描述语言的途径

穷举法

文法（产生式系统）描述

用规则生成语言中合法的句子
形式语言学也称代数语言学，用来精确描述语言及其结构。
形式语法：G=(N, ∑,S,P)

- N 非终结符号有限集
- ∑终结符号有限集
- V 为总词汇表，V=∑∪N，且∑∩N=∅
- P重写规则的有限集合
- S句子符或初始符

最右推导
最左推导

正则文法

P中所有规则满足：A-→Bx或A→x，其中A,B∈N，x∈∑

正则文法也称3型文法
分为：

1. 左线性正则文法
2. 右线性正则文法

eg:G=(N, ∑,S,P),N={S,A,B,C}，∑={a,b},
P:(1) S→aA (2)A→aA (3)A→bbB (4) B→bB (5)B→b
第三条规则可改写为 A→bB' B'→bB

例子属于右线性正则文法（因为规则中非终结符号出现在最右边）

可识别的语言为：
L（G)={a^n *b^m} n>=1,m>=3

上下文无关法CFG

P中所有规则满足：A→α，其中A∈N ，α∈N∪∑
也称2型文法

eg:G=(N, ∑,S,P),N={S,A,B,C}，∑={a,b,c},
P:(1) S→ABC (2)A→aA|a (3)B→bB|b (4) C→BA|c
可识别的语言为：
L（G)={a^n *b^m *a^k *c^α} n>=1,m>=1，k>=0，α∈{0,1}

上下文相关发CSG

P中所有规则满足：αAβ→αγβ，其中A∈N，α，γ，β∈（N∪∑）*，αβ可以是空字符，且γ至少包含一个字符，称G文法为上下文相关法（CSG）或称1型文法。2型文法是1型文法的特例。

规则左部不一定有一个非终结符，规则右端的长度不小于规则左部的长度

eg:G=(N, ∑,S,P),N={S,A,B,C}，∑={a,b,c},

P:(1) S→ABC　(2)A→aA|a　(3)B→bB|b　(4) BC→Bcc

可识别的语言为：
L（G)={a^n *b^m *c^2} n>=1,m>=1

无约束文法

P中所有规则满足：α→β，其中α∈(N∪∑)+，β∈(N∪∑)*,也称0型文法

自动机法

通过对输入的句子进行合法性检验来判断是否是语言中的句子

- 很难描述语言的结构
-

有限自动机FA/FSM

确定性有限自动机DFA

M是一个五元组
M=(∑，Q，δ，q0，F）
∑是输入符号的有穷集合
Q是状态的有限集合
q0∈Q是初始状态
F是终止状态集合，F包含于Q
δ是Q与∑的直积Q*∑到Q（下一个状态）的映射，它支配着有限状态控制的行为，被称为状态转移函数。

DFA确定的语言
δ(q0,x)=p,p∈F，那么句子x被M接受，被M接受的句子的全集称为由M定义的语言，记为T(M):
T(M)={x|δ(q0,x)∈F}

状态转化图构造：
每个状态为节点，用圆圈表示。有向弧表示在输入符号下的转移过程

不确定性有限自动机NFA

M是一个五元组

M=(∑，Q，δ，q0，F）
∑是输入符号的有穷集合
Q是状态的有限集合
q0∈Q是初始状态
F是终止状态集合，F包含于Q
δ是Q与∑的直积Q*∑到Q的幂集2^Q的映射，它支配着有限状态控制的行为，被称为状态转移函数。
不同：DFA中δ(q,a)是一个状态
NFA中δ(q,a)是一个状态集合
对于NFA，M有映射
δ(q,a)={q1,q2,……，qk},k>=1
M在状态q时，接受输入符号a时，可选择状态集中的任何一个状态作为下一个状态

NFA接受的语言：
若存在一个状态p,有p∈δ(q0,x)且p∈F，称句子x被NFA M所接受。被NFA M接受的所有句子的集合称为NFA M定义的语言
T(M)={x|p∈δ(q0,x)且p∈F}
L 是被NFA接受的语言，则存在一个DFA，能够接受L

下推自动机PDA

带有附加下推存储器的有限自动机
下推存储器是一个堆栈
PDA是一个七元组
M=(∑，Q，Γ，δ，q0，Z0，F）
Γ是下推存储器符号的有穷集合
Z0∈Γ是最初出现在下推存储器顶端的开始符号
δ是Q与∑的直积Q*(∑∪{ε})*Γ到Q*Γ的子集的映射。
δ(q,a,Z)={(q1,γ1),(q2,γ2)……，(qm,γm)}
q1,q2……qm∈Q，a∈∑，Z∈Γ，γ1,γ2,……γm∈Γ。

判断一种语言或句子是否被PDA接受标准：
1）终止状态接受标准
T(M)={x|x:(q0,Z0)|--(q,γ),γ∈Γ*,q∈F}
2）空存储器接受标准
N(M)={x|x:(q0,Z0)|--(q,ε),q∈Q}

线性界限自动机

是确定单带图灵机，与图灵机不同之处是其读/写头不能超越原来输入带上字符串的初始和终止位置。
线性界限自动机的存储空间被输入符号串的长度所限制。

六元组
M=(∑，Q，Γ，δ，q0，F）
∑是输入/输出带上字符的有穷集合
Γ是输入符号的有穷集合

F是终止状态集合，Q包含F
δ是Q*Γ到Q*Γ*{R,L,}子集的一个映射。
∑包括#和$分别表示输入链左端和右端的结束标志。
其格局ID与图灵机相同

图灵机T

与FA区别：可通过其读写头改变输入带上的字符。
图灵机T为一个六元组
M=(∑，Q，Γ，δ，q0，F）
∑是输入/输出带上字符的有穷集合，不包含空白符号B
Γ是输入符号的有穷集合，包含空符号B，Γ包含∑，Γ=∑∪{B}
F是终止状态集合，Q包含F
δ是Q*Γ到Q*（Γ-{B}）*{R,L,S}子集的一个映射，{R,L,S}表示左移右移和停止不动。

图灵机的一个格局（ID）可以用三元组（q,α,i)表示。
q∈Q
α是输入/输出带上的非空白部分，α∈（Γ-{B}）
i是整数，表示T的读/写头到α左端（起始位置）的距离。