数据结构与算法设计

教材:

[1][殷] 殷人昆, 数据结构, 清华大学.

[2][王] 王晓东,计算机算法设计与分析,电子工业.

[3][S] 唐常杰等译, Sipser著, 计算理论导引, 机械工业.

参考资料:

[4][严]严蔚敏等,数据结构, 清华大学.

[5][C] 潘金贵等译, Cormen等著, 算法导论, 机械工业.

[6][M] 黄林鹏等译, Manber著, 算法引论-一种创造性方法, 电子.

[7][刘] 刘汝佳等, 算法艺术与信息学竞赛, 清华大学.

计算理论基础第1章作业

1.1 下图给出了两台DFA M_1和M_2的状态图。

回答下述关于这两台机器的问题。

a. 它们的起始状态是什么?

b. 它们的接受状态集是什么?

c. 对输入aabb，它们经过的状态序列是什么?

d. 它们接受字符串aabb吗?

e.它们接受字符串ε吗?

答: a. M_1的起始状态为q_1, M_2的起始状态为q_1.

b. M_1的接受状态集为$\{q_2\}$, M_2的接受状态集为$\{q_1, q_4\}$.

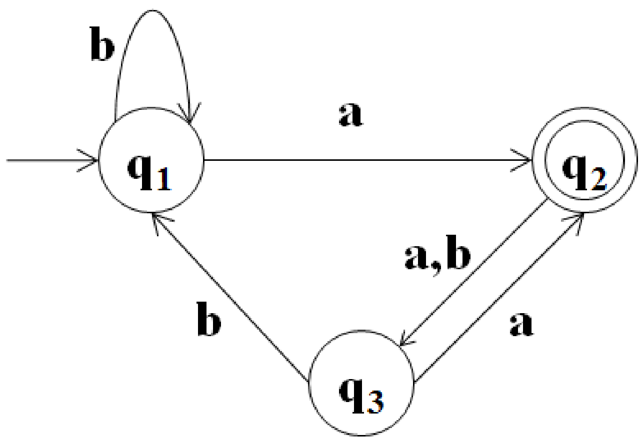
c. 对于输入aabb, 经过的状态序列

M_1 : q_1, q_2, q_3, q_1, q_1.

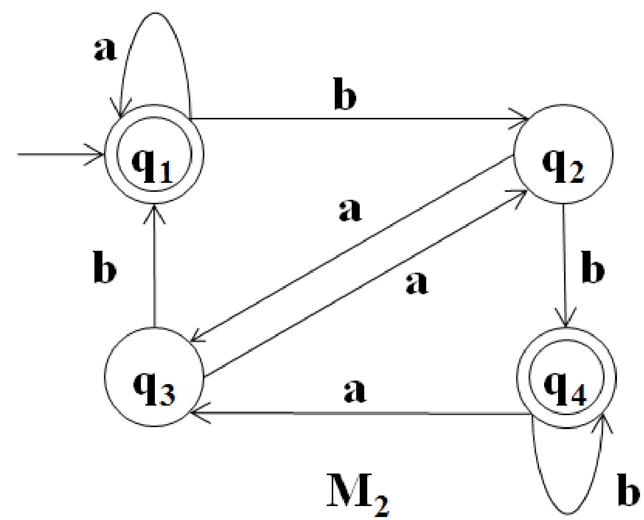
M_2 : q_1, q_1, q_1, q_2, q_4.

d. M_1不接受aabb, M_2接受aabb.

e. M_1不接受ε, M_2接受ε.



M_1



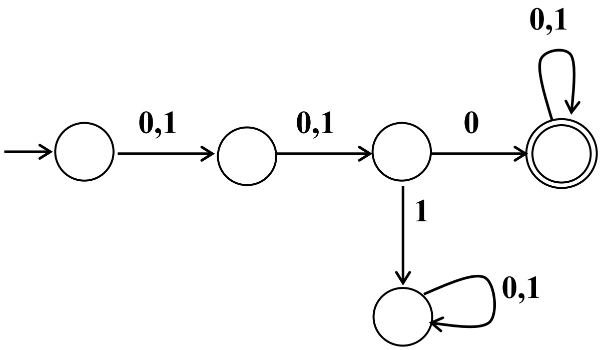
M_2

计算理论基础第1章作业

1.6 画出识别下述语言的DFA状态图. 字母表为{0,1}

d. { w | w的长度不小于3, 并且第3个符号为0};

解：状态图如下

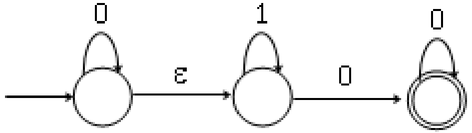


1.7. 给出下述语言的NFA, 并且符合规定的状态数.

字母表为{0,1}

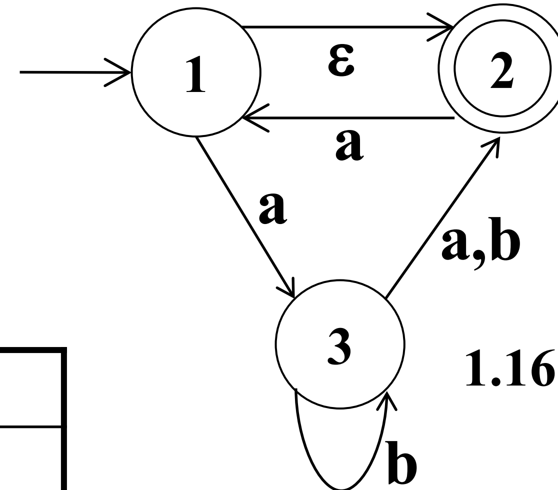
e. 语言$0^*1^*0^*0$, 3个状态.

解：状态图如下



计算理论基础第1章作业

1.16(b) 将如右图的非确定有限自动机转换成等价的确定有限自动机.



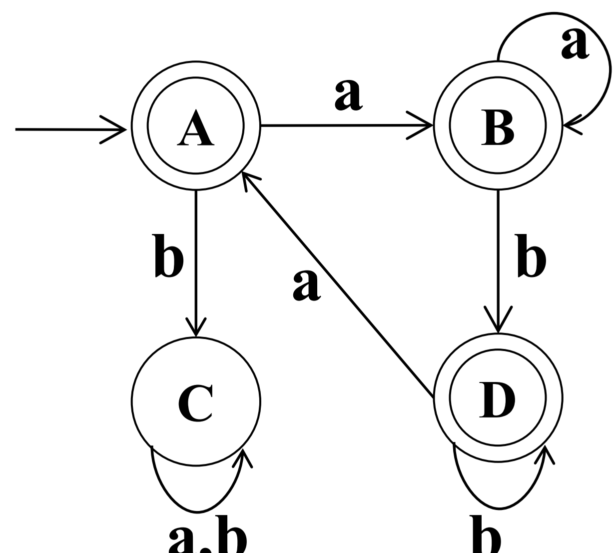
1.16(b)题图

解: 根据NFA的状态图,

得到与它等价的如下DFA状态转移表

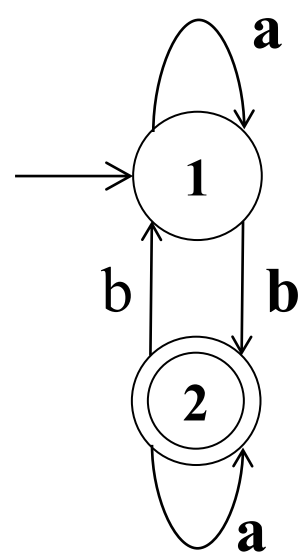
编号	δ	a	b
A*	{1,2}	{1,2,3}B	∅ C
B*	{1,2,3}	{1,2,3}	{2,3} D
C	∅	∅	∅
D*	{2,3}	{1,2}	{2,3}

对应状态转移图为



计算理论基础第1章作业

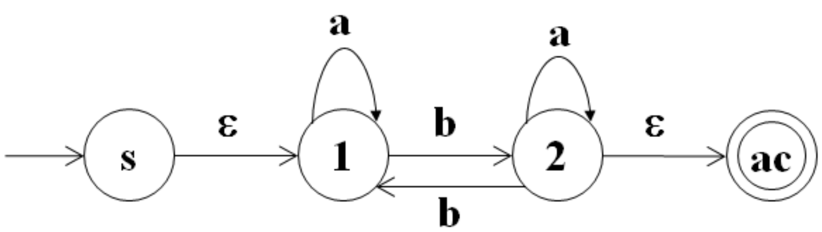
1.21(a) 将如右图的有限自动机转换成等价的正则表达式.



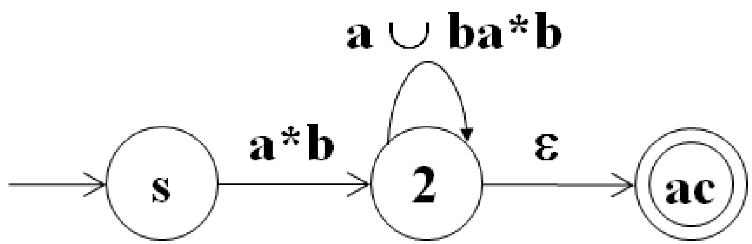
1.21(a)题图

解:

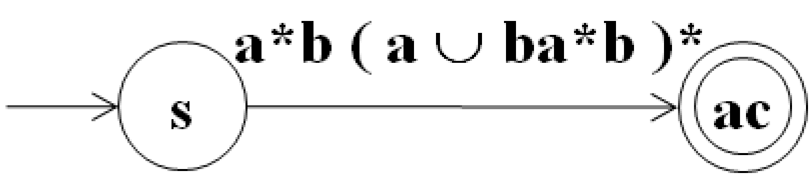
1. 添加s, ac,



2. 去掉状态1



3. 去掉状态2



说明: 答案不唯一. 错误: (a*b) (a ∪ ba*b*)*, 例abba是反例

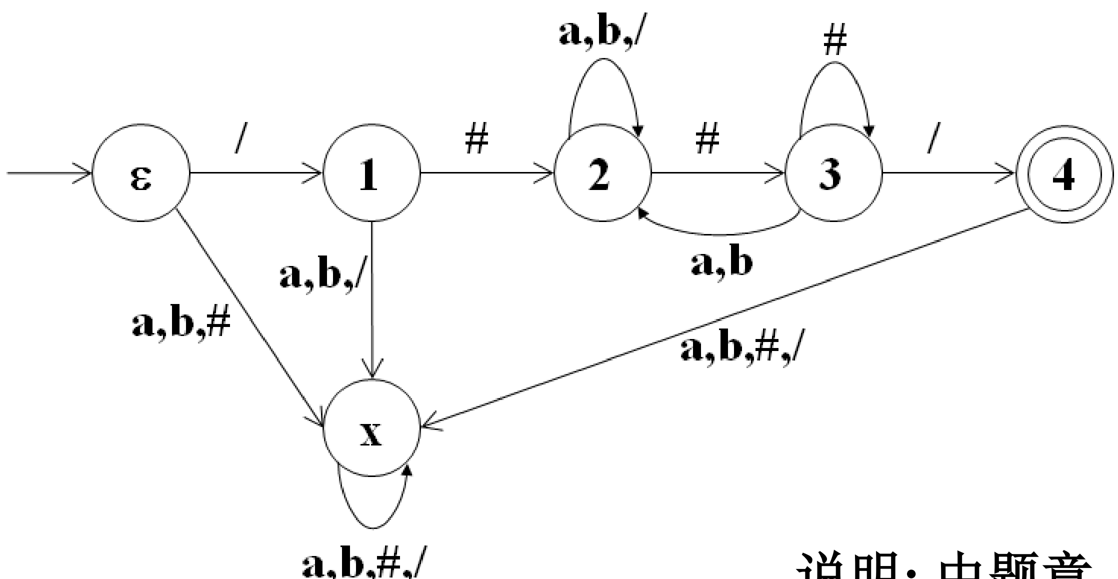
正确: a*ba* ∪ a*b(a*ba*b)* a* 或 (a ∪ ba*b)* ba* 或 a*b(a*ba*b)*a* 等

计算理论基础第1章作业

1.22 在某些程序设计语言中, 注释出现在两个分隔符之间, 如/#和#/. 设C是所有有效注释串形成的语言. C中的成员必须以/#开始, #/结束, 并且在开始和结束之间没有#/. 为简便起见, 所有注释都由符号a和b写成; 因此C的字母表 Σ={a, b, /, #}. a. 给出识别C的DFA. b. 给出产生C的正则表达式.

解: a. 设关键信息: ε, /, /#, /#..#, /#..#/, error, 分别对应状态ε,1,2,3,4,x,

根据关键信息之间的关系得如下状态图:



说明: 由题意, 允许/#///#/出现

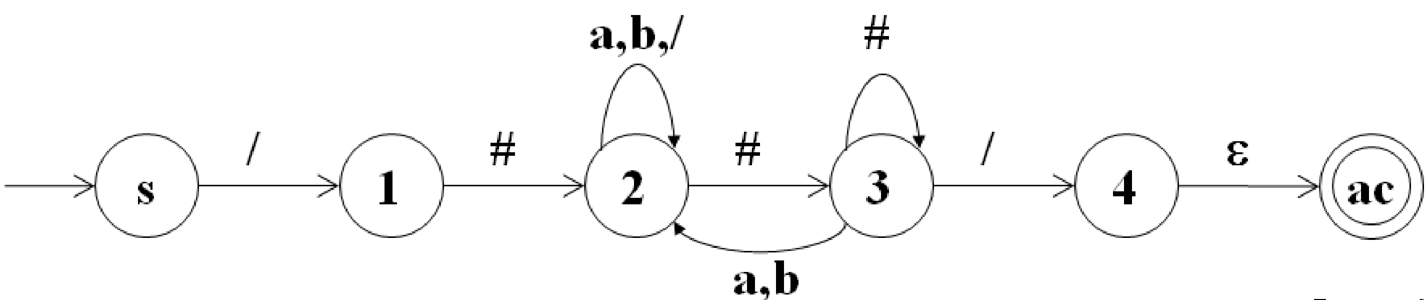
计算理论基础第1章作业

1.22 在某些程序设计语言中，注释出现在两个分隔符之间，如/#和#/. 设C是所有有效注释串形成的语言. C中的成员必须以/#开始，#/结束，并且在开始和结束之间没有#/. 为简便起见，所有注释都由符号a和b写成；因此C的字母表

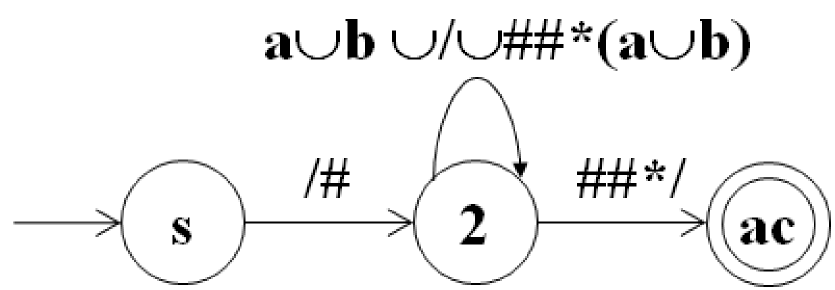
Σ={a, b, /, #}. a. 给出识别C的DFA. b. 给出产生C的正则表达式.

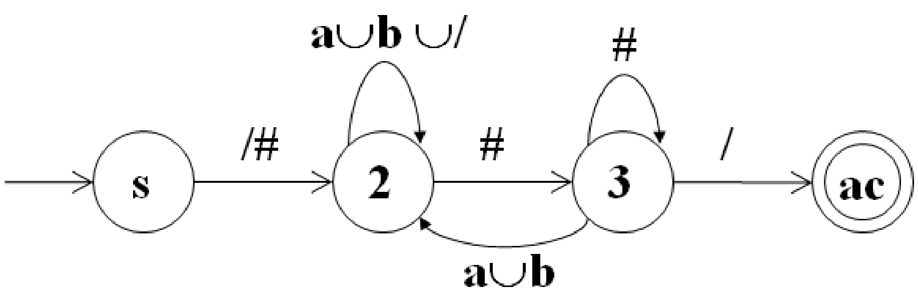
解：b. 产生C的正则表达式 为 /#(a ∪ b ∪ / ∪ ##*(a∪b))*##*/

说明：答案不唯一,作者新版答案/#(#*(a ∪ b) ∪ /)*##*/. 中间段#后面不能是/.



1. 添加s, ac
2. 去掉x





3. 去掉状态1， 4. 去掉状态4   5. 去掉状态3， 6. 去掉状态2得答案

计算理论基础第1章作业

1.29 使用泵引理证明下述语言不是正则的。

b. $A = \{ www \mid w \in \{a,b\}^* \}$

证明: ∵ $\forall p>0$, 令 $w=a^p b a^p b a^p b$,

$\forall x,y,z(|y|>0, |xy| \le p, w=xyz)$

令 $i=0$, $xz = a^{p-|y|} b a^p b a^p b \notin A$

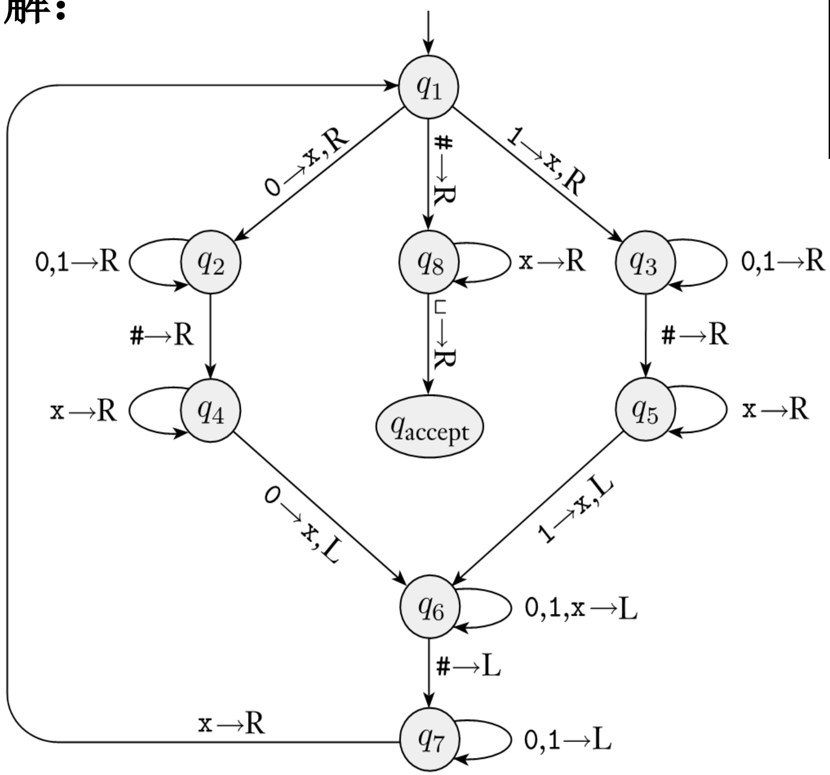
∴ 根据正则语言泵引理, A非正则语言.

计算理论第3章作业

3.2 对于识别{w|w=u#u, u∈{0,1}*}的图灵机M_1 (见左图)，在下列输入串上，给出M所进入的格局序列.

c. 1##1，d. 10#11，e. 10#10

解:



c.

$q_1$1##1,
xq_3##1,
x#q_5#1,
x##q_r1,
拒绝

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_r_,
拒绝

e.

$q_1$10#10,
x$q_3$0#10,
x0q_3#10,
x0#$q_5$10,
x0q_6#x0,
x$q_7$0#x0,
q_7x0#x0,
x$q_1$0#x0,
xxq_2#x0,
xx#q_4x0,
xx#x$q_4$0,
xx#$q_6$xx,

e. 继续

xxq_6#xx,
xq_7x#xx,
xxq_1#xx,
xx#$q_8$xx,
xx#xq_8x_,
xx#xxq_8_,
xx#xx_q_{ac}_
接受

补充说明: 没有画出的箭头指向拒绝状态, 假设这些箭头都不改写右移且q_r是拒绝状态.

计算理论第3章作业

3.8 下面的语言都是字母表{0,1}上的语言, 以实现水平的描述给出判定这些语言的图灵机:

b. B = {w|w所包含的0的个数是1的个数的两倍}

c. C = {w|w所包含的0的个数不是1的个数的两倍}

解:

b. 构造如下的图灵机

M = “对于输入串w,
1) 从左至右扫描, 重复以下步骤直到带上没有0或没有1
2) 从左至右扫描, 删除遇到的第一个1.
3) 从左至右扫描, 删除遇到的前两个0, 若没有则拒绝.
4) 若带上既没有0也没有1, 则接受; 否则, 拒绝.”

若输入w中0的个数是1的个数的两倍, 则停机接受; 否则, 停机拒绝.
所以一方面M的语言是B; 另一方面M对所有输入串都能停机, 是判定器.
补充细节说明: (仅为方便理解, 不用写在答案上)
可以针对第一个符号使用不同关于左端标记和删除标记。
若第一个符号是0, 则可第一次扫描时改写为$, 需要删除时改写为!;
若第一个符号是1, 则可第一次扫描时改写为%, 需要删除时改写为*.
这样当读到$,!,%,*都表示到了最左端.

计算理论第3章作业

3.8 下面的语言都是字母表{0,1}上的语言, 以实现水平的描述给出判定这些语言的图灵机:

b. B = {w|w所包含的0的个数是1的个数的两倍}

c. C = {w|w所包含的0的个数不是1的个数的两倍}

解:

c. 构造如下的图灵机

M = "对于输入串w,
1) 从左至右扫描, 重复以下步骤直到带上没有0或没有1
2) 从左至右扫描, 删除遇到的第一个1.
3) 从左至右扫描, 删除遇到的前两个0, 若没有则接受.
4) 若带上既没有0也没有1, 则拒绝; 否则, 接受."

一方面M的语言是C; 另一方面M对所有输入串都能停机, 是判定器.

计算理论第3章作业

3.15b 证明图灵可判定语言类在连接运算下封闭.

证明: 设A和B是图灵可判定语言, 则有判定器T和Q使得 L(T)=A, L(Q)=B, 构造如下的图灵机

M = “对于输入串w, 设w长度为n, 即$w=w_1w_2\ldots w_n$,

1) 对于i = 0 到 n,

2) 令$x=w_1\ldots w_i$, $y=w_{i+1}\ldots w_n$, (其中若i=0则x=ε, 若i=n则y=ε)

3) 在x上运行T, 在y上运行Q,

4) 若两个都接受, 则停机接受.

5) 停机拒绝.”

若w属于A°B, 则M会停机接受; 否则, M会停机拒绝.

所以是判定器而且M的语言是A°B. 证毕

计算理论第3章作业

3.16d证明图灵可识别语言类在交运算下封闭.

证明: 设A和B是图灵可识别语言, 则有图灵机T和Q使得 L(T)=A, L(Q)=B,

构造如下的图灵机

M = "对于输入串w,

1) 在w上运行T, 在w上运行Q,

2) 若两个都接受, 则接受; 否则拒绝."

若w属于A∩B, 则M会停机接受; 否则, M会不停机或停机拒绝.

所以M的语言是A∩B. 证毕

3.21 设多项式 $c_1x^n + c_2x^{n-1} + \ldots + c_nx + c_{n+1}$**有根** $x = x_0$, c_{max}**是**c_i**的最大绝对值. 证明**

$$|x_0| \le (n+1)\, c_{max} / |c_1|$$

解: 不妨设$c_1 \ne 0$.

若 $|x_0| \le 1$, **则** $|x_0| \le 1 \le c_{max} / |c_1| \le (n+1)\, c_{max} / |c_1|$, **性质成立**

若 $|x_0| > 1$, **则由** $c_1x_0^n + c_2x_0^{n-1} + \ldots + c_nx_0 + c_{n+1} = 0$, **得**

$$c_1x_0^n = -(c_2x_0^{n-1} + \ldots + c_nx_0 + c_{n+1}),$$

$$|c_1|\,|x_0|^n < (n+1)c_{max}|x_0|^{n-1},$$

$$|x_0| < (n+1)\, c_{max} / |c_1|.$$

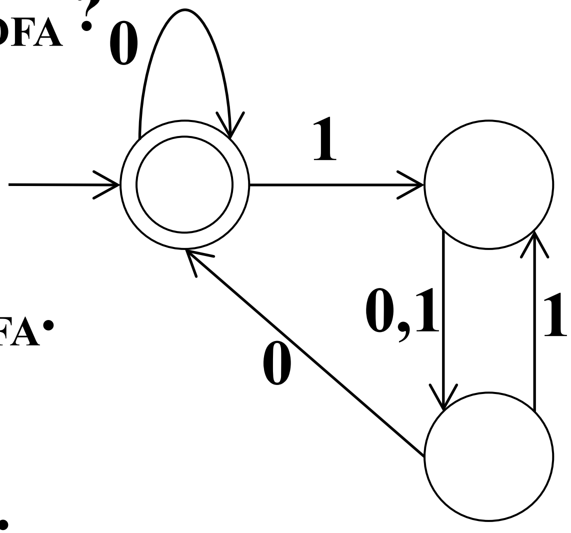
计算理论第4章作业

4.1 对于右图所示的DFA M, 回答下列问题，并说明理由

a. $<M,0100> \in A_{DFA}$?　　b. $<M,011> \in A_{DFA}$?

c. $<M> \in A_{DFA}$?

e. $<M> \in E_{DFA}$?　　f. $<M,M> \in EQ_{DFA}$?



解: a) 因为M接受0100, 所以$<M,0100> \in A_{DFA}$.

b) 因为M不接受011, 所以$<M,011> \notin A_{DFA}$.

c) $<M>$不符合A_{DFA}的编码, 所以$<M> \notin A_{DFA}$.

e) 因为M的语言包含0,00等字符串, 所以M的语言非空, 所以$<M> \notin E_{DFA}$.

f) 因为M和M自己语言相同, 所以$<M,M> \in EQ_{DFA}$.

计算理论第4章作业

4.2 考虑一个DFA和一个正则表达式是否等价的问题。

将这个问题描述为一个语言并证明它是可判定的。

解: 一个DFA是否与一个正则表达式是否等价可以表示为如下的语言:

A = { <M,R> | M是一个DFA, R是一个正则表达式, 满足 L(M) = L(R) }.

构造如下的图灵机

P = "对输入<M,R>, M是DFA, R是正则表达式,

1) 将R转换为等价的NFA, 再转换为等价的DFA T,

2) 构造DFA Q使得 $L(Q) = (L(M) \cap L(T)^c) \cup (L(T) \cap L(M)^c)$

3) 检查Q的语言是否空(起始状态是否与某个接受状态连通)

4) 若不连通, 则接受; 否则拒绝."

若M与R等价, 则Q的语言空, P会接受<M,R>; 否则Q的语言非空, P会拒绝.

所以P是判定器, 而且P的语言是A. 证毕

补充说明：正则语言对补, 交, 并都是封闭的, 所以L(M)与L(T)的对称差仍正则

计算理论第4章作业

4.3 设 $ALL_{DFA} = \{\langle A\rangle \mid A$是一个识别$\Sigma^*$的DFA$\}$. 证明$ALL_{DFA}$可判定.

证明:构造如下的图灵机

P = “对输入$\langle A\rangle$, A是DFA

1) 标记所有与起始状态连通的状态.

2) 若所有有标记的状态都是接受状态, 则接受; 否则拒绝.”

若A的所有被标记状态都是接受状态, 则所有输入都会被接受, A的语言是Σ^*; 否则, A的语言不是Σ^*.

P是判定器, 且P的语言是ALL_{DFA}, 所以ALL_{DFA}是可判定语言.

证毕

计算理论第4章作业

4.15 设A = {<R> | R是一个正则表达式，
其所描述的语言中至少有一个串w以111为子串 }.
证明A是可判定的。

证明: 构造一个DFA T使得L(T)={w|w以111为子串},
构造如下的图灵机
P = “对输入<R>, R是正则表达式,
1) 构造与R等价的DFA Q.
2) 构造DFA S使得$L(S) = L(R) \cap L(T)$
3) 标记S中与起始状态连通的所有状态
4) 若有接受状态被标记, 则接受; 否则拒绝.”

若S有接受状态被标记, 则L(R)与L(T)交非空, $<R> \in A$;
否则L(R)与L(T)交为空, $<R> \notin A$.
所以P是判定器且L(P) = A. 所以A是可判定语言.

计算理论第7章作业

7.9 无向图中的三角形是一个3团。证明TRIANGLE∈P，其中TRIANGLE={ <G> | G包含一个三角形 }。

证明: 构造如下图灵机

H = "对于输入<G>, G是一个无向图,

1) 设G有n个顶点, 且顶点为x_1, x_2,…, x_n.

2) 若 n < 3, 则拒绝.

3) 对 i = 1 到 n-2,

4) 对 j = i+1 到 n-1,

5) 对 k = j+1 到 n,

6) 检查x_i, x_j, x_k,是否是一个三角形(即是否三条边相连)

7) 若是, 则接受.

8) 拒绝."

计算理论第7章作业

7.11 若图G的节点重新排序后，G可以变得与H完全相同，则称G与H是同构的。令ISO = {<G,H> | G和H是同构的图}。 证明ISO∈NP。

证明:构造如下非确定图灵机

N= “对于输入<G,H>, G和H都是图,

1) 若G和H顶点数不同, 则拒绝.

2) 设G的顶点为x_1, x_2,…, x_n, H的顶点为y_1, y_2,…, y_n.

3) 非确定的选择1到n的排列p.

4) 对 i = 1 到 n-1

5) 对 j = i+1 到 n

6) 若 $(x_i,x_j)\in E(G)$ 异或 $(y_{p(i)},y_{p(j)})\in E(H)$ 为真, 则拒绝

7) 接受.”。

若G, H同构,则 N一定有分支接受; 否则, N所有分支拒绝.

N的所有分支都在都在$O(n^2)$时间内运行.

所以, N是ISO的多项式时间非确定判定器, ISO∈NP.

计算理论第7章作业

7.21 令Double-SAT = { <ϕ> | ϕ至少有两个满足赋值 }。证明Double-SAT是NP完全的。

证明:

(1) Double-SAT$\in$NP

构造如下非确定图灵机

N= "对于输入<ϕ>, ϕ是布尔公式,

(a) 非确定地产生两组不同赋值s,t

(b) 若既有在赋值s下ϕ=1, 又有在赋值t下ϕ=1, 则接受; 否则, 拒绝"

因为N的语言是Double-SAT, 且N在多项式时间内运行, 所以Double-SAT$\in$NP.

(2) 证明SAT可以多项式时间映射归约到Double-SAT.

对任意布尔公式ϕ, 添加一个新变量a, 构造函数$f(\phi) = \phi \wedge (a \vee \neg a)$。

首先, f可在多项式时间内计算完成。

其次, f是SAT到Double-SAT的映射归约, 即ϕ可满足$\Leftrightarrow$f(ϕ)有两个满足赋值:

若ϕ有可满足赋值s, 则在赋值s和a=1下f(ϕ)=1, 在赋值s和a=0下f(ϕ)=1, 从而有两个不等赋值; 若f(ϕ)有可满足赋值s, 则从s中去掉a的赋值, 必然也是ϕ的可满足赋值. 所以f是从SAT到Double-SAT的多项式时间映射归约。

由(1)和(2) 及SAT是NP完全问题知, Double-SAT是NP完全问题。

计算理论第7章作业

7.22 令HALF-CLIQUE = { <G> | G是无向图, 包含结点数至少为m/2的完全子图, m是G的结点数}. 证明HALF-CLIQUE是NP完全的.

说明: 书上的答案只是要点，考试时需要给出完整的答案.

证明:

(1) HALF-CLIQUE∈NP

构造如下非确定图灵机

N= "对于输入<G>, G是无向图,有m个顶点

(a) 非确定地产生一个m/2个顶点的子集

(b) 若这个子集中的任意两个顶点之间都有边相连, 则接受; 否则, 拒绝".

因为N的语言是HALF-CLIQUE，且N是在多项式时间运行，所以HALF-CLIQUE∈NP。

计算理论第7章作业

(2) 证明CLIQUE可以多项式时间映射归约到HALF-CLIQUE.
对任意<G,k>，其中G是一个无向图，k是一个正整数。构造函数f(<G,k>) = G’。
设G有m个顶点。按如下方式构造G’：
若k=m/2，则G=G’；
若k>m/2，则在G中增加2k-m个新顶点，这些新顶点都是孤立点，得到G’；
若k<m/2，则增加m-2k个新顶点，这些新顶点之间两两都有边相连，新顶点与G的所有顶点之间也都相连。
首先，f可在多项式时间内计算完成。
其次证明f是CLIQUE到HALF-CLIQUE的映射归约，即证明G有k团⇔G’(设有m’个顶点)有m’/2个顶点的团：
若G有k团，当k=m/2时，G’=G, m’=m，则G’也有k=m’/2团；当k>m/2时，m’=2k，G’中也有k=m’/2团；当k<m/2时，m’=2m-2k，G中的k团加上新添的m-2k个顶点形成m-k=m’/2团。
若G’有m’/2团，当k=m/2时，G’=G, m’=m，则G也有k=m’/2团；当k>m/2时，m’=2k，G中也有k=m’/2团；当k<m/2时，m’=2m-2k，G’中的m-k团至多有m-2k个新添顶点，去掉新添顶点至少还有k个顶点，所以G中有k团。

由(1),(2)和CLIQUE是NP完全问题知，HALF-CLIQUE是NP完全问题。