

回顾：Chapter 1

1、何谓源程序、目标程序、翻译程序、编译程序和解释程序？它们之间可能有何种关系？

源程序：用源语言编写的程序。

目标程序：源程序经翻译程序过加工处理后生成的程序。

翻译程序：将源程序转换为与其逻辑上等价的目标程序。

编译程序：源语言为高级语言，目标语言为汇编语言或机器语言的翻译程序。

① 先翻译后执行 ② 执行速度快 ③ 多次运算

解释程序：源语言程序作为输入，但不产生目标程序，而是边解释边执行源程序本身。

① 边解释边执行 ② 有利于程序的调试 ③ 1次运算

回顾：chapter1

2、一个典型的编译系统通常由有哪些部分组成？
各部分的主要功能是什么？

编译程序 { 词法分析
语法分析
语义分析与中间代码生成
中间代码优化
目标代码生成及优化

Chapter 1

① 词法分析(Lexical Analysis):

从左到右一个字符一个字符的读入源程序，对构成源程序的字符串进行扫描和分解，从而识别出一个个单词（也称单词符号或简称符号）。

② 语法分析(Syntax Analysis):

在词法分析的基础上将单词序列分解成各类语法短语，如“程序”，“语句”，“表达式”等等。

③ 语义分析(Syntactic Analysis):

语义分析是在语法分析程序确定出语法短语后，审查有无语义错误，并为代码生成阶段收集类型信息。

Chapter 1

④ 中间代码生成(Generation of intermediate code):

完成语法分析和语义处理工作后，编译程序将源程序变成一种内部表示形式，这种内部表示形式叫做中间语言或称中间代码，它是一种结构简单、含义明确的记号系统。

⑤ 中间代码优化(Optimization of intermediate code):

为了使生成的目标代码更为高效，可以对产生的中间代码进行变换或进行改造，这就是代码的优化。

⑥ 目标代码生成及优化

(Generation and Optimization of code):

目标代码生成是编译器的最后一个阶段。在生成目标代码时要考虑以下几个问题：计算机的系统结构、指令系统、寄存器的分配以及内存的组织等。

第二章 形式语言与自动机理论基础

- 2.1 文法的引入---预备知识
- 2.2 文法的讨论
- 2.3 文法和语言的定义
- 2.4 分析树和二义性
- 2.5 形式语言概观

2.1 预备知识

- 2.1.1 字母表
- 2.1.2 符号串
 - 一、符号串的定义
 - 二、术语
 - 三、符号串的运算
 - 四、符号串集合的运算

2.1.1 字母表

字母表是**符号**的**非空有穷**集合；例： $\Sigma = \{a, b, c\}$
任何程序语言都有自己的字母表。

1. 计算机语言：由符号“0”和“1”组成的字母表： $\Sigma = \{0, 1\}$

2. Pascal语言字母表： $\Sigma =$
 $\{A \sim Z, a \sim z, 0 \sim 9, +, -, *, /, <, =, >, :$
 $, ', ", ;, ., \uparrow, (,), \{, \}, [,] \}$

3. C语言字母表： $\Sigma =$
 $\{A \sim Z, a \sim z, 0 \sim 9, +, -, *, /, <, =, >, _ , \&, \wedge,$
 $\sim, \backslash, :, ', ", ;, ., ?, (,), \{, \}, [,], \text{空格}, !, \#, \% \}$

2.1.2 符号串

一. 符号串的定义

由**字母表** Σ 中的符号所组成的任何**有穷序列**被称之为该字母表上的符号串。**空符号串**：无任何符号的符号串，记作 ϵ

符号串的形式定义：

- (1) 字母表 Σ 上字符是 Σ 上的符号串。
- (2) 若 x 是 Σ 上的符号串，而 a 是 Σ 的元素，则 xa 是 Σ 上的符号串。
- (3) y 是 Σ 上的符号串，当且仅当它由 (1) 和 (2) 导出。

二. 术语

(设**s**是符号串banana)

前缀: 移走s的尾部的零个或多个符号所得符号串
 ϵ , b, ba, ban, bana, banan, banana

后缀: 删去s的头部的零个或多个符号所得符号串
banana, anana, nana, ana, na, a, ϵ

子串: 从s中删去一个前缀和一个后缀所得符号串
banana, anana, banan

真前缀、真后缀和真子串: 不是**s**和 ϵ 的前缀、后缀和子串

子序列: 从s中删去零个或多个符号(不要求是连续)而得到的符号串。如baaa

长度: 是符号串中符号的个数。例如 $|aab|=3$, $|\epsilon|=0$

语 言： 确定字母表上字符串的任何集合

例如：

不含任何元素的空集合 \emptyset ，即 $\{\}$

只含有空符号串的集合 $\{\epsilon\}$

符合C语法的程序组成的集合
(C语言)

符合英文语法的句子组成的集合

三. 符号串的运算

1. **连接**: 设 x 和 y 是符号串, 它们的连接 xy 是把符号串 y 写在符号串 x 的之后得到的符号串。

例如 $x=ba, y=nana, xy=banana$.

$$\varepsilon x = x \varepsilon = ba$$

2. **方幂**: $x^0=\varepsilon \quad x^1=x \quad x^2=xx \quad \dots \quad x^n=x^{n-1}x$

例如 $x=ba \quad x^1=ba \quad x^2=baba \quad x^3=bababa, \dots$

四. 语言 (符号串集合) 的运算

(语言L和M)

1. **合并**: $L \cup M = \{s \mid s \in L \text{ or } s \in M\}$

属于L或属于M的符号串s所组成的集合

2. **连接**: $LM = \{st \mid s \in L \text{ and } t \in M\}$

s属于L和**t**属于M的所有符号串**st**组成的集合

$$\{\varepsilon\} \mathbf{L} = \mathbf{L} \{\varepsilon\} = \mathbf{L}$$

3. **方幂**: $L^0 = \{\varepsilon\} \quad L^1 = L \quad \dots \quad L^n = L^{(n-1)} L$
($n > 0$)

4. 语言L的正闭包，记作 L^+

$$L^+ = L^1 \cup L^2 \cup L^3 \cup L^4 \cup \dots$$

5. 语言L的Kleene闭包(自反闭包)，记作 L^*

$$L^* = L^0 \cup L^+ = L^0 \cup L^1 \cup L^2 \cup L^3 \cup \dots$$

例： $A = \{x, y\}$

$$A^+ = \{ \underbrace{x, y}_{A^1}, \underbrace{xx, xy, yx, yy}_{A^2}, \dots \}$$

$$A^* = \{ \underbrace{\epsilon}_{A^0}, \underbrace{x, y}_{A^1}, \underbrace{xx, xy, yx, yy}_{A^2}, \dots \}$$

例：（语言 $L = \{A \sim Z, a \sim z, _ \}$ $D = \{0 \sim 9\}$ ）

1. $L \cup D = \{A \sim Z, a \sim z, 0 \sim 9\}$

2. LD 所有一字母后跟一数字组成的符号串构成的集合

3. L^4 所有的四个字母的符号串构成的集合

4. $L(L \cup D)^*$ 所有字母打头的字母和数字符号串构成的集合

5. D^+ 所有长度大于等于1的数字串构成的集合

回顾：Chapter 2---文法的引入

- 英语 —> 句子 —> 单词 —> 字母
- 程序设计语言 —> 程序 —> 单词 —> 基本符号
- 基本符号集合：构成一个语言的一切基本符号组成一个集合
- 字母表（符号集合或基本符号集合） Σ
 - 字母表是符号的 非空有穷 集合
- 符号（元素）
- 符号串： Σ 上的有穷序列
 - 符号串中符号的顺序很重要
 - 空串 ε

回顾：Chapter 2---文法的引入

- 符号串运算：
 - 连接与方幂
 - 特殊： $a^0 = \varepsilon$
- 符号串集合的运算：
 - 合并、连接、方幂、闭包
 - 特殊1： $V^0 = \{\varepsilon\}$
 - 特殊2： Φ

回顾: Chapter 2

例1: $\Sigma = \{a, b, c\}$, 则_____是 Σ 上的符号串。

例2: 设 $z = abc$,

那么 z 的前缀是_____，

z 的后缀是_____，

z 的长度是_____。

例3: $x = AB$, $x^3 =$ _____

例4: 若 集合 $A = \{ab, cde\}$, $B = \{0, 1\}$

则 $AB =$ _____

回顾: Chapter 2

字母表的闭包 Σ^* (Closure Set) :
 Σ 上的一切符号串组成的集合

字母表的正闭包 Σ^+ (Positive Closure Set):
 Σ 上的除 ϵ 外的所有符号串组成的集合

例5: $\Sigma=\{0\}$, 则 $\Sigma^*=$ _____

$\Sigma=\{a, b\}$, 则 $\Sigma^*=$ _____

例6:多项选择题

Σ^* 的元素的数量 ()

A. 可数的



B. 不可数的



C. 无穷多的

D. 有穷多的

例7：辨析

- Φ

- ε

- $\{\varepsilon\}$

例8：连线练习

英语 —> 句子 —> 单词 —> 字母

程序设计语言 —> 程序 —> 句子 —> 单词

—> 基本符号

字母表

构成语言的基本符号

元素

语言的基本符号集

符号串

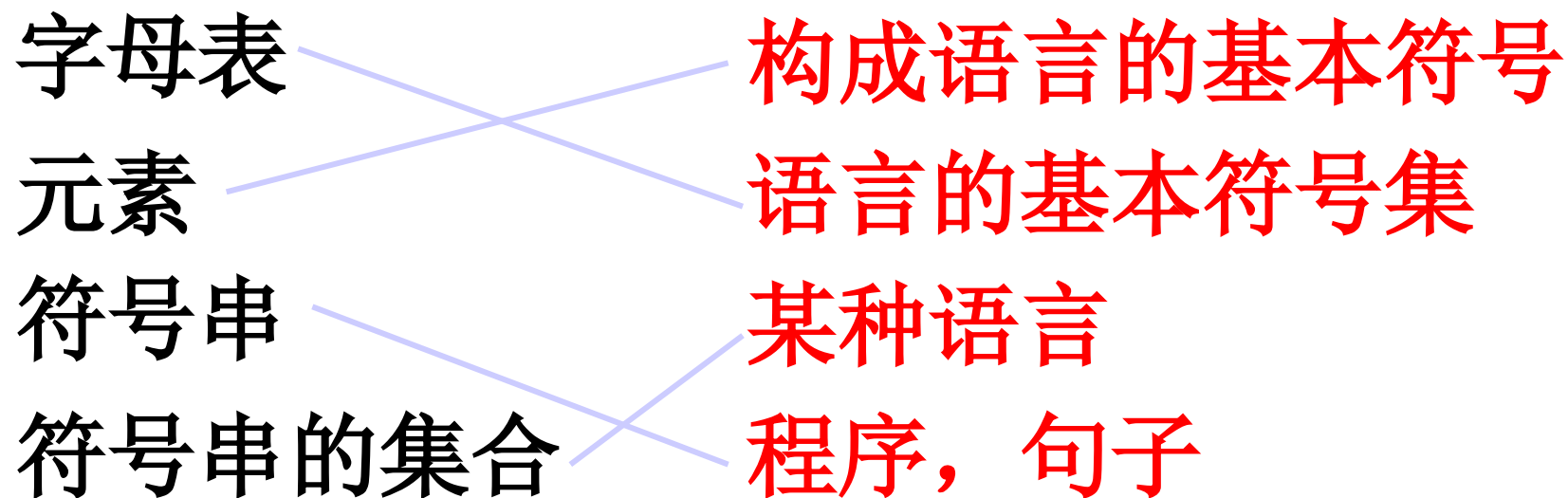
某种语言

符号串的集合

程序，句子

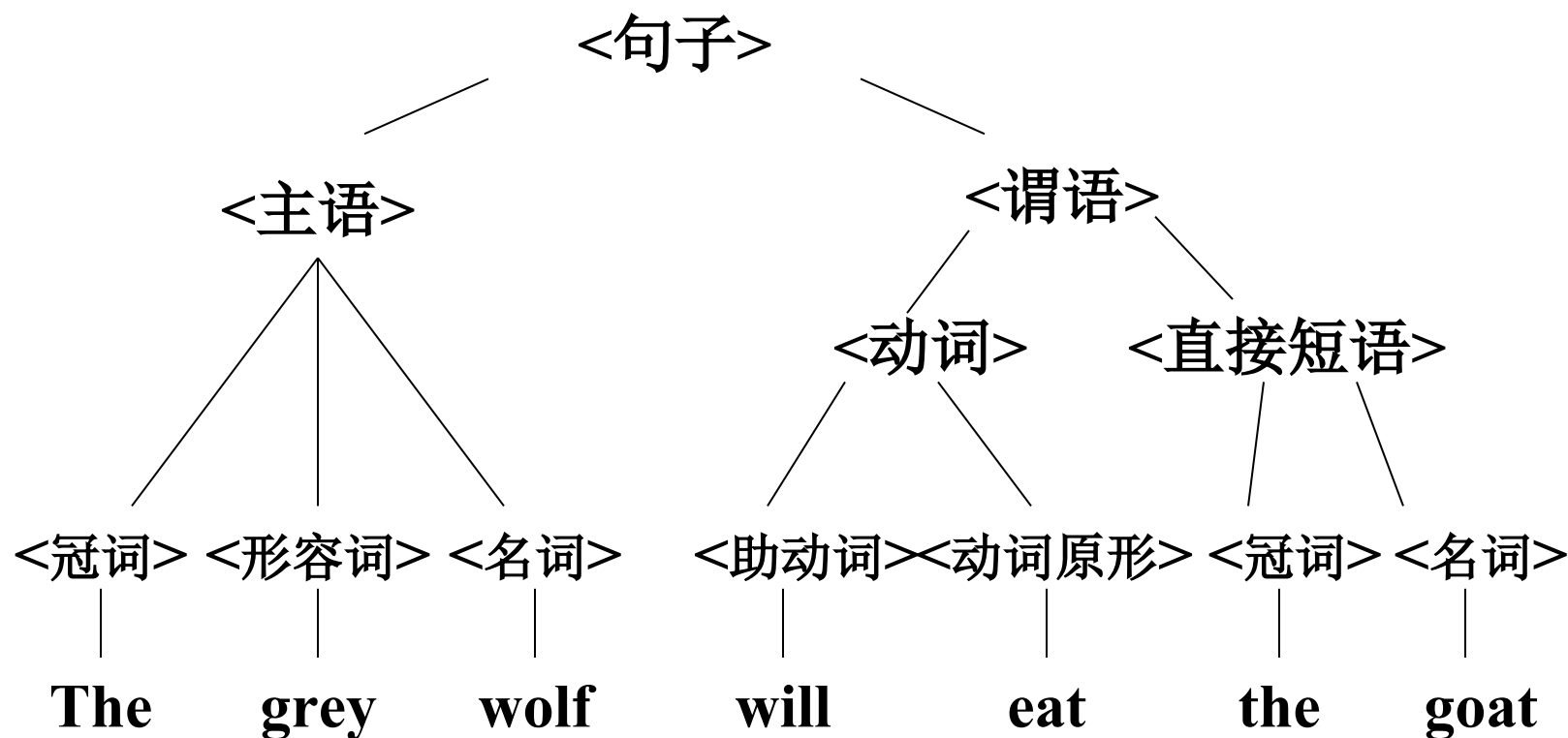
例8：连线练习

英语 —> 句子 —> 单词 —> 字母
程序设计语言 —> 程序 —> 单词 —> 基本符号



2.2 文法的讨论

例：有一英文句子：“The grey wolf will eat the goat.”。这是一个在语法、语义上都正确的句子。如何用语法单位如<句子>、<主语>等推导出该句子？



BNF范式表示法： 如果用符号“ \rightarrow ”(或“ $::=$ ”)表示“定义为”，用符号“|”表示“或”， $\langle \rangle$ 表示语法成分

$\langle \text{句子} \rangle \rightarrow \langle \text{主语} \rangle \langle \text{谓语} \rangle$ (1)

$\langle \text{主语} \rangle \rightarrow \langle \text{冠词} \rangle \langle \text{形容词} \rangle \langle \text{名词} \rangle$ (2)

$\langle \text{冠词} \rangle \rightarrow \text{the}$ (3) $\langle \text{形容词} \rangle \rightarrow \text{grey}$ (4)

$\langle \text{谓语} \rangle \rightarrow \langle \text{动词} \rangle \langle \text{直接宾语} \rangle$ (5)

$\langle \text{动词} \rangle \rightarrow \langle \text{助动词} \rangle \langle \text{动词原形} \rangle$ (6)

$\langle \text{助动词} \rangle \rightarrow \text{will}$ (7) $\langle \text{动词原形} \rangle \rightarrow \text{eat}$ (8)

$\langle \text{直接宾语} \rangle \rightarrow \langle \text{冠词} \rangle \langle \text{名词} \rangle$ (9)

$\langle \text{名词} \rangle \rightarrow \text{wolf}$ (10) $\langle \text{名词} \rangle \rightarrow \text{goat}$ (11)

$\langle \text{名词} \rangle \rightarrow \text{wolf} | \text{goat}$

由规则推导句子：有了一组规则之后，可以按照一定的方式用它们去**推导或产生句子**。

推导方法：从一个要识别的符号开始推导，即用相应规则的**右部**来替代规则的**左部**，每次仅用一条规则去进行推导。

<句子> \Rightarrow <主语><谓语>

\Rightarrow <冠词> <形容词> <名词> <谓语>

.....

这种推导一直进行下去，直到所有带< >的符号都由终结符号替代为止。

推导句子所需的四部分

- 1、终结符号集 $V_T = \{\text{the, gray, wolf, will, eat, goat}\}$
- 2、非终结符号集 $V_N = \{\langle \text{句子} \rangle, \langle \text{主语} \rangle, \langle \text{谓语} \rangle, \langle \text{冠词} \rangle, \langle \text{形容词} \rangle, \langle \text{名词} \rangle, \langle \text{动词} \rangle, \langle \text{直接宾语} \rangle, \langle \text{助动词} \rangle, \langle \text{动词原形} \rangle\}$
- 3、语法规则集 $P = \{\langle \text{句子} \rangle \rightarrow \langle \text{主语} \rangle \langle \text{谓语} \rangle, \dots\}$
- 4、开始符号 $S = \langle \text{句子} \rangle$

2.3 文法和语言的形式定义

- 一. 文法的形式定义
- 二. 直接推导和推导
- 三. 语言的形式定义
- 四. 短语、直接短语、句柄

一. 文法和语言的形式定义

定义1.1: 一个上下文无关文法G是一个四元组 $G = (V_T, V_N, S, P)$ ，其中：

1. V_T 是一个非空有穷终结符号集合；
2. V_N 是一个非空有穷的非终结符号集合，
且 $V_T \cap V_N = \Phi$ ，表示一类具有某种性质的符号
3. $S \in V_N$ 开始符号。
4. P 是一个产生式的非空有穷集合，每个产生式的形式是 $A \rightarrow \alpha$ ，其中 $A \in V_N$ ， $\alpha \in (V_T \cup V_N)^*$
开始符号S至少必须在某个产生式的左部出现一次

$(V_T \cup V_N)^*$ 表示什么集合？

一般情况下（缺省）符号指称：

- 1、 A, B, C 等，表示非终结符号
- 2、 a, b, c, d 等，表示终结符号
- 3、 α, β, γ 等，表示文法符号串（终结符号和非终结符号组成的符号串）

例 简单算术表达式的文法G

$G = (\{a, +, *, (,)\}, \{ \langle \text{表达式} \rangle, \langle \text{项} \rangle, \langle \text{因子} \rangle \}, \langle \text{表达式} \rangle, P)$

$P: \langle \text{表达式} \rangle \rightarrow \langle \text{表达式} \rangle + \langle \text{项} \rangle \mid \langle \text{项} \rangle$

$\langle \text{项} \rangle \rightarrow \langle \text{项} \rangle * \langle \text{因子} \rangle \mid \langle \text{因子} \rangle$

$\langle \text{因子} \rangle \rightarrow (\langle \text{表达式} \rangle) \mid a$

(用E、T、F分别代替 $\langle \text{表达式} \rangle$ 、 $\langle \text{项} \rangle$ 、 $\langle \text{因子} \rangle$)

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

二. 直接推导和推导

令 $G = (V_T, V_N, S, P)$, $S \Rightarrow \alpha A \beta$; 若 $A \rightarrow \gamma \in P$, 且 $\alpha, (V_T \cup V_N)^*$ 称 $\alpha A \beta$ **直接推出** $\alpha \gamma \beta$, 表示成 $\alpha A \beta \Rightarrow \alpha \gamma \beta$ 同时也称 $\alpha \gamma \beta$ 是 $\alpha A \beta$ 的**直接推导**, 或称 $\alpha \gamma \beta$ **直接归约**到 $\alpha A \beta$

如果存在一个直接推导序列:

$$\alpha_0 \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_n \quad (n > 0)$$

表示成 $\alpha_0 \xRightarrow{+} \alpha_n$, 称从 α_0 到 α_n 的**长度为n的推导**。

$\alpha_0 \xRightarrow{*} \alpha_n$ 表示从 α_0 出发, 经0步或若干步, 可推导出 α_n ($\alpha_0 = \alpha_n$ 或 $\alpha_0 \xRightarrow{+} \alpha_n$)

$$E \rightarrow E+T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid a$$

推导： $E \Rightarrow E+T \Rightarrow T+T \Rightarrow F+T$
 $\Rightarrow a+T \Rightarrow a+F \Rightarrow a+a$

最左推导和最右推导

例如 $E \Rightarrow E+T \Rightarrow T+T \Rightarrow F+T \Rightarrow a+T \Rightarrow a+T * F$
 $\Rightarrow a+F * F \Rightarrow a+a * F \Rightarrow a+a * a$

特点: $\alpha A \beta \Rightarrow \alpha \gamma \beta$ ($A \rightarrow \gamma$), $\alpha \in V_T^*$ (最左推导)

每一步推导都是对最左非终结符号进行替换

$E \Rightarrow E+T \Rightarrow E+T * F \Rightarrow E+T * a \Rightarrow E+F * a$
 $\Rightarrow E+a * a \Rightarrow T+a * a \Rightarrow F+a * a \Rightarrow a+a * a$

特点: $\alpha A \beta \Rightarrow \alpha \gamma \beta$ ($A \rightarrow \gamma$), $\beta \in V_T^*$ (最右推导)

每一步推导都是对最右非终结符号进行替换, 也称规范推导, 其归约称为规范归约

三. 语言的形式定义

设文法 $G = (V_T, V_N, S, P)$ 。如果 $S \xRightarrow{*} \alpha$, 则称 α 是一个**句型**。仅含终结符号的句型是一个**句子**。

语言 $L(G)$ 是由文法 G 产生的所有句子所组成的集合:

$$L(G) = \{ \alpha \mid S \xRightarrow{+} \alpha \text{ 且 } \alpha \in V_T^* \}$$

例子: 一个文法 $G = (\{a, b\}, \{S\}, S, P)$ 其中 P :

$$S \rightarrow aSb \mid ab$$

$$\begin{aligned} \text{则} \quad S &\Rightarrow aSb \Rightarrow aaSbb \Rightarrow a^3Sb^3 \\ &\Rightarrow \dots \Rightarrow a^{n-1}Sb^{n-1} \Rightarrow a^n b^n \end{aligned}$$

总结：语言、字母表、字母表闭包

- 语言是由句子组成的集合
- 字母表 Σ 上的一个语言是 Σ 上的一些符号串的集合
- 字母表的闭包 Σ^* 是 Σ 上的一切符号串组成的集合
- 字母表 Σ 上的每个语言是 Σ^* 的一个子集

例： 以下是字母表 $\Sigma=\{a,b\}$ 上的语言

- $A=\{ab,aabb,aaabbb,\dots,a^n b^n,\dots\}$

$$\{w|w \in \Sigma^* \text{ 且 } w=a^n b^n, n \geq 1\}$$

- $B=\{a,aa,aaa,\dots\}$

$$\{w|w \in \Sigma^* \text{ 且 } w=a^n, n \geq 1\}$$

关于语言的补充

- $\{\varepsilon\}$ 是一个语言。 Φ 是一个语言。
- 有关语言的运算： 既然将语言定义为一个集合，那么有关集合的运算也适合语言
 - 设 L 是(Σ 上的)一个语言，
 M 是(Σ 上的)一个语言，
则语言 L 和 M 的并，交，差，补是一个语言

课堂练习1：文法**G**为：

$$N \rightarrow D|ND$$

$$D \rightarrow 0|1|2|3|4|5|6|7|8|9$$

(1) **G**的语言是什么？

(2) 给出句子**0127**、**34**和**568**的最左和最右推导。

课堂练习1：文法G为：

$$N \rightarrow D | ND$$

$$D \rightarrow 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$$

(1) G的语言是什么？

(2) 给出句子0127、34和568的最左和最右推导。

G的语言是：0~9的数字组成的任意非空串

$$L(G) = \{x | x \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^+\}$$

课堂练习2： 写一文法，使其语言是奇数集。
要求不以0打头。

简单的情况：一位

复杂的情况：分三部分

末尾：以1|3|5|7|9结尾

开头：除了0的任意数字

中间部分：空或者任意数字串

课堂练习2: 写一文法, 使其语言是奇数集。
要求不以0打头。

(一位): $D \rightarrow 1|3|5|7|9$

复杂的情况: 分三部分

末尾: 以1|3|5|7|9结尾

$D \rightarrow 1|3|5|7|9$

开头: 除了0的任意数字

$B \rightarrow 2|4|6|8|D$

中间部分: 空或者任意数字串

$C \rightarrow CA|\epsilon$

所以题目要求的文法G[N]可以写成:

$A \rightarrow 0|B$

$N \rightarrow BCD|D$

$D \rightarrow 1|3|5|7|9$

$B \rightarrow 2|4|6|8|D$

$C \rightarrow CA|\epsilon$

$A \rightarrow 0|B$

2.4 分析树(语法树) 和二义性

- 一. 分析树的定义
- 二. 画出分析树
- 三. 子树
- 四. 二义性文法的定义

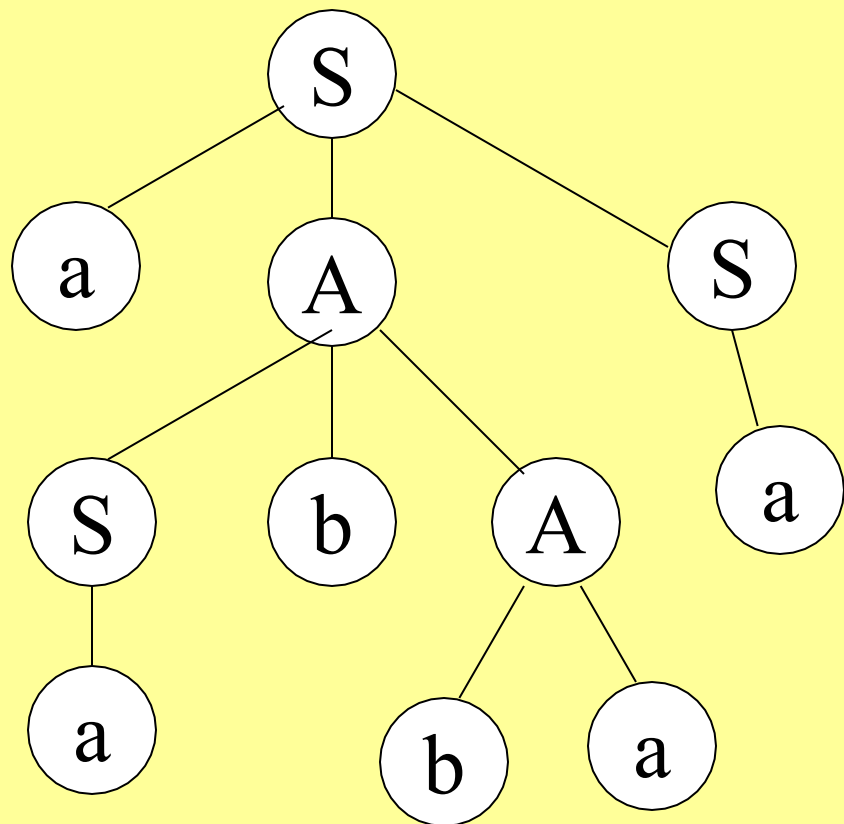
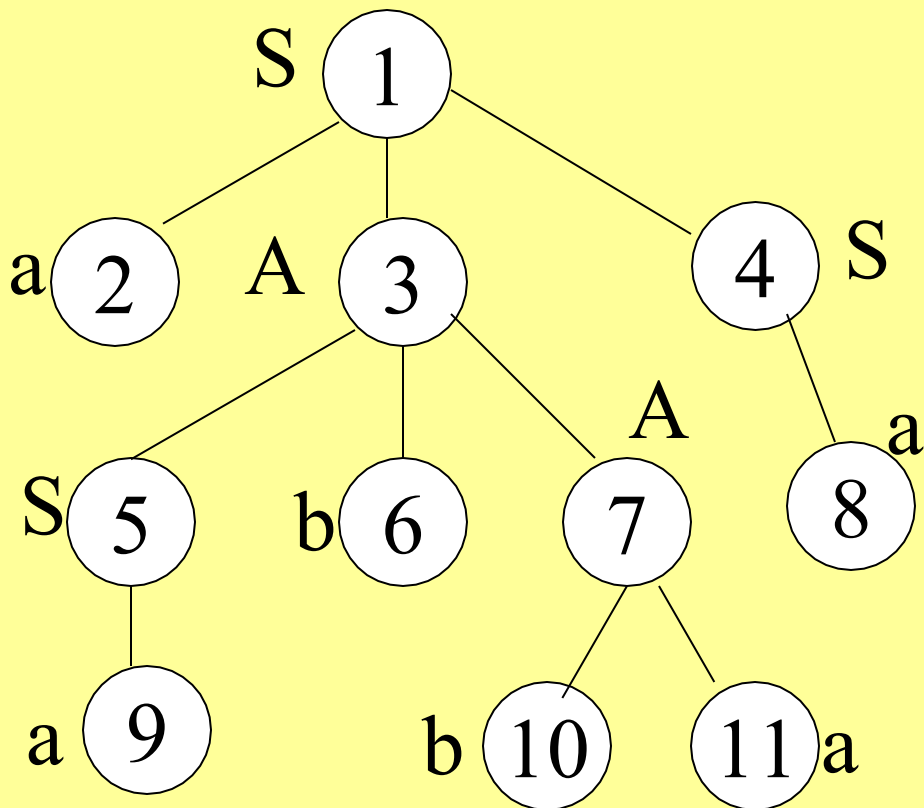
一. 分析树（语法树）的定义

设 $G = (V_T, V_N, S, P)$ 是一个上下文无关文法, G 的一棵**分析树**应满足如下条件:

1. 每个结点有个**标记**, 是 $V_T \cup V_N \cup \{ \epsilon \}$ 中的符号
2. 根的标记是 S
3. 如果结点是**内部结点**, 则其标记 **A 必在 V_N 中**
4. 如果编号为 n 的结点其标记为 A , n_1, n_2, \dots, n_k 是结点 n 的从左到右的儿子, 并分别有标记 X_1, X_2, \dots, X_k , 则 $A \rightarrow X_1 X_2 \dots X_k$ 必须是 P 的产生式
5. 如果结点 n 有标记 ϵ , 那么结点 n 是叶子, 且是它父亲唯一的儿子, 其他叶子结点是终结符号

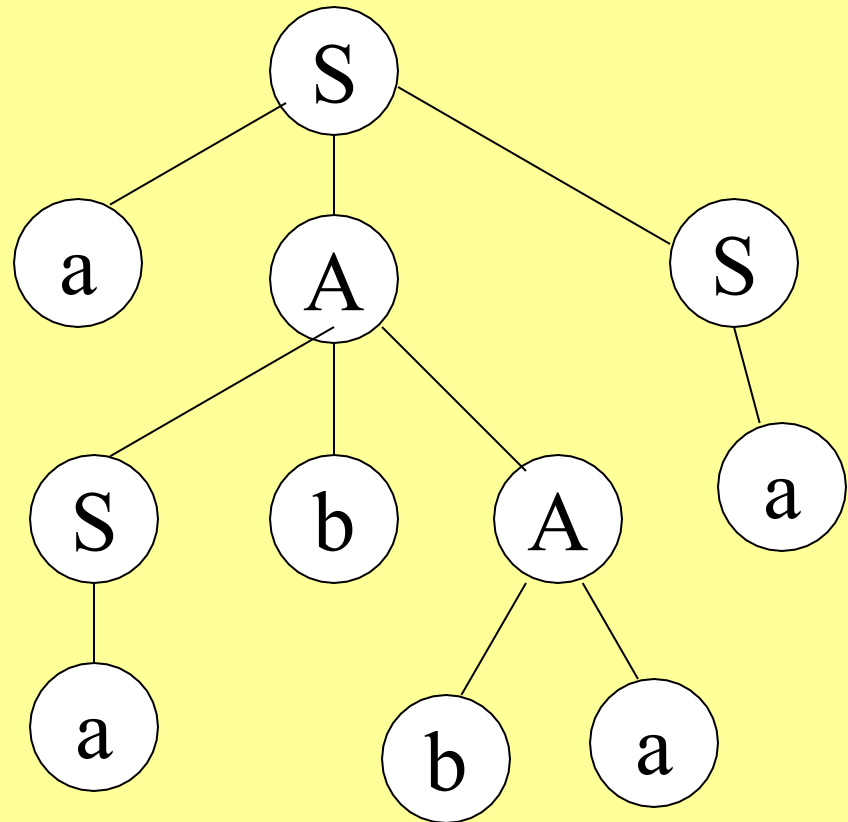
$G(S): (1) S \rightarrow aAS \mid a \quad (2) A \rightarrow SbA \mid SS \mid ba$

句子aabbbaa的分析树

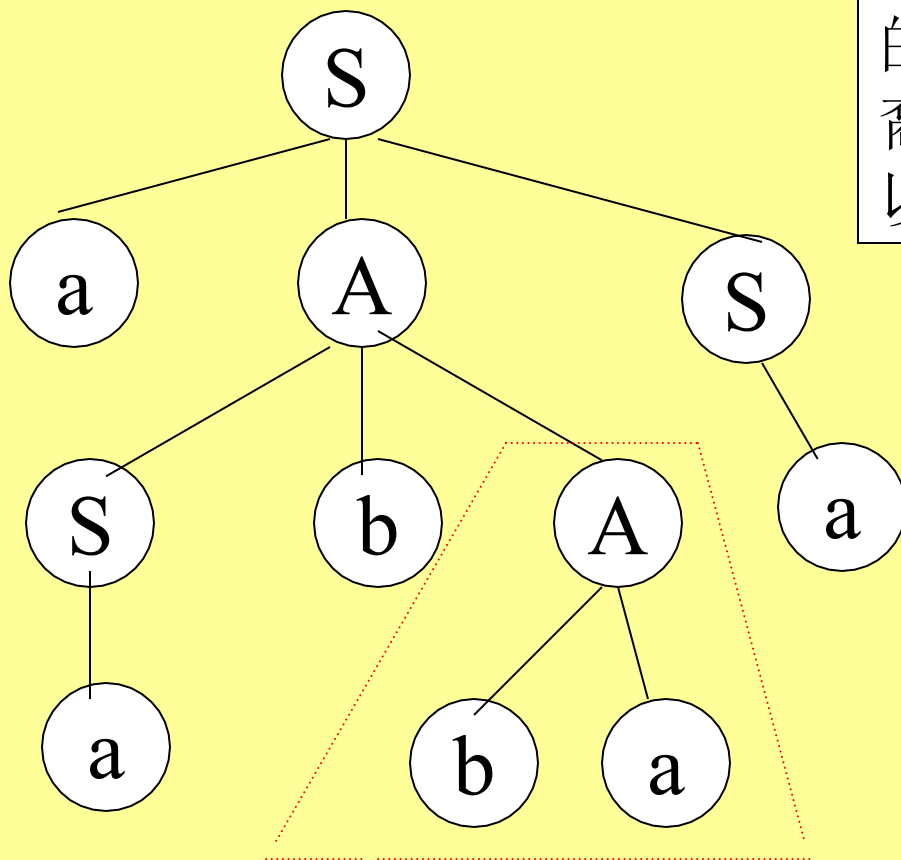


二. 画分析树 (自顶向下)

S
$\Rightarrow aAS$
$\Rightarrow aSbAS$
$\Rightarrow aabAS$
$\Rightarrow aabbaS$
$\Rightarrow aabbbaa$



三. 子树



一棵分析树中一个特有的结点连同它的全部后裔，连接这些后裔的边以及这些结点的标记

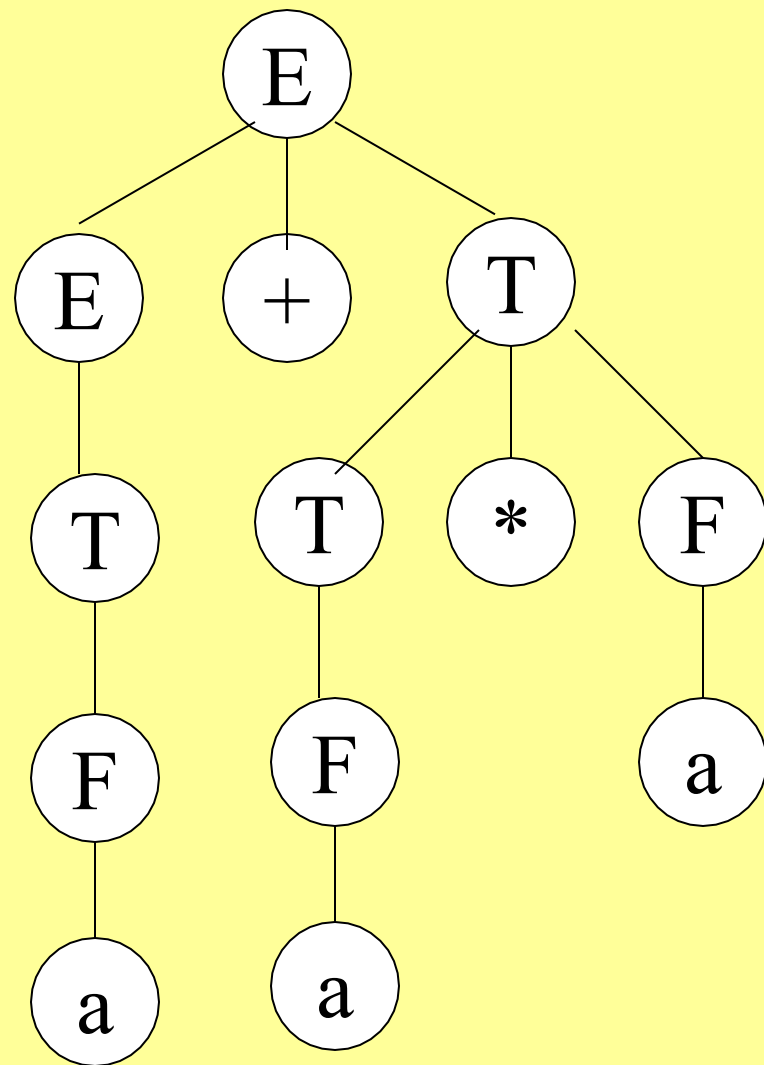
对表达式文法G和句子a+a*a, 给出最左推导过程
语法树

$G = (\{a, +, *, (,)\}, \{ \langle \text{表达式} \rangle, \langle \text{项} \rangle, \langle \text{因子} \rangle \},$
 $\langle \text{表达式} \rangle, P)$

P: (用E、T、F分别代替 $\langle \text{表达式} \rangle$ 、 $\langle \text{项} \rangle$ 、 $\langle \text{因子} \rangle$)

$$\begin{array}{lcl} E \rightarrow E+T & | & T \\ T \rightarrow T * F & | & F \\ F \rightarrow (E) & | & a \end{array}$$

例



E

$\Rightarrow \underline{E+T}$

$\Rightarrow \underline{T+T}$

$\Rightarrow \underline{F+T}$

$\Rightarrow \underline{a+T}$

$\Rightarrow \underline{a+T}*F$

$\Rightarrow \underline{a+F}*F$

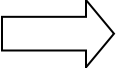
$\Rightarrow \underline{a+a}*F$

$\Rightarrow \underline{a+a}*a$

给定文法 $G=(\{a,b,c,d,e\},\{S,A,B\},S,P)$
其中P:

$S \rightarrow aAcBe$ $A \rightarrow b$ $A \rightarrow Ab$ $B \rightarrow d$

给出句子 $abbcde$ 的最右推导过程。

S  **aAcBe**

 **aAcde**

 **aAbcde**

 **abbcde**

四. 文法的二义性（ambiguity）的定义

引例

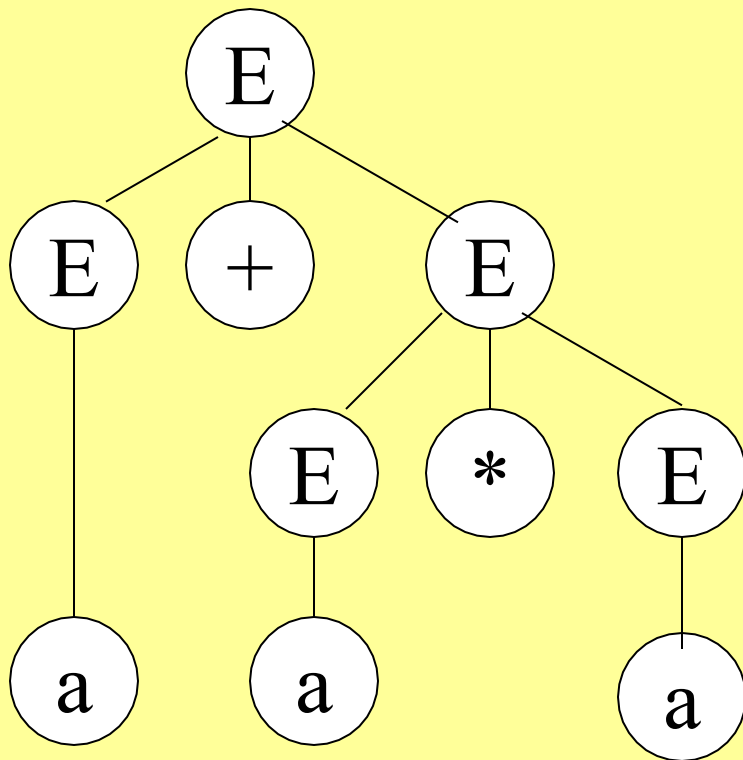
文法 G 产生式如下：

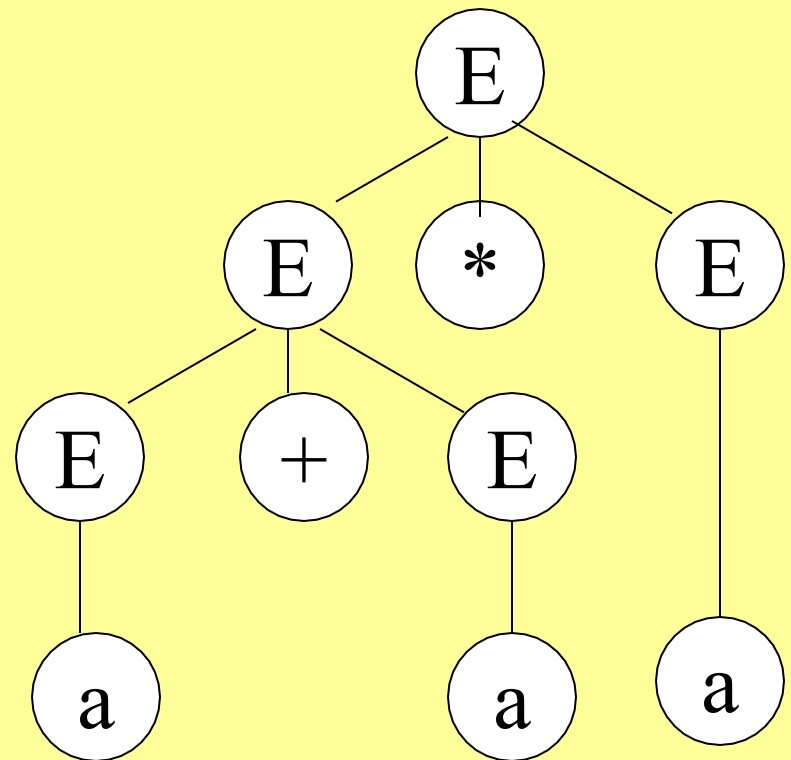
$$E \rightarrow E + E \mid E * E \mid (E) \mid a$$

对于句子 $a + a * a$, 有如下两个最左推导：

$$E \Rightarrow E + E \Rightarrow a + E \Rightarrow a + E * E \Rightarrow a + a * E \Rightarrow a + a * a$$

$$E \Rightarrow E * E \Rightarrow E + E * E \Rightarrow a + E * E \Rightarrow a + a * E \Rightarrow a + a * a$$

$$\begin{aligned}
 E &\Rightarrow E + E \Rightarrow a + E \\
 &\Rightarrow a + E * E \Rightarrow a + a * E \\
 &\Rightarrow a + a * a
 \end{aligned}$$


$$\begin{aligned}
 E &\Rightarrow E * E \Rightarrow E + E * E \\
 &\Rightarrow a + E * E \Rightarrow a + a * E \\
 &\Rightarrow a + a * a
 \end{aligned}$$


如果一个文法的句子**存在两棵语法树**, 则称该句子是二义性的; **换言之**, 无二义性文法的句子**只有一棵语法树**, 尽管推导过程可以不同。

如果一个文法**包含二义性句子**, 则称这个文法是二义性的; 否则, 该文法是无二义性的。

不存在一个算法, 它能在有限步骤内, 确切地判定一个文法是否是二义的; 但能给出一组充分条件, 满足这组充分条件的文法是无二义性的。

在能驾驭的情况下, 可以使用二义性文法. 文法的二义性和语言语义的二义性不是相同概念

课堂练习

考虑文法

$S \rightarrow aSbS \mid bSaS \mid \epsilon$

- (a) 为句子abab构造两个不同的最左推导，
以此说明该文法是二义的。
- (b) 为abab构造对应的最右推导。
- (c) 为abab构造对应的分析树。
- (d) 这个文法产生的语言是什么？

Answer:

(a) $S \Rightarrow lm \ aSbS$
 $\Rightarrow lm \ abS$
 $\Rightarrow lm \ abaSbS$
 $\Rightarrow lm \ ababS$
 $\Rightarrow lm \ abab \text{-----} \textcircled{1}$

$S \Rightarrow lm \ aSbS$
 $\Rightarrow lm \ abSaSbS$
 $\Rightarrow lm \ abaSbS$
 $\Rightarrow lm \ ababS$
 $\Rightarrow lm \ abab \text{-----} \textcircled{2}$

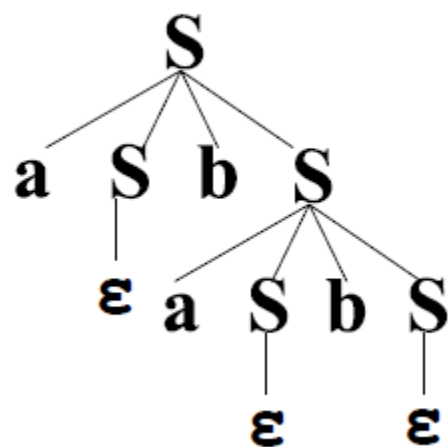
可知，对于句子 abab 存在两个不同的最左推导，所以该文法是二义的

(b)

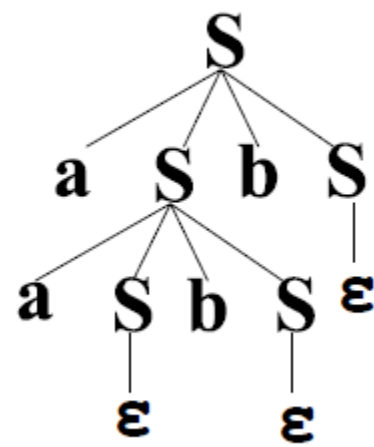
$S \Rightarrow rm \ aSbS$
 $\Rightarrow rm \ aSb$
 $\Rightarrow rm \ abSaSb$
 $\Rightarrow rm \ abSab$
 $\Rightarrow rm \ abab \text{-----} \textcircled{3}$

$S \Rightarrow rm \ aSbS$
 $\Rightarrow rm \ aSbaSbS$
 $\Rightarrow rm \ aSbaSb$
 $\Rightarrow rm \ aSbab$
 $\Rightarrow rm \ abab \text{-----} \textcircled{4}$

(c) ①④对应的分析树:



②③对应的分析树:



(d) 该文法产生 a、b 个数相等的 ab 串 (含空串)

2.5 形式语言概观

N. Chomsky把文法分为**四种类型**，即**0型**、**1型**、**2型**、**3型**。差别在于对产生式施加了不同限制

0型： $G = (V_T, V_N, S, P)$

规则形式： $\alpha \rightarrow \beta$ $\alpha, \beta \in (V_T \cup V_N)^*$, $\alpha \neq \varepsilon$

0型文法产生的语言称为0型语言

1型（上下文有关）： $G = (V_N, V_T, S, P)$

产生式形式 $\alpha A \beta \rightarrow \alpha \gamma \beta$

$A \in V_N$, $\alpha, \beta \in (V_T \cup V_N)^*$, $\gamma \in (V_T \cup V_N)^+$

（仅 $S \rightarrow \varepsilon$ 除外，但此时S不得出现在任何产生式的右部），则称文法G为1型文法或上下文有关文法。

2型（上下文无关）：规则形式： $A \rightarrow \beta$

$$A \in V_N, \beta \in (V_T \cup V_N)^*$$

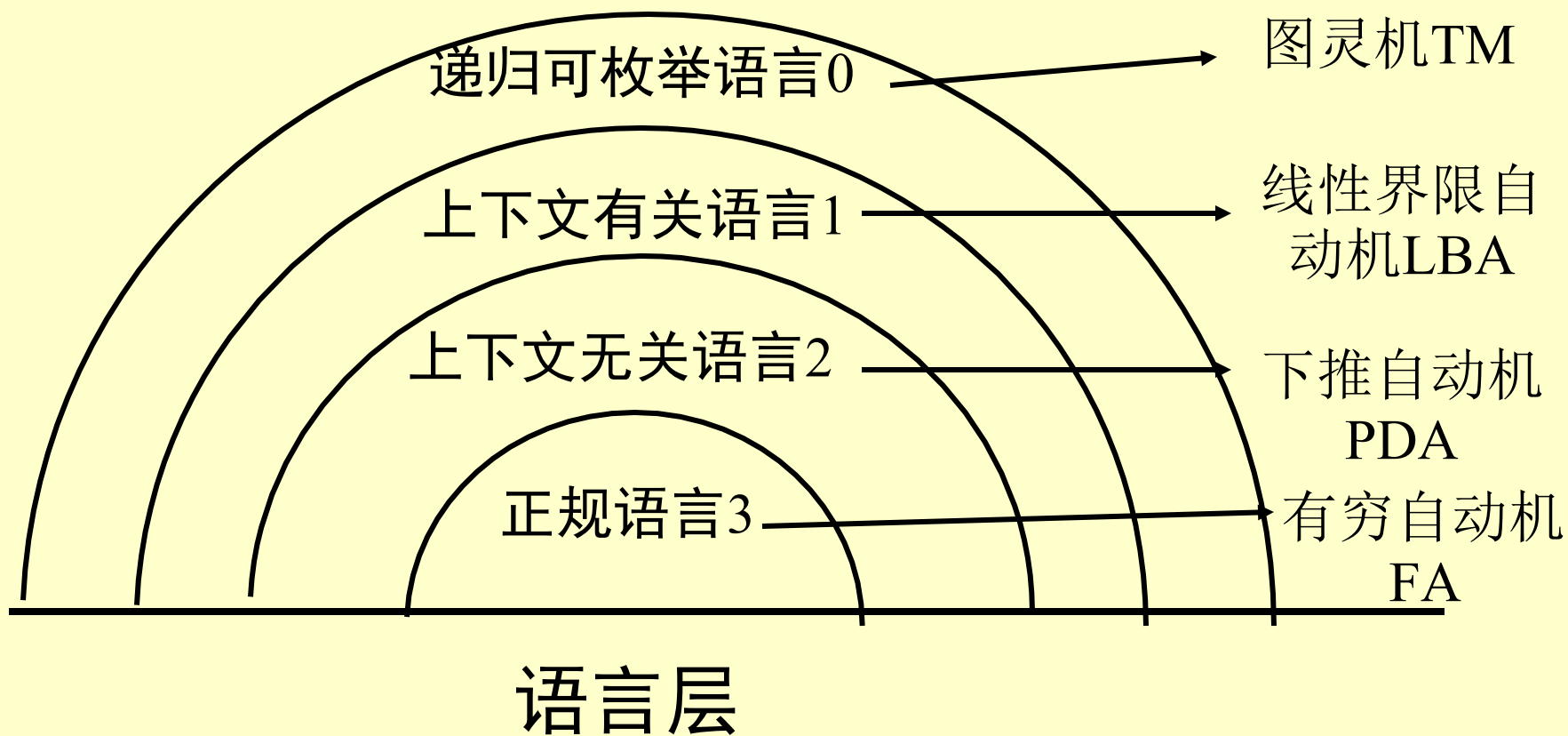
2型文法产生的语言称为**2型语言（上下文无关）**

3型（正规文法）：左线性和右线性文法

$A \rightarrow aB$ 或 $A \rightarrow a$ （右线性） $A, B \in V_N, a \in V_T \cup \{\epsilon\}$

$A \rightarrow Ba$ 或 $A \rightarrow a$ （左线性）

3型文法产生的语言称为**3型语言（正规语言）**



小结

- 掌握符号串和符号串集合的运算、文法和语言的定义
- 几个重要概念：分析树（语法树）、文法的二义性。
- 了解文法和语言的分类。