图灵机的行为学习*

邓 晶　白 硕

(国家智能计算机研究开发中心　北京　100080)

摘要　图灵机计算时读写头在带子上的活动,包括移动及改写字符,称为图灵机的行为.本文证明,图灵机的行为是可学习的.即存在一个学习过程,它能根据一个图灵机的行为序列,学到另一个也能产生同样行为序列的图灵机.

关键词　学习,行为,行为序列,行为学习,功能学习.

1. 学习

本文研究图灵机计算时的行为是否是可学习的.

学习是被很多学科,包括认知科学、语言学、人工智能等研究的对象.虽然直观上人们心里都清楚什么是"学习",可即使在人工智能领域里,对学习的定义都远未统一.我们这里只讨论对黑箱的学习,即根据可观测量的历史记录,对未来进行预测.

这种对黑箱的学习可表示为:

$$O_t = S(I_t) \qquad (t=0,1,2,\ldots)$$

$$g_{t+1} = f(g_t, I_t, O_t), \qquad (t=0,1,2,\ldots)$$

其中,黑箱 S 是一个输入输出系统. I_t 是 S 从时刻 0 到 t 的总输入. O_t 是 S 从时刻 0 到 t 的总输出. f 是一个学习机制, g_t 是在时刻 t, f 对 S 作的猜测, f 根据已经生成的 g_t, 黑箱 S 的可观测量 I_t 和 O_t 作新的预测 g_{t+1}.

如果存在整数 n, 当 $t>n$ 时, $g_t = g_{t+i}$ $(i\geqslant 0)$, 而且对任意的时刻 t_m, $g_t(t_m) = O(t_m)$, 则 S 是可学习的,否则 S 是不可学习的.**

2. 图灵机的行为学习

上面我们描述了对黑箱的学习.然而大多数情况下,学习对象都不是"纯粹"的黑箱,学习系统都是在已经假定了学习对象的一些性质的前提下来学习的.这篇文章讨论的学习对象就是图灵机.学习前我们已经默认它有了状态集、指令集等一个图灵机应有的性质.在这些性质之外的部分,才是需要学习的"黑箱".

对于自动机来说,通常认为只有它的输入输出是可观测量,而其余部分都是不可观测

* 作者邓晶,1971 年生,硕士生,主要研究领域为人工智能,计算语言学.白硕,1956 年生,研究员,主要研究领域为人工智能,计算语言学.

本文通讯联系人:邓晶,北京 100080,国家智能计算机研究开发中心

本文 1995-07-13 收到修改稿

** 这种学习的定义类似于 *GOLD* 的语言辨识的定义,见文献[1].

的.所以对自动机的学习过程就是试图构造另一个自动机，它具有和原自动机完全相同的输入输出序列.这种输入输出序列可称为功能序列.于是自然可以问，一个自动机的功能序列是否是可学习的？

图灵机是公认的具有较强计算能力的一类自动机.它的功能序列是不是可学习的呢？

有一种办法似乎可以学到图灵机的功能序列：因为图灵机的状态和指令集都是有限的，那么图灵机的数量一定是可数无穷多的.我们可以枚举出所有的图灵机，看它们是否有和原图灵机相同的功能序列.这样下去不是总可以找到具有原图灵机功能的图灵机吗？可实际上，因为图灵机的停机问题不可判定，它的功能序列很可能得不到，从而这种“枚举学习”的办法不可能实施.* E. GOLD 已经指出，图灵机的功能序列不可学习.

但是图灵机是否就是不可学习的呢?从上面对“学习”的理解来看，学习能力取决于可观测量的范围.不同范围的可观测量对应着不同的学习能力.如果我们不仅认为图灵机的计算结果是可观测量，而且认为，它在进行计算时的行为——读写头在带子上的移动、对字符的改写也都是可观测量的话，学习能力就会有本质性的提高.

我们把图灵机进行计算时读写头的移动和改写字符称为图灵机的行为，并认为它们是图灵机的可观测量.把图灵机进行计算时的一系列(连续的)行为称为图灵机的行为序列.区别于图灵机的功能学习，我们把这种基于行为序列来学习一个图灵机的过程称为图灵机的行为学习.

我们将在下一章证明存在一个学习过程 A，对任意的图灵机 T 产生的行为序列 L，经过有限的学习步骤，A 都将给出一个图灵机 T_A，T_A 也能产生行为序列 L.因此，图灵机的行为序列是可学习的.但是，这个学习过程并不能判断经过多少步后就可以学到行为序列.也就是说，虽然图灵机的行为序列在有限步内就可学习到，但在学习中的任何时刻，学习系统自己都不能判定它是否已经学到了这个行为序列.

下面我们将给出图灵机行为序列的严格定义以及图灵机行为可学习性的证明.

1　图灵机的行为序列是可学习的

1.1　图灵机的行为序列

不失一般性，假设要学习的是一个确定性的图灵机，并且它只有唯一的正常结束状态 q_0:**

$$T=(K,\Sigma,\Gamma,\delta,q_1,\{q_0\}).$$

设 $qa\Rightarrow bpD$ 是当 T 在状态 q 时读到字符 a 所执行的动作.T 把字符 a 改为 b，把状态改为 p.读写头移动的方向为 D.

定义 1. 定义 $a\Rightarrow(b,D,*)$是图灵机的外部行为.它是可观测量.这里把它写作四元组$(a,b,D,*)$.其中 D 是读写头的移动，分为 S(不动)，L(左移)，R(右移)3 种.加入 * 是为了标记一个完整的计算过程的结束(从初态到终态).* 有‘\$’(正常结束)和‘#’(非正常结束)2 种标记.* 为空字符表示计算还没有结束.必要时用‘ε’代表空字符.

---

* 假设我们根据图灵机 T 的输入输出来构造图灵机 T'，当有个输入使 T 总不停机的话，T' 应该怎么办？它既不能使自己对这个输入永不停机，又不能使它有可能停机.

** 此定义引自文献[2].

定义 2. T 计算时的一系列(连续的)外部行为称作一个行为序列(见表 1、表 2).

表 1 T_1 的行为序列

p	$abD*$	p	$abD*$	p	$aDd*$	p	$abD*$
1	11R	2	11R	3	01R	4	11R
5	11R	6	00L	7	11L	8	11L
9	11L	10	11L	11	11L	12	00R
13	10R	14	11S $	15	11R	16	11R
17	01R	18	00L	19	11L	20	11L
21	11L	22	00R	23	10R	24	11S $
25	01R	26	11R	27	00L	28	11L
29	11L	30	00R	31	10R	32	11S $
…	…	…	…	…	…	…	…

表中列出了 3 个功能段:1～14,15～24,25～32

表 2 T_2 的行为序列

p	$abD*$	p	$abD*$	p	$aDd*$	p	$abD*$
1	aXR	2	bYL	3	XXR	4	YYR
5	BYS $	6	aXR	7	aaR	8	bYL
9	aaL	10	XXR	11	aXR	12	YYR
13	bYL	14	YYL	15	XXR	16	YYR
17	YYR	18	BYS $	19	aXR	20	aaR
21	bYL	22	aaL	23	XXR	24	aXR
25	YYR	26	bYL	27	YYL	28	XXR
29	YYR	30	YYR	31	bbS #	32	aXR
33	aaR	34	aaR	35	bYL	36	aaL
37	aal	38	XXR	39	aXR	40	aaR
41	YYR	42	bYL	43	YYL	44	aaL
45	XXR	46	aXR	47	YYR	48	YYR
49	BBS $	50	aXR	51	aaR	52	aaR
53	bYL	54	aaL	55	aaL	56	XXR
57	aXR	58	aaR	59	YYR	60	bYL
61	YYL	62	aaL	63	XXR	64	aXR
65	YYR	66	YYR	67	bYL	68	YYL
69	YYL	70	XXR	71	YYR	72	YYR
73	YYR	74	BYS $	…	…	…	…
…	…	…	…	…	…	…	…

表中列出了 5 个功能段:1～5,6～18,19～31,32～49,50～74

无穷长的行为序列可由无穷个有穷长的功能段,或有穷个有穷长的功能段加最后一个无穷长的功能段构成,甚至可由单独一个无穷长的功能段构成.

定义 3. 定义 $beh(T)$ 为 T 的所有行为序列的集合.

为了描述便利,我们用 q_{-1} 表示 T 的非正常结束状态,并设它和 q_0 都是 T 的结束状态.

定义 4. 定义 $w(T)=\{q_1,\ldots,q_{n_k}\}$ 为 T 的工作状态集.

定义 5. 对指令 $\delta:qa\rightarrow bpD$,定义 $\delta^{(i)}=a$ 为指令 δ 的输入. $\delta^{(s)}=\langle b,D,*\rangle$ 为 δ 的预期外部行为. $*$ 的含义与定义 2 相同. $\delta^{(c)}=p$ 为 δ 的预期状态转换.

一个行为序列可分成很多功能段,每个功能段表示图灵机的一个完整的计算过程,它的结束由外部行为中的‘$*$’(‘$’或‘#’)标识. 如果图灵机对某个输入停机的话,它会产生有

穷长的功能段，否则会产生无穷长的功能段(见表1、表2).

1.2 图灵机的行为学习

设所有确定性图灵机的集合是 DTM.

定义 6. 如果存在一个过程 A，对任意的 $L\in beh(T)$，$(T\in DTM)$，A 都能构造 T_A，使得：$L\in beh(T_A)$，那么图灵机的行为是可学习的.

当然，如果 L 是有限长的话，行为学习没有什么意义. 我们考虑的都是当 L 为无限长时的可学习性. 以后谈到 L 都默认它是无限长的.

我们先给出行为序列的学习过程 A，然后证明由 A 得到的图灵机 T_A 具有行为序列 L.

简单地讲，A 先假定 T_A 只有一个工作状态 q_1，然后根据所读到的行为来构造 T_A 的指令. 当发现所读到的行为与 T_A 已有的指令集冲突时，则通过修改指令或增加状态来解决这个冲突. 如此不断地构造和修改指令，T_A 的 δ 集能满足越来越长的行为序列. 这样下去，我们能证明经过有限步后，δ 集终将满足整个行为序列.

假设 $|w(T_A)|=n_k$，T_A 有了一个已学到的指令序列

$$\Delta=(\Delta_1,...,\Delta_m).$$

每个 Δ_i 由一条指令 $\delta_i:q_la\Rightarrow bq_kD$ 和构造这条指令时读到的最后一个行为在 L 里的序号 p_i 构成：

$$\Delta_i=(\delta_i,p_i),\qquad \delta_i:q_la\Rightarrow bq_kD.$$

Δ 中的所有指令构成 T_A 的指令集：$D(T_A)$.

定义 7. 设 T_A 的当前状态为 q，所读的行为是 $(a,b,D,*)$. 当 Δ 中存在以 qa 为左部的指令 δ_i 时，如果 $\delta_i^{(s)}=\langle b,D,*\rangle$，称指令 δ_i 满足所读的行为，这时也称 $D(T_A)$ 满足所读的行为；如果 $\delta_i^{(s)}\neq\langle b,D,*\rangle$，则称 δ_i 与所读的行为冲突.

定义 8. 对于图灵机 T 的某个行为序列 L，$Tx\in DTM$，如果 $D(Tx)$ 与 L 的某个行为冲突，则称 Tx 与 T 在 L 上不一致，否则称 Tx 与 T 在 L 上一致(这时也称 Tx 满足 L).

以下是对图灵机 T 的行为序列 L 的学习过程 A：

设 q 是 T_A 的当前状态，p 是正要读的行为在行为序列 L 里的序号. 初始化 $n_k=1$，$q=q_1$，$p=1$.

(1)读入 T 的行为 $(a,b,D,*)$，扩充 T_A 的 Σ,Γ. *

如果 $D(T_A)$ 中没有以 qa 为左部的指令，转入(2)，否则，转入(3).

(2)构造 $\Delta_{m+1}=(\delta_{m+1},p_{m+1})$. 其中，

$$\delta_{m+1}:qa\rightarrow bp_jD,\quad p_{m+1}=p$$

如果 $*=$'\$'，则　　$q_j=q_0,q=q_1$；

如果 $*=$'#'，则　　$q_j=q_{-1},q=q_1$；

否则　　$q_j=q_1,q=q_1$.

$p+1$(p 指向下一个行为)，转到(1).

---

* 可以精确地扩充 $\Sigma_{T_A},\Gamma_{T_A}$，使 $\Sigma_{T_A}\subseteq\Sigma_T,\Gamma_{T_A}\subseteq\Gamma_T$：记住 T 执行行为 $(a,b,D,*)$ 时读写头在带子上的位置. 当 $*=$'#'时 Σ_{T_A} 和 Γ_{T_A} 不变. 否则，$\Gamma_{T_A}=\Gamma_{T_A}\cup\{a,b\}$. 如果该位置第1次被 T 读到，$\Sigma_{T_A}=\Sigma_{T_A}\cup\{a\}$.

(3)当 δ_i 满足所读的行为时，使 $q=\delta_i^{(c)}$，$p+1$，转(1).

否则，表明 δ_i 与所读的行为冲突，这时需要修改 Δ.

修改的过程是：从 Δ_m 到 Δ_1，按递降顺序选择预期状态转换不是 q_0，q_{-1}，或 q_{n_k} 的指令. 如果 δ_j：$q_1a\to bq_kD(1\leqslant j\leqslant m)$ 是所选择的指令的话，把 q_k 改为 q_{k+1}，从 Δ 中去掉 Δ_{j+1} 到 Δ_m，$q=q_{k+1}$，$p=p_{j+1}+1$，转(1).

如果选不到指令(δ_1 到 δ_m 都不能修改)，说明 n_k 个状态数已不能构造满足 L 的指令集. 这时增加状态数：n_k+1，$\Delta=\varnothing$，$q=q_1$，$p=1$，转(1).

以上描述了学习过程 A，下面证明 $L\in beh(T_A)$.

定理 1. 对给定了 Σ,Γ，且 $|K|=n$ 的图灵机的数量是有限的.

证明：显然.

推论 1. Σ,Γ 分别包含于一个有限字符集，状态数均小于 n 的图灵机的数量是有限的.

引理 1. 在 A 的学习过程中，L 上的任何一个行为都将被 T_A 满足.

证明：因为 A 总是可以用增加状态来使 T_A 的 δ 集满足所读的行为. 命题成立.

引理 2. 在 A 的学习过程中，如果 T_A 已满足了 L 上的第 p 个行为，那么 T_A 是满足 L 上从序号 1 到序号 p 的子序列的状态数最少的图灵机.

证明：学习过程 A 只有尝试了原状态数下所有满足已读行为的 $D(T_A)$ 后，T_A 才增加新的状态. 命题成立.

定理 2. 图灵机 T 的行为序列 L 是可学习的.

证明：设被学的图灵机 $T=\{K_T,\Sigma_T,\Gamma_T,q_1,\delta_T,\{q_0\}\}$，

A 学到的图灵机 $T_A=\{K_{T_A},\Sigma_{T_A},\Gamma_{T_A},q_1,\delta_{T_A},\{q_0,q_{-1}\}\}$.

设 L 上 $\in\Sigma_T$ 的字符的集合是 Σ_{T_L}，$\Sigma_{T_L}\subseteq\Sigma_T$.

L 上 $\in\Gamma_T$ 的字符的集合是 Γ_{T_L}，$\Gamma_{T_L}\subseteq\Gamma_T$.

生成 L 时 T 所达到的状态的集合是 K_{T_L}. $|K_{T_L}|<|K_T|<+\infty$.

设 $\{T_x\}=\{T_d\,|\,T_d\in DTM,\Sigma_{T_d}\subseteq\Sigma_{T_L},\Gamma_{T_d}\subseteq\Gamma_{T_L},|K_{T_d}|\leqslant|K_{T_L}|,|\delta_{T_d}|\leqslant|\delta_T|\}$.

由推论 1，$|\{T_x\}|<+\infty$.

设 T_b 是与 T 在行为序列 L 上不一致的图灵机. 用 $d_f(L,T,T_b)$ 表示 L 上第 1 个与 T_b 的 δ 集冲突的行为的序号. 因为 $|\{T_x\}|<+\infty$，所以 $\{T_x\}$ 中与 T 在 L 上不一致的图灵机的个数也是有限的，设它们的集合为 $\{T_{df}\}$.

$$F_{df}=\{d_f(L,T,T_{df})\,|\,T_{df}\in\{T_{df}\}\}.$$

因为 $|\{T_{df}\}|<+\infty$，所以 F_{df} 中必有一个最大的数，设为 $max(F_{df})$.

根据过程 A，由引理 1，T_A 能满足第 $max(F_{df})$ 个行为. 这时的 T_A，要么已经与 T 在 L 上一致，要么 $d_f(L,T,T_A)>max(F_{df})$.

此时必有 $|K_{T_A}|>|K_T|$. 可因为 T 满足第 $max(F_{df})$ 个行为，由引理 2，$|K_{T_A}|\leqslant|K_T|$. 矛盾.

所以此时 $L\in beh(T_A)$，即图灵机 T 的行为序列是可学习的. 证毕.

推论 2. 过程 A 学到的 T_A 是满足所学行为序列 L 的状态数最少的图灵机.

2 学习的实例

这里给出 2 个行为学习的例子. 我们根据本文的学习过程编了程序. 下面 2 个例子的数

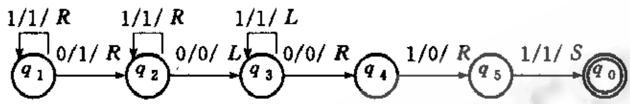
据都取自程序的运行结果.

2.1 行为序列 1 的学习

行为序列 1(表 1)是由图灵机 T_1 对输入 11011，110，01，... 产生的. T_1 的功能是把字符串$\overbrace{11110}^{m个1}\ \overbrace{11111}^{m个1}$转换成字符串$\overbrace{111111111}^{m+n个1}$. T_1 的指令集和状态转换图如下.

δ 集　$q_1 1 \to 1 q_1 R$　$q_1 0 \to 1 q_2 R$　$q_2 1 \to 1 q_2 R$　$q_2 0 \to 0 q_3 L$

　　　$q_3 1 \to 1 q_3 L$　$q_3 0 \to 0 q_4 R$　$q_4 1 \to 0 q_5 R$　$q_5 1 \to 1 q_0 S$



T_1 的学习过程如下：

$$\begin{array}{l} n_k=1 \\ \Delta_1(q_1 1 \to 1 q_1 R, 1) \\ p=2 \end{array}$$

读到第 1 个行为，构造指令序列 Δ，p 指向下一个行为. 〈a〉

$$\begin{array}{l} n_k=1 \\ \Delta_1(q_1 1 \to 1 q_1 R, 1) \\ p=3 \end{array} \Rightarrow \begin{array}{l} n_k=1 \\ \Delta_1(q_1 1 \to 1 q_1 R, 1) \\ \Delta_2(q_1 0 \to 1 q_1 R, 3) \\ p=4 \end{array}$$

第 2 个行为满足 Δ，读到第 3 个行为时构造 Δ. 〈b〉

$$\begin{array}{l} n_k=1 \\ \Delta_1(q_1, 1 \to 1 q_1 R, 1) \\ \Delta_2(q_1 0 \to 1 q_1 R, 3) \\ p=6 \end{array} \Rightarrow \begin{array}{l} n_k=2 \\ \varnothing \\ p=1 \end{array}$$

第 6 个行为与 δ_2 冲突，因为 δ_1 与 δ_2 都不能修改，Δ 为空，T_A 增加一个状态，$n_k=2$，p 重新指向 1. 〈c〉

$$\begin{array}{l} n_k=2 \\ \Delta_1(q_1 1 \to 1 q_1 R, 1) \\ \Delta_2(q_1 0 \to 1 q_1 R, 3) \\ p=6 \end{array} \Rightarrow \begin{array}{l} n_k=2 \\ \Delta_1(q_1 1 \to 1 q_1 R, 1) \\ \Delta_2(q_1 0 \to 1 q_2 R, 3) \\ p=4 \end{array}$$

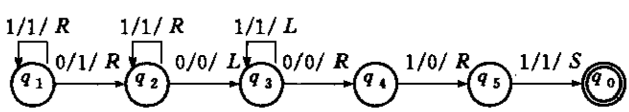
当 p 再次指向 6 时，δ_2 又与行为冲突，这时 δ_2 可修改. 〈d〉

$$n_k=5$$

$\Delta_1(q_1 1 \to 1 q_1 R, 1)$　$\Delta_2(q_1 0 \to 1 q_2 R, 3)$　$\Delta_3(q_2 1 \to 1 q_2 R, 4)$　$\Delta_4(q_2 0 \to 0 q_3 L, 6)$

$\Delta_5(q_3 1 \to 1 q_3 L, 7)$　$\Delta_6(q_3 0 \to 0 q_4 R, 12)$　$\Delta_7(q_4 1 \to 0 q_5 R, 13)$　$\Delta_8(q_5 1 \to 1 q_0 S, 14)$

Δ 对应的状态转换图：



如此下去，当学完第 1 个功能段($P=15$)，Δ 和 T_A 状态转换图如上，它已和 T 完全相同. 以后一直到 $P=32$，Δ 不变. 〈e〉

2.2　行为序列 2 的学习

行为序列 2(表 2)是由图灵机 T_2 对输入 $ab, aabb, aabbb, aaabb, aaabbb, \ldots$ 产生的. T_2 接受上下文无关语言 $\{a_nb_n \mid n\geq 1\}$. T_2 的指令集和状态转换图如下：

δ 集　$q_1a\to Xq_2R$　　$q_2a\to aq_2R$

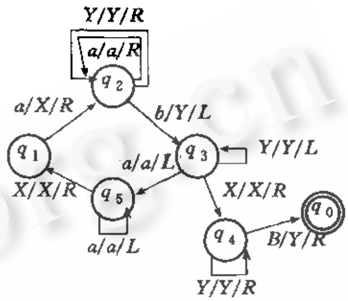
$q_2Y\to Yq_2R$　　$q_2b\to Yq_3L$

$q_3Y\to Yq_3L$　　$q_3X\to Xq_4R$

$q_3a\to aq_5L$　　$q_5a\to aq_5L$

$q_5X\to Xq_1R$　　$q_4Y\to Yq_4R$

$q_4B\to Yq_0R$



T_2 的学习过程见下：

$n_k=1$

Δ_1　$(q_1a\to Xq_1R, 1)$

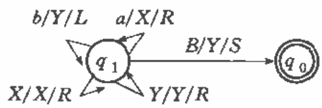
Δ_2　$(q_1b\to Yq_1L, 2)$

Δ_3　$(q_1X\to Xq_1R, 3)$

Δ_4　$(q_1Y\to Yq_1R, 4)$

Δ_5　$(q_1B\to Yq_0S, 5)$

$p=6$



学完第 1 个功能段后的 Δ 与状态转换图　〈a〉

$n_k=3$

Δ_1　$(q_1a\to Xq_2R, 1)$　Δ_2　$(q_2b\to Yq_1L, 2)$

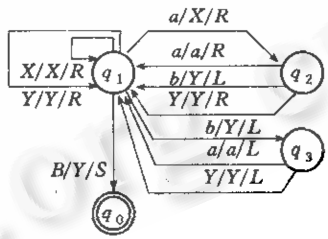
Δ_3　$(q_1X\to Xq_1R, 3)$　Δ_4　$(q_1Y\to Yq_1R, 4)$

Δ_5　$(q_1B\to Yq_0S, 5)$　Δ_6　$(q_2a\to aq_1R, 7)$

Δ_7　$(q_1b\to Yq_3L, 8)$　Δ_8　$(q_3a\to aq_1L, 9)$

Δ_9　$(q_2Y\to Yq_1R, 12)$　Δ_{10}　$(q_3Y\to Yq_1L, 14)$

$p=19$



学完第 2 个功能段后的 Δ 和 T_A 的状态转换图　〈b〉

$n_k=3$

Δ_1　$(q_1a\to Xq_2R, 1)$　Δ_2　$(q_2b\to Yq_1L, 2)$

Δ_3　$(q_1X\to Xq_1R, 3)$　Δ_4　$(q_1Y\to Yq_1R, 4)$

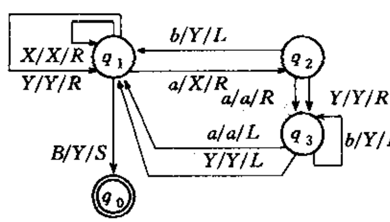
Δ_5　$(q_1B\to Yq_0S, 5)$　Δ_6　$(q_2a\to aq_3R, 7)$

Δ_7　$(q_3b\to Yq_3L, 8)$　Δ_8　$(q_3a\to aq_1L, 9)$

Δ_9　$(q_2Y\to Yq_3R, 12)$　Δ_{10}　$(q_3Y\to Yq_1L, 14)$

Δ_{11}　$(q_1b\to bq_{-1}S, 31)$

$p=32$

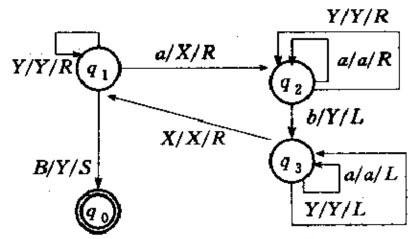


学完第 3 个功能段后的 Δ 和 T_A 的状态转换图　〈c〉

$n_k=3$

Δ_1　$(q_1a\to Xq_2R, 1)$　Δ_2　$(q_2b\to Yq_3L, 2)$

Δ_3 $(q_3X\rightarrow Xq_1R,3)$ Δ_4 $(q_1Y\rightarrow Yq_1R,4)$
Δ_5 $(q_1B\rightarrow Yq_0S,5)$ Δ_6 $(q_2a\rightarrow aq_2R,7)$
Δ_7 $(q_3a\rightarrow aq_3L,9)$ Δ_8 $(q_2Y\rightarrow Yq_2R,12)$
Δ_9 $(q_3Y\rightarrow Yq_3L,14)$ Δ_{10} $(q_1b\rightarrow bq_{-1}S,31)$
Δ_{11} $(q_2B\rightarrow bq_{-1}S,49)$
$p=50$



学完第 4 个功能段后的 Δ 和 T_A 的状态转换图. 如果假设 δ 中不存在的指令都把状态转到非正常结束,那么这时的 T_A 已和 T 行为一致. (d)

3 讨 论

学习是针对可观测量的学习. 黑箱不能观测的部分,可以称为内部状态,但这只是一个名称,在评价学习方面,它没有任何意义(对观测者而言,可观测量就是黑箱 S. 讨论在 S 的可观测量外更"本质"的 S 以及它是否能被学到,可观测量是否完全代表了它,都是没有意义的). 我们要学习的图灵机 T 的可观测量就是它的行为序列 L. 因此学到了 L,就可以认为学到了图灵机 T.

实际上的学习对象都不是纯粹的"黑箱",我们总是对学习对象的非观测部分作了一些假定. 那些假定是学习前就"知道"的,它们是非观测的,也是非学习的. 假定之外的部分,才是需要学习的"黑箱". 正是假定了行为序列是由图灵机产生的,我们才能让学到的对象 T_A 有 Σ,δ 等. T_A 的这些属性是我们默认的部分,它们不能被学习改变. 由于有了这些条件的约束,A 所学习的行为序列就不会是任意的,A 能经过有限次的学习步骤获得行为序列的"规律",从而能"再生"它.

目前为止,我们知道的对黑箱 S 的学习有 3 种:(1)S 可学习,并且学习可检验,我们可知道什么时候学习能完成. (2)S 可学习,但学习不能检验. (3)S 不可学习.

具有第 1 种学习的对象很少. 一般它们的可观测量都是有限的. 图灵机的行为学习是属于第 2 种学习. 按照学习过程 A,如果还没有学到原图灵机的行为序列,我们必然会在有限步内发现这个事实. 但在任何时候,我们都不能肯定是否已经学到了原图灵机的行为. 也就是说,原图灵机的行为序列中存在一个序号为 p 的行为,在满足了序号小于 p 的所有行为以后,学到的图灵机就不会再改变,而且它还将满足序号大于 p 的所有行为. 可我们不知道 p 到底是多少.

并不是说凡是学习对象是可枚举的都能用本文的方法学到. 事实上,学习要受很多具体因素影响. 虽然图灵机可枚举,但它的功能学习却是不可能的.

这种行为学习的方法可相应地运用于对 LR 分析器的学习. 我们知道,LR 分析器是由文法规则和分析策略组成的,它的整个分析策略就是 LR 分析表. 分析表由动作和状态构成,每个状态根据输入的符号执行不同的动作(包括移进、归约、接受和出错)和状态转移.

因为可以从一个未知的 LR 分析器输出的句子的生成树看到分析表的动作的外部表现(例如移进和运用什么规则进行归约等),所以可以把分析树当作 LR 分析器黑箱的外部行为,而把分析表里的状态当作 LR 分析器的内部状态,分析表里的项当作黑箱内的指令.

因为分析表中的状态和项都是有限的,根据本文的行为学习过程的原理,我们能够得出

结论:*LR* 分析器是行为可学习的.

4　结束语

本文讨论了黑箱的学习.并相应于功能学习,提出了行为学习的概念,证明了图灵机产生的行为序列是可学习的.也就是说,我们能根据图灵机在计算时可观测的动作来构造一个和它行为一致的图灵机.按照行为学习的原理,LR 分析器也是行为可学习的.

参考文献

1　Mark E Gold. Language identification in the limit. Information And Control, 1967, **10**(5):447～474.
2　何成武.自动机理论及其应用.北京:科学出版社,1990.

THE BEHAVIOR LEARNING OF TURING MACHINES

Deng Jing　Bai Shuo

(*National Research Center for Intelligent Computing Systems　Beijing*　100080)

Abstract　The observable computing actions of a Turing machine, including moving and writing, can be categorized as the behavior of the Turing machine. In this paper, it is shown that the behavior of a Turing machine is learnable. That is, there exists a learning procedure which, given the behavior of a Turing machine, will yield a new Turing machine with the same behavior in a finite number of steps. According to the same mechanism, the behavior of an LR parser is also learnable.

Key words　Learning, behavior, behavior sequence, behavior learning, function learning.