计算理论导论

习题七: 图灵机与可判定性

中国人民大学 信息学院 崔冠宇 2018202147

1. 3.2 This exercise concerns TM M_1, whose description and state diagram appear in Example 3.9. In each of the parts, give the sequence of configurations that M_1 enters when started on the indicated input string.

d. 10#11.

解: d. 计算格局 (configuration) 序列为:

$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#xq_{reject}1。

2. 3.15 Show that the collection of decidable languages is closed under the operation of

b. concatenation. **c.** star. **e.** intersection.

证明: 设 $A = L(M_A)$ 和 $B = L(M_B)$ 是两个图灵可判定语言，其中 M_A 和 M_B 是判定它们的 TM。

b. 判定思路：构造一台图灵机 M_{AB} 判定 AB。对于任意输入 w，

1. M_{AB} 非确定性地将 w 切成 x 和 y 两段；
2. 将 x 作为 M_A 的输入，模拟 M_A，若 M_A 接受 x，则进行下一步，否则 M_{AB} 拒绝 w；
3. 将 y 作为 M_B 的输入，模拟 M_B，若 M_B 接受 y，则 M_{AB} 接受 w，否则 M_{AB} 拒绝 w。

由于 w 有限长，M_{AB} 尝试有限种切分方法，而每种方法都会在有限步得到接受或拒绝的结果（因为两个子机器都是判定器），因此 M_{AB} 会在有限步内给出接受或拒绝的结果，从而能够判定 AB。

c. 判定思路：构造一台图灵机 M_{A^*} 判定 A^*。对于任意输入 w，

1. 若当前输入为空串，接受，否则 M_{A^*} 非确定性地从当前位置切下一段字符串 x；

2. 将 x 作为 M_A 的输入，模拟 M_A，若 M_A 接受 x，返回第 1 步，否则 M_{A^*} 拒绝 w；

由于 w 有限长，每次将切下的字符串在 M_A 上模拟都会在有限步内终止，因此 M_{A^*} 能判定 A^*。

e. 判定思路：构造一台图灵机 $M_{A\cap B}$ 判定 $A \cap B$。对于任意输入 w，

1. 将 w 作为 M_A 的输入，模拟 M_A，若 M_A 接受 w，则进行下一步，否则 $M_{A\cap B}$ 拒绝 w；
2. 将 w 作为 M_B 的输入，模拟 M_B，若 M_B 接受 w，则 $M_{A\cap B}$ 接受 w，否则拒绝 w。

因为 M_A 和 M_B 都是判定器，对于任意输入 w 都会在有限步内给出接受或拒绝的结果，从而 $M_{A\cap B}$ 能够判定 $A \cap B$。

3. **3.16** Show that the collection of Turing-recognizable languages is closed under the operation of

b. concatenation. **c.** star. **e.** homomorphism.

证明:

b. 识别思路：构造一台图灵机 M_{AB} 识别 AB。对于任意输入 w，

1. M_{AB} 非确定性地将 w 切成 x 和 y 两段；
2. 将 x 作为 M_A 的输入，模拟 M_A，若 M_A 接受 x，则进行下一步，若拒绝 x，则 M_{AB} 拒绝 w；
3. 将 y 作为 M_B 的输入，模拟 M_B，若 M_B 接受 y，则 M_{AB} 接受 w，若拒绝 y，则 M_{AB} 拒绝 w。

对于属于 AB 的字符串 w，显然会经过有限步停止，因此 M_{AB} 识别 AB。

c. 识别思路：构造一台图灵机 M_{A^*} 识别 A^*。对于任意输入 w，

1. M_{A^*} 非确定性地从当前位置切下一段字符串 x；
2. 将 x 作为 M_A 的输入，模拟 M_A，若 M_A 接受 x，返回第 1 步，否则拒绝；

对于属于 A^* 的字符串 w，显然会经过有限步停止，因此 M_{A^*} 识别 A^*。

e. 识别思路：设 h 是同态映射，构造一台图灵机 N 来识别 $h(A)$。对于任意输入 w，

1. N 非确定性地枚举 x；

2. 若 $h(x) = w$，在 M 上运行 x，若 M 接受 x，则 N 接受 w，否则拒绝。

对于属于 $h(A)$ 的字符串，由于是非确定性枚举 x，一定能找到 $h(x) = w$ 从而接受。

4. **3.18** Show that a language is decidable iff some enumerator enumerates the language in the standard string order.

证明:

($\Rightarrow$) 若 A 是可判定的，设图灵机 M_A 判定它，构造一台枚举器 E 以标准字典序枚举它：

1. E 枚举字典序下一个字符串 w；

2. 用 M_A 判定 w 是否属于 A，若 M_A 接受 w，则 E 输出 w，否则不输出 w；

3. 回到步骤 1。

($\Leftarrow$) 若 E 是以标准字典序枚举某语言的枚举器，构造一台判定 A 的图灵机 M_A。对于任意输入 w：

1. M_A 让 E 枚举下一个字符串 x（若没有下一个字符串，则 M_A 拒绝 w）；

2. 若 $x < w$（大小关系指字典序，下同），回到步骤 1；

3. 若 $x = w$，M_A 接受 w；

4. 若 $x > w$，M_A 拒绝 w。

显然 M_A 判定了 A。

5. Here is an informal description of a 2-Stack Nondeterministic Pushdown Automata (2-NPDA): the machine is just like an ordinary NPDA, except there is a second stack that behaves just like the first. At each step, the machine reads a symbol from the tape (possible ε), pops specified symbols from each of the two stacks (either may be ε), pushes specified symbols onto each of the two stacks (either may be ε), and moves into a specified state. The machine accepts if there is some computation on its input string that causes it to reach an accept state.

(a) Give an informal description of a 2-NPDA that decides the language:

$$L = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n | n \geq 0\}$$

(b) Prove that 2-NPDAs are equivalent to Turing Machines. That is, show language L is decided by a 2-NPDA if and only if it is decided by a Turing Machine.

解:

(a) 非形式化的描述：构造一台 2-NPDA P 来判定 L：

1. 初始状态为状态 0，向两栈分别压一个字符 `#`，进入状态 1；
2. 在状态 1，若遇到 `a`，入栈 A，保持状态 1；
3. 在状态 1，若遇到 `b`，入栈 B，进入状态 2；
4. 在状态 2，若遇到 `b`，入栈 B，保持状态 2；
5. 在状态 2，若遇到 `c`，栈 A 和栈 B 分别往外弹一个字符，进入状态 3；
6. 在状态 3，若遇到 `c`，栈 A 和栈 B 分别往外弹一个字符，保持状态 3；
7. 在状态 3，若读完所有输入后两栈顶都为 `#`，接受，否则拒绝。

(b) 证明思路：证明二者可以相互模拟。下面用两个引理来从两方面证明结论。

引理 1. *对任意 2-NPDA N，都存在一台 (3-tape) NTM M 能够模拟 N。*

证明. 对于任意 2-NPDA N，讨论 NTM M 模拟 N 各操作的方法（对于输入部分，约定读写头指向当前读取字符；对于栈部分，约定读写头指向栈顶字符）：

1. 压栈操作。压栈可以先将 2、3 带读写头右移，然后在右侧空白格上写一个字符；

2. 弹栈操作。弹栈可以将 2、3 带最右侧非空白字符抹掉成空白字符，然后读写头左移。

M 对 N 的模拟过程：

1. 将 N 的所有输入写到带子 1 上；

2. 对 N 的每一步计算，即对一个状态转移 $\texttt{i}, (\texttt{a}, \texttt{b}) \to (\texttt{c}, \texttt{d})$（读到字符 i，弹出两栈顶元素 a 和 b，压入 c 和 d）可以用以下方式模拟：

   (a) 1 号带读头右移至下一个字符；

   (b) 2、3 号带读写头将当前指向的 a 和 b 抹成空白符，读写头左移（如果不对应，此机器“死亡”）；

   (c) 2、3 号带读写头向右移，写上 c 和 d。

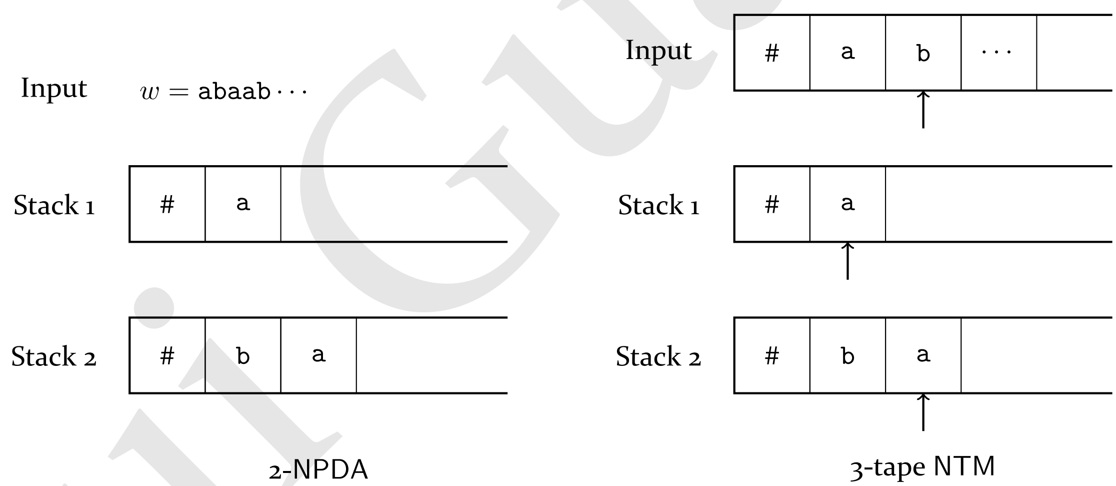

图 1: 2-NPDA 用 NTM 模拟

□

引理 2. *对任意 (single-tape) TM M，都存在一台 2-NPDA N 能够模拟 M。*

证明. 对于任意 single-tape TM M，讨论 2-NPDA N 模拟 M 各操作的方法：（约定栈 1 存放 M 读写头左侧所有字符，栈 2 存放读写头指向字符以及右侧所有非空白字符）

1. 读写头写一个字符：将栈 2 的栈顶元素弹出，然后压入新字符；

2. 读写头右移：将栈 2 的栈顶元素弹出，然后压到栈 1 中，若此时栈 2 已经到栈底，则再向栈 2 压入一个空白符 ␣（左移则相反）。

N 对 M 的模拟过程：

1. 将 M 的输入依次压入栈 1 中，然后逐个弹出依次压入栈 2 中；
2. 对 M 的每一步计算，即对一个状态转移 $\mathtt{i} \rightarrow \mathtt{j}, \mathsf{L}/\mathsf{R}$ （读到字符 `i`，将其改写成 `j`，然后左移/右移）可以用以下方式模拟：
   (a) 将栈 2 顶部的 `i` 弹出（如果不对应，则该机器拒绝），然后压入字符 `j`；
   (b) 用上述的方法进行左移/右移。

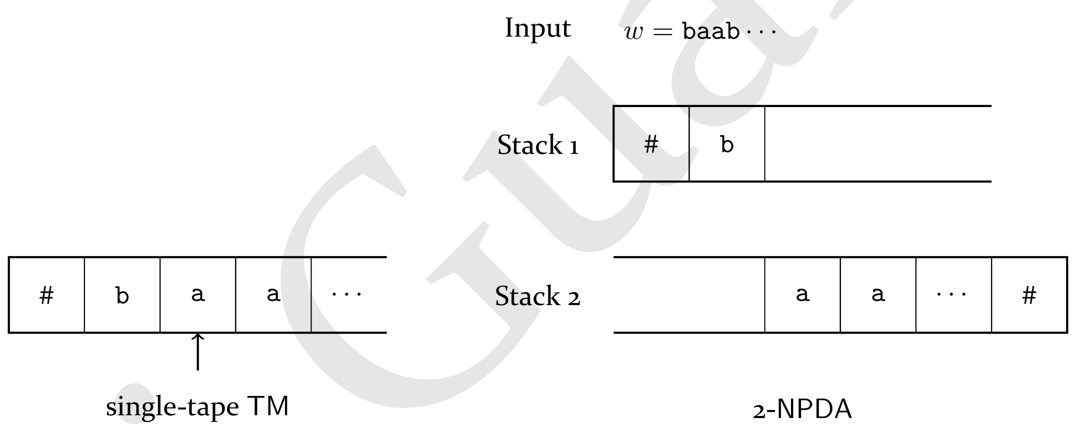

图 2: TM 用 2-NPDA 模拟

□

根据上面的两个引理，我们可以得到：2-NPDA 可以用 3 带 NTM 模拟，而后者可以用单带 TM 模拟，又单带 TM 可以用 2-NPDA 模拟，于是我们就得到二者等价的结论。

6. Consider the following problem: you are given an NFA and a PDA and you would like to know whether there exists a string that they both accept. Formulate this problem as a language and prove it is decidable.

解:

形式化后的语言：$L_{\mathsf{CFL}\cap\mathsf{Regular}\neq\emptyset}=\{\langle N,P\rangle | N$ 是 NFA, P 是 PDA, 且 $L(N)\cap L(P)\neq\emptyset\}$。

可判定性的证明：构造一台 TM 来判定这个问题。对任意 NFA N 以及 PDA P，

1. 根据“每个 NFA 都存在一个等价的 DFA 与之对应”([Sipser, P55, **Theorem 1.39**])，将 N 转成与之等价的 DFA D；
2. 根据“上下文无关语言 G 与正则语言 R 的交集是上下文无关语言”([Sipser, P161, **Problems 2.18 a**])，利用 D 和 P 构造识别它们交语言的 PDA；
3. 根据“每一个 PDA 都有一个等价的 CFG”([Sipser, P121, **Lemma 2.27**])，将上述 PDA 转换为等价的 CFG；
4. 根据“上下文无关语言 G 是空语言”是可判定的（[Sipser, P199, **Theorem 4.8**]），判定该语言是否为空即可。

7. Consider the problem of determining whether the language of a given Turing machine contains at least 2016 strings. Formulate this problem as a language 2016TM and show that 2016TM is Turing-recognizable.

解:

形式化后的语言：$L_{\mathsf{2016TM}}=\{\langle M\rangle | M$ 是一台 TM 且 $|L(M)|\geq 2016\}$。

可识别性的证明：构造一台 TM 来识别这个问题，

1. 首先约定 Σ^* 的元素按字典序排成 $\{w_1,w_2,\cdots\}$；
2. 在第 i 轮运行 $w_1,\cdots,w_i$ 各 i 步；
3. 若某 w_k 被接受，则计数器加一，同时标记此串表示下次不再运行它；
4. 若计数器已经超过 2016，接受，否则回到第二步。

8. **4.20** Let A and B be two disjoint languages. Say that language C separates A and B if $A \subseteq C$ and $B \subseteq \overline{C}$. Show that any two disjoint co-Turing-recognizable languages are separable by some decidable language.

证明: 设 L_1 和 L_2 是不相交的两个 co-Turing-recognizable 的语言，即存在两个 TM M_1 和 M_2 分别识别 $\overline{L_1}$ 以及 $\overline{L_2}$。下面构造一台图灵机 M，对于任意输入 w，

1. 在 M_1 上运行 w 一步，若此时 M_1 接受 w，则 M 拒绝 w；
2. 在 M_2 上运行 w 一步，若此时 M_2 接受 w，则 M 接受 w。

断言：$L = L(M)$ 是分开 (seperate) L_1 和 L_2 的可判定语言。只需要分别证明以下两个引理。

引理 3. *L 是可判定的。*

证明. 只需要证明 M 对于任意输入 w 都会在有限步内停机即可。

用反证法，假设 M 对某输入 w 不停机，则根据 M 的定义，w 不会被 M_1 接受，也不会被 M_2 接受。因为 M_1 不接受 w，因此 $w \notin \overline{L_1}$，即 $w \in L_1$；同理 $w \in L_2$，因此有 $w \in L_1 \cap L_2$，但是这与 $L_1 \cap L_2 = \emptyset$ 矛盾。因此假设不成立，M 一定会在有限步内停机，从而 $L = L(M)$ 是可判定的。 □

引理 4. *L 分开 (separates) L_1 和 L_2。*

证明. 分两部分证明。

1. 对 $\forall w \in L_1$，因为 $w \notin \overline{L_1}$，所以 w 不会被 M_1 接受，因此 w 将被 M_2 接受，从而被 M 接受，因此 $w \in L = L(M)$，即 $L_1 \subseteq L$。
2. 对 $\forall w \in L_2$，因为 $w \notin \overline{L_2}$，所以 w 不会被 M_2 接受，因此 w 将被 M_1 接受，从而被 M 拒绝，因此 $w \notin L$，或说 $w \in \overline{L}$，即 $L_2 \subseteq \overline{L}$。

根据定义，L 分开了 L_1 和 L_2。 □