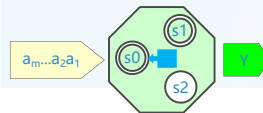


离散数学：形式语言与自动机：有限状态机

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

句子识别

- 给定一个字符串，判定是否属于给定语法G的语言L(G)
- 一般来说，这是个难解的问题
- 对于正则语法和正则语言，可以通过一个“识别机器”来判定



北京大学地球与空间科学学院/陈斌/2015

机器machine

- 一个系统，接受输入，改变自身的状态，产生输出
- 状态指为了完成机器的任务而对输入序列进行的一种临时归类
- 可以理解作为一种对记忆的模拟
- 在逐个接受输入字符的过程中，机器状态会发生多次改变，最终会停止在某个状态，并有输出产生
- 如果对于所有的输入，机器状态的数目有限，称为有限状态机 (Finite State Machine)

北京大学地球与空间科学学院/陈斌/2015

有限状态机

- 有限状态机是一个五元组 $M(A, S, Y, s_0, F)$
- A ：输入字符串的字母表
- S ：机器的有限状态集合
- $Y \subseteq S$ ：被称作“接受”的一些状态
- $s_0 \in S$ ：初始状态
- $F: S \times A \rightarrow S$ ：状态转移函数，指明在某个状态下接受输入字符所引起的状态变迁

北京大学地球与空间科学学院/陈斌/2015

有限状态机例子

- $A = \{a, b\}$
- $S = \{s_0, s_1, s_2\}$, $Y = \{s_0, s_1\}$, s_0 初始状态
- F如下表：

F	a	b
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_2	s_2

北京大学地球与空间科学学院/陈斌/2015

离散数学：形式语言与自动机：状态图

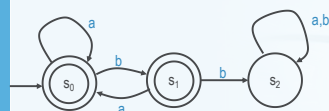
陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

状态图

- FSM通常用状态图来表示
- 状态图 $D=D(M)$ 是边带标记的有向图
- 节点为状态，转移函数决定边的走向和标记
- 接受状态用双圈表示
- 边的定义：如果 $F(s_i, a) = s_k$ ，则从节点 s_i 做标记为 a 的有向边到节点 s_k
- 初始状态 s_0 用一个特殊的无源箭头标识

北京大学地球与空间科学学院/陈域/2015

状态图例子：赋权有向图



北京大学地球与空间科学学院/陈域/2015

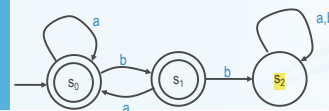
机器识别的语言

- M所识别的A上的所有字符串
- $w = a_1 a_2 \dots a_m \in A^*$,
- w确定了一个状态序列 $s_0, s_1, s_2, \dots, s_m$
- 其中 $F(s_{i-1}, a_i) = s_i$
- 也就是确定拟路径 $P = (s_0, a_1, s_1, a_2, s_2, \dots, a_m, s_m)$
- 如果 $s_m \in Y$ 则称M识别w

北京大学地球与空间科学学院/陈域/2015

从状态图看识别语言

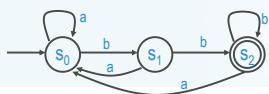
- 看看下图的例子是否识别：
- ababba、baab、空串
- 例子机器识别什么模式的字符串？
- 所有不包含连续两个b的字符串



北京大学地球与空间科学学院/陈域/2015

正则语言和有限状态机

- Kleen定理：字母表A上的形式语言L是正则的当且仅当存在一个有限状态机M使得 $L=L(M)$
- 例：构造自动机M，识别恰好以bb结尾的字符串
- 正则表达式： $(a|b)^*bb$



北京大学地球与空间科学学院/陈域/2015

离散数学：形式语言与自动机：泵引理

陈域 北京大学地球与空间科学学院 gischen@pku.edu.cn

泵Pumping引理

- 长度超过状态数目的接受串 w ，都可以表示成 $w=xyz$ 形式，而 xy^iz 都会被 M 接受
- 设 w 对应的状态序列为 $s_0, s_1, s_2, \dots, s_n$
- n 大于状态的数目 $|S|$ ，所以根据鸽笼原理必有 $s_i = s_j (i < j)$
- 令 $X = a_1 a_2 \dots a_i$, $Y = a_{i+1} \dots a_j$, $Z = a_{j+1} \dots a_n$
- x 以 s_i 状态结尾， xy 以 $s_j = s_i$ 状态结尾，所以 xy^m 也以状态 s_i 结尾， $xy^m z$ 结尾状态仍为 s_n



证明： $L = \{a^m b^m : m > 0\}$ 不是正则语言

- 假设 L 是正则语言，则有有限状态机 M 接受 L ，假设 M 状态个数为 k
- 令 $w = a^k b^k$ ，则一定有 $|w| > k$ ，根据泵引理， $w = xyz$ ，其中 y 非空，而且 xy^2z 也被 M 接受
- 如果 y 仅含有 a 或仅含有 b ，那么 xy^2z 中 a, b 的个数就不相等了，应该不属于 L
- 如果 y 同时含有 a 和 b ，那么 y^2 中一定会出现 a 在 b 之后的情况， xy^2z 也应该不属于 L
- 两个矛盾说明假设错误， L 不是正则语言
- 我们以前的例子 $a^m c b^m$ 也不是正则语言

离散数学：形式语言与自动机：机器同余

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

机器同余(machine congruence)

- 机器同余是一个等价关系 R
- 在有限状态机 $M(A, S, F)$ 的状态集合 S 上的等价关系 R
- 对于任意的 $s, t \in S$ ，如果 $s R t$ ，
- 能导出对于任意的输入符号 $x \in A$ ，都有 $F(x, s) R F(x, t)$
- 即：同一个等价类的状态对于任意的输入都转移到属于同一个等价类的状态

回顾抽象代数中的同余关系

- 在代数结构 $\langle S, \Delta \rangle$ 中， S 上的一个等价关系 \sim ，如果满足：
- $a \sim b$ 蕴涵 $\Delta a \sim \Delta b$ ，称 \sim 是 S 上关于一元运算 Δ 的同余关系
- 同余关系下，载体元素体现出一种聚类性质，同一类的元素经过运算，其结果还是同一类
- 而在有限状态机中的机器同余则表现了状态之间的一种聚类性质，同一类状态经过读入字符，其目标状态还是同一类
- 同余反映了某种等效性

离散数学：形式语言与自动机：商机器

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

商机器 (quotient machine)

- 相对于有限状态机 $M(A,S,F)$ 和 M 上的机器同余 R
- $S' = S/R = \{[s] | s \in S\}$
- 是 S 关于等价关系 R 的商集
- $F' = \{ \langle [s], [F(x,s)] \rangle \mid s \in S, x \in A \}$
- $M'(A,S',F')$ 称为 M 关于 R 的商机器 M/R
- 通常商机器比原来的机器简单一些，是机器化简的基础

北京大学地球与空间科学学院/陈斌/2015

商机器例子

- $A = \{a, b\}$, $S = \{s_0, s_1, s_2, s_3, s_4, s_5\}$, F 如表
- S 上等价关系 R 的关系矩阵如图
- 对照检查 F , R 是机器同余

F	a	b
s_0	s_0	s_4
s_1	s_1	s_0
s_2	s_2	s_4
s_3	s_5	s_2
s_4	s_4	s_3
s_5	s_3	s_2

$[s_0] = \{s_0, s_2\}$

$[s_1] = \{s_1, s_3, s_5\}$

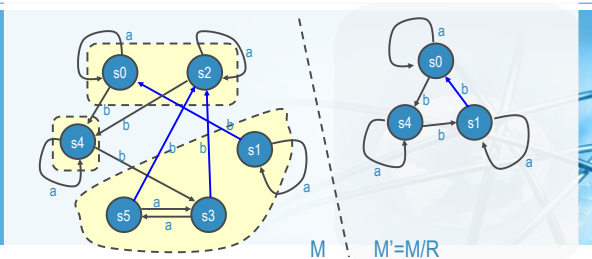
$[s_4] = \{s_4\}$

1	0	1	0	0	0
0	1	0	1	0	1
1	0	1	0	0	0
0	1	0	1	0	1
0	0	0	0	1	0
0	1	0	1	0	1

F'	a	b
$[s_0]$	$[s_0]$	$[s_4]$
$[s_1]$	$[s_1]$	$[s_0]$
$[s_4]$	$[s_4]$	$[s_1]$

北京大学地球与空间科学学院/陈斌/2015

商机器例子：状态图



北京大学地球与空间科学学院/陈斌/2015

离散数学：形式语言与自动机：商机器性质

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

商机器应用

- 有限状态机可能出现冗余的状态
- 不同的机器可能接受相同的语言
- 其中状态数目最少的机器为最简
- 通过构造机器同余及商机器来合并这些冗余状态，减少状态数目

北京大学地球与空间科学学院/陈斌/2015

对字符串相容的状态

- $s, t \in S$, $w \in A^*$
- 如果 $F(s, w) \in Y$ 并且 $F(t, w) \in Y$
- 或者 $F(s, w) \notin Y$ 并且 $F(t, w) \notin Y$
- 注： $F(s, w) = F(\dots F(F(F(s, a_1), a_2), a_3 \dots), a_n)$
- 称 s, t 是关于 w 相容的
- 定义： S 上的二元关系 R
- 如果 $s, t \in S$ 对于任何 $w \in A^*$ 都相容，则 $s R t$
- 说明 s, t 其实是关于句子识别的等效状态

北京大学地球与空间科学学院/陈斌/2015

关于相容的二元关系R

- › R是等价关系：
- › 首先，具有自反、对称性
- › 其次， sRt, tRu 则 $F(s,w), F(t,w), F(u,w)$ 要么都属于Y，要么都不属于Y，所以也有 sRu ，是传递的

北京大学地球与空间科学学院/陈域/2015

关于相容的二元关系R

- › R是机器同余：
- › 要证明对于任意 $s, t \in S, x \in A$ ，如果 sRt ，有 $F(s,x)RF(t,x)$
- › 设任意 $w \in A^*$ ， $w' = xw$
- › 因为 sRt ，则有 $F(s,w')$ 和 $F(t,w')$ 同时属于或不属于Y
- › 也就是 $F(F(s,x),w)$ 和 $F(F(t,x),w)$ 同时属于或不属于Y，即 $F(s,x)RF(t,x)$ ，所以R是机器同余

北京大学地球与空间科学学院/陈域/2015

商机器的性质

- › 由具有相容性质的机器同余，所构造的商机器M/R
- › 从初始状态开始，对任何字符串与原机器做出的是否接受结论是相同的
- › M和商机器M/R识别**相同**的语言
- › 由于商机器状态不多于原机器，M/R即M化简的结果
- › 但如何对于长度不限制的任意w找到这样的等价关系R？

北京大学地球与空间科学学院/陈域/2015

离散数学：形式语言与自动机：机器化简

陈域 北京大学地球与空间科学学院 gischen@pku.edu.cn

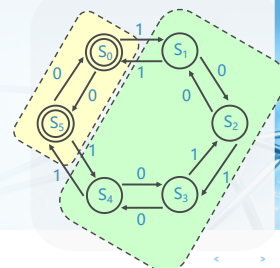
相容关系R的迭代算法

- › S的初始划分 $P_0 = \{Y, Y^c\}$
- › 假设目前已有划分 $P_k = \{S_1, S_2, \dots, S_m\}$ ，考察每个等价类 S_i
- › 如果 S_i 中的两个状态 s, t 在**所有**的输入 x 作用下都转移到**同一个**状态分块 A_j （取决于 x ），则据此构造 S_i 的进一步细分
- › 将所有 S_i 的细分合在一起，成为新的划分 P_{k+1}
- › 如果 $P_{k+1} = P_k$ ，算法停止，否则回到第2步
- › 根据R构造商机器M/R即为化简结果

北京大学地球与空间科学学院/陈域/2015

机器化简的例子

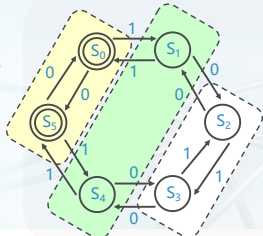
- › 从 $P_0 = \{\{s_0, s_5\}, \{s_1, s_2, s_3, s_4\}\}$ 开始
- › 首先先看： $\{s_0, s_5\}$
- › 对0：转到 $\{s_0, s_5\}$ ；对1：转到 $\{s_1, s_4\}$
- › 不分裂。
- › 再看 $\{s_1, s_2, s_3, s_4\}$
- › 对0： s_1, s_2, s_3, s_4 都转到 $Y \sim$
- › 对1： s_1, s_4 转到Y， s_2, s_3 转到 $Y \sim$
- › 分裂为 $\{s_1, s_4\}, \{s_2, s_3\}$



北京大学地球与空间科学学院/陈域/2015

机器化简的例子

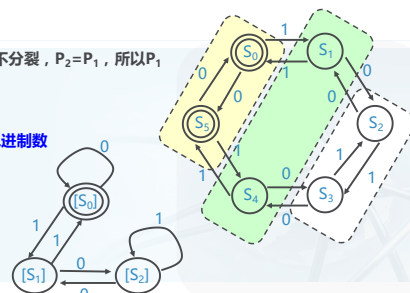
- › $P_1 = \{\{s_0, s_3\}, \{s_1, s_4\}, \{s_2, s_5\}\}$
- › 看 $\{s_0, s_5\}$:
› 对0转到 $\{s_0, s_5\}$; 对1转到 $\{s_1, s_4\}$; 不分裂 ;
- › 看 $\{s_1, s_4\}$:
› 对0转到 $\{s_2, s_3\}$; 对1转到 $\{s_0, s_5\}$; 不分裂 ;
- › 看 $\{s_2, s_3\}$:
› 对0转到 $\{s_1, s_4\}$; 对1转到 $\{s_2, s_3\}$; 不分裂 ;



北京大学地球与空间科学学院/陈斌/2015

化简的机器

- › P_1 中所有的分块都不分裂, $P_2 = P_1$, 所以 P_1 对应机器同余R
- › 商机器M/R
- › 识别能被3整除的二进制数
- › $[s_0]$ 被3整除的状态
- › $[s_1]$ 余1的状态
- › $[s_2]$ 余2的状态



北京大学地球与空间科学学院/陈斌/2015

离散数学：形式语言与自动机：带输出的机器

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

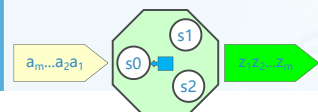
带输出的机器

- › 在有限状态机 $M(A, S, Y, s_0, F)$ 的基础上
- › 去掉接受状态集合 Y , 增加:
- › 输出字符的有限集合 Z
- › 输出函数 $G: S \times A \rightarrow Z$
- › 在状态转移的同时, 输出一个字符

北京大学地球与空间科学学院/陈斌/2015

带输出机器的工作方式

- › 输入字符串 $u = a_1 a_2 \dots a_m$ 印在一条输入带上
- › 机器M逐个读入输入带的字符
- › 进行状态转移 $V = s_0 s_1 s_2 \dots s_m$
- › 同时在输出带上逐个打印输出字符:
- › $W = z_1 z_2 \dots z_m$



北京大学地球与空间科学学院/陈斌/2015

实现二进制加法的有限状态机

- › 两个二进制数的加法:
- › 通过给较短的数前面补0, 使两个数具有相同的长度;
- › 将两个数对位组合, 并从低位开始输入到机器, 输入特定的“空白”表示输入结束
- › 输入: 11, 11, 00, 11, 01, 11, 10, b
- › 输出: 0, 1, 1, 0, 0, 1, 0, 1
- › 计算结束, 进入“停机”状态

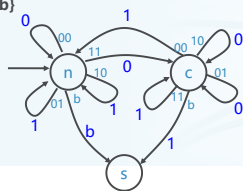
```

1101011
+0111011
10100110
  
```

北京大学地球与空间科学学院/陈斌/2015

实现二进制加法的有限状态机

- › $M=(A,S,Z,s_0,F,G)$, $A=\{00,01,10,11,b\}$
- › $S=\{\text{执行}(c), \text{初始}(n), \text{停止}(s)\}$
- › $Z=\{0,1,b\}$



北京大学地球与空间科学学院/陈斌/2015

离散数学：形式语言与自动机：程序实现状态机

陈斌 北京大学地球与空间科学学院 gischen@pku.edu.cn

用程序实现有限状态机

```
> status=s0;
> while(a=get_next_input())
> switch (status) of {
>   s0: switch(a) of {
>     'x' : status= s3; output( '0' ); break;
>     'y' : status= s4; output( '1' ); break;
>   }; break;
>   s1:...
> };
> return status;
```

北京大学地球与空间科学学院/陈斌/2015

有限状态机的限制

- › 不存在能作“操作数位数无限制”的二进制乘法的有限状态机
- › 现实中的计算机，内存有限，能够表达的状态有限，属于一种有限状态机
- › 能够做位数有限的乘法
- › 如果宇宙是有限的，理论上也是有限状态机
- › 经过足够长的时间，总会回到以前的某个状态，再重演历史
- › 是循环宇宙论的主要论据☹

北京大学地球与空间科学学院/陈斌/2015