第三/四章作业

袁晨圃

3.1

习题 1 (3.2 d)：此练习与图灵机M_1有关，例3.5给出了它的描述和状态图，在下列每一个输入串上，给出M_1进入的格局序列

d. 10#11

> $M_1 := (Q, \Sigma, \Gamma, \delta, q_1, q_{\text{accept}}, q_{\text{reject}})$
>
> - $Q = \{q_1, ..., q_8, q_{\text{accept}}, q_{\text{reject}}\}$
> - $\Sigma = \{0, 1, \#\}, \Gamma = \{0, 1, \#, x, \sqcup\}$
> - 开始、接受、拒绝状态分别为$q_1, q_{\text{accept}}, q_{\text{reject}}$

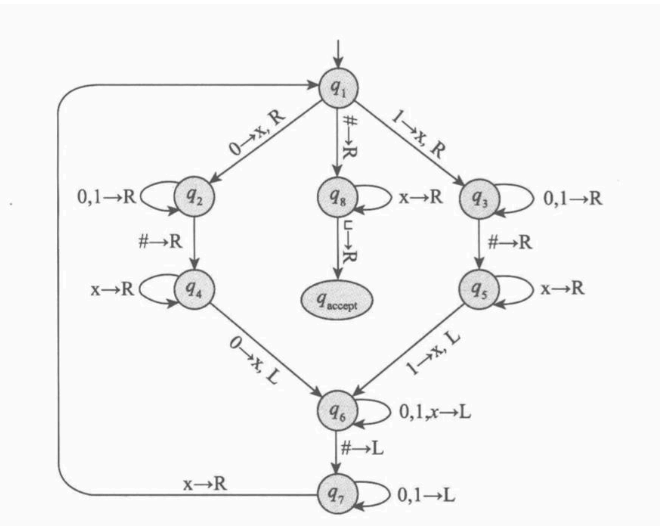


图 3-5 图灵机 M_1 的状态图

解：

$q_1 10\#11\sqcup$

$xq_3 0\#11\sqcup$

$x0q_3\#11\sqcup$

$x0\#q_5 11\sqcup$

$x0q_6\#x1\sqcup$

$xq_7 0\#x1\sqcup$

$q_7 x0\#x1\sqcup$

$xq_1 0\#x1\sqcup$

$xxq_2\#x1\sqcup$

$xx\#q_4 x1\sqcup$

$xx\#xq_4 1\sqcup$

$xx\#x1q_{\text{reject}}\sqcup$

习题 2 (3.7)：下面描述的不是一个合法的图灵机，解释为什么。

$M_{\text{bad}} =$“ 在输入$\langle p \rangle$上，其中p为变元$x_1, ..., x_k$上的一个多项式：
1. 让$x_1, \cdots, x_k$取所有可能的整数值
2. 对所有这些取值求p的值
3. 只要某个取值使得p为0，则接受，否则拒绝”

解：

如果不存在$x_1, ..., x_k$使得$p(x_1, ..., x_k) = 0$，那么这个机器就会无限循环下去，永远不会停止

设计不出由普通状态到q_{reject}的转移

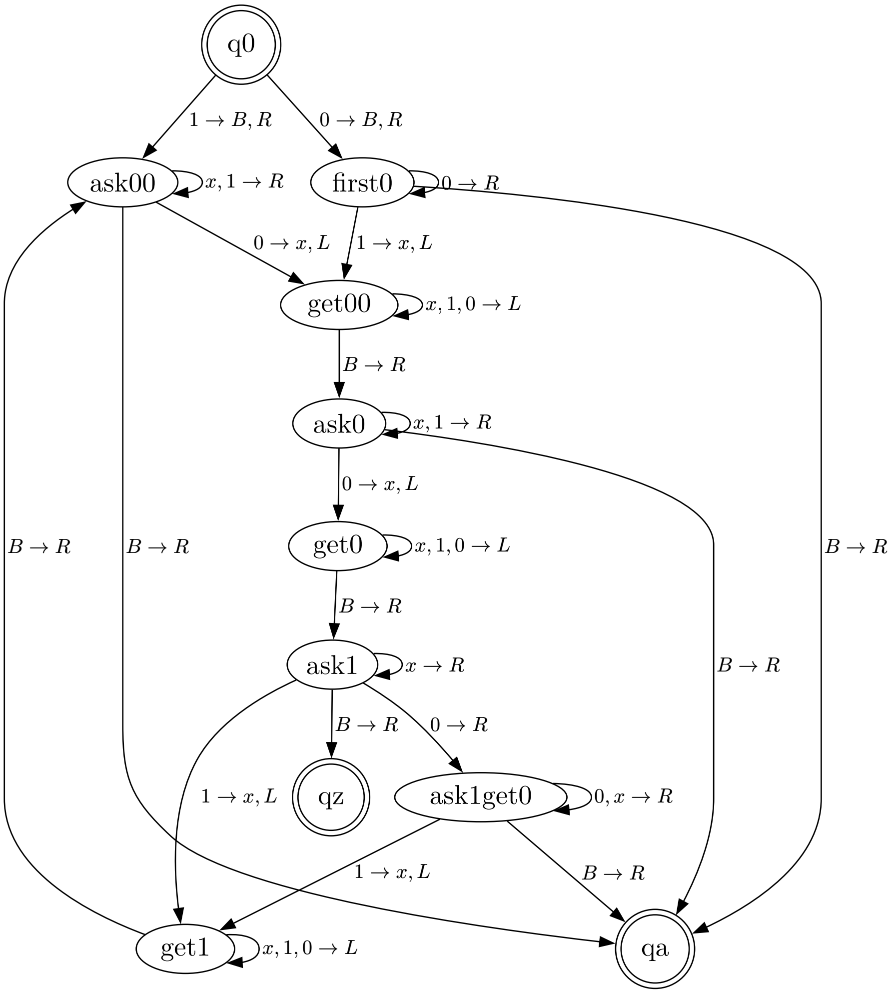
✓

习题 3 (3.8 c)：下面的语言都是字母表$\{0, 1\}$上的语言，以实现层次的表述给出判定这些语言的图灵机

c. $\{w \mid w$ 所包含的 0 的个数不是 1 的个数的两倍$\}$

解：

使用B表示空白字符⊔， `q0` 表示起始状态，`qa` 表示接受状态，`qz` 表示拒绝状态



run it: https://paste.ubuntu.com/p/MJSPk6jmFN/

实现层次!

$M =$ "对于输入串 w:

1. 扫描带子并对第一个未标记的 0 进行标记。如果没有发现未被标记的 0，则转到第 4 步。
2. 扫描带子并对第一个未标记的 0 进行标记。如果没有发现未被标记的 0，则接受，否则，读写头返回至带子的左端点。
3. 扫描带子并对第一个未标记的 1 进行标记。如果没有发现未被标记的 1，则接受，否则，读写头返回至带子的左端点，并转到第 1 步继续执行。
4. 读写头返回至带子的左端点，扫描带子以发现是否存在未被标记的 1。如果没有则拒绝，否则接受。"

习题 4 (°3.20): **以停留代替左移图灵机**和普通图灵机类似，只是它的转移函数具有下列形式：

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{R, S\}$$

在任何时候，机器可以将读写头向右移，或让其在原地不动，证明这样的图灵机与普通图灵机不等价，这样的图灵机识别什么语言类?

解:

a. 因为这样的图灵机无法识别语言$A = \{a^t b^t \mid t \in \mathbb{Z}\}$ (因为Γ有限，无法在一个单元格内编码任意的t)

所以跟普通的图灵机不等价

b. A是上下文无关语言，所以此种图灵机无法识别上下文无关语言类

下面证明它识别正则语言

证明：设L为正则语言，存在 DFA $D(Q_0, \Sigma_0, \delta_0, q_0, F_0)$识别它

构造图灵机$M = \left(Q, \Sigma, \Gamma, \delta, q_{\text{start}}, q_{\text{accept}}, q_{\text{reject}}\right)$，其中

- $Q = Q' \cup \left\{q_{\text{accept}}, q_{\text{reject}}\right\}$，$Q'$满足存在从$Q_0$到$Q'$的一一映射$f : Q_0 \to Q'$
- $\Sigma = \Sigma_0$
- $\Gamma = \Sigma_0 \cup \{\sqcup\}$
- $q_{\text{start}} = f(q_0)$
- $$\delta : \begin{cases} (f(q), a) \mapsto (f(\delta_0(q, a)), a, R), & \text{where } a \in \Sigma, q \in Q_0 \\ (f(q_F), \sqcup) \mapsto \left(q_{\text{accept}}, \sqcup, R\right), & \text{where } q_F \in F \\ (f(q), \sqcup) \mapsto \left(q_{\text{reject}}, \sqcup, R\right), & \text{where } q \in Q_0 \setminus F \end{cases}$$

□

反过来，证明 RSTM 可以转换为 DFA:

称新图灵机为 RSTM。

(1) 任给 RSTM $M = (\ldots, \delta, \ldots, q_{\text{accept}}, q_{\text{reject}})$，可以构造 DFA $D = (\ldots, \delta', \ldots, q'_{\text{accept}})$ 来模拟。使用接受态 q'_{accept} 和陷阱态 q'_{reject} 替换 q_{accept} 和 q_{reject}，其它状态名称不变。在 DFA 的语境下，由于不能立刻接受/拒绝，即使到达这些状态也仍需读取后续输入并保持状态不变，故 $\delta'(q', a) = q'$ ($q' \in \{q'_{\text{accept}}, q'_{\text{reject}}\}, \forall a$)。

转移函数设计思路为：
设当前状态为 r_0，读写头下字符为 a_0。推导转移过程 $\delta(r_i, a_i) = (r_{i+1}, a_{i+1}, \mathsf{S})$，直到停机或变为 R。则必为以下几种情况之一：

- 进入停机状态，且除了最后一步可 R 可 S，之前一直为 S。若停机时接受，令 $\delta'(r_0, a_0) = q'_{\text{accept}}$；若拒绝，令 $\delta'(r_0, a_0) = q'_{\text{reject}}$；
- 存在最小的 n 使得 $\delta(r_n, a_n) = (r_{n+1}, a_{n+1}, \mathsf{R})$，且直到 r_{n+1} 均不停机。令 $\delta'(r_0, a_0) = r_{n+1}$；
- 一直不停机，且一直为 S。令 $\delta'(r_0, a_0) = q'_{\text{reject}}$。

由于 $Q \times \Gamma$ 组合数有限，一定可以判断是哪种情况。

3.2

习题 5 (3.15 c)：证明图灵可识别语言在 $*$ 运算符下封闭

证明：设M识别语言L

下面构造图灵机M'识别L^*

对于串 w 枚举 w 的不同划分$\text{Devide}(w) = \left\{\left\{w_1, w_2, \ldots, w_{n_i}\right\} \mid w_1 w_2 \cdots w_{n_i} = w\right\}$

给划分 d_i 编号，设 d_i 有后继$\text{Succ}(d_i)$

对于每一个划分，在各个字串上执行M，最多k步，其中k是递增整数。若字串上均接受，则图灵机M'接受w，不断地循环这一过程

```
Limit(M, k, w):
  input w for M
  for _ in range (k) do
      run M for one step
      if Accept then
          return Accept
  return Reject
```

```
Check(d, k):
  for w in d do
      if Limit(M, k) = Reject then
          return false
  return true
```

```
M'(w):
k ← 1
d ← d_0
while true do
    if Check(d, k) then
        return Accept
    d ← Succ(d)
    k ← k + 1
```

使用高层次描述：

M' ="对于输入w：

a. 扫描带子，并非确定性（得分点）地考虑w 的每个分割方式$w = w_1 \ldots w_k$。

b. 从左至右在$w_1, \ldots, w_k$ 上分别模拟 M。对任一子串，若 M 拒绝，则拒绝；若 M 接受，则移至下一子串，并将 M 恢复至起始状态重新运行。若所有子串均接受，则接受。"

习题 6 (°3.13)：证明：一个语言是可判定的，当且仅当有枚举器以标准字符串顺序枚举这个语言

> 标准字符串顺序：以长度为第一关键字排序，长度相同则按字典序

解：

必要性：

设L可判定，则有图灵机M在输入上一定停机，且$\begin{cases} w \in L \Leftrightarrow M \text{ Accept} \\ w \notin L \Leftrightarrow M \text{ Reject} \end{cases}$

构造枚举器E:

```
E():
w ← ε
while true do
    if M(w) = Accept then
        yield w
    w ← Succ(w)
```

充分性：

设有枚举器E，构造图灵机如下：

```
Locate(w):
for i, s in enumerate(S) do   // S: strings sorted by standard order
    if s = w then
        return i
```

```
M(w):
k ← Locate(w)
while true do
    e ← E()
    k_e ← Locate(e)
    if e = w then:
        return Accept
    if k_e > k then:
        return Reject
```

高层次:

设 $s_1, s_2, \ldots$ 是 Σ^* 中按标准字符串顺序排列得到的字符串序列。(这样排列可以使得每个字符串有唯一确定的序列号，而字典序不然。)

(1) 若语言 L 可被图灵机 M 判定，则构造枚举器 E:
$E =$ "忽略输入。

1. 对 $i = 1, 2, \ldots$，重复下列步骤。
2. 在 s_i 上模拟 M。若 M 接受，则打印输出 s_i。"

由于 M 为判定器，在 M 上输入任意串都能在有限步内停机，故每个 L 中的串都能在有限步内输出。

(2) 若语言 L 可被枚举器 E 以标准字符串顺序枚举，当 L 是有限语言时，显然可以构造判定器判定 L；当 L 是无限语言时，对任意 $w \in L$，E 在有限步后必能输出 w 或某个顺序在 w 之后的字符串（因为若 E 不输出任何 w 之后的字符串，则 L 不是无限语言；若 E 不能在有限步内输出某个顺序在 w 之后的字符串，则该字符串不会出现在枚举器枚举的语言中）。如下构造判定器 M:
$M =$ "对于输入 w:

1. 模拟运行 E。每当 E 输出一个串时，将其与 w 比较。若该串为 w，则接受；若该串的顺序在 w 之后，则拒绝。"

注：需要讨论 L 是否为有限语言，因为若 L 为有限语言，枚举 L 的枚举器 E 也可能在枚举完所有字符串后不停机，从而无法判断 E 是否后续会枚举出想要的字符串。

习题 7 (*3.12)：证明每一个无穷图灵可识别语言都有一个无穷可判定子集

证明：设有枚举器 E 枚举无穷图灵可识别语言 L

构造枚举器 E'，运行 E 但仅当得到的字符串比之前输出的字符串更大时才输出

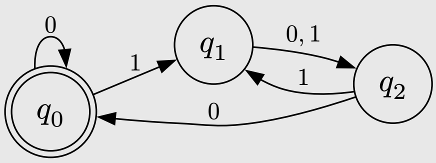
根据习题 6，枚举器 E' 枚举的语言 L' 是可判定的

又 L 无穷，所以对任意 w 总能找到 $w' > w, w \in L$，所以 L' 也是无穷的

correct

3.3

习题 8 (4.1)：对于下图所示 DFA M，回答下列问题并说明理由



a. $\langle M, 0100\rangle \in A_{\mathrm{DFA}}$?

b. $\langle M, 011\rangle \in A_{\mathrm{DFA}}$?

c. $\langle M\rangle \in A_{\mathrm{DFA}}$?

d. $\langle M, 0100\rangle \in A_{\mathrm{REX}}$?

e. $\langle M\rangle \in E_{\mathrm{DFA}}$?

f. $\langle M, M\rangle \in EQ_{\mathrm{DFA}}$?

解：

a. M 接受 0100 $\Rightarrow$ 属于

b. M 拒绝 011 $\Rightarrow$ 不属于

c. M 接受 $\varepsilon \Rightarrow$ 属于 输入格式不符（跟 $\langle M, \varepsilon\rangle$ 不同）

d. 将 M 的编码作为正则表达式不能派生 0100 $\Rightarrow$不属于

e. $L(M) \neq \emptyset$，所以$\langle M\rangle \notin E_{\mathrm{DFA}}$

f. $L(M) = L(M) \Rightarrow$ 属于

习题 9 (4.3)：设 $ALL_{\mathrm{DFA}} = \{\langle A\rangle \mid A$ 是一个 DFA,且 $L(A) = \Sigma^*\}$，证明ALL_{DFA}是可判定的

证明：由于 DFA 在补运算封闭，对于输入的 $\langle A\rangle$，先构造 $\langle\overline{A}\rangle$补充：交换接受状态和非接受状态，然后因为 E_{DFA} 是可判定的，所以能判定 $L\left(\overline{A}\right)$ 是否为 $\emptyset$，等价于 $L(A)$ 是否为 Σ^*

习题 10 (°4.24)：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 $C = \{x \mid \exists y(\langle x, y\rangle \in D)\}$

证明：

必要性：

若 C 是图灵可识别的，由图灵机 M 识别，对于 $w \in C$ 若 k 步内图灵机 M 接受 w 则 $\langle w, k\rangle \in D$

由于一定停机，所以 D 是可判定的

充分性：

给定输入 x 和可判定语言 D

通过标准字符串顺序枚举所有的字符串 y，若$\langle x, y\rangle \in D$，则接受，反之继续枚举

这样构造出来的是图灵可识别语言 $\{x \mid \exists y(\langle x, y\rangle \in D)\}$。

高层次描述：

(1) 若 C 是图灵可识别语言，存在图灵机 M_C 识别 C。构造判定器 M_D 判定语言 D:
$M_D =$ "对于输入 $\langle x, y\rangle$，其中 x 为字符串，y 为自然数：

1. 在 x 上模拟 M_C 运行 y 步。若在 y 步内接受，则接受，否则拒绝。"

若 $x \in C$，一定 $\exists y$ 使得在 x 上模拟 M_C 运行 y 步接受，否则不存在这样的 y。故 C 和 D 关系成立。

(2) 若存在可判定语言 D 满足要求，存在判定器 M_D 判定 D。设 $s_1, s_2, \ldots$ 为 M_D 的 Σ^* 上以标准字符串顺序排列的字符串序列。构造图灵机 M_C:
$M_C =$ "对于输入 x，其中 x 为字符串：

1. 对 $i = 1, 2, \ldots$，重复下列步骤。
2. 在 $\langle x, s_i\rangle$ 上模拟 M_D。若接受，则接受。"

若 $x \in C$，必存在 s_i 使得在 $\langle x, s_i\rangle$ 上模拟 M_D 接受。由于 M_D 为判定器，必可在有限时间内出现接受结果。否则，若 $x \notin C$，M_C 不停机。因此，M_C 识别语言 C。

习题 11 (*4.12)：设 A 是由某些图灵机的描述构成的一个图灵可识别语言 $\{\langle M_1\rangle, \langle M_2\rangle, \ldots\}$，其中每个 M_i 都是判定器。求证：若判定器 M_i 的描述在 A 中，则存在可判定语言 D，但它不能被任何 M_i 所判定。

提示：考虑 A 的一个枚举器

证明：

因为可判定语言有无限多个，如果 A 有限则显然成立

考虑 A 的一个枚举器 E，它能枚举出所有的 $\langle M_i\rangle$，其中 M_i 是判定器

尝试构造一个跟任意一个 M_i 判定的语言都不同的语言 D

设图灵机 M_D 满足以下条件：

- 对于输入 w，调用 $\text{Locate}(w)$ 得到它在标准字符串顺序中的编号 x
- 调用 E，输出 $\langle M_x\rangle$，然后运行 M_x 输入 w，若 M_x 拒绝则接受，接受则拒绝

这样 M_D 判定（以上两步都会停机）语言 D，但 D 不能被任何 M_i 所判定（若 $\langle M_i\rangle \in A$，则必定有一个枚举顺序编号 x，以及对应的标准字符串顺序为 x 的串 w_x ，而上述构造出的语言在 w_x 停机状态下肯定跟 M_i 不相同）

✓