abab 非成员：abb，aa

c. a\*∪b\* 成员：aaa，bbb 非成员：ab，ba

d. (aaa)\* 成员：aaa，aaaaaa 非成员：a，aa

e.Σ\*aΣ\*bΣ\*aΣ\* 成员：aba，aaba 非成员：aa，abb

f. aba∪bab 成员：aba，bab 非成员：a，b

g. (ε∪a)b 成员：b，ab 非成员：a，bb

h. (a∪ba∪bb) Σ\* 成员：a，bb 非成员：ε，b

1.21 使用引理2.32中叙述的过程，把图2-38中的有穷自动机转换成正则表达式。

b

b

a,b

a

a

1

2

3

b

a

b

1

2

a

a), b),

解: a) a\*b(a∪ba\*b)\*

b) ε∪(a∪b)a\*b[(aa∪ab∪b)a\*b]\*(a∪ε).

(注：答案不唯一)

1.29利用泵引理证明下述语言不是正则的。

a. A1={0n1n2n | n≥0}。

证明：假设A1是正则的。设p是泵引理给出的关于A1的泵长度。

令S=0p1p2p,

∵S是A1的一个成员且S的长度大于p，所以泵引理保证S可被分成3段S=xyz且满足泵引理的3个条件。根据条件3，y中只含0，xyyz中，0比1、2多，xyyz不是A1的成员。违反泵引理的条件1，矛盾。

∴A1不是正则的。

b. A2={www | w∈{a,b}\*}.

证明：假设A2是正则的。设p是泵引理给出的关于A2的泵长度。

令S=apbapbapb,

∵S是A2的一个成员且S的长度大于p，所以泵引理保证S可被分成3段S=xyz且满足泵引理的3个条件。根据条件3，y中只含a，所以xyyz中第一个a的个数将比后两个a的个数多，故xyyz不是A2的成员。违反泵引理的条件1，矛盾。

∴A2不是正则的。

c. A3={a2n | n≥0}.(在这里，a2n表示一串2n个a .)

证明：假设A3是正则的。设p是泵引理给出的关于A3的泵长度。

令S= a2p，  
∵S是A2的一个成员且S的长度大于p，所以泵引理保证S可被分成3段S=xyz且满足泵引理的3个条件。即对任意的i≥0，xyiz都应在A3中，且xyiz与xyi+1z的长度都应是2的幂. 而且xyi+1z的长度应是xyiz的长度的2n倍(n≥1)。于是可以选择足够大的i，使得|xyiz|=2n>p. 但是|xyi+1z|-|xyiz|=|y|≤p. 即|xyi+1z|<2n+1, 矛盾。

∴A3不是正则的。

1.30下面“证明”0\*1\*不是正则语言，指出这个“证明”中的错误。(因为0\*1\*是正则的，所以一定错误。) 采用反证法证明。假设0\*1\*是正则的。令P是泵引定理给出的关于0\*1\*的泵长度。取S为字符串0p1p。S是0\*1\*的一个成员，但是例2.38已证明S不能被抽取。于是得到矛盾，所以0\*1\*不是正则的。

解：在例2.38中的语言是{0n1n | n≥0}，取S为字符串0p1p，S确实不能被抽取；但针对语言0\*1\*，S是能被抽取的。将S分成三段S=xyz，由泵引理的条件3，y仅包含0，而xyiz属于语言0\*1\*，即S能被抽取，故题设中的“证明”不正确。

1.24有穷状态转换器(FST)是确定性有穷自动机的一种类型。它的输出是一个字符串，而不仅仅是接受或拒绝。图2—39是两台有穷状态状态转换器T1和T2的状态图。

b/1

a/0

q3

q2

q1

b/0

a/1

b/1

a/1

2/1

0/0

1/0

0/0

q1

q2

1/1

2/1

T1 T2

FST的每一个转移用两个符号标记，一个指明该转移的输入符号，另一个指明输出符号。两个符号之间用斜杠“/”把它们分开。在T1中，从q1到q2的转移有输入符号2和输出符号1。某些转移可能有多对输入-输出，比如T1中从q1到它自身的转移。FST在对输入串w计算时，从起始状态开始，一个接一个地取输入符号w1…wn，并且比照输入标记和符号序列w1…wn=w进行转移。每一次沿一个转移走一步，输出对应的输出符号。例如，对输入2212011，机器T1依次进入状态q1, q2, q2, q2, q2, q1, q1, q1和输出1111000。对输入abbb，T2输出1001。给出在下述每一小题中机器进入的状态序列和产生的输出。

a. T1对输入串011, 输出：000, 状态序列：q1, q1, q1, q1.

b. T1对输入串211, 输出：111, 状态序列：q1, q2, q2, q2.

c. T1对输入串0202, 输出：0101, 状态序列：q1, q1, q2, q1, q2。

d. T2对输入串b, 输出：1, 状态序列：q1, q3.

e. T2对输入串bbab, 输出：1111, 状态序列：q1, q3, q2, q3, q2.

f. T2对输入串bbbbbb, 输出：110110, 状态序列：q1, q3, q2, q1, q3, q2, q1。

g. T2对输入串ε, 输出：ε, 状态序列：q1。

1.25 给出有穷状态转换器的形式定义。

解：有穷状态转换器FST是一个五元组(Q,Σ,Г,δ,q0)

* 1. Q是一个由穷集合，叫做状态集
  2. Σ是一个由穷集合，叫做输入字母表
  3. Г是一个由穷集合，叫做输出字母表
  4. δ：Q×Σ🡪Q×Г是转移函数
  5. q0是起始状态

FST计算的形式定义：  
M=(Q,Σ,Г,δ,q0)是一台由穷状态转换器，w=w1w2…wn是输入字母表上的一个字符串。若存在Q中的状态序列：r0, r1, …rn和输出字母表上的一个字符串s=s1…sn, 满足下述条件：

1. r0=q0;
2. δ(ri,wi+1)=(ri+1, si+1), i=0,1,…,n-1

则M在W的输入下输出s.

1.26利用你给练习2.20的答案，给出练习2.19中画出的机器T1和T2的形式描述。

解：有穷状态转换器T1的形式描述为：

T1=({q1, q2 }, {0，1，2},δ, q1, {0,1})

其中转移函数为：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 | 2 |
| q1 | q1/0 | q1/0 | q2/1 |
| q2 | q1/0 | q2/1 | q2/1 |

有穷状态转换器T2的形式描述为：

T2=({{q1, q2, q3}, {a, b},δ, q1, {0,1})

其中转移函数为：

|  |  |  |
| --- | --- | --- |
|  | a | b |
| q1 | q2/1 | q3/1 |
| q2 | q3/1 | q1/0 |
| q3 | q1/0 | q2/1 |

1.27 给出一台具有下述行为的FST的状态图。它的输入、输出字母表都是{0,1}。它输出的字符串与输入的字符串偶数为相同、奇数位相反。例如，对于输入0000111，它应该输出1010010。

解：

1/0

0/1

1/1

0/0

q1

q2

1.46 证明：

1. 假设{0n1m0n|m,n≥0}是正则的，p是由泵引理给出的泵长度。取s＝0p1q0p，q>0。由泵引理条件3知，|xy|≤p，故y一定由0组成，从而字符串xyyz中1前后0的数目不同，即xyyz不属于该语言，这与泵引理矛盾。所以该语言不是正则的。
2. 假设{0n1n|n≥0}的补集是正则的，则根据正则语言在补运算下封闭可得{0n1n|n≥0}是正则的，这与已知矛盾，故假设不成立。所以该语言不是正则的。

记c={0m1n|m≠n}，┐c为c的补集，┐c∩0\*1\*={0n1n|n≥0}，已知{0n1n|n≥0}不是正则的。若 ┐c是正则的，由于0\*1\*是正则的，那么┐c∩0\*1\*也应为正则的。这与已知矛盾，所以 ┐c不是正则的。由正则语言在补运算下的封闭性可知c也不是正则的。

1. {w | w∈{0,1}\*不是一个回文}的补集是{w | w∈{0,1}\*是一个回文}，设其是正则的，令p是由泵引理给出的泵长度。取字符串s=0p1q0p，显然s是一个回文且长度大于p。由泵引理条件3知|xy|≤p，故y只能由0组成。而xyyz不再是一个回文，这与泵引理矛盾。所以{w | w∈{0,1}\*是一个回文}不是正则的。由正则语言在补运算下的封闭性可知{w | w∈{0,1}\*不是一个回文}也不是正则的。

1.31 对于任意的字符串w=w1w2…wn，w的反转是按相反的顺序排列w得到的字符串，记作wR，即wR=wn…w2w1。对于任意的语言A，记 AR={wR | ∈A}证明：如果A是正则的，则AR也是正则的。

证明：

因为A是正则语言，所以可以用NFA来表示该语言，现在来构造AR的NFA，将NFA A中的接受态变为中间态，起始态变为接受态，再添加一新的起始态，并用ε箭头连接至原来的接受态，其它所有的箭头反向。 经过变换后得到NFA变成描述AR的NFA.

1.32 令

0

0

0

0

0

1

,

0

1

0

,

…

,

,

1

1

1

Σ3=

∑3包括所有高度为3的0和1的列向量。∑3上的字符串给出三行0和1的字符串。把每一行看作一个二进制数，令

B={ w∈∑3\* | w最下面一行等于上面两行的和}

例如，

, 而

∈B

∉B

0

0

1

1

0

0

1

1

0

0

0

1

1

0

1

证明B是正则的。

证明：由题设B的定义可知，w上面两行之和减去下面一行结果为零，由此可设计NFA M (Q, Σ, δ, q0, F)，其中∑=∑3。Q={q0,q1}。q0状态表示上一次运算的进位为0，q1状态表示上一次运算的进位为1。

δ由下表给出：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0  0  0 | 0  0  1 | 0  1  0 | 0  1  1 | 1  0  0 | 1  0  1 | 1  1  0 | 1  1  1 |
| q0 | {q0} | ∅ | ∅ | {q0} | ∅ | {q0} | {q1} | ∅ |
| q1 | ∅ | {q0} | {q1} | ∅ | {q1} | ∅ | ∅ | {q1} |

F={q0}

q0

q1

0

0

1

1

1

0

1

0

1

1

0

1

0

0

0

1

1

1

1

0

0

0

1

0

,

,

,

,

状态图为：

所以可知自动机M识别的是语言BR，因此BR是正则的。由题2-24的结论可知B也是正则的。

1.33 令

,

,

,

Σ2=

0

0

0

1

1

1

1

0

Σ2包含所有高度为2的0和1的列。Σ2上的字符串给出两行0和1的字符串。把每一行看作一个二进制数，令

C={ w∈∑2\* | w下面一行等于上面一行的3倍 }。

证明C是正则的。可以假设已知问题2.24中的结果。

证明：如下的NFA识别CR：其中q0状态表示上一次运算的进位为0，

q0

q1

q2

0

0

1

1

1

1

0

0

0

1

1

0

q1状态表示上一次运算的进位为1, q2状态表示上一次运算的进位为2。

如下的NFA识别C：其中状态q0,q1,q2分别表示目前读到的下面的数减上面

q0

q1

q2

0

0

1

1

0

1

1

0

1

1

0

0

的数的3倍余0,1,2的情况。

,

,

,

Σ2=

0

0

0

1

1

1

1

0

1.34令

Σ2包含所有高度为2的0和1的列。Σ2上的字符串给出两行0和1的字符串。把每一行看作一个二进制数，令

D={ w∈∑2\* | w上一行大于下一行 }。

证明D是正则的。

证明：由题设可设计自动机M=(Q, Σ, δ, q, F)，其中Q={q1,q2}，F={q2},

转移函数与状态图为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0  0 | 0  1 | 1  0 | 1  1 |
| q1 | {q1} | ∅ | {q2} | {q1} |
| q2 | {q2} | {q2} | {q2} | {q2} |

q1

q2

0

0

1

1

0

1

1

0

1

1

0

0

1

0

,

,

,

,

1.35设∑2与问题2.26中的相同。把每一行看作0和1的字符串，令E={w∈∑2\* | w的下一行是上一行的反转}，证明E不是正则的。

证明：假设E是正则的，令p是有泵引理给出的泵长度。

1

0

p

0

1

p

选择字符串s= 。于是s能够被划分为xyz且满足泵引理的条件。

1

0

根据条件3，y仅能取包含 的部分，当添加一个y时，xyyz不属于E. 所以E不是正则的.

1.36 令Bn={ak | k是n的整数倍}。证明对于每一个n≥1, Bn是正则的。

证明：设字母表∑为{a}，则an是一正则表达式。所以，(an)\*也是正则表达式。由题意Bn=(an)\*,即Bn可以用正则表达式表示。所以，Bn也是正则的。

1.37 令Cn={x | x是一个二进制数，且是n的整数倍}。证明对于每一个n≥1, Cn是正则的。

证明：下面的DFA识别Cn：(正向读)

M=( Q, {0,1} , δ , q0 , F ), 其中Q＝{0,1,2,…，n-1},

δ( i,1)=2i+1 mod n, δ( i,0 )=( 2i mod n), i=0,1,2,…，n-1,

起始状态为0, F＝{0}.

这里i表示当前数mod n余i.

下面的DFA识别CnR：(反向读)

M=( Q, {0,1} , δ , q0 , F ), 其中Q＝{(i,j)|i,j=0,1,2,…，n-1},

δ((i,j),1)=( 2i mod n, (2i+j)mod n ), δ((i,j),0)=( 2i mod n, j ), i,j=0,1,2,…，n-1

起始状态为(1,0), F＝{ (i,0) | i=0,1,…，n-1}.

这里(i,j)表示当前数mod n余j, 而下一位所表示单位数mod n余i(例如，若读下一位将达到k位，则下一位所表示单位数为10k-1).

1.38 考虑一种叫做全路径NFA的新型有穷自动机。一台全路径NFA M是5元组 ( Q, ∑, δ, q0, F). 如果M对x∈∑\*的每一个可能的计算都结束在F中的状态，则M接受x。注意，相反的，普通的NFA只需有一个计算结束在接受状态，就接受这个字符串。证明：全路径NFA识别正则语言。

证明：一个DFA一定是一个全路径NFA。所以下面只需证明，任意全路径NFA，都有一个与之等价的DFA。

设有一全路径NFA N=( Q, Σ, δ, q0, F)，构造一个新与N等价的全路径NFA

N1＝( Q1, Σ, δ1, q0, F), 其中Q1=Q∪{s}, s∉Q。对于任意q∈Q1, a∈Σε

δ(q,a)， q ≠ s, 且δ(q,a) ≠∅;

δ1(q,a) = {s}, q ≠ s, 且δ(q,a) =∅;

{s}, q=s.

现在来构造一个与全路径NFA N1等价的DFA M＝(Q2, Σ, δ2, q1, F2). 其中

1) Q2=Power(Q1), 即Q1的所有子集组成的集合(也即Q1的幂集)；

2) 定义函数E: Q2🡪Q2为：对任意R∈Q2, ；

3) q1=E(q0);

4) 对于任意的R属于Q2, a属于Σ, .

5) F2={ R∈Q2 | R⊆F}。

综上所述，DFA等价于全路径NFA。

1.40如果存在字符串z使得xz=y,则称字符串x是字符串y的前缀。如果x是y的前缀且x≠y,则称x是y的真前缀。下面每小题定义一个语言A上的运算。证明：正则语言类在每个运算下封闭。

a) NOPREFIX(A)={ω∈A|ω的任意真前缀都不是A的元素}

b)NOEXTEND(A)={ ω∈A|ω不是A中任何字符串的真前缀}

证明：假设DFA M=( Q, Σ, δ, q0, F)识别语言A。

a) 构造NFA N1=( Q, Σ, δ1, q0, F)识别语言NOPREFIX(A)。其中，

对任意q∈F, a ∈Σε, 

所以，即NOPREFIX(A)为正则语言，亦即正则语言类A在NOPREFIX(A)运算下封闭。█

b) 构造NFA N2=( Q, Σ, δ, q0, F2)识别语言NOEXTEND(A)。F2构造如下：

对M中的任一接受状态qi, 若存在一条从它出发到达某接受状态(含本身)的路径，则将状态qi改为非接受状态; 最后剩下的接受状态集记为F2. 所得的NFA N2即识别NOEXTEND(A)。所以，NOEXTEND(A)为正则语言，亦即正则语言类A在运算NOEXTEND(A)下封闭。█

1.48 证明：构造NFA N如下：

0

1

0

1

1

0

1

1

0

0

1

10

0

01

由于该NFA识别D，故D是正则语言。

1.50参见练习1.24中给出的有穷状态转换器的非形式定义。证明不存在FST对每一个输入w能够输出wR，其中输入和输出的字母表为{0,1}。

证明：假设存在一台FST对每个输入w能够输出wR。

则对于输入串w1 =100, w2=101.

该FST可分别输出w1R=001，w2R=101，于是对于它的起始状态和输入字符1，会输出1和0两个不同字符，这与FST是确定性有穷自动机相矛盾。

所以，不存在一台FST对每个输入w能够输出wR。

1.51

证明: 1) 自反性。即对任意字符串x，x≡Lx。这是因为对于每个字符串z均有xz和xz同时是或不是L的成员。

2) 对称性。即对任意字符串x和y, 若x≡Ly，则y≡Lx。这是因为若x≡Ly，则对于每个字符串z, xz和yz同时是或不是L的成员，那么yz和xz同时是或不是L的成员, 于是y≡Lx。

3) 传递性。即对任意字符串x,y和z, 若x≡Ly且y≡Lz，则x≡Lz。这是因为对任意字符串u, 由x≡Ly知xu和yu同时是或不是L的成员, 由y≡Lz知yu和zu同时是或不是L的成员, 所以xu和zu同时是或不是L的成员, 此即x≡Lz。

综上所述，≡L是自反的，对称的，传递的，所以是一个等价的关系。

1.53 令Σ={0，1，+，=}和

ADD={ x=y+z | x,y,z是二进制整数，且x是y与z的和}

证明ADD不是正则的。

证明：假设ADD是正则的。设P是泵引理给出的关于ADD的泵长度。令s为1P=10P-1+1P-1。于是s能够被划分成xyz，且满足泵引理的条件。根据条件3，y=1i, i>0. 所以xyyz为1P+i=10P-1+1P-1∉ADD。故ADD不是正则的。

1.54证明：语言F={aibjck | i,j,k≥0且若i=1，则j=k}满足泵引理的3个条件，虽然它不是正则的。解释这个事实为什么与泵引理不矛盾。

证明：对任意正数p>1，设S是F中的一个成员且S的长度不小于p。

将S分成3段S=xyz。

1. i=0，j=0. 此时S=ck (k>0)。

取x=ε, y=c, z=ck-1. 则对任意i≥0, xyiz∈F。

1. i=0，j>0. 此时S=bjck(j>0, k≥0).

取x=ε, y=b, z=bj-1ck. 则对任意i≥0, xyiz∈F。

1. i=1. 此时S=abjck(j≥0, k≥0).

取x=ε, y=a, z=bjck. 则对任意i≥0, xyiz∈F。

1. i=2. 此时S= a2bjck(j≥0, k≥0).  
   取x=ε, y=a2, z=bjck. 则对任意i≥0, xyiz∈F。
2. i>2. 此时S= aibjck(i≥3, j≥0, k≥0).

取x=ε, y=a, z=ai-1bjck. 则对任意i≥0, xyiz∈F。

综上所述，语言F满足泵引理的3个条件。这一事实不与泵引理矛盾是因为泵引理是正则语言的必要不充分条件。

1.55 求最小泵长度。

a. 0001\*的最小泵长度为4.

因为对任何s 若|s|=3则s只含有0，不能被抽取。

若|s|≥4时，把s划分为x=000 y=1 z为其余部分，则xyiz∈0001\*。

b. 0\*1\*最小泵长度为1.

若|s|≥1时,S分两种情况：

* 1. S以0开头,令x=ε, y=0, z为其余则xyiz∈0\*1\*.
  2. S以1开头,令x=ε, y=1, z为其余则xyiz∈0\*1\*.

c. (01)\*最小泵长度为2.

若|s|≥2时,令x=ε, y=01, z为其余, 则xyiz∈(01)\*.

d. 01，其最小泵长度为3。  
因为语言中只有有限个字符串时，任何一个字符串都不能被抽取。所以有限语言的泵长度为其最长字符串的长度加1。此时没有比泵长度长的字符串，前提假所以命题真。

e. ε，其最小泵长度为1.

理由类似于d中所述。

2.39 证明：设对于每一个k>1，Ak={ w | w包含子串1k-1}⊆{1}\*，下面证明Ak能被一台k个状态的DFA识别，而不能被只有k－1个状态的DFA识别。

显然，Ak能被k个状态的DFA

M=({q1,q2,…,qk}, {1}, δ, q1, {qk}).

识别, 其中δ(qi,1)=qi+1(i=1,2,…,k-1), δ(qk,1)=qk。

假设AK可以被只有k-1个状态的DFA　M1识别。

考虑这样一个输入串s=1k-1，设M识别s的状态序列是r1, r2,…rk,由于M的状态只有k-1个，根据鸽巢原理，r1,r2,…,rk中必有两个重复的状态，假设是ri=rj (0≤i<j≤k)，此时可知，字符串x=1j-i把M从ri带到了rj。由于ri,rj是同一个状态，所以去掉s中的x子串，M仍可识别s的剩余部分，即M识别1k-1-(j-i)，这与M1识别Ak相矛盾。故Ak不能被只有k-1个状态的DFA M识别。

所以，对于每一个k>1，存在AK，使得AK能被K个状态的DFA识别，而不能被只有k-1个状态的DFA识别。

**第二章**

2.1 略。

2.2 a. 利用语言A={ambncn | m,n≥0}和A={anbncm | m,n≥0}以及例3.20，证明上下文无关语言在交的运算下不封闭。

b. 利用(a)和DeMorgan律(定理1.10)，证明上下文无关语言在补运算下不封闭。

证明：a.先说明A,B均为上下文无关文法，对A构造CFG C1

S→aS|T|ε

T→bTc|ε

对B,构造CFG C2

S→Sc|R|ε

R→aRb

由此知 A,B均为上下文无关语言。

但是由例3.20, A∩B={anbncn|n≥0}不是上下文无关语言，所以上下文无关语言在交的运算下不封闭。

b.用反证法。假设CFL在补运算下封闭，则对于(a)中上下文无关语言A,B，,**也为**CFL，且CFL对并运算封闭，所以**也为**CFL，进而知道**为**CFL，由DeMorgan定律**＝**A∩B，由此A∩B是CFL,这与(a)的结论矛盾，所以CFL对补运算不封闭。

2.3 略。

2.4和2.5 给出产生下述语言的上下文无关文法和PDA，其中字母表Σ={0,1}。

1. {w | w至少含有3个1}

ε,1→ε

1, ε→1

0, ε→ε

ε,1→ε

ε,1→ε

S→A1A1A1A

A→0A|1A|ε

1. {w | w以相同的符号开始和结束}

S→0A0|1A1

1,ε→1

1,ε→ε

0,ε→ε

0,ε→0

1,1→ε

0,0→ε

A→0A|1A|ε

1. {w | w的长度为奇数}

1,ε→ε

0,ε→ε

1,ε→ε

0,ε→ε

S→0A|1A

A→0B|1B|ε

B→0A|1A

1. {w | w的长度为奇数且正中间的符号为0}

S→0S0|1S1|0S1|1S0|0

1,ε→0

0,ε→ε

0,ε→0

1,0→ε

0,0→ε

ε,ε→$

ε,$→ε

1. {w | w中1比0多}

1,ε→1

ε,1→ε

0,ε→0

ε,1→ε

ε,ε→$

ε,$→ε

1,0→ε

0,1→ε

S→A1A

A→0A1|1A0|1A|AA|ε

1. {w | w=wR}

S→0S0|1S1|1|0

1,ε→1

0,ε→ε

0,ε→0

1,1→ε

0,0→ε

ε,ε→$

ε,$→ε

1,ε→ε

ε,ε→ε

1. 空集

S→S

2.6 给出产生下述语言的上下文无关文法：

1. 字母表{a,b}上a的个数是b的个数的两倍的所有字符串组成的集合。

S→bSaSaS|aSbSaS|aSaSbS|ε

b．语言{anbn|n≥0}的补集。见问题3.25中的CFG:

S→aSb|bY|Ta

T→aT|bT|ε

c．{w#x | w, x∈{0,1}\*且wR是x的子串}。

S→UV

U→0U0|1U1|W

W→W1|W0|#

V→0V|1V|ε

d．{x1#x2#…#xk|k≥1, 每一个xi∈{a,b}\* , 且存在i和j使得xi＝xjR}。

**S**→UVW

U→A|ε

A→aA|bA|#A|#

V→aVa|bVb|#B|#

B→aB|bB|#B|#

W→B|ε

2.8 证明上下文无关语言类在并,连接和星号三种正则运算下封闭。

1. A∪B

方法一：CFG。设有CFG G1＝(Q1,Σ,R1,S1)和G2=(Q2,Σ,R2,S2)且L(G1)=A, L(G2)=B。构造CFG G=(Q,Σ,R,S0)，其中

Q= Q1∪Q2∪{S0}, S0是起始变元，R= R1∪R2∪{S0→ S1|S2}.

方法二：PDA。

设P1=(Q1,Σ,Γ1,δ1,q1,F1)识别A，P2=(Q1,Σ,Γ2,δ2,q2,F2)是识别B。

则如下构造的P=(Q,Σ,Γ,δ,q0,F)识别A∪B，其中

1. Q=Q1∪Q2∪{q0}是状态集，
2. Γ＝Γ1∪Γ2，是栈字母表，
3. q0是起始状态，
4. F＝F1∪F2是接受状态集，
5. δ是转移函数，满足对任意q∈Q, a∈Σε,b∈Γε

δ(q,a,b)=

1. 连接AB

方法一：CFG。设有CFG G1＝(Q1,Σ,R1,S1)和G2=(Q2,Σ,R2,S2)且L(G1)=A, L(G2)=B。构造CFG G=(Q,Σ,R,S0)，其中

Q= Q1∪Q2∪{S0}, S0是起始变元，R= R1∪R2∪{S0→ S1S2}.

方法二：PDA。

设P1=(Q1,Σ,Γ1,δ1,q1,F1)识别A，P2=(Q1,Σ,Γ2,δ2,q2,F2)是识别B，而且P1满足**在接受之前排空栈**(即若进入接受状态，则栈中为空)这个要求。

则如下构造的P=(Q,Σ,Γ,δ,q1,F)识别A∪B，其中

1. Q=Q1∪Q2是状态集，
2. Γ＝Γ1∪Γ2，是栈字母表，
3. q1是起始状态，
4. F＝F1∪F2是接受状态集，
5. δ是转移函数，满足对任意q∈Q, a∈Σε,b∈Γε

δ(q,a,b)=

1. A\*

方法一：CFG。设有CFG G1＝(Q1,Σ,R1,S1)，L(G1)=A。构造CFG G=(Q,Σ,R,S0)，其中Q= Q1 ∪{S0}, S0是起始变元，R= R1∪{S0→S0S0|S1|ε}.

方法二：PDA。

设P1=(Q1,Σ,Γ1,δ1,q1,F1)识别A，而且P1满足**在接受之前排空栈**(即若进入接受状态，则栈中为空)这个要求。

则如下构造的P=(Q,Σ,Γ,δ,q1,F)识别A\*，其中

1. Q=Q1∪{q0}是状态集，
2. Γ＝Γ1，是栈字母表，
3. q0是起始状态，
4. F＝F1∪{q0}是接受状态集，
5. δ是转移函数，满足对任意q∈Q, a∈Σε,b∈Γε

δ(q,a,b)=

2.9 证明在3.1节开始部分给出的文法G2中，字符串the girl touches the boy with the flower有两个不同的最左派生，叙述这句话的两个不同意思。

<句子>

⇒<名词短语><动词短语>

⇒<复合名词><动词短语>

⇒<冠词><名词><动词短语>

⇒a\_<名词><动词短语>

⇒a\_girl\_<动词短语>

⇒a\_girl\_<复合名词>

⇒a\_girl\_<动词>< 名词短语>

⇒a\_girl\_touches\_< 名词短语>

⇒ a\_girl\_touches\_<复合名词><介词短语>

⇒a\_girl\_touches\_<冠词><名词><介词短语>

⇒a\_girl\_touches\_the\_<介词><复合名词>

⇒a\_girl\_touches\_the\_boy\_<介词短语>

⇒a\_girl\_touches\_the\_boy\_<介词><复合名词>

⇒a\_girl\_touches\_the\_boy\_with\_<复合名词>

⇒a\_girl\_touches\_the\_boy\_with\_<冠词><名词>

⇒a\_girl\_touches\_the\_boy\_with\_the\_<名词>

⇒a\_girl\_touches\_the\_boy\_with\_the\_flower

含义是 ：女孩碰这个带着花的男孩

<句子>

⇒<名词短语><动词短语>

⇒<复合名词><动词短语>

⇒<冠词><名词><动词短语>

⇒a\_<名词><动词短语>

⇒a\_girl\_<动词短语>

⇒a\_girl\_<复合动词><介词短语>

⇒a\_girl\_<动词>< 名词短语><介词短语>

⇒a\_girl\_touches\_< 名词短语><介词短语>

⇒a\_girl\_touches\_<冠词><名词><介词短语>

⇒a\_girl\_touches\_the\_< 名词><介词短语>

⇒a\_girl\_touches\_the\_boy\_<介词短语>

⇒a\_girl\_touches\_the\_boy\_<介词><复合名词>

⇒a\_girl\_touches\_the\_boy\_with\_<复合名词>

⇒a\_girl\_touches\_the\_boy\_with\_<冠词><名词>

⇒a\_girl\_touches\_the\_boy\_with\_the\_<名词>

⇒a\_girl\_touches\_the\_boy\_with\_the\_flower

含义是： 女孩用花碰这个男孩

2.10 给出产生语言A={aibjck| i,j,k≥0且或者i=j或者j=k}的上下文无关文法。你给出的文法是歧义的吗？为什么？

解：下面是产生A的一个CFG:

S→UV|AB

U→aUb|ε

V→cV|ε

A→aA|ε

B→bUc|ε

这个CFG是歧义的，因为字符串abc有如下两种不同的最左派生：

S⇒UV⇒aUbV⇒abV⇒abcV⇒abc

S⇒AB⇒aAV⇒aV⇒abVc⇒abc

2.11 给出识别2.10中语言A的下推自动机的非形式描述。

解：其非形式描述为：

此PDA有两个非确定性的分支：一个分支先读a，并且每读一个a将一个a推入栈中，当碰到b时，每读一个b从栈中弹出一个a，若没有a可弹出则拒绝，最后读c且不改变栈中的内容，若此时栈为空则接受。另一个分支也是先读a，但不改变栈中内容，当碰到b时，每读一个b将一个b推入栈中，再读c, 每读一个c从栈中弹出一个b，若没有a可弹出则拒绝，若c读完后栈为空则接受。开始时，读输入串的字符，非确定性的选择一个分支运行，若有一个分支接受则接受，否则拒绝。

2.14 设有上下文无关文法G:

S→TT|U

U→0U00|#

T→0T|T0|#

1. 用普通的语言描述L(G)。
2. 证明L(G)不是正则的。

解： a. A={0i#0j#0k | i, j, k≥0}∪{0i#02i | i≥0}。

b. 取s=0p#02p, 则对于任意划分s=xyz(|xy|≤p, |y|>0), xynz=0p+i#02p∉A, 所以不是正则的。

2.15 用定理2.6中给出的过程，把下述CFG转换成等价的乔姆斯基范式文法。

A→BAB|B|ε

B→00|ε

解：添加新起始变元S0, 去掉B→ε

S0→A S0→A

A→BAB|B|ε A→BAB|AB|BA|B|ε

B→00|ε B→00

去掉A→ε, 去掉A→B

S0→A S0→A

A→BAB|AB|BA|B|BB A→BAB|AB|BA|00|BB

B→00 B→00

去掉S0→A, 添加新变元

S0→BAB|AB|BA|00|BB S0→VB|AB|BA|UU|BB

A→BAB|AB|BA|00|BB A→VB|AB|BA|UU|BB

B→00 B→UU

V→BA

U→0

2.x 先说明如何把正则表达式直接转换成等价得上下文无关文法，然后利用问题xxx的结果，给出每一个正则语言都是上下文无关文法的一个证明。

解：设有正则表达式T，将其转化为上下文无关文法。

1. 首先添加S→T，且令S为起始变元。
2. 根据T的不同形式，按如下方式添加规则
3. 若T＝a∈Σ， 则在规则集中添加T→a,
4. 若T＝ε， 则在规则集中添加T→ε,
5. 若T＝∅， 则在规则集中添加T→T，
6. 若T＝A∪B， 则在规则集中添加T→A|B，
7. 若T＝A·B, 则在规则集中添加T→AB，
8. 若T＝A\*, 则在规则集中添加T→A,和A→AA|ε

3) 若有新添加的变元，则根据其所对应的表达式转第2步，直到没有新的变元出现。

下面来证明每一个正则语言都是上下文无关的

由正则语言与正则表达式的等价性可知：对于一个正则语言都存在一个描述他的正则表达式，由前述讨论可知，这个正则表达式可转换为一个与之等价的上下文无关的文法。因此，此正则语言能由这个上下文无关文法产生。所以正则语言是上下文无关的。

2.18 a. 设C是上下文无关语言，R是正则语言。证明语言C∩R是上下文无关的。

1. 利用a证明语言

A={w | w∈{a,b,c}\*, 且含有数目相同的a,b,c}

证明：a. 设P=(Q1,Σ,Γ,δ1,q1,F1)是一个识别C的PDA，N=(Q2,Σ,δ2,q2,F2)是识别R的一台NFA。下面构造识别C∩R的PDA如下S=(Q,Σ,Γ,δ,q0,F):

* + 1. Q=Q1×Q2, 是状态集，
    2. q0=(q1,q2)是起始状态，
    3. F= F1×F2, 是接受状态集，
    4. δ是转移函数，满足对任意q∈Q1,r∈Q2,a∈Σε,b∈Γε,

δ((q,r),a,b)={((q’,r’),c) | q’∈δ1(q,a), (r’,c)∈δ2(r,a,b)}.

b. A∩a\*b\*c\*={anbncn|n≥0}, 若A是上下文无关的，则由a中命题知{anbncn|n≥0}也是上下文无关的，矛盾。

2.26 证明：设G是一个Chomsky范式CFG，则对任一长度≥2的字符串w∈L(G), w的任何派生恰好有2n－1步。

证明：(用数学归纳法)当n=2时，对于Chomsky范式CFG，长度为2的字符串必由3步派生，此时命题为真。

假设当2≤n≤k时命题为真。当n=k+1时，对于长度为k+1的字符串s=s1s2…sk+1，存在S0的一个直接派生S0⇒AB，使得A产生子串s1s2…sp，B产生子串sp+1sp+2…sk+1，其中1≤p≤k+1。由归纳假设，A产生子串s1s2…sp需要2p-1步派生，而B产生子串sp+1sp+2…sk+1需要2(k+1-p)-1步派生。因此，S0产生字符串s共需2(k+1)-1步派生。即当n=k+1时命题亦真。

因此，由G产生长度为n≥2的字符串需要2n-1派生。

2.30 用泵引理证明下述语言不是上下文无关的：

1. {0n1n0n1n | n≥0}

假设语言上下文无关，设p为泵长度，取S=0p1p0p1p.由泵引理知，S可以划分为uvxyz且满足泵引理条件。考察子串vxy,首先，它一定横跨S的中点，否则，若vxy位于S的前半段，由于|vxy|≤p, 则uv2xy2z中1将是后半段字符串的第一个字符，即不可能是0n1n0n1n的形式。同理，vxy也不可能位于S后半段的位置上。但是,若vxy跨越S的中点时，将S抽取成uxz，它形如0p1i0j1p,i，j不可能都为p，即uxz不可能是0n1n0n1n的形式。

综上，可知S不能被抽取，因此，该语言不是上下文无关的。

b. {0n#02n#03n | n≥0}

假设语言上下文无关，设p为泵长度，取S=0p#02p#03p, 由泵引理知，S可以划分为uvxyz且满足泵引理条件。考察子串vxy。首先，v,y中不能有#，否则S抽取成uxz后，其中#个数≤1。由条件3，vxy或者位于第2个#之前，或者位于第1个#之后。下面讨论这两种情况：

1. 此时，uv2xy2z中第3部分0的个数保持不变，而前面部分的0的个数至少有一个增加，因此，uv2xy2z不在该语言中。
2. 此时，uv2xy2z中第1部分0的个数保持不变，而第2，3部分0的个数至少有一个增加，也有uv2xy2z不在该语言中。

因此，该语言不是上下文无关的。

c. {w#x | w,x∈{a,b}\*, w是x的子串}

假设该语言上下文无关，设p为泵长度，取S=0p1p#0p1p, 由泵引理知，S可以划分为uvxyz且满足泵引理条件。显然，v,y中不包含#，不然抽取后的S=uxz中将不含#从而不在该语言中。

子串vxy不能落在#的左侧，否则uv2xy2z中#左侧的字符串长度将大于右边。

子串vxy不能落在#的右侧，否则uxz中#左侧的字符串长度也会大于右边。

现在假设#∈x，则必存在不全为0的i,j，使得vy=1i0j，下面分两种情况考虑：

1. j≠0, 则uxz=0p1p-i#0p-j1p不在该语言中
2. j=0, 则uv2xy2z中#左侧的字符串长度大于右边，不在该语言中。

因此，该语言不是上下文无关的。

d. {x1#x2#…#xk | k≥2, 每一个xi∈{a,b}\*, 且存在xi＝xj}

假设该语言上下文无关，设p为泵长度，取S=apbp#apbp, 由泵引理知，S可以划分为uvxyz且满足泵引理条件。显然，v,y中不包含#，不然抽取后的S=uxz中将不含#从而不在该语言中。

子串vxy不能落在#的左侧，否则uv2xy2z中#左侧的字符串长度将大于右边。

子串vxy不能落在#的右侧，否则uxz中#左侧的字符串长度也会大于右边。

于是vxy跨越#两侧.此时，S经抽取成uxz后，具有apbi#ajbp的形式，其中，i,j不全为p。因此，uxz不在该语言中。

综上该语言不是上下文无关的。

2.34 考虑语言B=L(G), 其中G是练习3.13中给出的文法。定理3.19关于上下文无关语的泵引理称存在关于B的泵长度p。p的最小值等于多少？要求给出证明。

证明：p的最小值为4。令D={0i#0j#0k | i, j, k≥0}, E={0i#02i | i≥0}, 则L(G)=D∪E。

L(G)中长度为1的字符串仅有#,不能被抽取。

L(G)中长度为2的字符串仅有#,也不能被抽取。

D中长度≥3的字符串必含有0，所以一定可以被抽取。

E中长度≥3的字符串也可以被抽取(E中没有长度等于3的字符串)。只需令uvxyz分别为v=0,x=#,y=00, u,z为两边其余的字符串即可。注意到此时|vxy|＝4。

但是泵长度不能等于3。因为若p=3，则要求|vxy|≤3，但是对于E中的字符串来说，每在#左边抽取1个0，则同时必须在#右边抽取2个0，即|vy|≥3，又x必须包含#, 所以任何有效的抽取必有|vxy|≥4。

综上所述，p的最小值为4。

2.35 设G是一个含有b个变元的乔姆斯基范式CFG。证明：如果G产生某个字符串使用了至少2b步的派生，则L(G)是无穷的。

证明：设G产生字符串S需要至少2b步。

T

R

R

u

v

x

y

z

由于一个分支点(变元)对应一次派生，所以S的语法树τ至少有2b个分支点。又由于在乔姆斯基范式下，一个分支点至多有两个子分支点(这意味着层数≤n的结点个数≤2b－1)，从而τ的由分支点组成的子树的高度大于等于b+1。τ有一条路径上分支点(变元)个数≥b+1。由鸽巢原理：必有某个变元R在该路径上出现至少两次。不妨假设R出现在路径上最下面的b+1个变元中。

按上图所示，将S划分为uvxyz，在每一个R的下面有一棵子树，它产生S的一部分。上面的R有一棵较大的子树，产生vxy，下面的R有一棵较大的子树，产生x，由于这两棵子树由同一个变元产生，因而可以相互替换。这里采用如下操作：反复用较大子树去替换较小子树，则我们得到对于任意n>1, uv nxy nz∈L(G)。

所以若我们能证明v,y不全为ε，则L(G)是无穷的。

事实上，由乔姆斯基范式的特点，上面一个R的派生选择的必为R→AB形式的规则，而且不可能有Aε和Bε，所以v,y不全为ε。从而命题得证。

2.26 给出一个语言，它不是上下文无关的、但满足泵引理的三个条件。要求给出证明。

解：令A＝{aibjckdl | i,j,k,l≥0, 且当i=1时，j=k=l}，则：

* 1. A满足泵引理的三个条件。取泵长度p＝1。对A中任意长度大于1的字符串s= aibjckdl ，分情况可以如下抽取：

若i=1,则j=k=l, 取v=a,uxy=ε，z= bjcjdj, 则对∀n≥0，uvnxynz= aibjcjdj ∈A；

若i≥2，且j,k,l不同时为0，不妨设j≠0，取u=ai,v=b,xy=ε，z= bj-1ckdl, 则对∀n≥0，uvnxynz= aibj-1+nckdl ∈A；

若i≥2，且j=k=l=0, 取u=ai-1,v=a,xyz=ε, 则对∀n≥0，uvnxynz= ai-1+n∈A；

若i=0，则j,k,l不同时为0，不妨设j≠0，取v=b,uxy=ε，z= bj-1ckdl, 则对∀n≥0，uvnxynz=bj-1+nckdl ∈A。

* 1. A不是上下文无关语言。事实上，若A为上下文无关语言，另取正则语言R=ab\*c\*d\*，由问题3.17可知，L∩R是上下文无关的。但这与L∩R={abncndn | n≥0}不是一个上下文无关语言矛盾。因此，A不是上下文无关的。

**第三章**

3.3 修改定理3.10以得到推论3.12的证明，即证明一个语言是可判定的当且仅当有非确定的TM判定它。

证明：若M是一个确定型判定器则，则M也是一个非确定型判定器。

现在设N是一个非确定的判定器，将构造一个与之等价的确定型判定器M。模拟过程使用深度搜索。

设N的不确定性分支的最大个数为b。

M有三个带：一个输入带，一个工作带，一个地址带。M按深度优先方式搜索N的不确定计算分支树。

M= “输入w，

1. 初始化，第一带上为w, 第二带为空，第三带为1;
2. 将第一带的内容复制到第二带上，
3. 按当前地址位数字选择N的一个不确定性分支，在第二带上模拟N运行一步；
4. 若当前地址位为i<b，且当前选择无效或按当前选择进入拒绝状态，则将当前地址位改为i+1, 转第2步；
5. 若当前地址位为i＝b，且当前选择无效或按当前选择进入拒绝状态，则将当前地址位改为空格, 左移并将当前地址位改为空格直到找到一个地址位其值<b，将当前地址位改为i+1, 转第2步；若到了地址带的最左端仍有当前地址位为b，则拒绝；
6. 若N进入接受状态，则接受；否则，右移一格，将空格上写入1，转第三步。”

由于N是非确定型判定器，所以对任意输入，由本题的提示M一定会停机。

3.4 给出枚举器的形式定义。

解：枚举器E=(Q,Σ,Γ,δ,q0,qaccept,qreject), 其中转移函数δ为：

δ：Q×Γ→Q×Γ×{L,R}×Σ\*

δ (q,a)=(r,b,s1,c)

表示若E处于状态q,且在工作带上读到a，则状态转移为r，当前格改写为b并按s1作相应左或右移，打印带上写下字符串c，其中若c等于ε，则不打印。

另外E的起始格局只能是q0v，这里v表示一个空格。

3.x 检查图灵机的形式定义，回答下列问题并解释你的推测：

1. 图灵机能在它的带子上写下空白符吗

b．带字母表Γ和输入字母表Σ能相同吗？

c．图灵机的读写头能在连续的两步中处于同一个位置吗？

d．图灵机能只包含一个状态吗？

解：

a．能。因为空白符属于带字母表Γ；

b．不能。因为空白符不属于输入字母表Σ；

c．能。当读写头处于左端点时，如果转移是向左转移，因为不准机器从带的左端点移出，所以下一个格局读写头仍在左端点。

d．不能。因为qaccept≠qreject，至少应有两个状态。

3.6 解：因为M不一定是判定器，可能会在运行某个si时不停机，则L(M)中按字典序大于si的字符串不可能被枚举出来。

3.7 解：因为图灵机的一个本质要求是一步之内，只能做有限的工作，而第1)步中取所有可能的值，这有无限多种情况。

3.8

构造具有3条带的图灵机。

对于问题a.

1. w 先读入第一条带，然后读到0就把0写入第2条带，读到1就把1写入第3条带，直到读到空格为止。
2. 然后把3个读写头都返回到最左边。
3. 开始读第2条带和第3条带，每次都是读一个字符，如果同时遇上空格符，则接收，否则拒绝。

对于问题b:

1. 和a的第1步相同。
2. 和a的第2步相同。
3. 每次读带3的一个字符就读带2的两个字符，如果同时遇上空格符，就接收，否则拒绝。

对于问题c:

1. 和a的第1步相同。
2. 和a的第2步相同。
3. 每次读带3的一个字符就读带2的两个字符，如果**同时**遇上空格符，就**拒绝**，否则接受。

3.x由题知,1-pda代表一个栈的下推自动机,2-pda代表两个栈的下推自动机。如果能用2-pda模拟一个图灵机,而我们已经知道图灵机比下推自动机强大，那么就有2-pda比1-pda更强大。设有TM S。下面构造2-pda P，且记其两个栈分别为A，B：

P＝“对于输入w,

1) 将w读入栈A，再全部从栈A中依次弹出并读入栈B。

2) 模拟S在w上运行。记录S的当前状态，并且栈B的栈顶字符为S的读写头所指方格的字符：

* 1. 若S执行一个右移δ(q,a)=(r,b,R)，则将栈B的栈顶字符a替换为b，弹出b，推入栈A，记录S的当前状态r。
  2. 若S执行一个左移δ(q,a)=(r,b,L)，首先将栈B的栈顶字符a替换为b；若栈A不空，则将栈A弹出一个字符推入栈B；若栈A为空(对应于S处于工作带最左端)，则不作栈操作。记录S的当前状态r。

3) 若S进入接受状态，则进入接受状态。”

由上知.我们用2-pda模拟了图灵机,所以2-pda比1-pda强大.

下面用一个四带TM S来模拟一个3-pda P，要求P**进入接受状态之前排空栈，并且或者推入或者弹出**：

记P的三个栈为A,B,C。S的四个带，第一带用来模拟P的输入带，第二,三,四带分别模拟栈A,B,C：

S＝“对于输入字符串w，

* 1. 初始化，第一带放入w，第二，三，四带为空。
  2. 模拟P分别在四个带上按如下方式动作：
     1. 若P在输入带上读一个非空字符，则读第一带字符并右移一格；

若P在输入带上读一个ε，则在第一带上不读字符，且读写头保持不动。

* + 1. 栈A,B,C中若有弹出，则在相应带上当前格改写为空格，并左移一格；若有推入a，则在相应带上右移一格，并写入a。

3) 若P进入接受状态，则接受。”

3.10证明双无限带图灵机识别图灵可识别语言。

证明思路：利用双无限单带图灵机模拟普通单带图灵机时，只需要设计一个左端点。我们的证明是让它在第一格左边标记“$”作为左端点。

证明：

首先用单带双无限带图灵机模拟普通单带图灵机：

设有普通图灵机M1＝( Q1, Σ, Γ1, δ1, q0, qaccept, qreject )。 下面构造与之等价的单带双无限带图灵机M2＝( Q2, Σ, Γ2, δ2, qs, qaccept, qreject )，其中

Q2=Q1∪{qs,qt}, qs为新的起始状态；Γ2＝Γ1∪{$}.

对任意q∈Q2, a∈Γ2,



再用普通单带图灵机模拟单带双无限带图灵机：

设有单带双无限图灵机M1＝(Q,Σ,Γ,δ,q0,qaccept,qreject)，下面构造普通单带图灵机M2：

M2=“输入串w，

1. 将带上字符串改写为$w, 将读写头放在w的第一个字符上，
2. 按照M1的转移函数运行，
3. 每当读写头移到了$上，就将$右边的所有字符右移一格，并在$右边的方格里写上空格符，再将读写头放在这个方格上，转第二步，
4. 若进入接受状态，则接受；若进入拒绝状态则拒绝。”

也可以用普通双带图灵机模拟单带双无限带图灵机：

设有单带双无限图灵机M1＝(Q,Σ,Γ,δ,q0,qaccept,qreject)，下面构造普通双带图灵机M2：

M2=“输入串w，

1. 在第一个带上放上$, 读写头放在$上；

第二个带子上放入#w, 读写头放在第二个方格上。

1. 当第一带读写头位于$上，而第二带读写头不在＃上时，在第二带上按照M1的转移规则运行，直到停机或读写头移到#上。若进入接受状态则接受，若进入拒绝状态则拒绝。若读写头移到#上，则将第一带上读写头右移一格。
2. 当第二带读写头位于#上，而第一带读写头不在$上时，在第一带上按照M1的转移规则运行，但是每一步要将读写头移动方向反向，直到停机或读写头移到$上。若进入接受状态则接受，若进入拒绝状态则拒绝。若读写头移到$上，则将第二带上读写头右移一格，转第二步。”

3.11**只写一次**图灵机是一个单带图灵机，它在每个带方格上最多只能**改变**其内容一次(包括带上得输入区域)。证明图灵机模型的这个变形等价于普通的图灵机模型。

**证明**：普通单带图灵机总是可以模拟**只写一次**图灵机。下面说明怎样用一个只写一次TM T 模拟一个普通单带TM S。

T=“对于输入w=w1w2…wn,

1) 在w1w2…wn上并模拟S运行。

2) 每当S要改写工作带时(例如设S要将当前方格内容改写为b，并且左移或右移一格)，T按如下方式动作：

a. 将当前方格改写为“b\*”，在带子右边第一个空格写下“#”。

b. 将“#”左边的字符抄写到“#”右边(注：每抄写一个字符，需要将此格字符改写一次以作上标记，但是“b\*”不用再作另外标记，而且将“b\*”抄写为“b~”)。

c. 找到带有标记“~”的字符，再模拟S左移或右移一格。

然后继续模拟S。

3) 若S接受则接受；若S拒绝则拒绝。”

3.12

对于普通图灵机N，构造与之等价的带左复位的图灵机E：

E＝“对于输入w,

1. 在w上模拟N的一步动作：  
   若N要将当前格由a改为b且右移，则照此动作；  
   若N要将当前格由a改为b且左移，则将当前格由a改为b#且复位：
   1. 以~标记当前位，右移一格；
   2. 若当前位没有标记#，则复位，右移直到找到标记有~的字符，去掉此标记，右移一格转步(a);
   3. 若当前位有标记#，则去掉标记#并复位，右移或直到找到标记有~的字符，去掉此标记；
2. 若N进入接受状态，则接受；若进入拒绝状态，则拒绝；否则转第一步。”

L(E)=L(N)。因此左复位的图灵机识别图灵可识别语言类。

3.13 以停留代替左移的图灵机识别什么语言类？

解：正则语言类。首先一个DFA可以被一个以停留代替左移的图灵机模拟。下面证明一个以停留代替左移的图灵机S=(Q,Σ,Γ,δ,q1,qaccept,qreject)，可以被一个NFA M=(Q1,Σ,δ1,q0,F)模拟。

M的构造如下：

令Q1= Q×Γε∪{qend}， F={qend}，q0=(q1,ε),

1. 对任意q∈Q-{qaccept,qreject}, a∈Σ,令

δ1( (q,ε) , a )={(q,a)};

1. 对任意q∈Q-{qaccept,qreject}, a∈Γ,

若有转移δ(q,a)=(r,b,R)，则令δ1( (q,a), ε )={(r, ε)};

若有转移δ(q,a)=(r,b,S)，则令δ1( (q,a), ε )={(r,b)};

1. 对任意a∈Γε, b∈Γ，

令δ1( (qaccept,a) , b )={(qaccept, ε)};

1. 对任意q∈Q，令Sq=(Q,Σ,Γ,δ,q,qaccept,qreject),

若ε∈L(Sq)，则令δ1( (q,ε) , ε )={qend}。

其中第一类转移是用来读字符的。

第二类转移是用来模拟S的读写头的移动的。由于S没有左移，所以右移一格之前改写的内容b可以舍去。

第三类转移用来处理当S已进入接受状态，但是字符串还没有读完的情况的。即先由δ1( (qaccept,a) , b )={ (qaccept,ε) }读完所有剩余字符，再由第四类转移中的δ1((qaccept,ε),ε)={qend}进入接受状态。

第四类转移用来处理S的读写头移出输入区域的情况的，在这种情况下，S是进入接受状态，还是进入拒绝状态，还是不停机，完全取决于进入空白区域时的状态q：即若ε∈L(Sq)，则S最终会进入接受状态；若ε∉L(Sq)，则S最终会进入拒绝状态或不停机。

3.15 证明可判定语言类在下列运算下封闭。

a. 并。

证明：设M1，M2为识别可判定语言A1,A2的判定器。构造图灵机M：

M＝“输入w，

1. 分别在w上运行M1和M2，每运行一步M1就运行一步M2。
2. 若M1和M2中有一个接受，则接受。

若都拒绝，则拒绝。”

M为识别A1∪A2的判定器。所以可判定语言类对并运算封闭。

b. 连接。

证明：设M1，M2为识别可判定语言A1,A2的判定器。构造图灵机M：

M＝“输入w，

1. 列出所有将w分成两段的方式(|w|+1种).
2. 对于每一种分段方式，在第一段上运行M1，在第二段上运行M2。若都接受，则接受。
3. 若没有一种分段方式被接受则拒绝。”

M为识别A1A2的判定器。所以可判定语言类对连接运算封闭。

c. 星号。

证明：设M1为识别可判定语言A的判定器。

M＝“输入w，

1. 列出w的所有分段的方式(2|w|-1种)。
2. 对于每一种分段方式，重复下列步骤：
3. 分别在每一段上运行M1，若每一段都能被M1接受，则接受。
4. 若没有一种分段方式被接受则拒绝。”

M为识别A\*的判定器。所以可判定语言类对星号运算封闭。

d. 补。

证明：设M1=(Q,Σ,Γ,δ,q0, q1, q2)为识别可判定语言A的判定器，其中q1为接受状态，q2为拒绝状态。令M=(Q,Σ,Γ,δ,q0, q2, q1)，其中q2为接受状态，q1为拒绝状态。则M为识别的判定器。所以可判定语言类对补运算是封闭的。

e. 交。

证明：设M1，M2为识别可判定语言A1,A2的判定器。构造图灵机M：

M＝“输入w，

1. 分别在w上运行M1和M2，每运行一步M1就运行一步M2。
2. 若M1和M2中都接受，则接受。

若M1和M2中有一个拒绝，则拒绝。”

M为识别A1∩A2的判定器。所以可判定语言类对交运算是封闭的。

**3.16 证明图灵可识别语言类在下列运算下封闭：**

a．并 b．连接 c．星号 d．交

**证明：**要证这四种运算下图灵可识别语言类封闭，只需设计出图灵机来识别此种语言。设A和B是图灵可识别语言，A=L(M1)，B=L(M2)，M1和M2是两个图灵机。

a．M=“对于输入w：

1) 在输入w上并行运行M1和M2；

2) M1和M2中有一个停机且接受，则接受；若都停机且拒绝，则拒绝。”

M识别A1∪A2。所以图灵可识别语言类对并运算是封闭的。

b. M＝“输入w，

1. 出所有将w分成两段的方式(|w|+1种).
2. 对于i=1,2,…重复以下步骤：
3. 对于每一种分段方式，在第一段上运行M1i步，在第二段上运行M2 i步，或者直到停机。若都接受，则接受。”

M识别A1A2。所以图灵可识别语言类对连接运算是封闭的。

c．M＝“输入w，

1. 列出w的所有分段的方式(2|w|-1种).
2. 对于i=1,2,…重复以下步骤：
3. 对于每一种分段方式，分别在每一段上运行M1 i步，或者直到停机。

若M1接受所有段，则接受。”

M识别A\*。所以图灵可识别语言类对星号运算是封闭的。

d．M= “对于输入w：

1. 在输入w上运行M1。若M1接受，则转(2)；若M1拒绝，则拒绝。
2. 在w上运行M2。若M2接受，则接受；若M2拒绝，则拒绝。”

M识别A∩B。所以图灵可识别语言类对并运算封闭。

3.21

1) 由cmax≥|c1|知，当|x|≤1，则欲判定不等式明显成立。

2) 当|x|>1时，由

c1xn + c2xn-1 + …+ cnx + cn+1 = 0

⇒ c1x =－(c2 + …+ cnx2-n + cn+1x1-n)

⇒ |c1| |x| = |c2 + …+ cnx2-n + cn+1x1-n|

< |c2| +…+ |cn||x|2-n + |cn+1| |x|1-n

≤ |c2| +…….|cn| + |cn+1||x0| 批注[cwzl]: 〡x〡>=1，当a<=0时，〡x〡a<1

≤ n cmax

< (n + 1) cmax

⇒ |x| < (n + 1) cmax / |c1|.

由(1)和(2)可知结论成立。

3.22 解：A是可判定的。

因为若上帝存在，则s=0，A={0}是可判定的。

若上帝不存在，则s=1，A={1}是可判定的。

**第四章**

4.2 考虑DFA和正则表达式是否等价的问题。将这个问题表示为一个语言并证明它可判定。

解：设EQD-R={<A,B>|A是DFA,B是正则表达式，L(A)=L(B)}。构造如下TM，

F=“对于输入<A,B> , A是DFA，B是正则表达式，

1. 将正则表达式B转化为等价的DFA C。
2. 检测A与C是否等价(EQDFA可判定)。
3. 若等价，则接受；否则拒绝。”

F判定EQD-R。

4.3 设ALLDFA={<A> | A是一个识别Σ\*的DFA}。证明ALLDFA可判定。

证明：

方法一：设计一个使用标记算法的TM M,

M=“对于输入<A>,其中A是一个DFA:

1. 去掉除起始状态以外的所有无用状态。标记起始状态。
2. 重复下列步骤，直到没有新的状态可被标记。
3. 对于每一个未标记状态，如果有一个到达它的转移是从某个已被标记过的状态出发的，则将其标记。
4. 如果所有的标记状态都是接受状态，则接受；否则拒绝”

方法二：构造DFA B，使得L(B)= Σ\*。

M＝“对于输入<A>,其中A是一个DFA:

1. 检测<A,B>是否等价(EQDFA可判定)。
2. 若等价，则接受；若不等价，则拒绝。”

4.4 AεCFG={<G> | G是一个派生ε的CFG}。证明AεCFG可判定。

证明：M=“输入<G>，G是一个CFG，

1. 构造与G等价的Chomsky文法P。
2. 若P的规则集中有S0→ε(其中S0为起始变元)，则接受；否则拒绝。”

M判定AεCFG。

4.9 设INFTYDFA={<A>|A是一个DFA，且L(A)是一个无限语言}。证明INFTYDFA是可判定的。

证明：要证一个DFA识别一个无限语言，只需证此DFA的状态图中有回路。  
     M=“对于输入<A>，其中A是一个DFA：

1. 若无接受状态则拒绝。
2. 去掉A中所有无用状态(没有从它出发到达任何接受状态的路径)。  
   若起始状态被去掉，则拒绝。
3. 重复下一步，直到没有新的状态被标记。
4. 考察每一个未标记的状态，如果没有到它的转移(射入的箭头)，则将其标记并去掉所有从它出发的转移(射出的箭头)。
5. 若所有状态都被标记，则拒绝；否则接受。”

4.11 设A={<M>|M是DFA，它不接受任何包含奇数个1的字符串}。证明A是可判定的。

证明：构造DFA N，使L(N)={含奇数个1的字符串}。构造图灵机

F=“对于输入<M>，其中M是DFA，

1. 构造DFA D，使L(D)=L(M)∩L(N)。
2. 检测L(D)是否为空。(EDFA可判定检测)。
3. 若L(D)＝∅，则接受；否则拒绝。”

4.12设A ={<R，S>｜R和S是正则表达式，且L(R)⊆L(S) }，证明A是可判定的。

解：T=“输入<R，S>，R和S是正则表达式,

1. 构造DFA C，使。
2. 用定理5.4检查L(C)是否为空。
3. 若L(C)为空，则接受；否则拒绝。”

T判定A。

4.13 “检查一个CFG是否派生1\*中某个串问题”

解： LX={<G>|G是{0，1}\*上的CFG，且1\*∩L(G)≠∅}

证明：构造TM T

T＝“对于输入<A>，A为CFG

1. 将终结符“1”和“ε”做标记。
2. 重复下列步骤，直至无可做标记的变元。
3. 如G有规则A→U1U2…Un，且U1U2…Un中每个符号都已做过标记，则将A做标记，其中Ui为终结符或非终结符。
4. 如果起始变元没有被标记则拒绝，否则接受。”

T判定LX。

4.15 设A ={<R>｜R是正则表达式，其所描述的串中至少有一个串以111为子串}。证明A是可判定的。

证明：构造自动机B，使得L(B)={所有以“111”为子串的字符串}。

S=“在输入<G>上，其中G是一个正则表达式,

1. 将G转化为与之等价的自动机A。
2. 构造C，使得L(C)= L(A)∩ L(B)。
3. 检测C是否为空(EDFA可判定)。
4. 若C为空，则拒绝；若C不为空，则接受。”

4.16 检查两个DFA在长度小于或对于某个数的所有串上运行，并以此方法证明EQDFA是可判定的。计算一个这样的数。

证明：对于长度k，构造TM

Mk=“输入<A，B>，A，B是DFA

1) 列出所有长度小于或等于k的串；

1. 在这些串上分别运行A，B；
2. 若A，B同时接受或拒绝，则M接受；若A，B不同时接受或拒绝，则M拒绝。”

因为所有长度≤k的串是有限的，Mk一定是可以进入停机状态，所以Mk是一个判定器。而且Mk判定

{<A,B> | A,B是DFA,Ck={x∈Σ\*| |x|≤k},且L(A)∩C=L(B) ∩C}。

构造TM M：

M=“输入<A，B>，A，B是DFA

1. 计算A和B的状态数p,q。
2. 在<A，B>上运行Mpq。
3. 若Mpq接受，则接受；否则拒绝。”

下面证明M判定EQDFA。

首先注意到A的任意状态与B的任意状态组成的不同状态对有p×q个。

对于Σ\*上的任意串w=w1w2w3…wL，设在A上运行得到状态序列P1P2P3…PL 和在B上运行得到状态序列Q1Q2Q3…QL。若L>pq, 则在L个状态对(Pi, Qi),i=1,2,…,L, 中必有相同的。不妨设Pi＝Pj, Qi＝Qj, 且i<j.

令u=w1w2…wiwj+1wj+2…wL。则有w∈L(A)⇔u∈L(A)，因为这都取决于PL是否属于A的接受状态集。和w∈L(B)⇔u∈L(B)，因为这都取决于QL是否属于B的接受状态集。

若|u|>pq，则按此方法继续减小长度，最后会得到存在v (|v|≤pq), w∈L(A)⇔v∈L(A)和w∈L(B)⇔v∈L(B)。

即L(A)=L(B)⇔[L(A)∩Cpq]=[L(B) ∩Cpq]。于是

M接受<A,B>⇔[L(A)∩Cpq]=[L(B) ∩Cpq] ⇔ L(A)=L(B)，

此即M判定EQDFA。

**4.17** 设C是一个语言。证明C是图灵可识别的，当且仅当存在一个可判定语言D，使得C={ x | ∃y (<x，y>∈D)}。

证明：设M是识别语言C的图灵机，N是语言D的判定器，即C=L(M)，D=L(N)；

(1) 首先证明必要性，即“若C是图灵可识别的，则存在一个可判定语言D，使得C={ x | ∃y (<x，y>∈D)}”。

N=“对于输入<x，k>，x是一个串，k是自然数且k≥1，

1. 在x上模拟M运行k步或直到停机。
2. 若M接受，则接受；若M不接受，则拒绝。”

（注：这里实际D={<x，k> | x在k步之内被M接受}）

(2) 下面证明充分性，即“若D是可判定语言，则存在图灵可识别语言C，满足C={x | ∃y (<x，y>∈D)}”。

M=“输入串x，

1. 取一个y，在<x，y>上运行D；
2. 若D接受，则接受；若D不接受，则找下一个y(按字典序)，重复第一步。”

综上，命题得证。

4.18 设A和B是两个不交的语言。称语言C分离A和B，如果A⊆C且B⊆。证明任意两个不交的补图灵可识别语言都可以由某个可判定语言分离。

证明：设识别A的补的TM为T，识别B的补的TM为S。注意到L(T)∪L(S)= Σ\*。

D=“输入w，

1. 在w上分别运行T和S，直到有一个首先进入接受状态。
2. 若T进入接受状态，则拒绝；若S进入接受状态，则接受。”

D是判定器，而且A⊆L(D)，B⊆。

4.19 设S={<M> | M是DFA，且只要接受w，就接受wR}。证明S可判定。

证明：构造如下TM：

D=“输入<M>,

1. 构造DFA M1使得L(M1)= L(M)R。
2. 检测M1与M是否等价。
3. 若等价，则接受；否则拒绝。”

D判定S。

4.22下推自动机的一个无用的状态是在任何输入上它都不会进入的状态。考虑检查一个下推自动机是否有无用状态问题。将这个问题形式化为一个语言，并证明它是可判定的。

证明：S={<P> | P是PDA，且没有无用状态}。构造如下判定器D：

D=“输入<P>，P=(Q,Σ,Γ,δ,F)是PDA，

1. 对于P中的每一个状态q，重复以下步骤。
2. 构造PDA P1=(Q,Σ,Γ,δ,F1)，其中F1={q}。
3. 检测L(P1)是否为空。若L(P1)=∅，则拒绝。
4. 若没有一个为空，则接受。”

4.28设A是由某些图灵机的描述构成的一个图灵可识别语言{<M1>,<M2>,…}，其中每个Mi都是判定器。求证：存在可判定语言D，它不能由任何一个其描述在A中出现的判定器来判定。

证：设E是A的枚举器。考虑判定器F：

F＝“对输入串x，

1. 初始化，令y＝ε。
2. 运行E，每当E输出一个串(判定器)时，计算y按字典序的下一个串，仍将此串记为y。若y≠x,继续运行E。
3. 若y与x相同，则记E此时输出的判定器为<Mi>。
4. 对串x运行Mi，若Mi接受则拒绝；若Mi拒绝则接受。”

L(F)即为所求语言。

**第五章**

5.1 证明EQCFG是不可判定的。

解：只须证明ALLCFG≤mEQCFG 即可。

构造CFG G1，使L(G1)=∑\*。设计从ALLCFG到EQCFG的归约函数如下：

F=“对于输入＜G＞，其中G是CFG：

1. 输出＜G,G1＞。”

若＜G＞∈ALLCFG，则<G,G1>∈EQCFG 。

若＜G＞∉ALLCFG，则<G, G1>∉EQCFG。

F将ALLCFG 归约到EQCFG 即ALLCFG≤mEQCFG

∵ALLCFG是不可判定的，

∴EQCFG是不可判定的。

5.2证明EQCFG是补图灵可识别的。

证明：注意到ACFG={<G,w>|G是能派生串w的CFG}是可判定的。构造如下TM：

F=“输入<G,H>，其中G,H是CFG，

1. 对于字符串S1, S2,…，重复如下步骤。
2. 检测Si是否可以由G和H派生。
3. 若G和H中有一个能派生w，而另一个不能，则接受。”

F识别EQCFG的补。

5.3 略。

5.4 如果A≤mB且B是正则语言，这是否蕴涵着A也是正则语言？为什么？

解：否。例如：

对非正则语言A={0n1n|n≥0}和正则语言B={0}，可以构造一个可计算函数f使得：

f(w)=

于是w∈A⇔f(w)∈B,故A≤mB。

5.5 证明ATM不可映射规约到ETM。

证明：反证法

假设ATM≤mETM，则有。而ATM的补不是图灵可识别的，从而可知ETM的补也不是图灵可识别的。

下面构造一个识别ETM的补的图灵机S：

S=“输入<M>，M是TM,

1. 对i=1,2,…重复下一步。
2. 对S1,S2,…,Si模拟M运行i步，若有接受，则接受。”

S可识别ETM的补，所以ETM的补是图灵可识别的，与上面由假设得到的ETM的补不是图灵可识别的矛盾。所以ATM不可映射规约到ETM。

5.6 略。

5.7证明：如果A是图灵可识别的，且A≤m，则A是可判定的。

证：∵A≤m≤mA

且A为图灵可识别的

∴也为图灵可识别的

∴由A和都是图灵可识别的可知A是可判定的.

5.8 解：在由<M,w>构造相应骨牌簇时，添加如下一类骨牌：

若M中有一个左移δ(q,a)=(r,b,L)，则添加一张骨牌：

。

并且第一张骨牌改为。

问题

5.x 证明所有的图灵可识别问题都映射可规约到ATM。

证明：设问题A是图灵可识别的，且M是识别A的TM。构造一个可计算函数f使得f(w)=<M,w>， 则有

w∈A⇔f(w)∈ ATM。

这说明A≤m ATM。

5.9 证明S={<M>|M是TM且满足：只要它接受w，就接受wR}不可判定。

证明：对任意<M,w>，其中M是TM，w是串，令f(<M,w>)是如下TM：

F=“输入x，

1. 若x≠01或10，则拒绝。
2. 若x=01，则接受。
3. 若x=10，则在w上运行M。若M接受，则接受。”

可以看到<M,w>∈ATM⇔ f(<M,w>)∈S。ATM≤mS，所有S不可判定。

5.10 证明S={<D,w>|双带TM M在输入w上运行时会在第二条带上写下一个非空白符}是不可判定的。

证明：对任意<M,w>，其中M是单带确定TM，w是字符串。令f(<M,w>)=<D,w>，其中D是如下的双带TM：

D=“输入x，

1. 初始化，x放在第一带上，第二带为空。
2. 在第一带上模拟M运行。
3. 若M接受，则在第二带上写下一个非空白符，并接受；若M拒绝，则拒绝。”

从D的构造可以看出<M,w>∈ATM⇔<D,w>∈S，即ATM≤mS，所以S不可判定。

5.13 USELESSTM ={<N>| N是TM,并且N有无用状态}。

求证USELESSTM不可判定。

证明：构造ETM到USELESSTM的规约函数:

F=“对于输入<M>,M=(Q,Σ,Γ,δ,q0,qaccept,qreject)是TM,

1. 构造TM N

N=(Q,Σ1,Γ1,δ1,q0,qaccept,qreject)，其中，

Σ1=Σ∪{$},Γ1=Γ1∪{$}。设Q={q0,q1,q2,…,qn,qreject ,qaccept }。

对任意q∈Q，a∈Γ1：



2) 输出<N>。”

对于任意TM M，如上构造的TM N，除接受状态外，每个状态均非无用状态(若在输入$上运行N，则N遍历q0,q1,q2,…,qn，最后进入qreject并停机)。构造N的目的就是消除M中任何非接受状态为无用状态的可能。因此有：

<M>∈ETM⇒ <N>∈USELESSTM

<M>∉ETM⇒ <N>∉USELESSTM

所以ETM≤mUSELESSTM而ETM不可判定，因此USELESSTM不可判定。

5.14 考虑这样的问题：检查图灵机在输入w上，当其读写头处于带最左方格时，是否曾经试图将读写头向左移。将这个问题形式化为一个语言，并证明它是不可判定的。

解：此问题可以形式化为一个语言S：

S={<M,w> | TM M在输入w上，当其读写头处于带最左方格时，曾经试图将读写头向左移}

为证明S是不可判定的，可以证明ATM≤mS。构造一个可计算函数f:∑\*→∑\*,使得对每个<M,w>，其中M是TM，w是串，f(<M,w>)=<M’,w>，其中

M’=“输入x,

1. 将工作带上内容改为$x。
2. 读写头置于x的第一个字符，模拟M运行。
3. 每当读写头移到$，保持状态，右移一格。
4. 若M进入接受状态，读写头左移到$,再左移一次，停机，接受；若M进入拒绝状态，则拒绝。”

于是<M,w>∈ATM⇔<M’,w>∈S。

5.15 证明S={<M,w>|图灵机M在输入w上运行时有左移}可判定。

证明：构造如下TM F：

F=“输入<M,w>，M是TM，w是串，

1. 计算M的状态数，记为p。
2. 在w上模拟M，直到有左移，或停机，或已运行了|w|+p步。
3. 若有左移，则接受；若停机但无左移，则拒绝；若无左移且不停机，则拒绝。”

需要说明的是在|w|+p步运行中无左移也不停机的情况。由于无左移，M运行|w|步以后进入空白区域。由于此后右移使得每次读写头所指向的都是空格，而且此后运行的p步至少会有一个状态出现两次，所以不停机意味着M进入了循环，也就不会出现左移。总之，F判定S。

5.17 证明PCP在一元字母表上,即在字母表∑={1}上,是可判定的.

证明：构造识别该语言的图灵机如下:

S=“对输入的骨牌序列<P>,

扫描骨牌序列。若所有的骨牌的上面的1的个数都大于下面的1的个数，或都小于下面的1的个数，则拒绝。否则，接受。”

S判定这样的PCP。

5.18证明PCP在二元字母表上，即在字母表Σ={0，1}上，是不可判定的。

证明：要想证明该PCP(记为PCP2)是不可判定的，只须证明ATM≤mPCP2。为此需要利用定理5.11的证明过程和规约的传递性：

首先，把书中的PCP任一实例P映射到PCP2的实例P2

设计从P到P2的规约函数如下：

F=“对输入<P>，其中P是PCP：

1. 造 PCP P2，对P中所有骨牌中包含的字符串进行Huffman编码，形成一一对应的只含0和1的字符串(也可进行定长编码)。
2. 输出<P2>。”

F将PCP映射规约到PCP2，即PCP≤mPCP2；

其次，有定理5.11有ATM≤mPCP；

根据规约的传递性可知，ATM≤mPCP2

∵ATM是不可判定的，

∴PCP2是不可判定的。

5.21 设AMBIGCFG＝{<G>|G是歧义的CFG}。证明AMBIGCFG是不可判定的。

证明：设AMBIGCFG是可判定的，且R判定AMBIGCFG构造S判定PCP

S＝“对于任一PCP输入，如p＝{[t1/b1],[t2/b2],...,[tk/bk]}

1) 利用如下规则构造一个CFG G：

S→T|B

T→t1Ta1|t2Ta2|...|tkTak|t1a1|...|tkak

B→b1Ba1|b2Ba2|...|bkBak|t1a1|...|bkak

2) 在<G>上运行R，如果R接受则接受，如果R拒绝则拒绝。”

其中ai保证在T→t1Ta1|t2Ta2|...|tkTak|t1a1|...|tkak不是歧义CFG。

这样，如果AMBIGCFG是可判定的，则有PCP可判定的，而PCP是不可判定的，导出矛盾。所以可以得到AMBIGCFG是不可判定的。

5.x 证明：举反例：

令A={<M>|M是图灵机}。则A满足问题xxx中的条件a,不满足条件b。但是A是可判定的，只要证明<M>是否符合图灵机的形式定义即可。因此，只满足条件a,不满足条件b不能导出不可判定。

令B={M1},其中M1是一台图灵机。则B满足问题xxx中的条件b,不满足条件a。但是B是可判定的。因此，只满足条件b,不满足条件a不能导出不可判定。

5.24证明：对任意字符串x，令f1(x)=0x。则有x∈ATM⇔f1(x)=0x∈J。即有ATM≤m J。进一步有≤m。由图灵不可识别知图灵不可识别。

对任意字符串x，令f2(x)=1x。则有x∉ATM⇔f2(x)=1x∈J。即有≤mJ。由图灵不可识别知J图灵不可识别。

5.25给出一个不可判定语言B的例子，使得B≤m。

解：可利用第10题的结果。令B为5.24中的J，则J≤m。构造归约函数如下

F=“输入w，

1. 对w的第一位取反，即0变1，1变0。
2. 输出w。”

则F把J映射归约到。而J 又是不可判定的。

5.26证明：

(a) 判定A2DFA的算法如下：

L=“对于输入<M，x>，其中M是2DFA，x是串，

1. 计算M的状态个数q，和x的长度n。
2. 在x上模拟M qn2步，或直至它停机；
3. 若M停机，则当M接受时接受，拒绝时拒绝。若未停机，则拒绝。”

因为M有q个状态，所以对长度为n的输入，M至多有qn2个不同格局(注意：带上的内容不会改变)。若模拟qn2步还未停机，则必定是进入了循环，该情况下应拒绝。

(b) 构造从ATM到E2DFA的补的映射归约函数。对于任意给定的<M,ω>，f(<M,ω>)=<B>是如下的2DFA的描述：

B=“输入x,

若x=#C1#C2#…#Cm# 是<M在ω上的接受计算历史>，即检查x满足：

* 1. C1是M在ω上的起始格局。
  2. 每个Ci+1都是Ci的合法结果。
  3. Cm是M的一个接受格局。

则接受。”

条件a,c较容易验证。验证b时，B的两个读头分别在Ci和Ci+1的相应位置上移动，验证结果是否适当。由B的构造可以看到

<M, ω>∈ATM⇔ f(<M, ω>)=<B>∉E2DFA

此即ATM映射可归约到E2DFA的补。由ATM不可判定，知E2DFA不可判定。

5.27 证明EQ2DIM-DFA不可判定。

证明：下面证明ATM映射可归约到EQ2DIM-DFA的补：

对于任意<M,w>，f(<M,w>)=<G,H>是如下的两个2DIM-DFA的描述，其中H满足L(H)=∅。为构造G，记格局序列C1,C2,…,Cm的编码为<C1,C2,…,Cm>，它是由格局序列C1,C2,…,Cm组成的矩形串。其由下至上分别是C1,C2,…,Cm，一个格局一行，较短的在右边补上适当数量的空格，四周是由#围成的方框。

G=“输入串x，

若x=<C1,C2,…,Cm>是<M在w上的接受计算历史>，即x满足：

1. C1是M在w上的起始格局。
2. 每个Ci+1都是Ci的合法结果。
3. Cm是M的一个接受格局。

则接受。”

由G,H的构造可以看到

<M,w>∈ATM⇔ f(<M,w>)=<G,H>∉EQ2DIM-DFA

此即ATM映射可归约到EQ2DIM-DFA的补。由ATM不可判定，知EQ2DIM-DFA不可判定。

**第七章**

7.3 a.

|  |  |  |
| --- | --- | --- |
| Operation | X | Y |
|  | 1274 | 10505 |
| X mod Y→X | 1274 | 10505 |
| X↔Y | 10505 | 1274 |
| X mod Y→X | 313 | 1274 |
| X↔Y | 1274 | 313 |
| X mod Y→X | 22 | 313 |
| X↔Y | 313 | 22 |
| X mod Y→X | 5 | 22 |
| X↔Y | 22 | 5 |
| X mod Y→X | 2 | 5 |
| X↔Y | 5 | 2 |
| X mod Y→X | 1 | 2 |
| X↔Y | 2 | 1 |
| X mod Y→X | 0 | 1 |
| X↔Y | 1 | 0 |

当Y=0时，输出X=1，所以1274和10505是互素的。

b.略。

7.4 对于字符串w=baba和下面的文法CFG G，试填写定理8.14中识别上下文无关语言的多项式时间算法中所描述的表。

S→RT

R→TR|a

T→RT|b

解：

|  |  |  |  |
| --- | --- | --- | --- |
| T | R,T | S | R,T,S |
|  | R | S | S |
|  |  | T | R,T |
|  |  |  | R |

7.5 下面的公式是可满足得吗？

φ=(x∨y)∧(x∨)∧(∨y)∧( ∨)

解:(x,y)共有四种取值：(true, true),(true, false),(false, true),(false, false).分别将其带入公式得：

若(x,y)=(true, true)，φ=(true∨true) ∧ (true∨false) ∧ (false∨true) ∧ (false∨false)=false

若(x,y)=(true, false)，φ=(true∨false) ∧ (true∨true) ∧ (false∨false) ∧ (false∨true)=false

若(x,y)=(false, true)，φ=(false∨true) ∧ (false∨false) ∧ (true∨true) ∧ (true∨false)=false

若(x,y)=(false, false),φ=(false∨false) ∧ (false∨true) ∧ (true∨false) ∧ (true∨true)=false

所以原公式不可满足。

7.6 证明P在并、连接和补运算下封闭。

(1) 并：

对任意 L1, L2∈P，设有na时间图灵机M1和nb时间图灵机M2 判定它们，且c=max{a,b}。

对L1∪L2 构造判定器M:

M=“对于输入字符串w :

1. 在w上运行M1，在w上运行M2。
2. 若有一个接受则接受，否则拒绝。”

时间复杂度：设M1为O(na),M2为O(nb)。令c=max{a,b}。

第一步用时O(na+nb) ，因此总时间为O(na+ nb)=O(nc),

所以L1∪L2属于P类，即 P在并的运算下封闭。

(2) 连接：

对任意 L1, L2 属于P 类，设有na时间图灵机M1和nb时间图灵机M2 判定它们，且c=max{a,b}。对L1L2 构造判定器M:

M=“对于输入字符串w=w1,w2,…,wn，

1. 对k=0,1,2,…,n重复下列步骤。
2. 在w1w2…wk上运行M1，在wk+1wk+2…wn上运行M2。
3. 若都接受，则接受。否则继续。
4. 若对所有分法都不接受则拒绝。“

时间复杂度：(n+1)×(O(na)+O(nb))=O(na+1)+O(nb+1)=O(nc+1)，所以L1L2属于P类，即 P在连接的运算下封闭。

(3)补：

对任意 L1属于P 类，设有时间O(na)判定器M1判定它，对构造判定器M:

M=“对于输入字符串w :

1. 在w上运行M1。
2. 若M1接受则拒绝，若M1拒绝则接受。”

时间复杂度为：O(na)。所以属于P类，即 P在补的运算下封闭 。

7.7 证明NP在并和连接运算下封闭。

(1) 并：

对任意 L1, L2∈NP，设分别有na时间非确定图灵机M1和nb时间非确定图灵机M2 判定它们，且c=max{a,b}。

构造判定L1∪L2的非确定图灵机M:

M=“对于输入字符串w :

1. 在w上运行M1，在w上运行M2。
2. 若有一个接受则接受，否则拒绝。”

对于每一个非确定计算分支，第一步用时为O(na)+O(nb)，因此总时间为O(na+nb)=O(nc)。 所以L1∪L2∈NP，即 NP在并的运算下封闭。

(2) 连接：

对任意 L1, L2∈NP，设分别有na时间非确定图灵机M1和nb时间非确定图灵机M2 判定它们，且c=max{a,b}。

构造判定L1L2的非确定图灵机M:

M=“对于输入字符串w :

1. 非确定地将分成两段x,y，使得w=xy。
2. 在x上运行M1，在y上运行M2。
3. 若都接受则接受，否则拒绝。”

对于每一个非确定计算分支，第一步用时O(n),第二步用时为O(na)+O(nb)，因此总时间为O(na+ nb)=O(nc)。 所以L1L2∈NP，即NP在连接运算下封闭。

* 1. 证明如果对数采用一进制编码而不是二进制编码,则素数判定是多项式时间可解的.换句话说,证明语言UNARY-PRIME={1n|n是素数}属于P.

证明:设x为被判定的数. 构造判定器M.

M=“输入1n，n为正整数,

1. 对k=2,3,…,n-1重复下列步骤。
2. 对于1n，每次消去k个1。若刚好可以消完，则拒绝。
3. 若都不能刚好消完，则接受。”

时间分析:

1. 最多执行n-2次,每次1步;
2. 对于每个k的值,执行步数小于n;
3. 需要一步

整个判定过程需要时间小于 (n-2)(1+n)=n2-n-2。运行时间为O(n2)。所以UNARY-PRIME∈P。

7.8 令CONNECTED＝｛<G>|G是连通的无向图｝。分析4.3.2节给出的算法，证明此语言属于P。

证明：判定CONNECTED的TM M的一个高水平描述：

M=“输入是图G的编码<G>：

1)选择G的第一个顶点，并作标记。

2)重复步骤(3)，直到没有新的顶点可以作标记。

3)对于G的每一个顶点，如果存在一条边，将其连到一个已作标记的顶点，则标记该顶点。

4)扫描G的所有顶点，确定它们是否都已作了标记，如果是，则接受，否则拒绝。”

分析该算法的执行，步骤1)执行一次。一个n个顶点连通的无向图最多有n(n-1)/2条边，所以每次执行步骤3，运行时间是O(n2)。为标记所有的n个顶点，步骤3至多需要执行n次。所以用于标记的总的运行时间是O(n3)。步骤4需要执行n步。

该算法的总步数1+ O(n3)+n即为O(n3),从而有该语言属于P。

7.9 无向图中的三角形是一个3-团。证明TRIANGLE∈P。其中

TRIANGLE＝{<G>|G包含3-团}

证明思路：采用相邻矩阵的形式存储图，有n个结点的无向图G存储在矩阵Rn×n中，并且满足：R[i,i]=0, R[i,j]=1(当结点i和结点j之间有一条边)，R[i,j]=∞(当结点i和结点j之间不存在边)。从矩阵的第一行开始逐行扫描矩阵各行，在每一行中找两个为1的元素，假设当前扫描到第i行,若R[i,j]=1且 R[i,k]=1，则检查矩阵中 R[j,k]是否为1，若成立则接受，否则继续搜索。

证明：以算法D实现这一思路。

D=“对输入<R>

1. 计算R的结点数n。若n≤2，则拒绝。否则转2）
2. For i=1 to n
3. For j=i+1 to n
4. 若R[i,j]=1 则
5. For k=j+1 to n
6. 若R[i,k]=1 则检查矩阵中R[j,k]是否为1，若是则接受
7. 若i,j,k同时为n，则拒绝。”

分析D，每一步都是在多项式时间内运行，步骤2最多运行n次，步骤2每运行一次步骤3最多运行n-i次,步骤3每运行一次步骤4最多运行n-j次。所以总计D执行O(n3)步。

7.11

证明：构造NTM如下：

N=“对于输入<G，H>，

1. 比较G与H节点数，若不相同，则拒绝。
2. 非确定地对G的节点进行重排。
3. 若G和H完全相同，则接受。否则拒绝”

N有n！个计算分支，每个长度为n，则此NTM有多项式时间，所以ISO∈NP。

问题

7.12 证明MODEXP∈P。

设二进制整数b= kmkm-1…k1k0，则b=k020+k121+k222+…+km2m(ki =0,1)。构造判定MODEXP的图灵机如下：

M=“对于输入<a,b,c,p>，a,b,c,p为二进制整数，

1. 以k0,k1,…,kn记b的从低位到高位的1至n+1位。
2. 令d0=a，对于i=1,2,…,n计算di=di-1×di-1 mod p。
3. 对于i=0,1,…,n，若ki=1，则令ci=di；若ki=0，则令ci=1。
4. 计算e=c0×c1×…×cn mod p。
5. 若e=c mod p，则接受，否则拒绝。”

设对于两个n位二进制整数，相乘再模一个至多n位的二进制整数，需要运行的时间为T。则第二步的时间为nT，第四步为nT，所以总的运行时间为O(nT)。这意味着MODEXP∈P。

7.14 证明P在星号运算下封闭。

证明：设M为判定A的时间f(n)图灵机，不妨设f(n)≥n。设计如下TM：

D=“对于输入y=y1y2…yn，

1. 若y=ε，则接受；
2. 对于i,j＝1,2,…,n重复(3)。
3. 在yiyi+1…yj上运行M。若接受，则令T(i, j)=“yes”。
4. 重复下面步骤直到表T不再改变。
5. 对于i,j＝1,2,…,n重复下面步骤。
6. 若T(i,j)=“yes”，转(5)。否则继续。
7. 对于k = i,i+1,…,j-1, 若T(i,k)=T(k+1,j)=“yes”,则令T(i,j)=“yes”。
8. 若T(1, n)=yes，则接受；否则拒绝。

运行时间：设O(n2f(n))+O(n3)＝O(n2f(n))。

7.15 证明NP在星号运算下封闭。

证明：设M为判定A的时间f(n)非确定图灵机，不妨设f(n)≥n。设计如下NTM：

D=“对于输入y=y1y2…yn，

1. 若y=ε，则接受；
2. 非确定地选择y的一个划分y=x1x2…xk，其中是x1,x2,…,xk字符串。
3. 对于i=1,2,…,k，重复下一步骤：
4. 在xi上运行M，若M拒绝，则拒绝。
5. 若都接受，则接受。”

以下略。若有兴趣讨论，可与刘庆晖(qhliu@bit.edu.cn)联系。