上下文有关语言

上下文有关语言

- 主要内容
  - **TM**与**PSG**的等价性。
  - 线性界限自动机(LBA)。
  - LBA作为CSL的识别器。
- 本章的内容是介绍性。

文法的乔姆斯基体系

- 文法G=(V，T，P，S)
- G叫做0型文法(type 0 grammar)，也叫做短语结构文法(phrase structure grammar, PSG)。
- L(G)叫做0型语言。也可以叫做短语结构语言(PSL)、递归可枚举集(recursively enumerable ，r.e.)。

文法的乔姆斯基体系

- G是0型文法。
- 如果对于$\forall \alpha \to \beta \in P$，均有$|\beta| \geq |\alpha|$成立，则称G为**1型文法(type 1 grammar)**，**或上下文有关文法(context sensitive grammar,CSG)**。
- L(G)叫做**1型语言(type 1 language)**或者**上下文有关语言(context sensitive language ,CSL)**。

文法的乔姆斯基体系

- G是1型文法
- 如果对于$\forall \alpha \to \beta \in P$，均有$|\beta| \geq |\alpha|$，并且$\alpha \in V$成立，则称G为2型文法(type 2 grammar)，或上下文无关文法(context free grammar,CFG)。
- L(G)叫做2型语言(type 2 language)或者上下文无关语言(context free language ,CFL)。

文法的乔姆斯基体系

- G是2型文法
- 如果对于$\forall \alpha \to \beta \in P$，$\alpha \to \beta$均具有形式

$$A \to w$$

$$A \to wB$$

其中A，B∈V，w∈T^+。则称G为3型文法(type 3 grammar)，也可称为正则文法(regular grammar ,RG)或者正规文法。L(G)叫做3型语言(type 3 language)，也可称为正则语言或者正规语言(regular language ,RL)。

TM与PSG的等价性

- 例 构造产生语言{0^n|n为2的非负整数次幂}的文法。

  设计思想：

  - 在文法中设置变量C，充当TM中的读头的作用，它从左到右扫描0，并且在每次遇到一个0时，都用00替换之，这使得当它从最左端移到最右端时，就完成了当前串的加倍工作，为了使串中的0再次被加倍，变量D充当将这个“读头”从右端移回到最左端的作用。为了标记出端点，文法用A、B分别表示串的最左端和最右端。

TM与PSG的等价性

- G_1： S→0

  产生句子0。

- S→AC0B

  产生句型AC0B，A、B分别表示左右端点，C为向右的倍增“扫描器”。

- C0→00C

  C向右扫描，将每一个0变成00，以实现0个数的加倍。

TM与PSG的等价性

- CB→DB

  C到达句型的右端点，变成D，准备进行从右到左的扫描，以实现对句型中0的个数的再次加倍。

- CB→E

  C到达句型的右端点，变成E，表示加倍工作已完成，准备结束。

TM与PSG的等价性

- 0D→D0 D移回到左端点。
- AD→AC

当D到达左端点时，变成C，此时已经做好了进行下一次加倍的准备工作。

- 0E→E0 E向左移动，以寻找左端点A。
- AE→ ε

E找到A后，一同变成ε，从而得到一个句子。

TM与PSG的等价性

另一个相关的文法

G_2： $S \rightarrow AC0B$

$C0 \rightarrow 00C$

$CB \rightarrow DB$

$0D \rightarrow D00$

$CB \rightarrow E$

$AD \rightarrow AC$

$AC \rightarrow F$

$F0 \rightarrow 0F$

$0E \rightarrow E0$

$AE \rightarrow \varepsilon$

$FB \rightarrow \varepsilon$

TM与PSG的等价性

定理 对于任一**PSG** G=(V，T，P，S)，存在TM M，使得L(M)=L(G)。

证明要点：

基本思想如下。

- M具有两条带，其中一条带用来存放输入字符串w，第二条带用来试着产生w。即，第二条带上存放的将是一个句型。我们希望该句型能够派生出w。在开始启动时，这个句型就是S。

TM与PSG的等价性

- 设第二条带上的句型为γ，M按照某种策略在γ中选择一个子串α，使α为G的某个产生式的左部，再按照非确定的方式选择α产生式的某一个候选式β，用β替换α。在需要时，利用适当的移动技术，让TM可以实现将句型中的α替换成β的工作。
- 当第二条带上的内容为一个终极符号行时，就把它与第一条带上的w进行比较，如果相等，就接受；如果不相等，就去寻找可以产生w的派生。

TM与PSG的等价性

- 由于G为PSG，所以，在整个“试派生”过程中，我们是无法总能根据当前句型的长度来决定该派生是否需要继续进行下去。这样一来，对于一个给定的输入字符串，如果它不是L(G)的句子，我们构造的TM可能会陷入永不停机的工作过程中。这从另一方面说明，短语结构语言不一定是递归语言。

TM与PSG的等价性

定理 对于任一TM M，存在PSG G=(V，T，P，S)，使得L(G)=L(M)。

证明要点：

①设TM M=(Q, Σ, Γ, δ, q_0, B, F)，L=(M)。

②让G可以产生Σ*中的任意一个字符串的变形，然后让G模拟M处理这个字符串。如果M接受它，则G就将此字符串的变形还原成该字符串。

③变形是让每个字符对应一个二元组。$\forall[a_1, a_1][a_2, a_2]\ldots[a_n, a_n] \in (\Sigma\times\Sigma)^*$，被看成$a_1a_2\ldots a_n$的两个副本。

TM与PSG的等价性

④G在一个副本上模拟M的识别动作，如果M进入终止状态，则G将句型中除另一个副本外的所有字符消去。

$G=(((\Sigma\cup\{\varepsilon\})\times\Gamma\cup\{A_1, A_2, A_3\}\cup Q, \Sigma, P, A_1)$

(1) $A_1\rightarrow q_0A_2$ 准备模拟M从q_0启动；

(2) $A_2\rightarrow[a, a]A_2$

$\forall a\in\Sigma$，A_2首先生成任意的形如$[a_1, a_1][a_2, a_2]\ldots[a_n, a_n]$的串；

TM与PSG的等价性

- **(3)　$A_2 \rightarrow A_3$**

　　在预生成双副本子串$[a_1, a_1][a_2, a_2]\ldots\ldots[a_n, a_n]$后，准备用$A_3$在该子串之后生成一系列的相当于空白符的子串，为G能够顺利地模拟M在处理相应的输入字符串的过程中，需要将读头移向输入串右侧的初始为B的地方做准备；

TM与PSG的等价性

- **(4)　$A_3 \to [\varepsilon, B] A_3$**

　　由于**M**在处理一个字符时，不知道将需要用到输入串右侧的多少个初始为**B**的带方格，所以，我们让A_3生成一系列的相当于空白符的子串$[\varepsilon, B][\varepsilon, B]\ldots\ldots[\varepsilon, B]$。在派生过程中，其个数依据实际需要而定；

TM与PSG的等价性

- **(5)** $A_3 \rightarrow \varepsilon$
- **(6)** $\forall a \in \Sigma \cup \{\varepsilon\}$，$\forall q, p \in Q$， $\forall X, Y \in \Gamma$， 如果 $\delta(q, X)=(p, Y, R)$， 则
  - $q[a, X] \rightarrow [a, Y]p$
  - **G**模拟**M**的一次右移;

TM与PSG的等价性

- **(7)** 对于∀**a, b**∈∑∪{ε}，∀**q, p**∈**Q**，∀**X, Y, Z**∈Γ，如果δ**(q, X)=(p, Y, L)**，则
  - **[b, Z]q[a, X] → p[b, Z] [a, Y]**
  - **G**模拟**M**的一次左移；
- **(8)** 对于∀**a**∈∑∪{ε}，∀**q**∈**F**则
  - **[a, X]q→qaq** **G**先将句型中的**[**、**]**、**X**等消除；
  - **q[a, X] →qaq**
  - **q→** ε 最后再消除句型中的状态**q**

LBA及其与CSG的等价性

- **线性有界自动机(linear bounded automaton，LBA)**
  - 非确定的TM。
  - 输入字母表包含两个特殊的符号¢和$，其中，¢作为输入符号串的左端标志，$作为输入符号串的右端标志。
  - LBA的读头只能在¢和$之间移动，它不能在端点符号¢和$上面打印另外一个符号。

LBA及其与CSG的等价性

- **LBA**可以被看成一个八元组，

$$M=(Q, \Sigma, \Gamma, \delta, q_0, ¢, \$, F)$$

其中，**Q**、Σ、Γ、δ、q_0、**F**与TM中的定义相同，$¢\in\Sigma$，$\$\in\Sigma$，**M**接受的语言

$$L(M)=\{w \mid w\in(\Sigma-\{¢, \$\})^* \ \& \ \exists\, q\in F \text{使得} \ q_0¢w\$ \vdash^* ¢\alpha q \beta \$\}。$$

LBA及其与CSG的等价性

定理 如果L的CSL， ε $\notin$L，则存在LBA M，使得L=L(M)。

证明要点：

① 设 CSG G=(V，T，P，S)，使得L=L(G)。。

②用一个两道TM模拟G。一道存放字符串¢w$，另一道用来生成w的推导。

LBA及其与CSG的等价性

③ CSG保证只用考察长度不超过|w|句型。

④ 将句型的长度限制在|w|以内，所以，M的运行不会超出符号¢和$规定的范围。

⑤ 对于任意输入，LBA均会停机，这表明CSL是递归语言。

LBA及其与CSG的等价性

定理 对于任意L，$\varepsilon \notin$ L，存在LBA M，使得L=L(M)，则L是CSL。

证明：主要是根据给定的LBA M构造出CSG G。这里的双副本串是形如$[a_1, q_0 \not\subset a_1][a_2, a_2]\ldots[a_n, a_n\$]$的符号行，当长度为1时，此符号行为$[a, q_0 \not\subset a\$]$。

LBA及其与CSG的等价性

（1）对于$\forall$a$\in\sum$-{$\not\subset$, \$}，

$A_1 \rightarrow [a, q_0 \not\subset a]A_2$

准备模拟M从q_0启动，生成形如$[a_1, q_0 \not\subset a_1][a_2, a_2]\ldots[a_n, a_n\$]$的双副本串(句型)中的$[a_1, q_0 \not\subset a_1]$，并将生成子串$[a_2, a_2]\ldots[a_n, a_n\$]$的任务交给$A_2$；

$A_1 \rightarrow [a, q_0 \not\subset a\$]$

生成双副本串$[a, q_0 \not\subset a\$]$；

LBA及其与CSG的等价性

（2）对于$\forall a \in \Sigma$-{$\not\subset$, \$}，

$A_2 \rightarrow [a, a]A_2$

A_2首先生成任意的形如$[a_1, q_0 \not\subset a_1][a_2, a_2]\ldots[a_n, a_n\$]$的双副本串中的子串$[a_2, a_2]\ldots[a_{n-1}, a_{n-1}]$;

（3）对于$\forall a \in \Sigma$-{$\not\subset$, \$}，

$A_2 \rightarrow [a, a\$]$

A_2 最 后 生 成 任 意 的 形 如 $[a_1, q_0 \not\subset a_1][a_2, a_2]\ldots[a_n, a_n\$]$的双副本中的子串$[a_n, a_n\$]$;

LBA及其与CSG的等价性

(4) 对于$\forall a \in \Sigma$-{\$}， $\forall q, p \in Q$， $\forall X, Y, Z \in \Gamma$， $X \neq \$$， 如果$\delta(q, X)=(p, Y, R)$， 则

- [a, q X][b, Z] →[a, Y] [b, p Z]
- G模拟M的一次右移；

(5) 对于$\forall a, b \in \Sigma$-{¢}， $\forall q, p \in Q$， $\forall X, Y, Z \in \Gamma$， 如果$\delta(q, X)=(p, Y, L)$， 则

- [b, Z] [a, q X] → [b, p Z] [a, Y]
- G模拟M的一次左移；

LBA及其与CSG的等价性

(6) 对于∀a∈Σ，∀q∈F，∀X, Y∈Γ-{B}， [a, XqY] →a

- 由于q为终止状态，所以可以消除它

(7) 对于∀a∈Σ-{¢, $}，∀X∈Γ-{B}，

- [a, X]b →ab
- a [b, X] →ab

小结

本章讨论TM与PSG的等价性，介绍了识别CSL的装置——LBA。

（1）对于任一PSG G=(V，T，P，S)，存在TM M，使得L(M)=L(G)；

（2）对于任一TM M，存在PSG G=(V，T，P，S)，使得L(G) =L(M)；

（3）LBA是一种非确定的TM，它的输入串被用符号¢和$括起来，而且读头只能在¢和$之间移动；

小结

(4)如果L是CSL， $\varepsilon \notin L$，则存在LBA M，使得L=L(M)；

(5)对于任意L， $\varepsilon \notin L$，存在LBA M，使得L=L(M)，则L是CSL。