预备知识

- 用一组数学符号和规则来描述语言的方式称为**形式描述**，所用的数学符号和规则称为**形式语言**
- 文法：程序语言的生成系统。描述语言的语法结构的形式规则，解决语言的有穷说明问题（根据有限条规则产生全部无限多句子）。包含对语法的描述，但不表达任何语义。
    - 词法：正规文法（正则文法）。3型文法。有限自动机。
    - 语法：上下文无关文法。2型文法。下推自动机。
    - 语义：上下文有关文法。1型文法。线性限界自动机。
    - 0型文法，短语结构文法。
- 自动机：程序语言的识别系统。以某字母表上的符号串为输入，**判别**该符号串是否为所描述语言的句子。
- 字母表：符号的非空有穷集合。
- 符号串：由字母表中的符号所组成的**任何有穷序列**被称之为该字母表上的符号串。空串记作ϵ
- 句子：字母表上符合某种规则构成的串
- 语言：句子的集合
- 方幂：$x^0=\epsilon, x^1=x, x^2=xx\cdots, x^n=x^{n-1}x$。例如$x$可以$=ba$
- L^+：语言L的正闭包。$L^+=L^1\cup L^2\cup L^3\cdots$
- L^*：语言的自反闭包。$L^+=L^0\cup L^1\cup L^2\cup L^3\cdots$

文法和语言的形式定义

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：

- V_T是一个非空有穷终结符号集合。
- V_N是一个非空有穷的非终结符号集合。
- $S\in V_N$是开始符合
- P是一个产生式的非空有穷集合，每个产生式的形式是$A\to\alpha, A\in V_N, \alpha\in(V_T\cup V_N)^*$。开始符号$S$至少必须在某个产生式的左部出现一次。

语言是由文法G产生的所有句子所组成的集合。

- 句型：如果$S\overset{*}{\Rightarrow}\alpha$，则$\alpha$是一个句型。
- 句子：仅含终结符号的句型是一个句子。
- 语言：$L(G)=\{\alpha|S\overset{+}{\Rightarrow}\alpha\&a\in V_T^*\}$

推导和规约

- 最左推导：任何一步$\alpha\to\beta$，都是对α中的**最左非终结符号**进行替换的。最右同理。
- 长度为n的推导：$\alpha_0\Rightarrow\alpha_1\Rightarrow\alpha_2\cdots\Rightarrow\alpha_n, n>0$，表示成$\alpha_0\overset{+}{\Rightarrow}\alpha_n$

  $\alpha_0\overset{*}{\Rightarrow}\alpha_n$表示从$\alpha_0$出发，经过0步或若干步，可推导出$\alpha_n$
- 最右推导=规范推导，最左规约=规范规约

递归

语言无限的前提：文法是递归的。

句型的分析

构造一个算法，判断所给的符号串是否为某文法的句型。

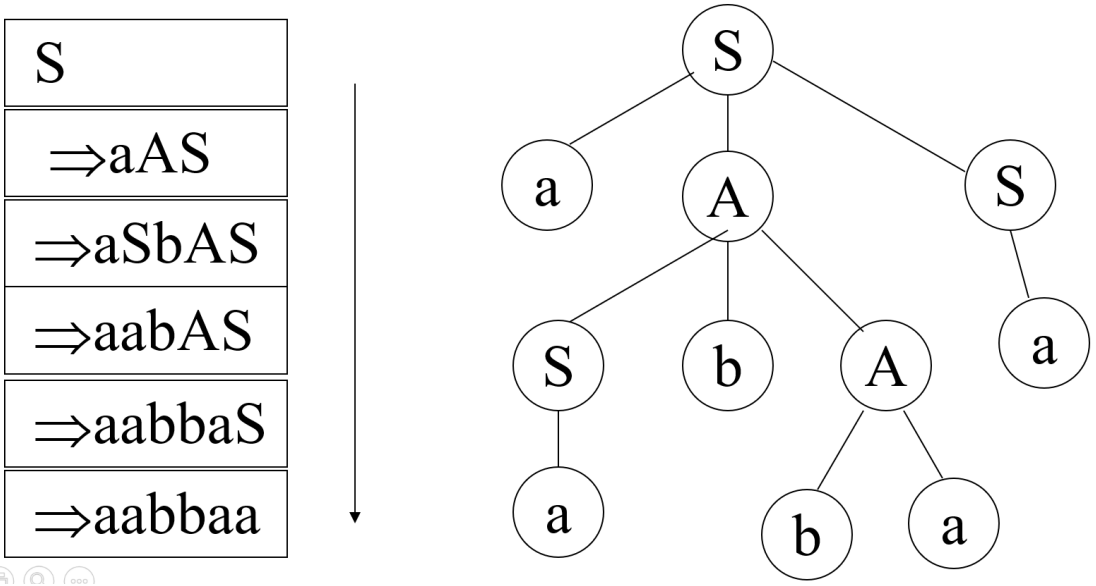
- 自顶向下：从开始符出发，试图推导出给定的符号串。
- 自底向上：推导的逆过程（归约），从已给的符号串出发，试图将其归约为开始符。

语法树

语法树是相对于句子而言的。一个句子才能生成对应的语法树。

- 树的根：开始符号S
- 树的内结点：非终结符A
- 树的叶子节点：文法符号
- 子树：直接推导
- 任一次全剪：句型
- 叶子$\in V_T$时，将叶子顺序排列：句子

G (S): (1) S→aAS | a　(2) A →SbA | SS | ba
句子aabbaa的分析树



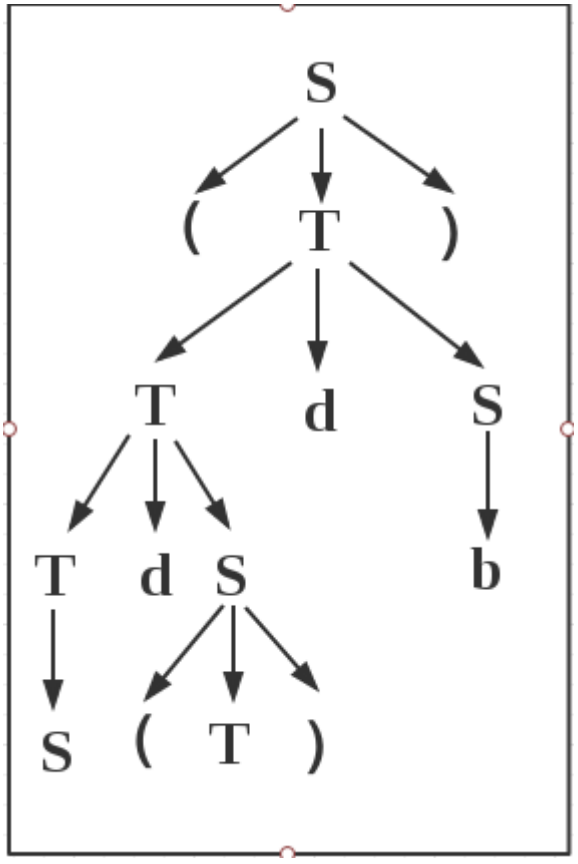
二义性

文法G的某一句子有两棵不同的树/有两个不同的最左或最右推导，则G是二义的。

短语、直接短语、句柄

- 短语：$S \overset{*}{\Rightarrow} \alpha A \delta, A \overset{+}{\Rightarrow} \beta$，则$\beta$是句型$\alpha\beta\delta$相对于非终结符$A$的短语。
    - 语法树最下面叶子节点组合一下就是了，注意要至少包含一个非终结符的全部叶子节点

- 直接短语：$A \Rightarrow \beta$
    - 从一个非终结符出发，能够直接到达的（不会经过其他非终结符）的叶子节点组合
- 句柄：一个句型的最左直接短语



- 短语:S，(T)，b，Sd(T)，Sd(T)db，（Sd(T)db)
- 直接短语:S，(T)，b
- 句柄:S

乔姆斯基四类文法

0型文法

- 短语文法（无限制文法），图灵机
- 产生式形如$\alpha \to \beta$，其中$\alpha \in (V_T \cup V_N)^*$，且至少包含一个非终结符；$\beta \in (V_T \cup V_N)^*$
- 左边有非终结符，右边有终结符。

1型文法

- 上下文有关文法，线性界限自动机
- 产生式形如$\alpha \to \beta$或者$aAb \to arb, |r| > 0$，其中$|\alpha| \leq |\beta|$，仅$\alpha \to \epsilon$除外
- 可以理解为，A在上下文分别为a或b时，可以转化为r。
- 判定
    - 式子左边可以有一个或多个字符，但必须有一个非终结符；
    - 式子右边可以有一个或多个字符，可以是终结符，也可以是非终结符
    - **左边长度必须小于等于右边**，空串除外。

2型文法

- 上下文无关文法，非确定下推自动机
- 产生式形如$A \to \beta$，其中$A \in V_N; \beta \in (V_T \cup V_N)^*$
- 判定
    - 左边必须有且仅有一个非终结符。
    - 2型文法所有产生式的右边可以含有若干个终结符和非终结符

3型文法

- 正规文法，有限自动机。
- 产生式形如：$A \to \alpha B$或$A \to \alpha$ (称为**右线性文法**)

  $A \to B\alpha$或$A \to \alpha$ (称为**左线性文法**)

  其中$\alpha \in V_T^*; A, B \in V_N$
- 注意，要么是左线性，要么是右线性，不能掺杂。产生式的右边非终结符的位置，要么在最左边，要么在最右边，而且最多只能有一个。
- 判定
    - 左边必须只有一个字符，且必须是非终结符；
    - 其右边最多只能有两个字符，要么是一个非终结符+终结符（终结符+非终结符），要么是一个终结符。
    - 对于3型文法中的所有产生式，若其右边有两个字符的产生式，这些产生式右边两个字符中终结符和非终结符的相对位置一定要固定。也就是说如果一个产生式右边的两个字符的排列是：终结符 + 非终结符，那么所有产生式右边只要有两个字符的，都必须满足终结符+非终结符。反之亦然。

从0到3，描述能力越来越弱，限制越来越强。

文法构造

找规律。尝试利用归纳法、递归的思维