第十一章 形式语言和自动机的层次结构

姚　刚

中国科学院信息工程研究所

目录

关于语言的问题

图灵机有很强的计算能力，它能接受很多语言，那么是否有图灵机不能接受的语言？图灵机可以接受正则语言、上下文无关语言等等，这些语言之间的关系是怎么样的？生成这些语言的文法之间有什么联系？在下面的讨论中，我们论证的结论仅对不包含空串ε的语言才成立。这是由于我们定义的图灵机是不接受空串的。在语言包含空串的情况下，重新阐述我们的结论是不困难的。

递归可枚举集

定义 (递归可枚举集)

一个语言L，如果存在一个接受它的图灵机，则称这个语言是递归可枚举的(*recursively enumerable*)。这表明存在一个图灵机M，对于所有的$w \in L$，满足

$$q_0 w \overset{*}{\vdash}_M x_1 q_f x_2,\ q_f \in F。$$

递归集合

定义 (递归集合)

在Σ上的语言L被称为是递归的(recurive)，如果存在一个图灵机M，该图灵机接受语言L，并且对Σ^+中的所有w都能停机。换句话说，一个语言是递归的当且仅当这个语言存在一个成员资格判定算法。

因为图灵机对不接受的输入不一定总会停机，所以图灵机接受的语言和存在成员资格判定算法的语言要严格区分。

递归语言的枚举过程

如果一个语言是递归的，那么存在一个简单的枚举过程。

假设M是一个判定语言L的成员资格的图灵机。我们有另一个图灵机先按照良序产生Σ^+中的字符串，然后将他们作为M的输入。而M被修改为仅仅把L中的串写在它的带上。

递归可枚举语言的枚举过程

对于递归可枚举语言有类似的枚举过程，但并不是像前面的那样简单。因为可能存在某个字符串，M永不停机，那么后面的串将无法枚举。

我们换种方法来执行，先生成w_1并对其作一次迁移；然后生成w_2并作一次迁移，接着对w_1作第二次迁移；接着生成w_3并作一次迁移，作w_2的第二次迁移，作w_1的第三次迁移；以此类推。L 中的所有符号串都能被M识别。

问题

给出的定义中并没有给出递归语言和递归可枚举语言的本质性信息。定义没有展示语言族成员的典型语言本质，也没有展示语言之间的关系与联系。我们马上会有这样的问题：是否存在语言是递归可枚举的，但不是递归的？是否存在语言可以被清楚地描述，但却不是递归可枚举的？对于这些问题，我们可以给出一些答案，但是却没有办法给出具体的例子。

定理

定理

如果S是一个无穷可数集合，那么它的幂集2^S不是可数的。

证明中使用的方法称为对角线法，在解决某些形式语言方面的问题时是一个有效的方法。

定理

对于任何非空的集合Σ，都存在非递归可枚举的语言。

对非递归可枚举语言的描述

既然所有能够被图灵机接受的语言都能用一种直接的算法方式描述，并且都是递归可枚举的，那么对于非递归可枚举语言的描述必然是非直接的。

但是，这种描述是可能的，证明的方法使用了对角线化理论的变种。

定理

定理

存在某种递归可枚举语言，它的补不是递归可枚举的。

定理

如果一个语言L和他的补$\overline{L}$都是递归可枚举的，那么两者都是递归的。如果L是递归的，那么$\overline{L}$也是递归的，同时两者都是递归可枚举的。

定理

定理

存在一个语言L，它是递归可枚举的，但它却不是递归的。换句话说，递归语言族是递归可枚举语言族的真子集。

递归可枚举语言与文法

为了考察递归可枚举的语言和文法之间的联系，我们再回过头来看文法的一般性定义。在最开始的文法的定义中，产生式规则可以采用任意形式，但是后来我们看到的特定文法都被加上了各种各样的限制。

如果采用一般性的方式并且不加任何限制，我们会得到无限制文法G。

无限制文法

定义 (无限制文法)

文法$G = (V, T, S, P)$被称为无限制文法(*unrestricted grammar*)当且仅当它的每个产生式都有形式:

$$u \to v,$$

这里$u \in (V \cup T)^+$，$v \in (V \cup T)^*$。

无限制文法与递归可枚举语言

无限制文法没有对产生式强加任何条件，唯一的限制就是ε不能作为一个产生式的左部。无限制文法比我们所研究的受限制文法，如上下文无关文法和正则文法，具有更强的表达能力。事实上，无限制文法对应了我们使用机械方法所能识别的最大语言族。

定理

任何有无限制文法生成的语言都是递归可枚举的。

由图灵机构造文法

给定一个图灵机$M = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$，我们来构造一个无限制文法$G$。既然图灵机的计算过程可以由如下瞬时描述序列表示：

$$q_0 w \overset{*}{\vdash} x q_f y, \tag{1}$$

当且仅当(1)成立，相应的文法具有如下特性：

$$q_0 w \overset{*}{\Rightarrow} x q_f y。\tag{2}$$

由图灵机构造文法(续)

对于所有满足条件(1)的w，我们需要在式(2)和我们需要的形式$S \overset{*}{\Rightarrow} w$之间建立联系。为此，文法大体有如下特征：

1. 对于所有的$w \in \Sigma^+$，S都可以推导出$q_0 w$。
2. 当且仅当式(1)成立，才可能有式(2)。
3. 当符号串$xq_f y$ $(q_f \in F)$生成时，文法将符号串转化为最初的w。

由图灵机构造文法(续)

完整的推导过程如下：

$$S \stackrel{*}{\Rightarrow} q_0 w \stackrel{*}{\Rightarrow} x q_f y \stackrel{*}{\Rightarrow} w。$$

这里第三步比较麻烦，问题是如何恢复被修改了的w。为此，对于所有的$a \in \Sigma \cup \{\Box\}$，$b \in \Gamma$以及所有满足$q_i \in Q$的$i$，引入变量$V_{ab}$和$V_{aib}$。$V_{ab}$是对符号$a$和$b$编码，而$V_{aib}$则同时对$a$、$b$和状态$q_i$编码。

构造产生式

对于第一步，可以有如下的产生式得到：

$$
\begin{aligned}
S &\rightarrow V_{\square\square}S|SV_{\square\square}|T, \\
T &\rightarrow TV_{aa}|V_{a0a},
\end{aligned}
$$

其中$a \in \Sigma$。

构造产生式(续)

第二步中，对于转换$\delta(q_i, c) = (q_j, d, R)$，构造产生式

$$V_{aic}V_{pq} \to V_{ad}V_{pjq},$$

对于转换$\delta(q_i, c) = (q_j, d, L)$，构造产生式

$$V_{pq}V_{aic} \to V_{pjq}V_{ad},$$

其中所有的$a, p \in \Sigma \cup \{\Box\}$，$q \in \Gamma$。

构造产生式(续)

第三步中，对于每一个$q_j \in F$，$a \in \Sigma \cup \{\square\}$，$b \in \Gamma$，我们引入产生式

$$V_{ajb} \to a,$$

对于所有的$a, c \in \Sigma \cup \{\square\}$，$b \in \Gamma$，引入产生式

$$cV_{ab} \to ca, \qquad V_{ab}c \to ac,$$

最后还需要一个特殊的产生式$\square \to \varepsilon$。

例子

对于图灵机$M = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$，令

$Q = \{q_0, q_1\}$， $\Gamma = \{a, b, \Box\}$，

$\Sigma = \{a, b\}$， $F = \{q_1\}$，

$\delta(q_0, a) = (q_0, a, R)$，

$\delta(q_0, \Box) = (q_1, \Box, L)$，

该图灵机接受$L(aa^*)$。

定理

定理

对于每一个递归可枚举语言L，存在一个无限制文法G，满足$L = L(G)$。

前面两个定理表明：无限制文法相应的语言族与递归可枚举的语言族是等同的。

上下文相关文法

定义 (上下文相关文法)

给定一个文法$G = (V, T, S, P)$，如果它的所有产生式都有如下形式：

$$x \to y,$$

这里$x, y \in (V \cup T)^+$，并且$|x| \leq |y|$，则称这个文法是上下文相关的(*context-sensitive*)。

一个特征

上下文相关文法的一个特征：在推导过程中，后继句型的长度是不会减小的，即上下文相关文法是不会收缩的(noncontracting)。

虽然为什么称这种文法为上下文相关文法不是很明显，但是可以看到这种文法可以用一种范式来改写，其中所有产生式有形式$xAy \to xvy$，即产生式$A \to v$仅当在A的左边是x，右边是y的这种上下文情况下才能被使用。

上下文相关语言

定义 (上下文相关语言)

语言L是一个上下文相关语言当且仅当存在一个上下文相关文法G，满足$L = L(G)$或者$L = L(G) \cup \{\varepsilon\}$。

上下文相关语言和上下文无关语言

因为上下文相关文法不允许有$x \to \varepsilon$的产生式，其不会生成包含$\{\varepsilon\}$的语言。每个不包含$\{\varepsilon\}$的上下文无关语言都可以由一个上下文相关文法的特例，比如乔姆斯基范式或者格里巴克范式来生成。

通过在上下文相关语言中(而不是在文法中)引入空串，我们可以说上下文无关语言的集合是上下文相关语言集合的一个子集。

例子

$L = \{a^n b^n c^n : n \geq 1\}$是一个上下文相关语言，构造一个上下文相关文法来表明这一点。

这个例子说明上下文无关语言的集合是上下文相关语言集合的一个真子集。在一般情况下，构造上下文相关文不是很容易，通常的解决办法实现在图灵机上获得程序，然后在构造一个等价的文法。

定理

定理

对于每个不包含空串的上下文相关语言L，都存在某个线性有界自动机M，满足$L = L(M)$。

定理

如果语言L被某个线性有界自动机M接受，那么就存在某个能够产生L的上下文相关文法。

定理

定理

每个上下文相关语言L都是递归的。

定理

存在不是上下文相关语言的递归语言。

结论

线性有界自动机不如图灵机的能力强，因为它们接受的只是递归语言的真子集。同时，线性有界自动机确实比下推自动机的能力强。上下文无关文法产生的上下文无关语言是上下文相关语言的真子集。线性有界自动机和上下文相关语言在本质上是等价的，而下推自动机和上下文无关语言在本质上是等价的。因此，所有下推自动机接受的语言同样也能被线性有界自动机所接受。

语言族

我们已经看到了很多种语言族，其中包括递归可枚举语言族L_{RE}、上下文相关语言族L_{CS}、上下文无关语言族L_{CF}和正则语言族L_{REG}。
此外还有确定型上下文无关语言族L_{DCF}和递归语言族L_{REC}。

这些语言族之间的关系可以用乔姆斯基层次结构(Chomsky hierarchy)表示。

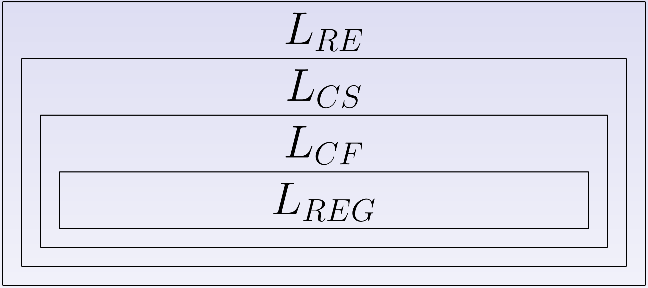
形式语言最初分类

形式语言最初分为四个类型，从0型到3型：

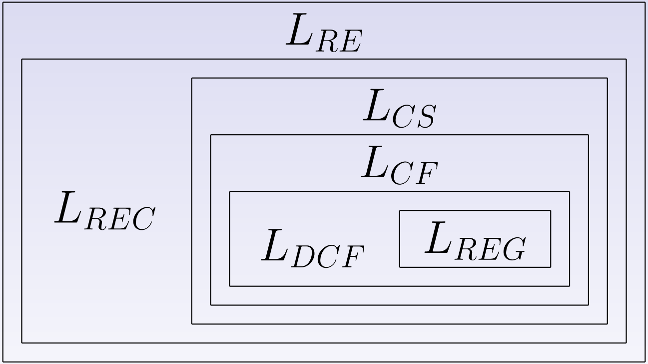
- 0型语言是由无限制文法生成的，即递归可枚举语言；
- 1型语言是由上下文相关语言构成；
- 2型语言是由上下文无关语言构成；
- 3型语言是由正则语言构成。

每个i型语言族是$i-1$型语言族的真子集。

图示(1)



图示(2)



例子

考察下面两个上下文无关语言：

$$L = \{w : n_a(w) = n_b(w)\}$$

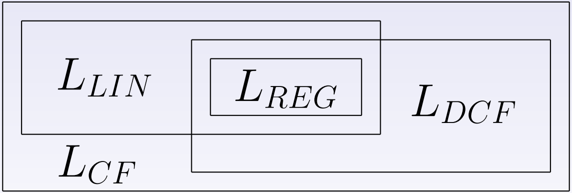
是确定型的，但不是线性的；

$$L = \{a^n b^n\} \cup \{a^n b^{2n}\}$$

是线性的，但不是确定型的。

关系图示

下图表明了正则语言、线性语言、确定型上下文无关语言和非确定性上下文无关语言之间的关系。



总结

我们现在已经知道了几种语言族及与它们相关的自动机之间的关系。我们建立了语言层次结构，并且把自动机按照它们的接受能力进行了分类。图灵机具有比线性有界自动机更强的能力，而这两者都具有比下推自动机更强的能力。在这个层次结构中，处于底层的是有穷接受器，它是我们研究的出发点。

谢谢！

主 讲 人： 姚 刚

电子邮箱： yaogang@iie.ac.cn