第二章 文法

- 对任何语言 $\mathbf{L}$，有一个字母表 $\boldsymbol{\Sigma}$，使得 $\mathbf{L} \subseteq \boldsymbol{\Sigma}^*$。
- **L**的具体组成结构是什么样的?

- 一个给定的字符串是一个给定语言的句子吗?如果不是，它在结构的什么地方出了错？进一步地，这个错误是什么样的错？如何更正？

- 这些问题对有穷语言来说，比较容易解决;
- 这些问题对无穷语言来说，不太容易解决。

- 语言的有穷描述。

描述语言的方式：

- 枚举法：如果一个语言仅包含有限个句子，则可以采用枚举法来描述此语言，将语言中每条句子都列举出来即可。
- 文法产生法：为每种语言定义一组文法规则，从而产生该语言中的每条句子。
- 自动机识别法：在这种方法中，每种语言对应一种自动机，由自动机来判定一个符号串是否在该语言中。

章节目录

2.1 启示

2.2 形式定义

2.3 文法的构造

2.4 文法的乔姆斯基体系

2.5 空语句

2.6 本章小结

2.1 启示

- 文法的概念最早是由语言学家们在研究自然语言理解中完成形式化。

举例：归纳如下句子的描述

(1) 哈尔滨是美丽的城市。

(2) 北京是祖国的首都。

(3) 集合是数学的基础。

(4) 形式语言是很抽象的。

(5) 教育走在社会发展的前面。

(6) 中国进入WTO。

2.1 启示

举例

归纳如下句子的描述：

(1) 哈尔滨是美丽的城市。

(2) 北京是祖国的首都。

(3) 集合是数学的基础。

(4) 形式语言是很抽象的。

(5) 教育走在社会发展的前面。

(6) 中国进入**WTO**。

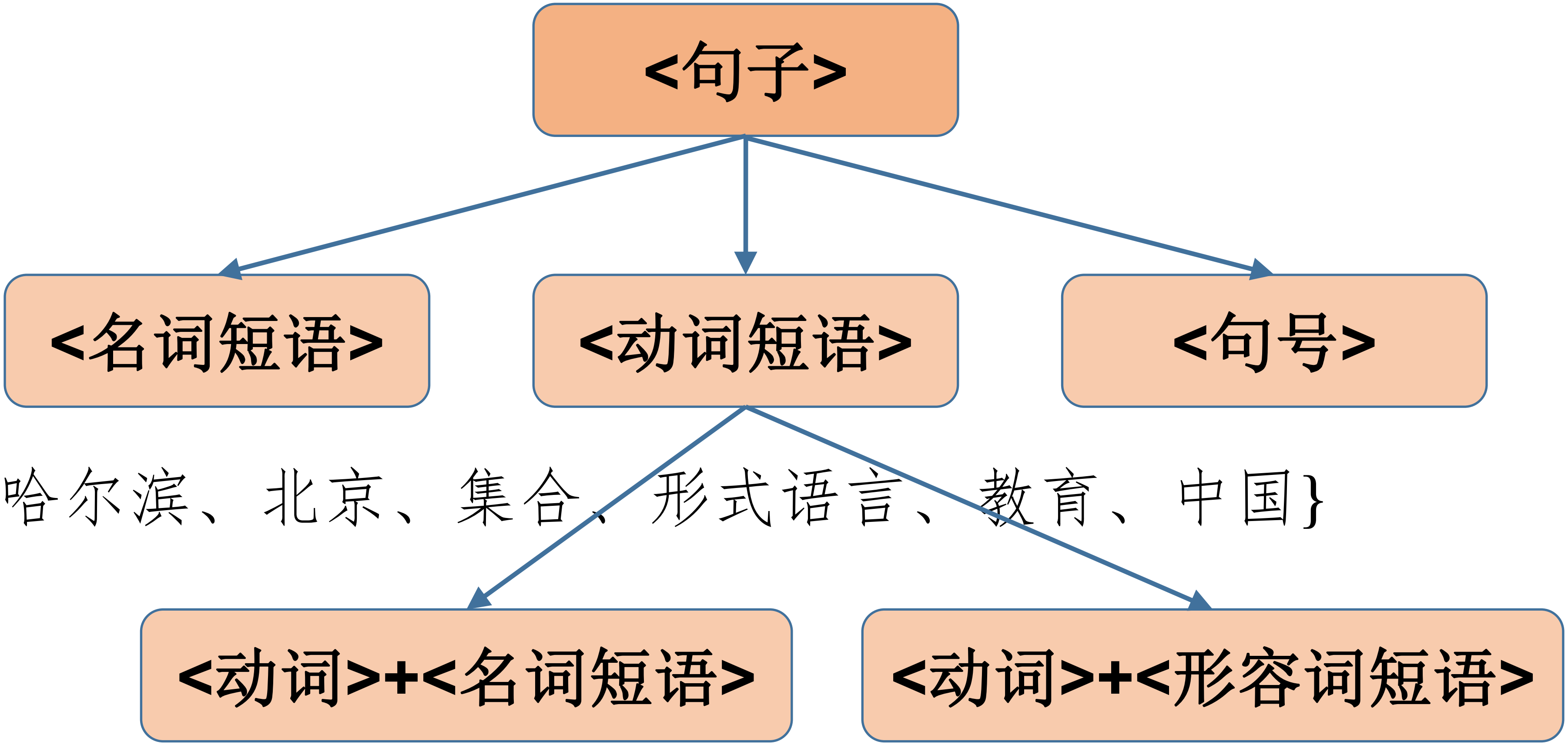
6个句子的主体结构：<名词短语><动词短语><句号>

2.1 启示

以上6个句子的主体结构：

1. <名词短语><动词短语><句号>
2. <名词短语>={哈尔滨，北京，集合，形式语言，教育，中国}
3. <动词短语>={是美丽的城市，是祖国的首都，是数学的基础，是很抽象的，走在社会发展的前面，进入WTO}
4. <句号>={。}

2.1 启示



{是美丽的城市、是祖国的首都、是数学的基础、走在社会发展的前面、进入WTO}

{是很抽象的}

2.1 启示

表示成$\alpha \rightarrow \beta$ 的形式

- <句子>→<名词短语><动词短语><句号>
- <动词短语>→<动词><名词短语>
- <动词短语>→<动词><形容词短语>
- <动词>→是
- <动词>→走在
- <动词>→进入
- <形容词短语>→很抽象的
- <名词短语>→哈尔滨
- <名词短语>→北京
- <名词短语>→集合

2.1 启示

表示成$\alpha \rightarrow \beta$的形式

- <名词短语>→形式语言
- <名词短语>→教育
- <名词短语>→中国
- <名词短语>→美丽的城市
- <名词短语>→祖国的首都
- <名词短语>→数学的基础
- <名词短语>→社会发展的前面
- <名词短语>→**WTO**
- <句号>→。

- 用尖括号<>把代表一个集合的词扩起来
- 括起来的部分称为语法成分
- 用符号→表示是、可以是
- 未用尖括号括起来部分表示语言的基本符号

2.1 启示

表示语言的四类对象：

① 一系列的“符号”，如〈动词短语〉

表示语言结构中某个位置上可以出现的一些内容。每个“符号”对应的是一个集合，在该语言的一个具体句子中，句子的这个位置上能且仅能出现相应集合中的某个元素。

② “符号”，如北京

它们是所定义语言的合法句子中出现的“符号”。仅仅表示自身，称为终极符号。

2.1 启示

表示语言的四类对象：

③ “规则”： $\alpha \rightarrow \beta$的形式

在产生语言的句子中被使用，称这些“规则”为产生式。

④ ⟨句子⟩

所有的“规则”，都是为了说明⟨句子⟩的结构而存在，相当于说，定义的就是⟨句子⟩。

章节目录

2.1 启示

2.2 形式定义

2.3 文法的构造

2.4 文法的乔姆斯基体系

2.5 空语句

2.6 本章小结

2.2 形式定义

文法

定义2-1 文法G是一个四元组G=(V，T，P，S)

- V：变量（Variable）/非终极符号的非空有穷集
- T：终极符号（Terminal）的非空有穷集
- P：产生式（Production）的非空有穷集
- S：文法G的开始符号（Start symbol），$S \in V$

文法

定义2-1 文法G是一个四元组G=(V，T，P，S)

- **V**：变量（**Variable**）/非终极符号的非空有穷集
- **T**：终极符（**Terminal**）的非空有穷集
- **P**：产生式（**Production**）的非空有穷集
- **S**：文法G的开始符号（**Start symbol**），$S \in V$

■∀A∈V，A叫做一个语法变量，简称为变量，也可叫做非终极符号(nonterminal)。

■A表示一个语法范畴，记为L(A)。

文法

定义2-1 文法G是一个四元组G=(V，T，P，S)

- V：变量（Variable）/非终极符号的非空有穷集
- T：终极符号（Terminal）的非空有穷集
- P：产生式（Production）的非空有穷集
- S：文法G的开始符号（Start symbol），$S \in V$

■∀a∈T，a成为终极符。

■T中的符号是语言的句子中出现的字符

■V∩T=Φ。

2.2 形式定义

文法

定义2-1 文法G是一个四元组G=(V，T，P，S)

- V：变量（Variable）/非终极符号的非空有穷集
- T：终极符号（Terminal）的非空有穷集
- P：产生式（Production）的非空有穷集
- S：文法G的开始符号（Start symbol），$S \in V$

■P中的元素均具有形式$\alpha \rightarrow \beta$，读作：α定义为β。

■其中$\alpha \in (V \cup T)^{+}$，$\beta \in (V \cup T)^{*}$，$\alpha$中至少有V中的一个元素出现。

■α称为产生式的左部，β称为产生式的右部。

文法

定义2-1 文法G是一个四元组G=(V，T，P，S)

- V：变量（Variable）/非终极符号的非空有穷集
- T：终极符号（Terminal）的非空有穷集
- P：产生式（Production）的非空有穷集
- S：文法G的开始符号（Start symbol），$S \in V$

■S是文法G的开始符号（入口）

■开始符号可以推导出语言的所有串

例2-1 以下四元组都是文法。

① $(\{A\}, \{0, 1\}, \{A \to 01, A \to 0A1, A \to 1A0\}, A)$。

② $(\{A\}, \{0, 1\}, \{A \to 0, A \to 0A\}, A)$

③ $(\{A, B\}, \{0, 1\}, \{A \to 01, A \to 0A1, A \to 1A0, B \to AB, B \to 0\}, A)$

④ $(\{A, B\}, \{0, 1\}, \{A \to 0, A \to 1, A \to 0A, A \to 1A\}, A)$

⑤ $(\{S, A, B, C, D\}, \{a, b, c, d, \#\}, \{S \to ABCD, S \to abc\#, A \to aaA, AB \to aabbB, BC \to bbccC, cC \to cccC, CD \to ccd\#, CD \to d\#, CD \to \#d\}, S)$

⑥ $(\{S\}, \{0, 1\}, \{S \to 00S, S \to 11S, S \to 00, S \to 11\}, S)$

约定

1. 对一组有相同左部的产生式

$$\alpha \rightarrow \beta_1, \alpha \rightarrow \beta_2, \cdots, \alpha \rightarrow \beta_n$$

可以简单地记为：$\alpha \rightarrow \beta_1 | \beta_2 | \cdots | \beta_n$，读作：α定义为$\beta_1$，或者$\beta_2$，…，或者$\beta_n$。$\beta_1, \beta_2, \ldots, \beta_n$称为候选式。

<名词短语> →集合|中国|形式语言|数学的基础|WTO

2.2 形式定义

约定

2. 一般地，按照如下方式使用符号：

- 英文字母表较为前面的大写字母，如A, B, C, …表示语法变量；
- 英文字母表较为前面的小写字母，如a, b, c, …表示终极符号；
- 英文字母表较为后面的大写字母，如X, Y, Z, …表示该符号是语法变量或者终极符号；
- 英文字母表较为后面的小写字母，如x, y, z, …表示由终极符号组成的行；
- 希腊字母α, β, γ, …表示表示由语法变量和终极符号组成的行

2.2 形式定义

例2-2 判断以下四元组是否满足文法的要求。

$(\{A, B, C, E\}, \{a, b, c\}, \{S \to ABC|abc, D \to e|a, FB \to c, A \to A, E \to abc|\varepsilon\}, S)$

主要问题在于：

- **$S, D, F \notin \{A, B, C, E\}$**
- **$e \notin \{a, b, c\}$**

2.2 形式定义

例2-2 以下四元组是否满足文法的要求。

({A, B, C, E}, {a, b, c}, {S → ABC|abc, D → e|a, FB→ *c*, A→A, E→abc|*ε*}, S)

可以进行如下4种修改：

（1）扩充V和T：将产生式中出现的，但在变量集和终极符号集中没有出现的符号放入语法变量集或者终极符号集中。

({A, B, C, E, S, D, F}, {a, b, c, e}, {S→ABC|abc, D→e|a, FB*c*, A→A, E→abc|*ε*}, S)

2.2 形式定义

例2-2 以下四元组是否满足文法的要求。

({A, B, C, E}, {a, b, c}, {S → ABC|abc, D → e|a, FB→ *c*, A→A, E→abc|ε}, S)

可以进行如下4种修改：

（2）仅将S扩充进V：保持S是开始符号，将S放入变量集，去掉含有不合法符号D, F, e的产生式

({A, B, C, E, S},{a, b, c},{S→ABC|abc, A→A, E→abc|ε}, S)

例2-2 以下四元组是否满足文法的要求。

({A, B, C, E}, {a, b, c}, {S → ABC|abc, D → e|a, FB→ *c*, A→A, E→abc|*ε*}, S)

（3）不扩充V和T：指定A为新开始符，去掉含有不合法符号的产生式

({A, B, C, E},{a, b, c},{A→A, E→abc|*ε*}, A)

（4）不扩充V和T：指定E为新开始符，去掉含有不合法符号的产生式

({A, B, C, E},{a, b, c},{A→A, E→abc|*ε*}, E)

推导

定义2-2 设G = (V, T, P, S)是一个文法，如果 $\alpha \to \beta \in P, \gamma, \delta \in (V \cup T)^*$，则称$\gamma\alpha\delta$在G中直接推导出$\gamma\beta\delta$。

$$\gamma\alpha\delta \Rightarrow_G \gamma\beta\delta$$

读作：$\gamma\alpha\delta$在文法G中直接推导出$\gamma\beta\delta$。

- “直接推导”可以简称为推导(derivation)，也称为派生。

归约

$$\gamma\alpha\delta \Rightarrow_G \gamma\beta\delta$$

与推导相对应，也可以称$\gamma\beta\delta$在文法G中直接归约成$\gamma\alpha\delta$，“直接归约”可以简称为归约 (reduction)。

$\Rightarrow$：正向推导，反向归约

2.2 形式定义

- 显然，$\Rightarrow_G$是$(V \cup T)^*$上的二元关系
- 用 $\Rightarrow_G^+$代表 $(\Rightarrow_G)^+$，用 $\Rightarrow_G^*$代表$(\Rightarrow_G)^*$，用 $\Rightarrow_G^n$，代表$(\Rightarrow_G)^n$。

$\alpha \Rightarrow_G^n \beta$：表示α在G中**经过n步推导出β**, β在G中经过n步归约成α。

即存在$\alpha_1, \alpha_2, \cdots, \alpha_{n-1} \in (V \cup T)^*$，使

$\alpha \Rightarrow_G \alpha_1, \alpha_1 \Rightarrow_G \alpha_2, \cdots, \alpha_{n-1} \Rightarrow_G \beta$。

2.2 形式定义

① 当n=0时，有$\alpha = \beta$，即$\alpha \Rightarrow_G^0 \alpha$。

② 当n≥1时，写为$\alpha \Rightarrow_G^+ \beta$：表示α在G中**经过至少1步推导出β**；β在G中经过至少1步归约成α。

③ 当n≥ 0时，写为$\alpha \Rightarrow_G^* \beta$：表示α在G中**经过若干步推导出β**；β在G中经过若干步归约成α。

分别用$\Rightarrow, \Rightarrow^n, \Rightarrow^+, \Rightarrow^*$代替$\Rightarrow_G, \Rightarrow_G^n, \Rightarrow_G^+, \Rightarrow_G^*$

2.2 形式定义

例2-3 设G=({A}, {a}, {A → a|aA}, A)

$A \Rightarrow a\underline{A}$	使用产生式A → aA	1次推导
$\Rightarrow aa\underline{A}$	使用产生式A → aA	2次推导
$\Rightarrow aaa\underline{A}$	使用产生式A → aA	3次推导
$\Rightarrow aaaa\underline{A}$	使用产生式A → aA	4次推导
…		
$\Rightarrow a\ldots a\underline{A}$	使用产生式A → aA	n-1次推导
$\Rightarrow a\ldots aaA$	使用产生式A → aA	n次推导
$\Rightarrow a\ldots aa$	使用产生式A → a	n次推导

可见，从A出发，通过n步可以推出a…aaA和a…aa两个不同的串，其中a…aa有n个a，a…aaA有n个a和一个A。

续例2-3 设G=({A},{a},{A → a|aA},A)

写出从字符串AAaaAAA到字符串aaaaaaaaaa的推导过程，用下划线标出每次推导被替换的符号。

$$
\begin{aligned}
AAaaA\underline{A}A &\Rightarrow A\underline{A}aaAaAA && \text{使用产生式}A\to aA\\
&\Rightarrow AaAaaAa\underline{A}A && \text{使用产生式}A\to aA\\
&\Rightarrow \underline{A}aAaaAaaA && \text{使用产生式}A\to a\\
&\Rightarrow aaAaaAaa\underline{A} && \text{使用产生式}A\to a\\
&\Rightarrow aa\underline{A}aaAaaa && \text{使用产生式}A\to a\\
&\Rightarrow aaa\underline{A}aaAaaa && \text{使用产生式}A\to aA\\
&\Rightarrow aaaaaa\underline{A}aaa && \text{使用产生式}A\to a\\
&\Rightarrow aaaaaaaaaa && \text{使用产生式}A\to a
\end{aligned}
$$

2.2 形式定义

- 为限制派生/推导的随意性，要求只替换符号串中最左边变元的派生过程，称为最左派生。
- 类似地，只替换最右的，称为最右派生。

- 任何派生都有等价的最左派生和最右派生。

2.2 形式定义

续例2-3 设G=({A},{a},{A → a|aA},A)

写出从字符串AAaaAAA到字符串aaaaaaaaaa的最左派生过程，用下划线标出每次推导被替换的符号。

$\underline{A}AaaAAA \Rightarrow a\underline{A}AaaAAA$ 使用产生式A→aA

$\Rightarrow aa\underline{A}AaaAAA$ 使用产生式A→aA

$\Rightarrow aaa\underline{A}AaaAAA$ 使用产生式A→aA

$\Rightarrow aaaa\underline{A}aaAAA$ 使用产生式A→a

$\Rightarrow aaaaaaa\underline{A}AA$ 使用产生式A→a

$\Rightarrow aaaaaaaa\underline{A}A$ 使用产生式A→a

$\Rightarrow aaaaaaaaa\underline{A}$ 使用产生式A→a

$\Rightarrow aaaaaaaaaa$ 使用产生式A→a

2.2 形式定义

例2-4 设G=({S, A, B},{0,1},{S→A|AB, A →0|0A, B→1|11}, S)，则有如下一些推导：

对于$n \geq 1$，

- $A \Rightarrow^n 0^n$ 首先连续n-1次使用产生式$A \rightarrow 0A$，最后一次使用产生式$A \rightarrow 0$；
- $A \Rightarrow^n 0^n A$ 连续n次使用产生式$A \rightarrow 0A$
- $B \Rightarrow 1$ 使用产生式$B \rightarrow 1$
- $B \Rightarrow 11$ 使用产生式$B \rightarrow 11$

续例2-4 设G=({S, A, B},{0,1},{S→A|AB, A →0|0A, B → 1|11}, S)，则语法范畴A和B所代表的集合分别为：

Callback:

- ∀A∈V，A叫做一个语法变量，简称为变量，也可叫做非终极符号(nonterminal)。
- A表示一个语法范畴，记为L(A)。

- 语法范畴A代表的集合

  L(A)={0, 00, 000, ….}=$\{0^n|n \geq 1\}$;

- 语法范畴B代表的集合

  L(B)={1, 11};

续例2-4 设G=({S, A, B},{0,1},{S→A|AB, A →0|0A, B→1|11}, S)，则有如下一些推导：

- 语法范畴S代表的集合

$$L(S)=L(A)\cup L(A)L(B)$$

$$=\{0, 00, 000, \ldots\}\cup\{0, 00, 000,\ldots\}\{1, 11\}$$

$$=\{0, 00, 000, \ldots\}\cup\{01, 001, 0001,\ldots\}\cup\{011, 0011, 00011, \ldots\}$$

2.2 形式定义

例2-5 设G=({A},{0,1},{A → 01|0A1}, A)，则有如下一些推导：

- $A \Rightarrow^{n} 0^{n}A\,1^{n}$ $\quad n \geq 0$
- $0^{n}A1^{n} \Rightarrow^{i} 0^{n+i}A\,1^{n+i}$ $\quad n \geq 0, i \geq 0$
- $0^{n}A1^{n} \Rightarrow^{*} 0^{m}A\,1^{m}$ $\quad n \geq 0, m \geq n$
- $0^{n}A1^{n} \Rightarrow^{+} 0^{m}A\,1^{m}$ $\quad n \geq 0, m \geq n+1$

- 语法范畴A代表的集合

  L(A)={01, 0011, 000111, ….}=$\{0^{n}1^{n}|n \geq 1\}$

几点结论

1. 对任意的$\boldsymbol{x} \in \boldsymbol{T}^{+}$，要使一个语法范畴D代表的集合为$\{\boldsymbol{x}^{\boldsymbol{n}} | \boldsymbol{n} \geq \mathbf{0}\}$，可用产生式组{D→ $\boldsymbol{\varepsilon}$|$\boldsymbol{x}$D}来实现。

2. 对任意的$\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{T}^{+}$，要使一个语法范畴D代表的集合为$\{\boldsymbol{x}^{\boldsymbol{n}}\boldsymbol{y}^{\boldsymbol{n}} | \boldsymbol{n} \geq \mathbf{1}\}$，可用产生式组{D→ $\boldsymbol{xy}$|$\boldsymbol{x}$D$\boldsymbol{y}$}来实现。

3. 对任意的$\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{T}^{+}$，要使一个语法范畴D代表的集合为$\{\boldsymbol{x}^{\boldsymbol{n}}\boldsymbol{y}^{\boldsymbol{n}} | \boldsymbol{n} \geq \mathbf{0}\}$，可用产生式组{D→ $\boldsymbol{\varepsilon}$|$\boldsymbol{xDy}$}来实现。

语言、句子与句型

定义2-3 设文法G=(V, T, P, S)，则L(G)称为文法G产生的语言，$\forall \omega \in L(G)$，ω称为G产生的一个句子。

$$L(G)=\{\omega | \omega \in T^* \text{且} S \Rightarrow^* \omega\}$$

定义2-4 设文法G=(V, T, P, S)，对于$\forall \alpha \in (V \cup T)^*$，如果$S \Rightarrow^* \alpha$，则称$\alpha$是G产生的一个句型。

2.2 形式定义

语言、句子与句型

- 句子ω是从S开始，在G中可以推导出来的终极符号行，它不含语法变量。
- 句型α是从S开始，在G中可以推导出来的符号行，它可能含有语法变量。

2.2 形式定义

续例2-5 给定文法G=({A},{0,1},{A →01|0A1}, A),

- $\mathbf{0}^{\boldsymbol{n}}\mathrm{A}\mathbf{1}^{\boldsymbol{n}}(\boldsymbol{n} \geq \mathbf{0})$是句型，不是句子
- $\mathbf{0}^{\boldsymbol{n}}\mathbf{1}^{\boldsymbol{n}}\ (\boldsymbol{n} \geq \mathbf{0})$是句型，是句子;

2.2 形式定义

方法一

例2-6 构造产生标识符的文法。

（假定：标识符是以字母开头的字母数字串。）

标识符可以递归定义为：

1. 大写英文字母表中的任意一个字母是一个标识符，小写英文字母表中的任意一个字母是一个标识符。
2. 如果α是一个标识符，则在α后接一个大写英文字母，或者一个小写英文字母，或者一个阿拉伯数字后仍然是一个标识符。
3. 只有满足1和2的才是标识符。

2.2 形式定义

方法一

例2-6 构造产生标识符的文法。

根据递归定义，可以构造产生式：

1. 大写英文字母表中的任意一个字母是一个标识符，小写英文字母表中的任意一个字母是一个标识符。

   <标识符> → <大写字母> | <小写字母>

2. 如果α是一个标识符，则在α后接一个大写英文字母，或者一个小写英文字母，或者一个阿拉伯数字后仍然是一个标识符。

<标识符> → <标识符><大写字母> | <标识符> <小写字母> | <标识符> <阿拉伯数字>

方法一

例2-6 构造产生标识符的文法。

根据递归定义，可以构造产生式：

3. <大写字母> →**A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z**

4. <小写字母> → **a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z**

5. <阿拉伯数字> → **0|1|2|3|4|5|6|7|8|9**

方法一

例2-6 构造产生标识符的文法。

G=({<标识符>，<大写字母>，<小写字母>， <阿拉伯数字>}，{0, 1, …,9, A, B, …Z, a, b, …, z}, P, <标识符>）

P={

<标识符> → <大写字母> **|** <小写字母>;

<标识符> → <标识符><大写字母> **|** <标识符> <小写字母> **|** <标识符> <阿拉伯数字>;

<大写字母> → **A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T |U|V|W|X|Y|Z**

<小写字母> → **a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z**

<阿拉伯数字> → **0|1|2|3|4|5|6|7|8|9**

}

方法二

例2-6 构造产生标识符的文法。

（假定：标识符是以字母开头的字母数字串。）

G′=({<标识符>，<头>，<尾>}，{0, 1, …,9, A, B, …Z, a, b, …, z}, P′, <标识符>）

P′={

<标识符> → <头> <尾>;

<头> → **A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q |R|S|T|U|V|W|X|Y|Z**

<头> → **a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z**

方法二

例2-6 构造产生标识符的文法。

<尾> → ε|0 <尾> |1 <尾> |2 <尾> |3 <尾> |4 <尾> |5 <尾> |6 <尾> |7 <尾> |8 <尾> |9 <尾>

<尾> → A <尾> |B <尾> |C <尾> |D <尾> |E <尾> |F<尾> |G<尾> <尾> → H <尾> |I <尾> |J <尾>|K <尾> |L <尾> |M <尾> |N <尾> <尾> → O<尾> |P<尾> |Q <尾> |R <尾> |S<尾>|T <尾> |U <尾> |<尾> → V <尾> |W<尾> |X<尾> |Y<尾> |Z <尾>

<尾> → a <尾> |b <尾> |c <尾> |d <尾> |e <尾> |f<尾> |g<尾>
<尾> → h <尾> |i <尾> |j <尾>|k <尾> |l <尾> |m <尾> |n <尾>
<尾> → o<尾> |p<尾> |q <尾> |r <尾> |s<尾>|t <尾> |u <尾>
<尾> → v <尾> |w<尾> |x<尾> |y<尾> |z <尾>

2.2 形式定义

续例2-6 在上述两个文法中推导标识符id8n23。

<标识符> → <大写字母> | <小写字母>;
<标识符> → <标识符><大写字母> | <标识符> <小写字母> | <标识符> <阿拉伯数字>;

1. 在文法G中：

<标识符> ⟹ <标识符><阿拉伯数字> ⟹ <标识符>3
⟹ <标识符><阿拉伯数字>3 ⟹ <标识符>23
⟹ <标识符><小写字母>23 ⟹ <标识符>n23
⟹ <标识符><阿拉伯数字>n23⟹ <标识符>8n23
⟹ <标识符><小写字母>8n23 ⟹ <标识符>d8n23
⟹ <小写字母>d8n23 ⟹ id8n23

从后向前

2.2 形式定义

续例2-6 在上述两个文法中推导标识符id8n23。

2. 在文法G′中：

<标识符> ⇒ <头><尾>

⇒ i<尾>

⇒ id<尾>

⇒ id8<尾>

⇒ id8n<尾>

⇒ id8n2<尾>

⇒ id8n23<尾>

⇒ id8n23

<标识符> → <头> | <尾>;

<头> → A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

<头> → a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

<尾> → ε|0 <尾> |1 <尾> |2 <尾> |3 <尾> |4 <尾> |5 <尾> |6 <尾> |7 <尾> |8 <尾> |9 <尾>

从前向后

章节目录

2.1 启示

2.2 形式定义

2.3 文法的构造

2.4 文法的乔姆斯基体系

2.5 空语句

2.6 本章小结

2.3 文法的构造

例2-7 构造文法G，使$L(G)=\{0, 1, 00, 11\}$

方法一：（直接由开始符号S定义）

$G_1=(\{S\}, \{0,1\}, \{S\to 0|1|00|11\}, S)$

方法二：（用两个变量A和B分别表示0和1）

$G_2=(\{S,A,B\}, \{0,1\}, \{S\to A|B|AA|BB, A\to 0, B\to 1\}, S)$

方法三：（变量在终结符的右侧）

$G_3=(\{S,A,B\}, \{0,1\}, \{S\to 0|1|0A|1B, A\to 0, B\to 1\}, S)$

2.3 文法的构造

例2-7 构造文法G，使L(G)={0, 1, 00, 11}

方法四：（增加冗余变量和终结符）

G_4 =({S,A,B,C},{0,1,2}, {S→A|B|AA|BB, A→0, B→1}, S)

方法五：（在方法四基础上增加冗余产生式）

G_5=({S,A,B,C}, {0,1,2}, {S→A|B|AA|BB, A→0, B→1,

CACS→21,C→11,C→2},S)

等价（equivalence）:设有两个文法G_1和G_2，如果$L(G_1)=L(G_2)$，则称G1和G2等价。

$$L(G_1)=L(G_2)=L(G_3)=L(G_4)=L(G_5)$$

文法的简化

对文法G=(V, T, P, S)，约定

- 一个文法的所有产生式中包含的符号，就是文法中所有可能有用的终极符和非终极符。
- 所列的第一个产生式的左部就是该文法的开始符号。
- 对于一个文法，只列出该文法的所有产生式即可

2.3 文法的构造

续例2-7 将以上五个文法简化。

- G_1=({S}, {0,1}, {S→0|1|00|11}, S)
- G_1： S→0|1|00|11

- G_2 =({S,A,B}, {0,1}, {S→A|B|AA|BB, A→0, B→1}, S)
- G_2： S→A|B|AA|BB, A→0, B→1

- G_3 =({S,A,B}, {0,1}, {A→0, S→0|1|0A|1B, B→1}, S)
- G_3： S→0|1|0A|1B, A→0, B→1

续例2-7 将以上五个文法简化。

- G_4 =({S,A,B,C},{0,1,2}, {S→A|B|AA|BB, A→0, B→1}, S)
- G_4： S→A|B|AA|BB, A→0, B→1

Callback:	G_2： S→A|B|AA|BB, A→0, B→1

- G_5 =({S,A,B,C}, {0,1,2}, {S→A|B|AA|BB, A→0, B→1, CACS→21,C→11,C→2}, S)
- G_5： S→A|B|AA|BB,A→0,B→1,CACS→21, C→11, C→2

例2-8 一些基本的文法。

- $L(G_6) = \{0^n | n \geq 1\}$

  G_6： $S \to 0|0S$

- $L(G_7) = \{0^n | n \geq 0\}$

  G_7： $S \to \varepsilon|0S$

- $L(G_8) = \{0^{2n}1^{3n} | n \geq 0\}$

  G_8： $S \to \varepsilon|00S111$

Note: 最基本的文法要记住!

2.3 文法的构造

例2-9 构造文法G_9，使$L(G_9) = \{\omega | \omega \in \{a, b, \cdots, z\}^+\}$。

- **G_9： S→A|AS**

  A→a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

规定:

- 不能用**A→a|b|c|...|z**去表示

  A→a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

- 不能用**S→a^n**$(n \geq 1)$去表示 **S**可以任意产生多个a

一般地，对于字母表Σ，当要产生语言Σ^+时，只用对Σ中的每个符号a安排产生式S→ a | aS 即可。

Σ^* ?

2.3 文法的构造

例2-10 构造文法G_{10}，使

$$L(G_{10}) = \{\omega\omega^T | \omega \in \{0, 1, 2, 3\}^+\}。$$

$$\omega = a_1 a_2 \cdots a_n, \quad \omega^T = a_n \cdots a_2 a_1$$

一个简单的想法:

- S → HE
- H → 0|1|2|3|0H|1H|2H|3H
- E → 0|1|2|3|E0|E1|E2|E3

2.3 文法的构造

例2-10 构造文法G_{10}，使

$$L(G_{10}) = \{\omega\omega^T | \omega \in \{0,1,2,3\}^+\}。$$

$$\omega = a_1a_2 \cdots a_n, \quad \omega^T = a_n \cdots a_2 a_1$$

观察句子的特点:

- $\omega\omega^T = a_1a_2 \cdots a_na_n \cdots a_2 a_1$

递归定义:

① 对$\forall a \in \{0,1,2,3\}$， $aa \in L(G_{10})$;

② 若$x \in L(G_{10})$，则对$\forall a \in \{0,1,2,3\}$，$axa \in L(G_{10})$;

③ $L(G_{10})$中不含不满足①②的任何其他串。

例2-10 构造文法G_{10}，使

$$L(G_{10}) = \{\omega\omega^T | \omega \in \{0,1,2,3\}^+\}。$$

根据递归定义中的前两个条件，有：

① 对$\forall a \in \{0,1,2,3\}$， $aa \in L(G_{10})$。

S→00 | 11 | 22 | 33

② 若$x \in L(G_{10})$，则对$\forall a \in \{0,1,2,3\}$，$axa \in L(G_{10})$。

S→0S0 | 1S1 | 2S2 | 3S3

G_{10}： S→00 | 11 | 22 | 33 | 0S0 | 1S1 | 2S2 | 3S3

G_{10}'： S→0 | 1 | 2 | 3 | 0S0 | 1S1 | 2S2 | 3S3产生的语言是？

2.3 文法的构造

例2-11 构造文法G_{11}，使

$$L(G_{11}) = \{\omega|\omega\text{是我们习惯的十进制有理数}\}。$$

1. 有理数分正负，而且正有理数前面的正号可以省略:S→R|+R|-R
2. 无符号有理数R又可以划分为无符号整数、无符号带小数（整数部分不为0）和无符号纯小数（整数部分为0）：R→N|B|P
3. 无符号带小数（整数部分不为0）B由用小数点隔开的整数和小数部分组成，小数部分用D表示：B→N.D
4. 无符号纯小数P是整数“0”后跟小数点“.”再跟小数部分： P→0.D

(下面解决N和D的定义/表示问题)

例2-11 构造文法G_{11}，使

$$L(G_{11}) = \{\omega|\omega\text{是我们习惯的十进制有理数}\}。$$

5. 整数部分N是由数字组成的符号行。无符号非0整数不希望是以0开头的；当为0时，又不希望用多于1个的一串0表示。M用来生成字母表{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}上的所有行。

   N→0|AM, A→1|2|3|4|5|6|7|8|9,

   M→ε|0M|1M|2M|3M|4M|5M|6M|7M|8M|9M

6. 小数部分D与N正好相反，除0外，不希望以0结束；当为0时，也不希望用多于1个的一串0表示：D→0|MA

例2-11 构造文法G_{11}，使

$$L(G_{11}) = \{\omega | \omega\text{是我们习惯的十进制有理数}\}。$$

G_{11}: $S \to R \mid +R \mid -R$

$R \to N \mid B \mid P$

$B \to N.D$

$P \to 0.D$

$N \to 0 \mid AM$

$D \to 0 \mid MA$

$A \to 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$M \to \varepsilon \mid 0M \mid 1M \mid 2M \mid 3M \mid 4M \mid 5M \mid 6M \mid 7M \mid 8M \mid 9M$

Note: 可以通过分析句子成分，分部分构造产生式

2.3 文法的构造

例2-12 构造产生算数表达式的文法G_{12}.

递归定义:

① 常数是算术表达式，变量是算术表达式。

② 若E_1、E_2是表达式,则$+E_1$、$-E_1$、E_1+E_2、E_1-E_2、E_1*E_2、E_1/E_2、$E_1\uparrow E_2$、$Fun(E_1)$是算术表达式，其中Fun为函数名。

③ 只有满足(1)和(2)的才是算术表达式。

G_{12}： $E \rightarrow id|c|+E|-E|E+E|E-E|E*E|E/E|E\uparrow E|Fun(E)$

用c表示常数，用id表示标识符（变量）

2.3 文法的构造

例2-13 构造产生语言 $\mathbf{L(G_{13}) = \{a^n b^n c^n | n \geq 1\}}$ 的文法。

构思1：

G：$S_1 \rightarrow ab \mid aS_1b$ 产生的语言$\{a^nb^n \mid n \geq 1\}$形式上看起来与语言$\{a^nb^nc^n \mid n \geq 1\}$比较接近。

G′：$S_2 \rightarrow c \mid cS_2$产生的语言是$\{c^n \mid n \geq 1\}$。

但是，语言$\{a^nb^nc^n \mid n \geq 1\}$并不是$\{a^nb^n \mid n \geq 1\}$和$\{c^n \mid n \geq 1\}$的乘积，即

$$\{a^nb^nc^n \mid n \geq 1\} \neq \{a^nb^n \mid n \geq 1\}\{c^n \mid n \geq 1\}$$

2.3 文法的构造

例2-13 构造产生语言 $\mathrm{L}(\mathrm{G}_{13}) = \{a^n b^n c^n | n \geq 1\}$ 的文法。

构思2:

考虑文法G：$\mathrm{S} \rightarrow \mathrm{abc} \mid \mathrm{aSbc}$

$\mathrm{S} \Longrightarrow \mathrm{abc}$

$\mathrm{S} \Longrightarrow \mathrm{aSbc} \Longrightarrow \mathrm{aabcbc}$

$\mathrm{S} \Longrightarrow \mathrm{aSbc} \Longrightarrow \mathrm{aaabcbcbc}$

文法G：$\mathrm{S} \rightarrow \mathrm{abc} \mid \mathrm{aSbc}$产生的语言为：$\{\mathrm{a}^n(\mathrm{bc})^n \mid n \geq 1\}$

焦点：交换b和c的位置。

例2-13 构造产生语言 $L(G_{13}) = \{a^n b^n c^n | n \geq 1\}$ 的文法。

G_{13}: S→aBC | aSBC，

CB→BC

aB→ab

bB→bb

bC→bc

cC→cc

G_{13}': S→abc|aSBc，

cB→Bc

bB→bb

续例2-13 在G_{13}中分别派生出abc, aabbcc和aaabbbccc。

G_{13}: $S \rightarrow aBC \mid aSBC$,

$CB \rightarrow BC$

$aB \rightarrow ab$

$bB \rightarrow bb$

$bC \rightarrow bc$

$cC \rightarrow cc$

$$S \Rightarrow \underline{aB}C \Rightarrow a\underline{bC} \Rightarrow abc$$

$$S \Rightarrow a\underline{S}BC \Rightarrow aaB\underline{CB}C$$

$$\Rightarrow a\underline{aB}BCC \Rightarrow aa\underline{bB}CC$$

$$\Rightarrow aab\underline{bC}C \Rightarrow aabb\underline{cC}$$

$$\Rightarrow aabbcc$$

$$\boxed{\begin{aligned} S &\Rightarrow a\underline{S}BC \Rightarrow a\underline{aB}CBC \\ &\Rightarrow aa\underline{bC}BC \Rightarrow aabcBC \quad \times \end{aligned}}$$

Note: 先将BC调整到正确的位置，再替换成终结符

续例2-13 在G_{13}中分别派生出abc, aabbcc和aaabbbccc。

G_{13}： $S \to aBC \mid aSBC$，

$CB \to BC$

$aB \to ab$

$bB \to bb$

$bC \to bc$

$cC \to cc$

$$S \Rightarrow a\underline{S}BC \Rightarrow aa\underline{S}BCBC \Rightarrow aaaB\underline{CB}CBC$$

$$\Rightarrow aaaBBC\underline{CB}C \Rightarrow aaaBB\underline{CB}CC$$

$$\Rightarrow aa\underline{aB}BBCCC \Rightarrow aaa\underline{bB}BCCC$$

$$\Rightarrow aaab\underline{bB}CCC \Rightarrow aaabbb\underline{C}CC$$

$$\Rightarrow aaabbb\underline{cC}C \Rightarrow aaabbbc\underline{cC}$$

$$\Rightarrow aaabbbccc$$

2.3 文法的构造

续例2-13 在G_{13}'中分别派生出abc, aabbcc和aaabbbccc。

G_{13}': $S \rightarrow abc \mid aSBc$,

$cB \rightarrow Bc$

$bB \rightarrow bb$

$$S \Rightarrow abc$$

$$S \Rightarrow a\underline{S}Bc \Rightarrow aab\underline{cB}c$$

$$\Rightarrow aa\underline{bB}cc \Rightarrow aabbcc$$

$$S \Rightarrow a\underline{S}Bc \Rightarrow aa\underline{S}BcBc$$

$$\Rightarrow aaab\underline{cB}cBc \Rightarrow aaabBc\underline{cB}c$$

$$\Rightarrow aaabB\underline{cB}cc \Rightarrow aaa\underline{bB}Bccc$$

$$\Rightarrow aaab\underline{bB}ccc \Rightarrow aaabbbccc$$

例2-14 构造产生语言$L(G_{14})$的文法。

$L(G_{14}) = \{\omega|\omega \in \{0,1\}^*$且$\omega$是以000为真后缀的串$\}$

G_{14}： $S \to A000,\ A \to 0|1|0A|1A$

$S \to A1000|A0000,\ A \to \varepsilon\ |0A|1A$

2.3 文法的构造

例2-15 构造产生语言L(G_{15})的文法。

$L(G_{15}) = \{\omega|\omega \in \{0,1\}^*$且$\omega$为不含有连续的0的串$\}$

空串	$S\rightarrow \varepsilon$
单0	$S\rightarrow 0$
连续的1串	$S\rightarrow A \quad A\rightarrow 1|1A$
0开头的0连接多于一个1的串	$S\rightarrow 0AS$
1开头的0连接多于一个1的串	$S\rightarrow AS$

G_{15}： $S\rightarrow \varepsilon|0|A|AS|0AS,\ A\rightarrow 1|1A$

G_{15}： $S\rightarrow \varepsilon|0|AS|0AS,\ A\rightarrow 1|1A$

2.3 文法的构造

例2-15 构造产生语言$L(G_{15})$的文法。

$$L(G_{15}) = \{\omega|\omega \in \{0,1\}^* 且\omega为不含有连续的0的串\}$$

S→ ε|0|S1|S10

S→ ε|0|1S|01S

S→ ε|0|01|1S|01S

2.3 文法的构造

课堂练习：构造产生下列语言的文法。

1. $L(G_1)=\{\varepsilon, 01, 0101, 010101, \ldots\}$
2. $L(G_2)=\{a^m b^{m+n} c^n | m, n \geq 0\}$
3. $L(G_3)=\{a^m b^{m+n+1} c^n | m, n \geq 1\}$
4. $L(G_4)=\{\omega | \omega \in \{0,1\}^*$，且$\omega$为所有包含子串001的串$\}$
5. $L(G_5)=\{\omega | \omega \in \{0,1\}^*$，且$\omega$为既没有连续的0，也没有连续的1的串$\}$

课堂练习讲解：构造产生下列语言的文法。

1. $L(G_1)=\{\varepsilon, 01, 0101, 010101, \ldots\}$

递归基础： ε

$S \to \varepsilon$

递归方法：每次添加一个01

$S \to S01$

$$\boxed{G_1: S \to \varepsilon | S01}$$

2.3 文法的构造

课堂练习讲解：构造产生下列语言的文法。

2. $L(G_2)=\{a^m b^{m+n} c^n | m,n \geq 0\}$

将语言拆分为两部分：$(a^m b^m)(b^n c^n)$

令变量A、B分别表示$a^m b^m$和$b^n c^n$

$S \to AB$

对A来说

- 递归基础：ε（m=0时）
- 递归方法：每次在A前面加一个a，后面加一个b

$A \to \varepsilon | aAb$

对B来说

- 递归基础：ε（n=0时）
- 递归方法：每次在B前面加一个b，后面加一个c

$B \to \varepsilon | bBc$

$$
\begin{aligned}
G_2: S &\to AB \\
A &\to \varepsilon | aAb \\
B &\to \varepsilon | bBc
\end{aligned}
$$

2.3 文法的构造

课堂练习讲解：构造产生下列语言的文法。

3. $L(G_3)=\{a^m b^{m+n+1} c^n | m, n \geq 1\}$

将语言中字符串分为三部分：$(a^m b^m) b (b^n c^n)$

令变量A、B分别表示$a^m b^m$和$b^n c^n$

$S \rightarrow AbB$

对A来说

- 递归基础：ab（m=1时）
- 递归方法：每次在A前面加一个a，后面加一个b

$A \rightarrow ab | aAb$

对B来说

- 递归基础：bc（n=1时）
- 递归方法：每次在B前面加一个b，后面加一个c

$B \rightarrow bc | bBc$

$$
\begin{aligned}
G_3: S &\rightarrow AbB \\
A &\rightarrow ab | aAb \\
B &\rightarrow bc | bBc
\end{aligned}
$$

2.3 文法的构造

课堂练习讲解：构造产生下列语言的文法。

4. $L(G_4)=\{\omega|\omega \in \{0,1\}^*$，且$\omega$为所有包含子串001的串$\}$

将语言拆分为三部分：001前的串、001、001后的串

令变量A、B分别表示001前的串和001后的串

对A来说，可以是任意的0、1串

- 递归基础：ε（001只要求是子串）
- 递归方法：每次在A后加一个0或1

B与A相同，可以是任意的0、1串

$S \rightarrow A001B$

$A \rightarrow \varepsilon|A0|A1$

$B \rightarrow \varepsilon|B0|B1$

$$G_4: S \rightarrow A001B$$
$$A \rightarrow \varepsilon|A0|A1$$
$$B \rightarrow \varepsilon|B0|B1$$

课堂练习讲解：构造产生下列语言的文法。

5. $L(G_5)=\{\omega|\omega \in \{0,1\}^*$，且$\omega$为既没有连续的0，也没有连续的1的串}

语言中字符串只能是0、1交替组成的字符串。

解法一：*令变量A表示由连续01组成的串（0开头1结尾）*

对A来说

- 递归基础：01
- 递归方法：在A后加一个01

$A \to 01|A01$

$G_5: S \to \varepsilon|0|1|A|A0|1A|1A0$

$A \to 01|A01$

语言中字符串可以是ε、0、1、0开头1结尾的01串(A)，0开头0结尾的01串(A0)、1开头1结尾的串(1A)、1开头0结尾的串(1A0)

$S \to \varepsilon|0|1|A|A0\ |1A|1A0$

课堂练习讲解：构造产生下列语言的文法。

5. $L(G_5)=\{\omega|\omega \in \{0,1\}^*$，且$\omega$为既没有连续的0，也没有连续的1的串$\}$

解法二：令变量B表示由连续10组成的串（1开头0结尾）

$$B \rightarrow 10|B10$$

$$S \rightarrow \varepsilon|0|1|B|B1|0B|0B1$$

$$\boxed{\begin{aligned} G_5': S &\rightarrow \varepsilon|0|1|B|B1|0B|0B1 \\ B &\rightarrow 10|B10 \end{aligned}}$$

章节目录

2.1 启示

2.2 形式定义

2.3 文法的构造

2.4 文法的乔姆斯基体系

2.5 空语句

2.6 本章小结

形式语言与自动机

计算模型/自动机：

FA	PDA	LBA	TM
Finite Automata	**Push Down Automata**	**Linear Bounded Automata**	**Turing Machine**
有穷状态自动机	下推自动机	线性有界自动机	图灵机

文法：

• 正则文法	• 上下文无关文法	• 上下文有关文法	• 短语结构文法
• 3型文法	• 2型文法	• 2型文法	• 0型文法

文法的分类

定义2-5 设文法G=(V, T, P, S)，则

- G叫做0型文法，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(phrase structure language, PSL)、递归可枚举集。

文法的分类

定义2-5 设文法G=(V, T, P, S)，则

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

文法的分类

定义2-5 设文法G=(V, T, P, S)，则

- 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**，或**上下文无关文法**(content free grammar, CFG)。
- L(G)叫做**2型语言**，或**上下文无关语言**(context free language, CFL)。

文法的分类

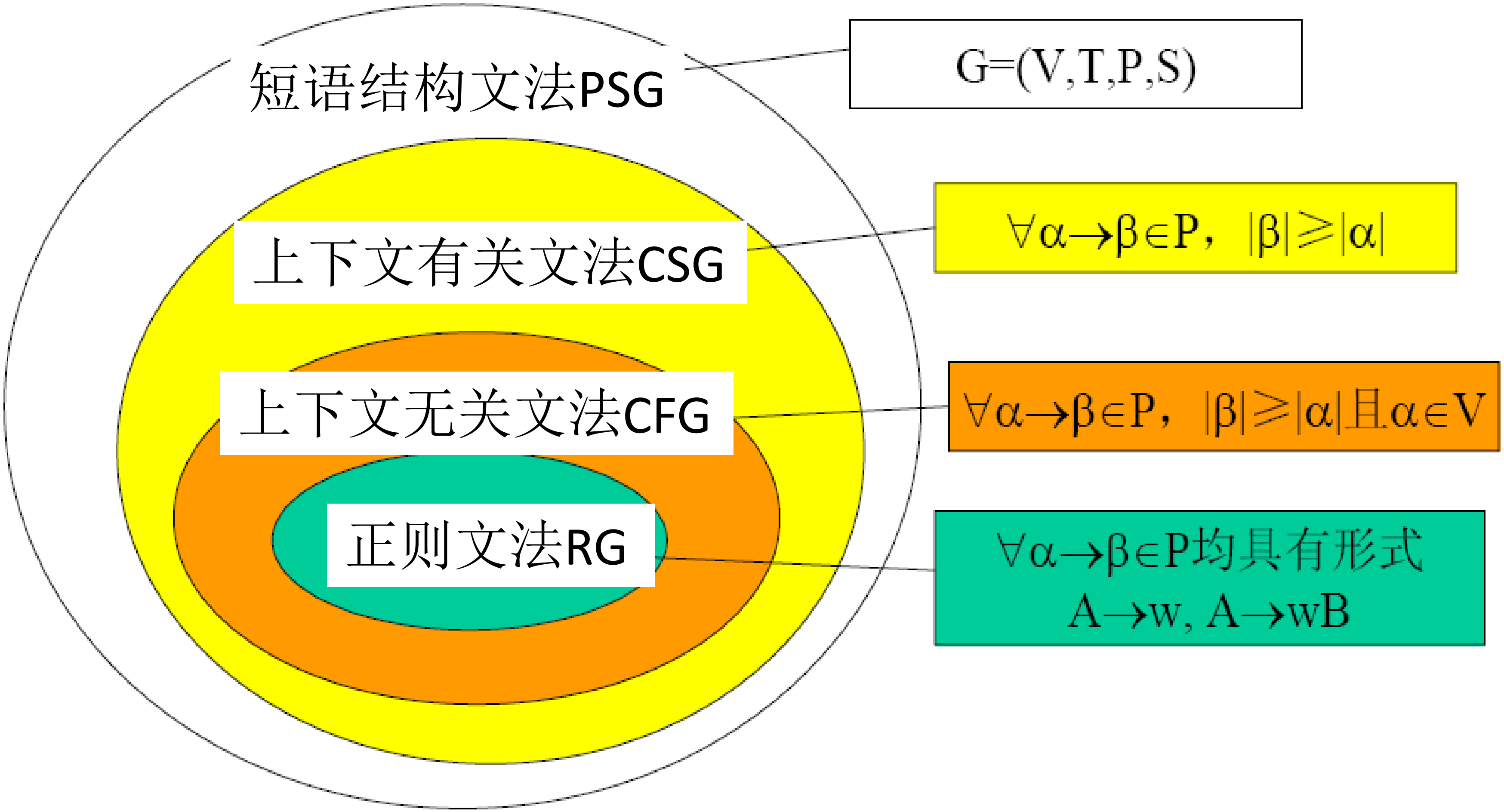
定义2-5 设文法G=(V, T, P, S)，则

- 如果对于$\forall \alpha \rightarrow \beta \in P$，$\alpha \rightarrow \beta$均具有形式 $A \rightarrow \omega$，$A \rightarrow \omega B$，其中$A, B \in V, \omega \in T^+$，则称G为3型文法，也可称为正则文法 **(regular grammar, RG)**。
- L(G)叫做3型语言，也可称为正则语言 **(regular language, RL)**。

2.4 文法的乔姆斯基体系

文法的分类

$$\alpha \in (V \cup T)^{+}, \ \beta \in (V \cup T)^{*}$$



0型—3型文法的区别在于：产生式的形式

2.4 文法的乔姆斯基体系

文法的分类

设文法G=(V,T,P,S)，则判断G是哪类文法的方法如下:

◆ G是短语结构文法;

◆如果所有产生式都有右边部分长度大于等于左边部分，那么G是上下文有关文法;

◆如果所有产生式的左边部分都是单个非终极符号，那么G是上下文无关文法;

◆如果所有产生式的右边部分都是以终极符号开始、含有至多一个非终极符号、如果有非终极符号则出现在最右边，那么G是正则文法。

2.4 文法的乔姆斯基体系

例2-16 判断文法的分类。

G1：$S \to 0|1|00|11$

G2：$S \to 0|1|0A|1B$, $A \to 0$, $B \to 1$

G3：$S \to 0|0S$

■上面文法是**PSG, CSG, CFG**和**RG**

G4：$S \to A|B|AA|BB$, $A \to 0$, $B \to 1$

G5：$S \to A|AS$

$A \to a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z$

G6：$S \to 00|11|22|33|0S0|1S1|2S2|3S3$

■上面文法是**PSG, CSG**和**CFG**，但不是**RG**

例2-16 判断文法的分类。

G7：$S \to aBC|aSBC$, $CB \to BC$, $aB \to ab$, $bB \to bb$, $bC \to bc$, $cC \to cc$

■上面文法是**PSG**和**CSG**，但不是**CFG**和**RG**

G8：$S \to A|B|BB$, $A \to 0$, $B \to 1$, $CACS \to 21$, $C \to 11$, $C \to 2$

G9：$S \to \boldsymbol{\varepsilon}|0S$

■上面文法是**PSG**，但不是**CFG**、**RG**和**CSG**。

2.4 文法的乔姆斯基体系

文法的分类

1. 如果一个文法G是RG，则它也是CFG、CSG和PSG；反之不一定成立。

2. 如果一个文法G是CFG，则它也是CSG和PSG；反之不一定成立。

3. 如果一个文法G是CSG，则它也是PSG；反之不一定成立。

文法的分类

1. RL也是CFL、CSL和PSL；反之不一定成立。

语言的分类

2. CFL也是CSL和PSL；反之不一定成立。
3. CSL也是PSL；反之不一定成立。

文法的分类

乔姆斯基文法体系

语言的分类

1. 当文法G是CFG时，L(G)可能是RL。
2. 当文法G是CSG时，L(G)可能是RL、CFL。
3. 当文法G是PSG时，L(G)可能是RL、CSL和CSL。

文法的分类

Callback: 例2-8 构造文法G，使L(G)={0, 1, 00, 11}	
G_1： S→0|1|00|11	RG
G_2： S→A|B|AA|BB	CFG

正则语言的判定

定理2-1 L是RL的充要条件是存在一个文法，该文法产生语言L，并且它的产生式要么是形如：A→a的产生式，要么是形如A→aB的产生式。其中A、B为语法变量，a为终极符号。

Callback: 定义2-5

如果对于$\forall \alpha \to \beta \in P$，$\alpha \to \beta$均具有形式$A \to \omega$，$A \to \omega B$，其中$A, B \in V, \omega \in T^+$，则称G为正则文法，L(G)称为正则语言。

正则语言的判定

定理2-1 L是RL的充要条件是存在一个文法，该文法产生语言L，并且它的产生式要么是形如：A→a的产生式，要么是形如A→aB的产生式。其中A、B为语法变量，a为终极符号。

证明： 充分性。

设存在文法G'，L(G')=L，且G'的产生式满足定理要求，即要么形如A→a，要么形如A→aB，则该文法满足正则语言的定义，G'是RG，则L是RL。

正则语言的判定

定理2-1 L是RL的充要条件是存在一个文法，该文法产生语言L，并且它的产生式要么是形如：$A \rightarrow a$的产生式，要么是形如$A \rightarrow aB$的产生式。其中A、B为语法变量，a为终极符号。

证明： 必要性。

思路：构造文法G’，使得G’的产生式满足定理的要求，再证明语言L(G’)=L。

（1）构造G’。由于L是RG，则一定存在RG G，使得L(G)=L。由定义2-5可知，G中的产生式要么形如$\boldsymbol{A}\to\boldsymbol{\omega}$, $\boldsymbol{A}\to\boldsymbol{\omega B}$，其中$\omega \in T^{+}$。

不妨设$\omega = a_1 a_2 \cdots a_n$，其中$n \geq 1$。则$A \to a_1 a_2 \cdots a_n$和$A \to a_1 a_2 \cdots a_n B$可改写如下：

$$
\begin{array}{ll}
A \to a_1 A_1 & A \to a_1 A_1 \\
A_1 \to a_2 A_2 & A_1 \to a_2 A_2 \\
A_2 \to a_3 A_3 & A_2 \to a_3 A_3 \\
\quad \cdots & \quad \cdots \\
A_{n-1} \to a_n & A_{n-1} \to a_n B
\end{array}
$$

（2）证明L(G)=L(G’)。

即需证对于$\forall x \in T^{*}$，$x \in$L(G) $\Leftrightarrow$ $x \in$ L(G’)。

1)$x \in$L(G) $\Rightarrow$ $x \in$ L(G'): 设$x = x_1x_2\cdots x_n$在G中可经一步推导得到，则必有A→ $x_1x_2\cdots x_n$在G的产生式集合中。由G'的构造方法，可知在G'中必存在产生式

$$
\begin{aligned}
&A \to x_1A_1\\
&A_1 \to x_2A_2\\
&A_2 \to x_3A_3\\
&\quad\cdots\\
&A_{n-1} \to x_n
\end{aligned}
$$

则在G'中，有$A \Rightarrow x_1A_1 \Rightarrow x_1x_2A_2 \Rightarrow \cdots \Rightarrow x_1x_2\cdots x_n$，即证得$x \in$ L(G')。假设在G中经k步推导得到的字符串$x \in$ L(G')，往证经k+1步推导的结果也在G'中。细节略。

2)同理再证$x \in$L(G') $\Rightarrow$ $x \in$ L(G)。细节略。证毕。

线性文法

<u>定义2-6</u> 设文法G=(V, T, P, S)，如果对于$\forall \alpha \to \beta \in P$，$\alpha \to \beta$均具有形式

$$A \to \omega,$$

$$A \to \omega B x,$$

其中$A, B \in V, \omega, x \in T^*$，则称G为线性文法，L(G)叫做线性语言。

右线性文法（right liner grammar）

定义2-7 设文法G=(V, T, P, S)，如果对于$\forall\alpha \rightarrow \beta \in P$，$\alpha \rightarrow \beta$均具有形式

$$A \rightarrow \omega,$$

$$A \rightarrow \omega B,$$

其中$A, B \in V, \omega \in T^{+}$，则称G为右线性文法，L(G)叫做右线性语言。

左线性文法（left liner grammar）

定义2-8 设文法G=(V, T, P, S)，如果对于$\forall \alpha \rightarrow \beta \in P$，$\alpha \rightarrow \beta$均具有形式

$$A \rightarrow \omega,$$

$$A \rightarrow B\omega,$$

其中$A, B \in V, \omega \in T^{+}$，则称G为左线性文法，L(G)叫做左线性语言。

2.4 文法的乔姆斯基体系

定理2-2 L是一个左线性语言的充要条件是存在文法G，G中的产生式要么是形如：$A \rightarrow a$的产生式，要么是形如$A \rightarrow Ba$的产生式。

其中A、B为语法变量，a为终极符号。

定理2-3 左线性文法与右线性文法等价。

思路：

要想证明本定理，需要完成如下工作：

- 对任意右线性文法G，我们能够构造出对应的左线性文法G′，使得L(G′)=L(G)；
- 对任意左线性文法G，我们能够构造出对应的右线性文法G′，使得L(G′)=L(G)。

例2-17 语言{0123456}的右线性文法和左线性文法的构造。（限定每个产生式右部有且仅有一个终极符）

(1) 右线性文法

G_r: $S_r \rightarrow 0A_r$

$A_r \rightarrow 1B_r$

$B_r \rightarrow 2C_r$

$C_r \rightarrow 3D_r$

$D_r \rightarrow 4E_r$

$E_r \rightarrow 5F_r$

$F_r \rightarrow 6$

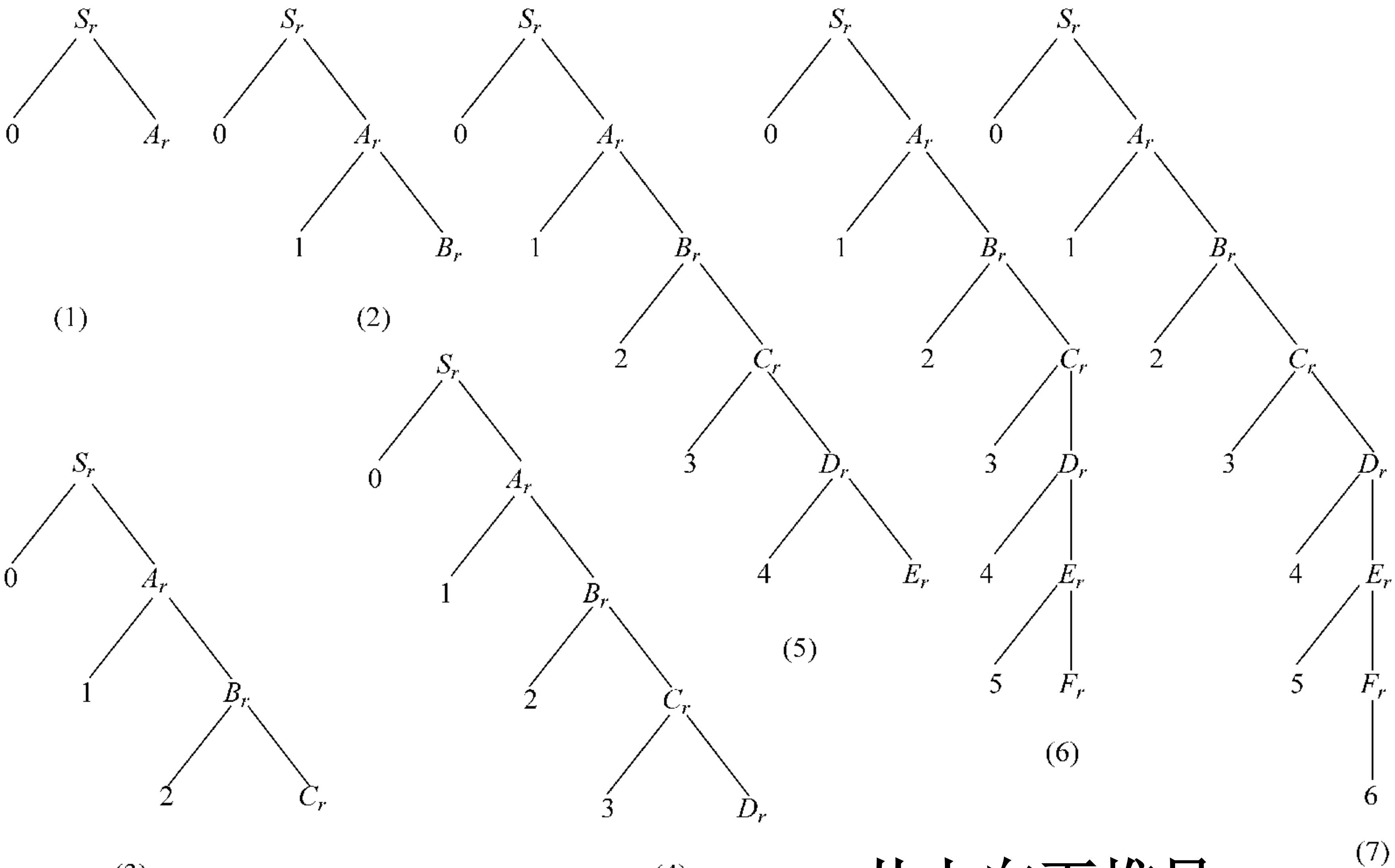
2.4 文法的乔姆斯基体系

例2-17 语言{0123456}的右线性文法和左线性文法的构造。（限定每个产生式右部有且仅有一个终极符）

(1) 右线性文法

0123456在文法G_r中的推导

$S_r \Rightarrow 0A_r$	使用产生式$S_r \rightarrow 0A_r$
$\Rightarrow 01B_r$	使用产生式$A_r \rightarrow 1B_r$
$\Rightarrow 012C_r$	使用产生式$B_r \rightarrow 2C_r$
$\Rightarrow 0123D_r$	使用产生式$C_r \rightarrow 3D_r$
$\Rightarrow 01234E_r$	使用产生式$D_r \rightarrow 4E_r$
$\Rightarrow 012345F_r$	使用产生式$E_r \rightarrow 5F_r$
$\Rightarrow 0123456$	使用产生式$F_r \rightarrow 6$



从上向下推导

例2-17 语言{0123456}的右线性文法和左线性文法的构造。（限定每个产生式右部有且仅有一个终极符）

(2) 左线性文法

$$
\begin{aligned}
G_l\text{：}\quad & S_l \rightarrow A_l 6 \\
& A_l \rightarrow B_l 5 \\
& B_l \rightarrow C_l 4 \\
& C_l \rightarrow D_l 3 \\
& D_l \rightarrow E_l 2 \\
& E_l \rightarrow F_l 1 \\
& F_l \rightarrow 0
\end{aligned}
$$

例2-17 语言{0123456}的右线性文法和左线性文法的构造。（限定每个产生式右部有且仅有一个终极符）

(2) 左线性文法

0123456在文法G_l中的推导

$S_l \Rightarrow A_l6$	使用产生式$S_r \rightarrow A_l6$
$\Rightarrow B_l56$	使用产生式$A_l \rightarrow B_l5$
$\Rightarrow C_l456$	使用产生式$B_l \rightarrow C_l4$
$\Rightarrow D_l3456$	使用产生式$C_l \rightarrow D_l3$
$\Rightarrow E_l23456$	使用产生式$D_l \rightarrow E_l2$
$\Rightarrow F_l123456$	使用产生式$E_l \rightarrow F_l1$
$\Rightarrow 0123456$	使用产生式$F_l \rightarrow 0$

例2-17 语言{0123456}的右线性文法和左线性文法的构造。（限定每个产生式右部有且仅有一个终极符）

(2) 左线性文法

但是当用计算机系统处理一个句子的时候，会希望按照该句子的符号出现的前后顺序来对句子进行处理，系统从前向后逐个地扫描句子的每一个符号。

采用归约的方式。

2.4 文法的乔姆斯基体系

例2-17 语言{0123456}的右线性文法和左线性文法的构造。（限定每个产生式右部有且仅有一个终极符）

- 0123456被归约成文法G_l的开始符号S

$\underline{0}123456$

$\Leftarrow \underline{F_l 1}23456$　　使用产生式$F_l \rightarrow 0$

$\Leftarrow \underline{E_l 2}3456$　　使用产生式$E_l \rightarrow F_l 1$

$\Leftarrow \underline{D_l 3}456$　　使用产生式$D_l \rightarrow E_l 2$

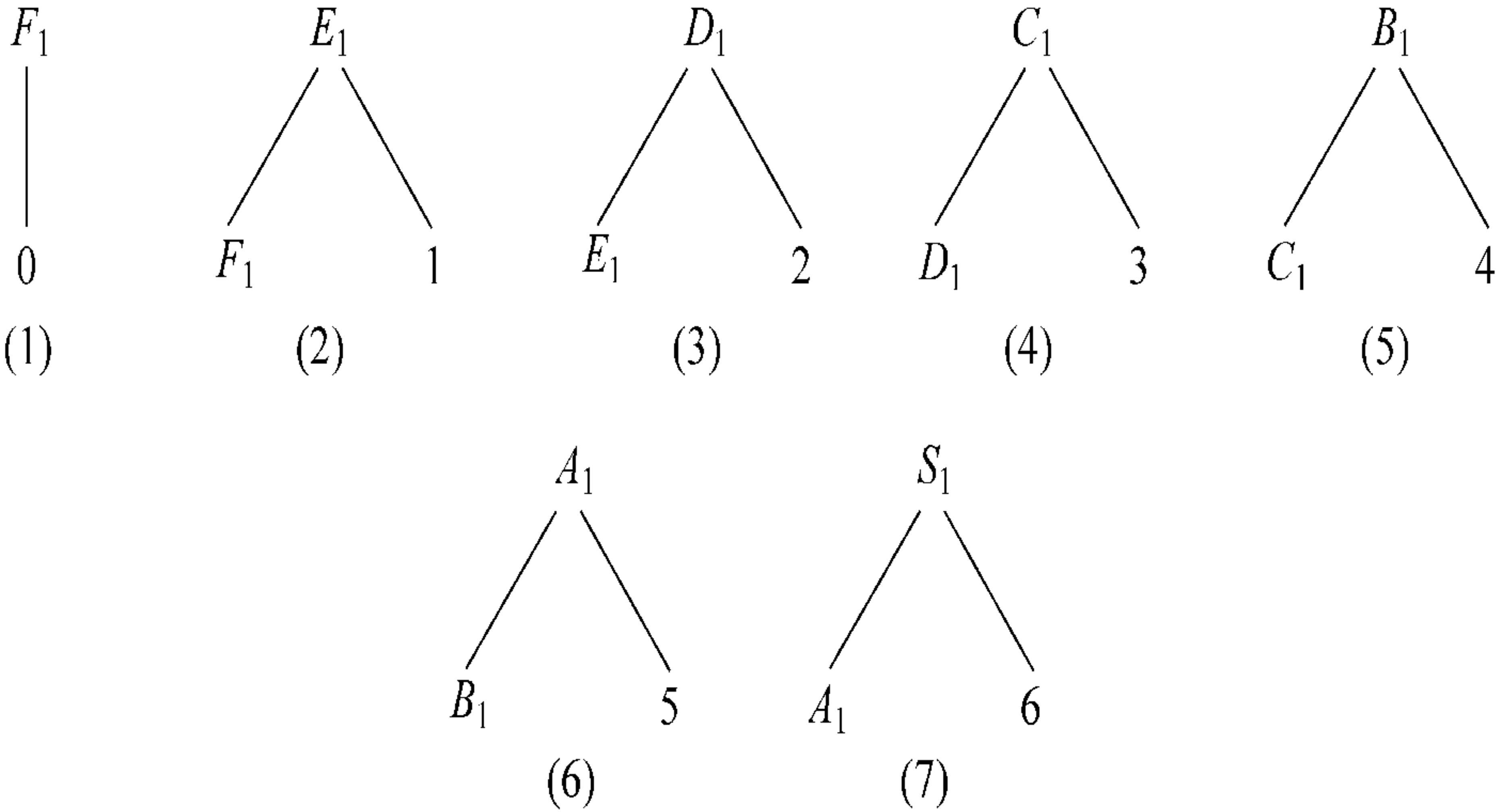
$\Leftarrow \underline{C_l 4}56$　　使用产生式$C_l \rightarrow D_l 3$

$\Leftarrow \underline{B_l 5}6$　　使用产生式$B_l \rightarrow C_l 4$

$\Leftarrow \underline{A_l 6}$　　使用产生式$A_l \rightarrow B_l 5$

$\Leftarrow S_l$　　使用产生式$S_l \rightarrow A_l 6$

2.4 文法的乔姆斯基体系



从下向上归约

定理2-4 左线性文法的产生式与右线性文法的产生式混用所得到的文法不是RG。

设有文法

G_{15}： $S \rightarrow 0A$

$A \rightarrow S1|1$

不难看出，$L(G_{15})=\{0^n1^n|n\geq 1\}$。

我们构造不出RG G，使得$L(G)= L(G_{15})=\{0^n1^n|n\geq 1\}$。

因为$L(G_{15})=\{0^n1^n|n\geq 1\}$不是RL。所以，$G_{15}$不是RG。

章节目录

2.1 启示

2.2 形式定义

2.3 文法的构造

2.4 文法的乔姆斯基体系

2.5 空语句

2.6 本章小结

2.5 空语句

<u>定义2-9</u> 形如$A \to \varepsilon$的产生式叫做空产生式，也可叫做ε产生式。

根据文法分类的定义：

■在RG、CFG、CSG中，都不能含有空产生式。

■所以，任何RL、CFL、CSL中都不含有空语句ε。

RG

■ 如果对于$\forall\alpha \to \beta \in P$，$\alpha \to \beta$均具有形式$A \to \omega$，$A \to \omega B$，其中$A, B \in V, \omega \in T^+$，则称G为3型文法(type 3 grammar)，也可称为正则文法(regular grammar, RG)。

CFG

■ 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(content free grammar, CFG)。

CSG

■ 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(content sensitive grammar, CSG)。

2.5 空语句

- 空语句ε在一个语言中的存在并不影响该语言的有穷描述的存在。除了为生成空语句ε外，空产生式可以不被用于语言中其他任何句子的推导中。
- 允许CSL、CFL、RL包含空语句后，还会给问题的处理提供一些方便。
- 允许在RG、CFG、CSG中含有空产生式，也就允许CSL、CFL、RL包含空语句ε。

2.5 空语句

设G=(V，T，P，S)是一个文法，如果S不出现在G的任何产生式的右部，则：

1. 如果G是CSG，则仍然称G=(V，T，P∪{S→ε}，S)为CSG；G产生的语言仍然称为CSL。
2. 如果G是CFG，则仍然称G=(V，T，P∪{S→ε}，S)为CFG；G产生的语言仍然称为CFL。
3. 如果G是RG，则仍然称G=(V，T，P∪{S→ε}，S)为RG；G产生的语言仍然称为RL。

2.5 空语句

定理2-5 下列命题成立：

(1) 如果L是CSL，则L∪{ε} 仍然是CSL。

(2) 如果L是CFL，则L ∪ {ε}仍然是CFL。

(3) 如果L是RL，则L ∪ {ε}仍然是RL。

2.5 空语句

定理2-6 下列命题成立：

(1) 如果L是CSL，则L−{ε}仍然是CSL。

(2) 如果L是CFL，则L−{ε}仍然是CFL。

(3) 如果L是RL，则L−{ε}仍然是RL。

2.5 空语句

- 对于任意文法G=(V，T，P，S)，G中的其他变量A，出现形如A→ε的产生式是不会改变文法产生的语言的类型的，而且这样一来，对我们进行文法的构造等工作还提供了很多方便。
- 所以，我们约定：对于G中的任何变量A，在需要的时候，可以出现形如A→ε的产生式。

章节目录

2.1 启示

2.2 形式定义

2.3 文法的构造

2.4 文法的乔姆斯基体系

2.5 空语句

2.6 本章小结

2.6 本章小结

- 介绍了文法的形式定义和推导、归约、文法定义的语言、句子、句型、文法的等价等重要概念；
- 讨论了如何根据语言的特点进行文法构造；
- 介绍了乔姆斯基体系，将文法划分成PSG、CSG、CFG、RG等4类。在这些文法中，线性文法是一类重要的文法；
- 空产生式对文法没有影响，空语句对语言没有影响。

Thanks!