第二章 上下文无关文法和语言

§2.1 文法和语言的表示

§2.2 文法和语言的定义

§2.3 句型的分析

§2.4 文法的实用限制和其他表示法

§2.5 文法和语言的Chomsky分类

§2.4 文法的实用限制和其它表示方法

§2.4.1文法的实用限制

1、不含无用产生式

设G=（V_n，V_t，P，S）是一文法，G中的符号 $x \in V_n \cup V_t$是有用的，则 x 必满足

①存在α、$\beta \in V^*$，有$S \overset{*}{\Rightarrow} \alpha x \beta$ （无用：不可到达）

②存在$\omega \in V_t^*$ 使$\alpha x \beta \overset{*}{\Rightarrow} \omega$ （无用：无法终止）

称符号x是有用的，否则是无用的

无用产生式：产生式的左部或右部含有无用符号。

例1： G[S]:

S →aA |Bb

A →aA |c

B →bB

C →cC |d

例2： G[S] ：

1) S→Be

2) B→Ce

3) B→Af

4) A→Ae

5) A→e

6) C→Cf

7) D→f

2、不含有害规则

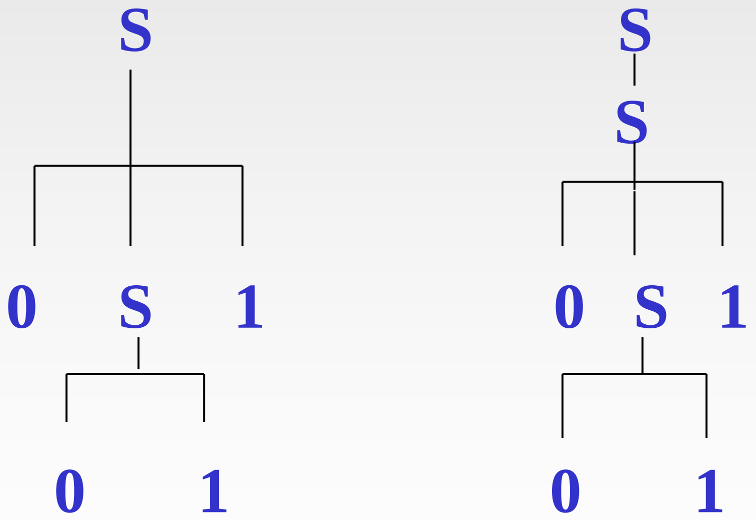
形如 U→U 的规则（原因①不必要②引起二义性）

例；$G_1[S]$：$S \to 0S1|01$　　　　G_1无二义性文法

$G_2[S]$：$S \to 0S1|01|S$　　　　G_2二义性文法

$$L(G_1) = L(G_2) = \{0^n 1^n \mid n >= 1\}$$

G2文法句子0011 的两棵不同语法树.



§2.4.2 ε-产生式的消除

如某L(G)中不含ε，可消除G中的全部ε产生式；

如某L(G)中含ε，肯定不能消除G中的全部ε产生式；

消除步骤：

1. 算法2.3，找出G中满足$A \overset{*}{\Rightarrow} \varepsilon$的所有A，构成集合W；

2. 算法2.4，若ε不属于L(G)，构造不含ε产生式的等价文法G'；

算法2.5，若ε属于L(G)，构造仅含S① → ε产生式的等价文法G_1(S①)；

算法2.3

设$G=(V_n, V_t, P, S)$

①作集合$W_1=\{A \mid A\rightarrow \varepsilon \in P\}$

②作集合序列$W_{k+1}= W_k \cup \{B\rightarrow \beta \in P$且$\beta \in W_k^+\}$

显然$W_k \leqq W_{k+1}$ $(K>=1)$，由于V_n有限，故必存在某i，使得$W_i=W_{i+1}=\ldots$，令$W=W_i$，对每个$A\in W$，$A\overset{*}{\Rightarrow}\varepsilon$

特别：当$S\in W$，则$\varepsilon \in L(G)$；否则，ε不属于$L(G)$。

例G[S]: $S \rightarrow aA$

$A \rightarrow BC$

$B \rightarrow bB \mid \varepsilon$

$C \rightarrow cC \mid \varepsilon$

求W

执行算法2.3

$W_1=\{B, C\}$

$W_2=\{A, B, C\}$

$W_3=W_2=W_1$ …

$W=\{A, B, C\}$

①作集合$W_1=\{A \mid A \rightarrow \varepsilon \in P\}$

②作集合序列$W_{k+1}= W_k \cup \{B \rightarrow \beta \in P$且$\beta \in W_k^+\}$

显然$W_k \leq W_{k+1}$ $(K>=1)$，由于V_n有限，故必存在某i，

使得$W_i=W_{i+1}=\ldots$，令$W=W_i$，对每个$A \in W$，$A \overset{*}{\Rightarrow} \varepsilon$

特别：当$S \in W$，则$\varepsilon \in L(G)$；否则，ε不属于$L(G)$。

算法2.4

设$G=(V_n, V_t, P, S)$，且ε不属于L(G)，则按下述算法构造

$G'=(V_n, V_t, P', S)$，使$L(G')=L(G)$；

①按算法2.3将V_n分为两个不相交的子集，W及V_n-W

②设$X\rightarrow X_1X_2...X_m$是P中的任一产生式，按下述规则将所有形

$Y\rightarrow Y_1Y_2...Y_m$的产生式放入P'中，对于一切$1<=i<=m$

（ⅰ）若X_i不属于W，即X_i属于$(V_n-W)\cup V_t$，则取$Y_i=X_i$；

（ⅱ）若X_i属于W，则分别取Y_i为X_i和ε，即如果$Y_1Y_2...Y_m$中有j个符号属于W，则将有2^j个形如$Y\rightarrow Y_1Y_2...Y_m$的产生式放入P'中，但若所有的$X_i$均属于W，却不能把所有的$Y_i$都取为ε。

文法G'与G等价且不含ε产生式.

例G[S]:S→aA

A→BC

B→bB | ε

C→cC | ε

执行算法2.3

W={A, B, C}

S不属于W

执行算法2.4

G[S]:S→aA | a

A→BC | B | C

B→bB | b

C→cC | c

算法2．5

设$G=(V_n, V_t, P, S)$，且ε属于$L(G)$，S不出现在任何产生式的右部，执行算法2.4得 $G'=(V_n, V_t, P', S)$，但$S\rightarrow\varepsilon$属于 G'．

否则，按下述算法先构造$G'=(V_n^{①}, V_t, P', S^{①})$，再构造$G_1=(V_n^{①}, V_t, P^{①}, S^{①})$，使$L(G_1)=L(G)$；

①引入新的符号$S^{①}$（$S^{①}$不属于V），作为G'的开始符号，并令$V_n^{①}=V_n\cup\{S^{①}\}$；

②作产生式集 $P'=P\cup\{S^{①}\rightarrow\alpha \mid S\rightarrow\alpha\in P\}$得到G'．

③对文法$G'=(V_n^{①}, V_t, P', S^{①})$，执行算法2.4消去P'中的全部ε产生式，并将$S^{①}\rightarrow\varepsilon$加入得到$P^{①}$，最终得到文法$G_1=(V_n^{①}, V_t, P^{①}, S^{①})$，$L(G_1)=L(G)$；

例G[S]:S→cS

S→AB

A→aAb |ε

B→Bb | ε

执行算法2.3

W={A, B, S}

S属于W

执行算法2.5

S出现在产生式的右边

P' = P ∪{ S① →cS | AB }

执行算法2.4

G[S①]: S① → cS | c | AB | A | B | ε

S → cS | c | AB | A | B

A → aAb | ab   B → Bb | b

§2.4.3 文法的其它表示方法

一、扩充的BNF表示

BNF：元符号 〈，〉，::=，(，→，)，|

扩充的BNF(EBNF)：〈，〉，::=(→)，|，（，），{，}，[，]

1、{ }

$\{t\}_n^m$　$t \in V^*$，符号串t自重复n到m次。

$\{t\}$　$t \in V^*$，符号串t自重复0到无穷次。

例:BNF: G[〈无符号整数〉]　　(含左递归)

〈无符号整数〉→〈数字〉 | 〈无符号整数〉〈数字〉

〈数字〉→0 |1 |2 |3 |… … | 9

扩充的BNF:

G[〈无符号整数〉]

〈无符号整数〉→〈数字〉{〈数字〉}　(不含左递归)

〈数字〉→0 |1 |2 |3 |… … | 9

2、[ ]

[t] , t $\in V^*$, t 符号串可有可无

例:BNF :S→if B then S

| if B then S else S

扩充的BNF:S→if B then S [else S]

3、( ) 规则中提取公因子

例:BNF: U→xy | xw | …… | xz

扩充的BNF: U→x(y | w | …… | z)

令:U′ → y | w | …… | z

则文法为:U → x U′

U′ → y | w | …… | z

例：BNF：G[〈标识符〉]

〈标识符〉→〈字母〉|〈标识符〉〈字母〉

|〈标识符〉〈数字〉

〈标〉→〈字〉|〈标〉(〈字〉|〈数〉)

〈字〉→a|b|…|z|A|…|Z

〈数〉→0|1|2|…|9

〈标〉=>〈标〉(〈字〉|〈数〉)

$=^{+}>$〈标〉(〈字〉|〈数〉)…… (〈字〉|〈数〉)

=>〈字〉(〈字〉|〈数〉)…… (〈字〉|〈数〉)

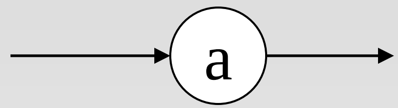
〈标〉 =>〈字〉{(〈字〉|〈数〉)}

〈标〉 =>〈字〉{〈字〉|〈数〉}

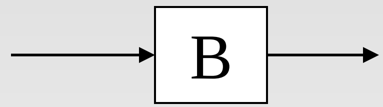
二、语法图

构造：规则右部的符号

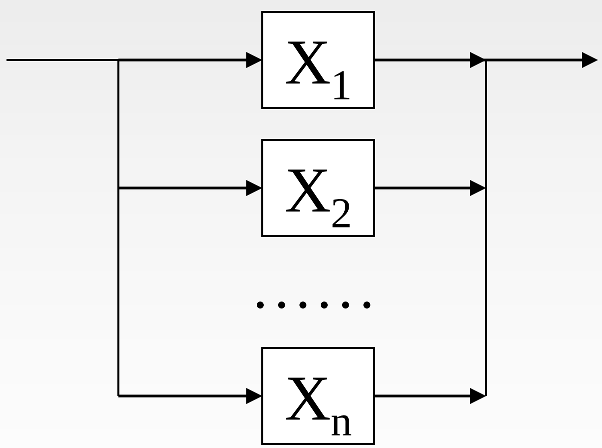
(1) $a \in Vt$



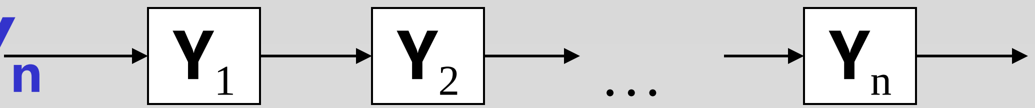
(2) $B \in Vn$



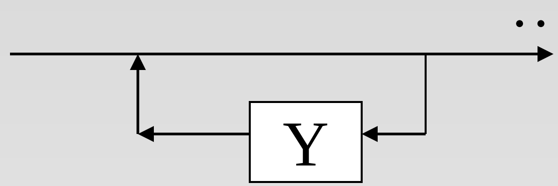
(3)形如 $U \rightarrow X_1 \mid X_2 \mid \ldots\ldots \mid X_n$



$X \rightarrow Y_1Y_2\ldots\ldots Y_n$



$X \rightarrow \{Y\}$

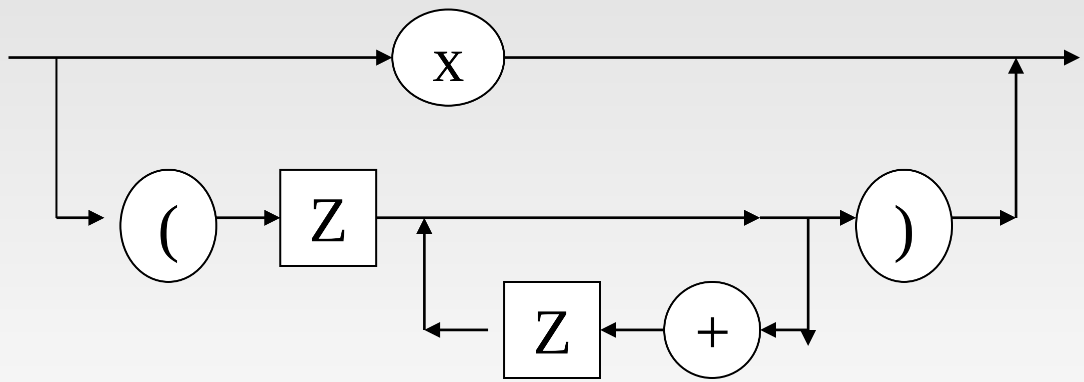


例： G[Z]:

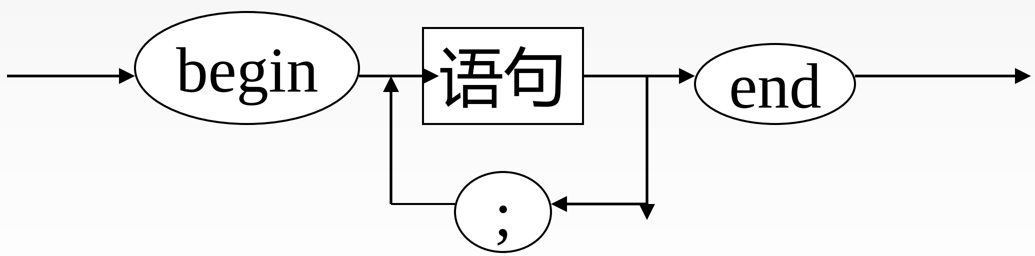
$Z \rightarrow x \mid (B)$

$B \rightarrow ZC$

$C \rightarrow \{+Z\}$



例Pascal语言中的〈分程序〉



第二章 上下文无关文法和语言

§2.1 文法和语言的表示

§2.2 文法和语言的定义

§2.3 句型的分析

§2.4 文法的实用限制和其他表示法

§2.5 文法和语言的Chomsky分类

§2.5 文法和语言的Chomsky分类

文法是一个四元组G=（V_n，V_t，P，Z）

乔姆斯基根据文法中**产生式的不同**，将文法分为四类，每一种文法对应一种语言。

- 0型文法：文法G中规则呈

$\alpha \to \beta \qquad \alpha \in V^+,\ \beta \in V^*$

也称短语结构文法（phrase structure grammar，PSG），确定的语言为0型语言L_0（**说明：对产生式基本无限制**）

0型文法的能力相当于图灵机，可以表征任何递归可枚举集，且任何0型语言都是递归可枚举的。

- 1型文法：文法G中规则是

$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$　　α_1，$\alpha_2 \in V^*$，

$A \in V_n$，$\beta \in V^+$

也称上下文有关文法(context-sensitive grammar, CSG)。

确定的语言为1型语言L_1，也称上下文有关语言。

等价定义：对任一产生式$\alpha \rightarrow \beta$，都有$|\beta| \geq |\alpha|$，仅仅$S \rightarrow \varepsilon$除外($\alpha$，$\beta \in V^+$)。

例如：aUb→aABBaab

➢ 2型文法：文法G中规则呈

$A \rightarrow \beta$ $\quad\quad$ $A \in V_n$，$\beta \in V^+$

也称上下文无关文法（context-free grammar, CFG）。非确定的下推自动机识别

确定的语言为2型语言L_2或上下文无关语言。

该文法相当于对1型文法中的规则形式加以限制，即要求$\alpha 1$和$\alpha 2$必须为空。

在语法分析中用于描述语法类

例：文法G[S]：$S \rightarrow AB$ $\quad A \rightarrow BS|0$ $\quad B \rightarrow SA|1$

➢ <u>3型文法</u>：文法G中规则呈：

- A→aB或A→a　　　A、**B** ∈Vn，a∈Vt，

称G为右线形正则文法。

- A→Ba或A→a　　　A、**B** ∈Vn，a∈Vt，

称G为左线形正则文法。

以上两者统称为3型文法或正规文法，确定的语言为3型语言L3或正规（正则）语言。

正则语言可用有限自动机来识别。

在词法分析中用于描述单词符号

例： 3型文法

G[S]: $S \rightarrow 0A \mid 1B \mid 0$

$A \rightarrow 0A \mid 1B \mid 0S$

$B \rightarrow 1B \mid 1 \mid 0$

G[I]: $I \rightarrow 1T$

$I \rightarrow 1$

$T \rightarrow 1T$

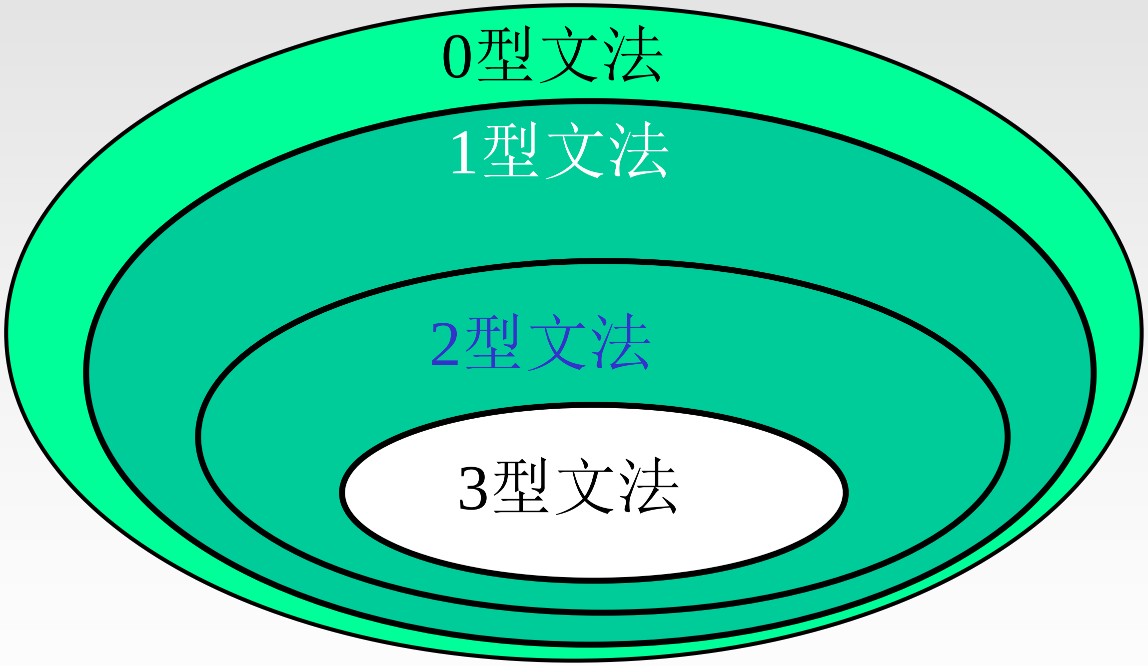
$T \rightarrow dT$

$T \rightarrow 1$

$T \rightarrow d$

§2.5 文法和语言的Chomsky分类

四类文法之间的逐级“包含”关系



说明：

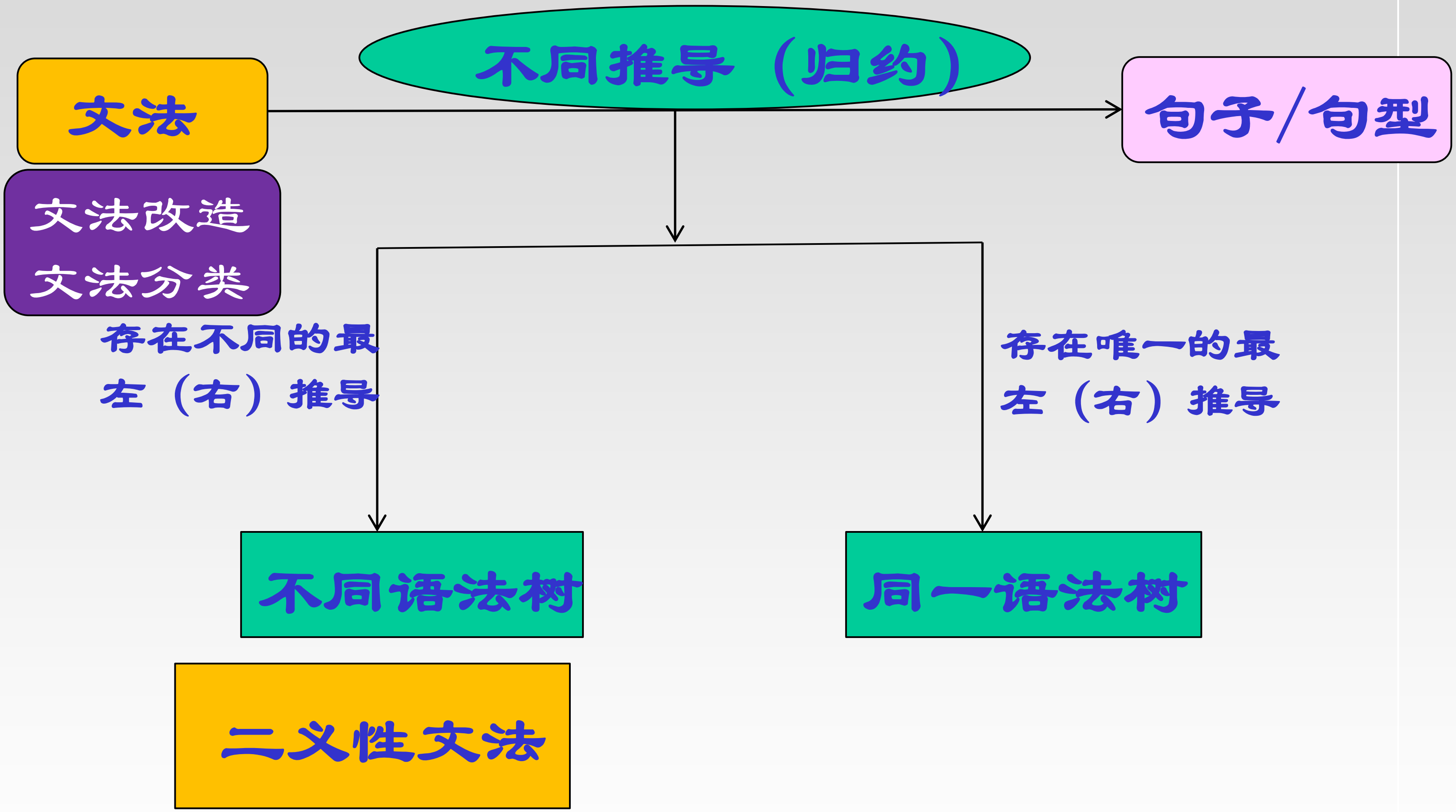
①由于对规则的限制逐渐增加，

因此L_0、L_1 、 L_2 、 L_3的语言范围逐渐减小。

②一个文法是正则的，必然是上下文无关的， 上下文无关文法有足够的能力描述程序设计语言的语法结构。

③本课主要讨论正则文法和上下文无关文法。

第二章回顾：



文法
文法改造
文法分类
不同推导（归约）
句子/句型
存在不同的最
左（右）推导
存在唯一的最
左（右）推导
不同语法树
同一语法树
二义性文法

习题：

① 证明G[S]：$S \rightarrow aSb|Sb|b$ 为二义性文法

解：找一个句子，可以生成两棵语法树

如aabbbb

习题：

②将下列文法改写成无二义性文法：

G[S]: $S \rightarrow SS \mid (S) \mid ()$

解：分析根源，再改写$S \rightarrow SS$为

G' [S]: $S \rightarrow TS \mid T$　　$T \rightarrow (S) \mid ()$

习题：

③写一个文法，使其语言L(G)是非零开头的正偶数集合

解：G[S]：$S \rightarrow XYZ|2|4|6|8$

$X \rightarrow 1|2|3|4|5|6|7|8|9$

$Y \rightarrow YX \mid Y0 \mid \varepsilon$

$Z \rightarrow 0|2|4|6|8$

第二章作业

- 2-2 (1) (2)　　2-3 (1) (2)
- 2-6　2-10
- 2-11 (2) (3)　2-14 (1) (2)
- 补充作业——课程平台下载
- 自己检查是否存在重复