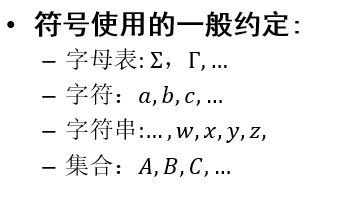
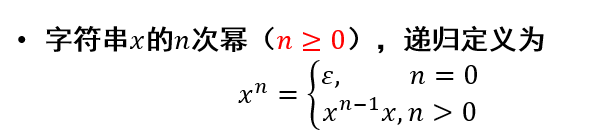
形式语言与自动机复习

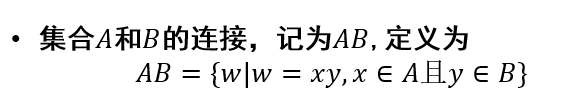
1. 基本概念
   1. 字母表：符号的非空有穷集
   2. 字符串：由字母表中的符号组成的有穷序列
   3. 空串：记为，有0个字符的字符串



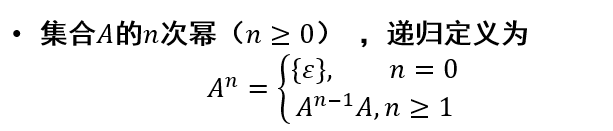
* 1. 字符串的长度：字符串中符号的个数
  2. 字符串x和y的连接：将首尾相接得到的新字符串的运算，记为xy。
  3. 字符串x的n次幂：

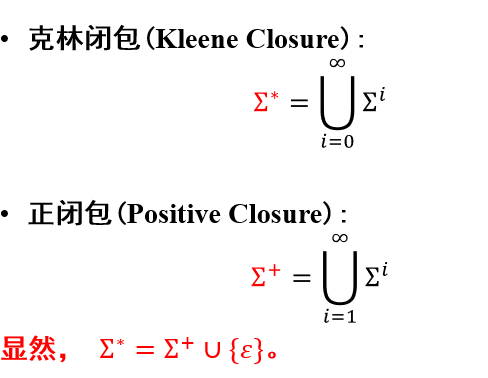


* 1. 集合A和B的连接，记为AB，定义为

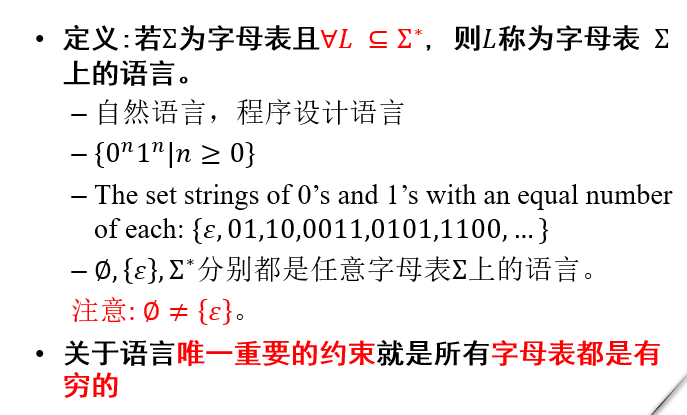


* 1. 集合A的n次幂，定义为：

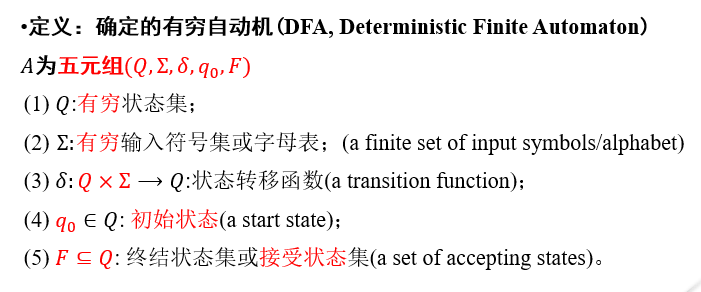




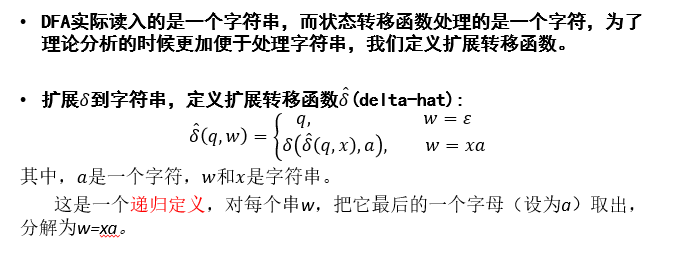
克林闭包是字母表上的所有字符串集合。



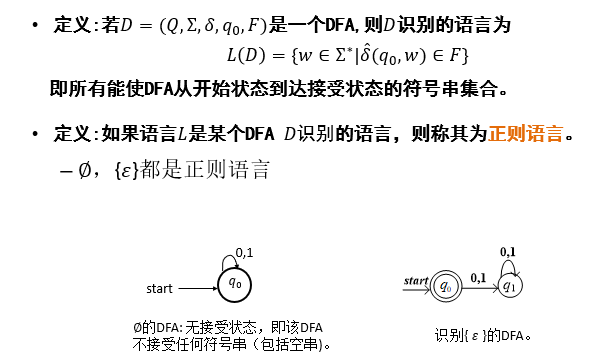
1. 有穷状态自动机
2. DFA



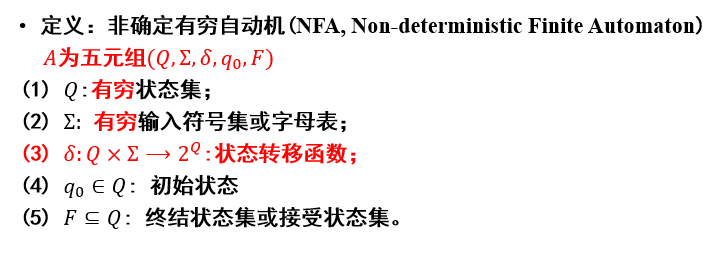
注意：，不为空，需要将所有的转移都写清楚。



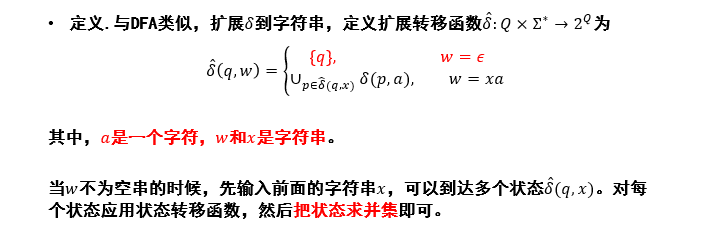
实际上是的映射。

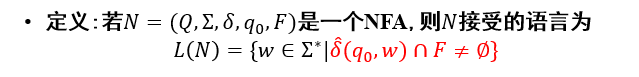


1. NFA



注意：变为，其中是的幂集，包括空集，同时也表明能同时处于多个状态，还能转移到多个状态之上。

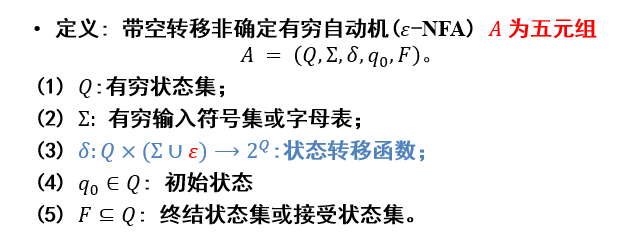




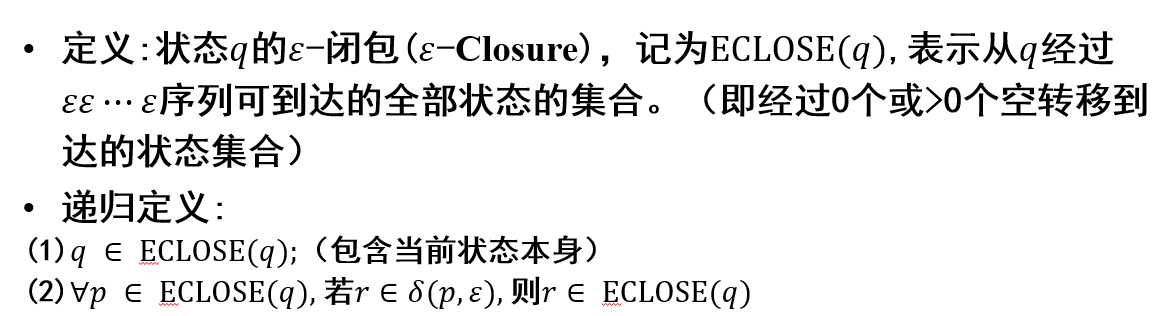
1. DFA和NFA的等价性

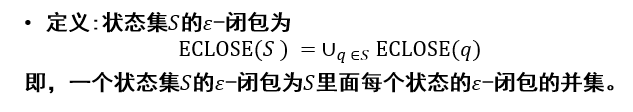
**定理：语言L被NFA接受，当且仅当L也能被DFA接受**

引出方法：**子集构造法**，提供了从NFA构造DFA的一种普遍的方法。



注意：相当于加了读空串时的转移方法。进一步简化了NFA的构造。

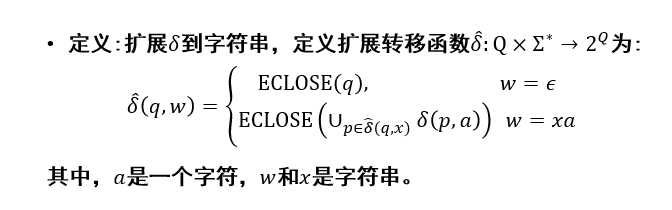


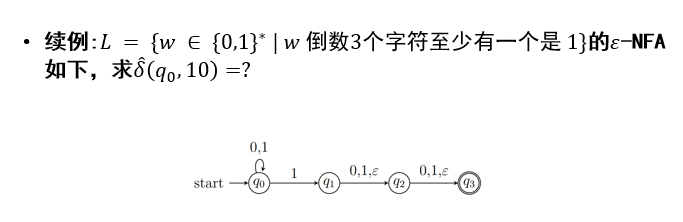


Text

Description automatically generated

在下对进行拓展：



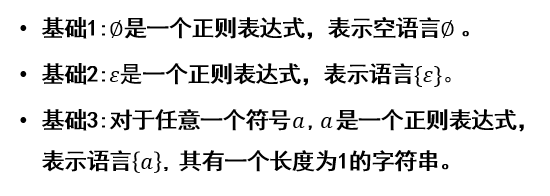


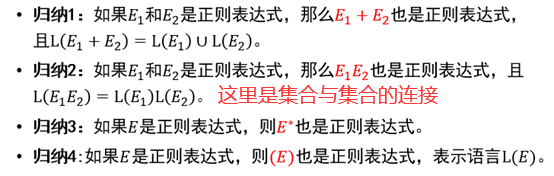
1. 和的等价性

定理：**如果语言𝐿被𝜀-NFA接受,当且仅当𝐿被DFA接受。**

**利用消除空转移的子集构造法构造DFA**

1. 正则表达式
2. 用处：通过代数式表示或产生正则语言，便于正则语言的计算机处理。
3. 递归定义：





示例：1+01\*

1. 有穷自动机和正则表达式

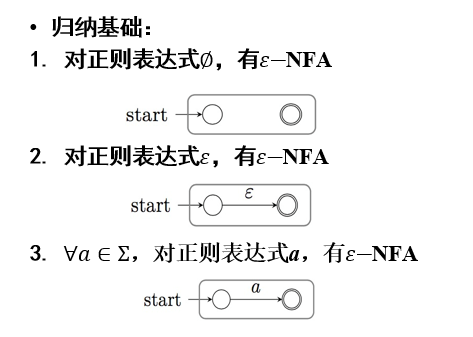
DFA、NFA、和正则表达式表示语言的能力等价。

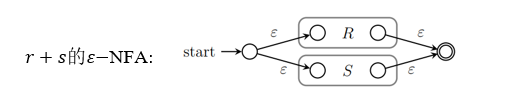
给出的方法：

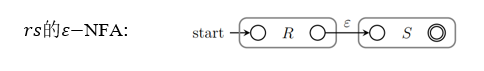
* + 1. 递归法
    2. 状态消除法

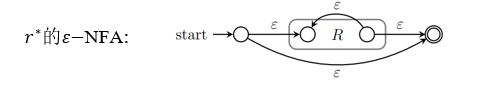
给出的方法

通过递归的方法逐步构建：

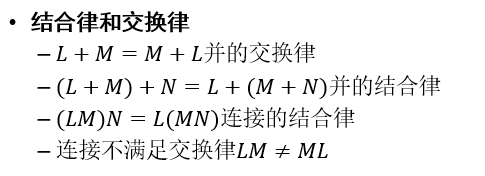


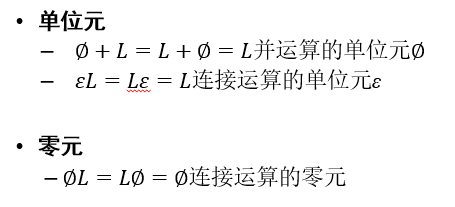




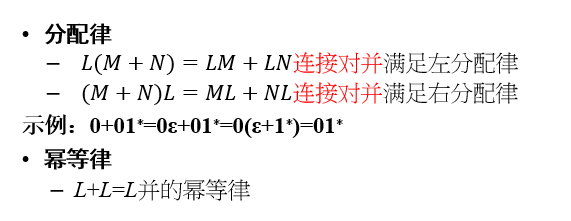


1. 正则表达式的代数运算

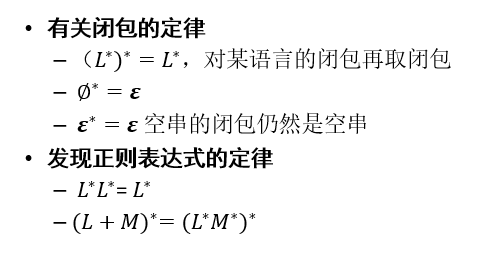




连接对并满足分配律



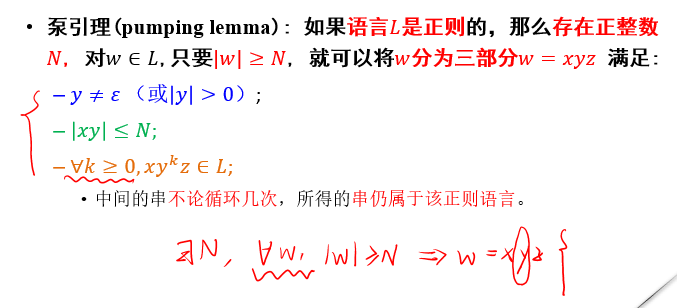
化简时可能用到的定律



1. 正则语言的性质

如何证明一个语言L不是正则语言？

使用泵引理：



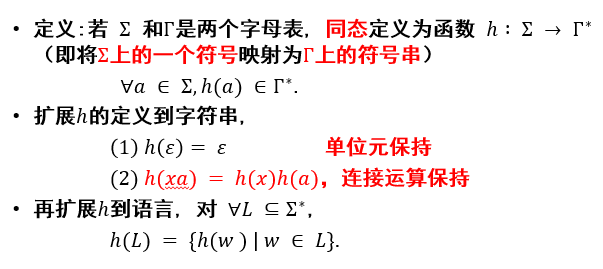
先假设不是正则语言，利用反证法进行证明。泵引理是一个必要条件。

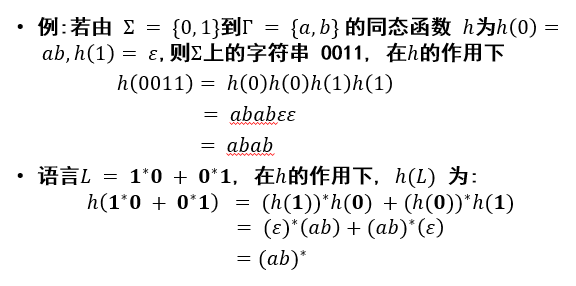
正则语言的封闭性：

正则语言经某些运算后得到的新语言仍然保持正则，称正则语言在这些运算下封闭。

* **正则语言和，在这些运算下封闭:**
  + 并: ，连接: ，闭包: （由定义得证）
  + 交: （由，通过封闭性得证）
  + 反转: （正则表达式归纳得证，或用DFA构造，将其反向后得证）
  + 同态: （正则表达式归纳得证）
  + 逆同态: （构造DFA得证）
  + 补: （将所有终止状态和非终止状态交换，所得的NFA得证）
  + 差: （由，由封闭性得证）

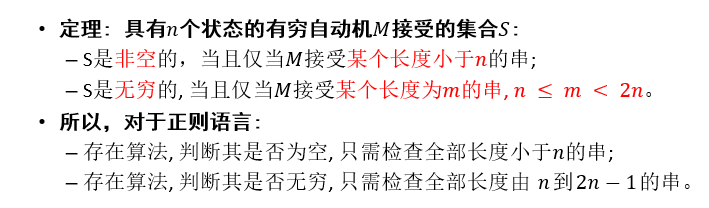
其中，特别注意同态：

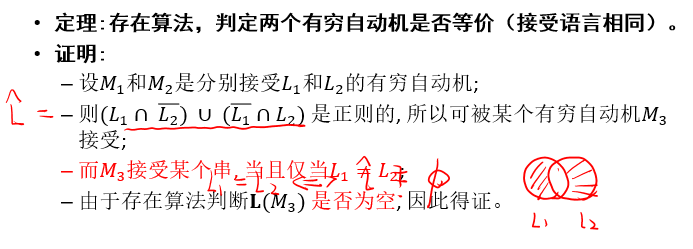




封闭性可以方便的判断正则表达式的运算是否还是正则表达式，并且一定程度上还可以用来证明语言L不是正则的。

正则语言的可判定性：

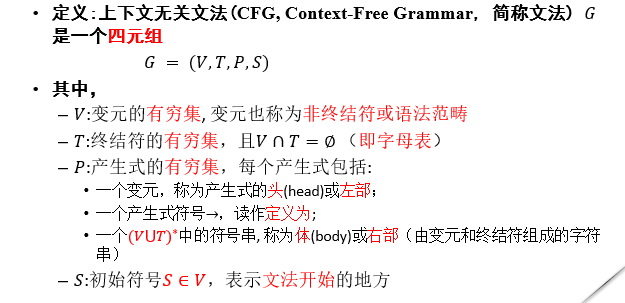


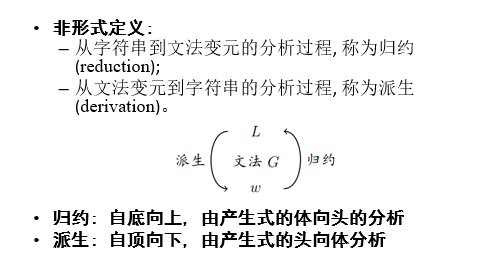


自动机的最小化：

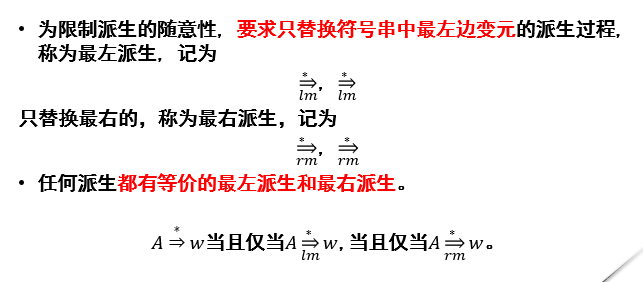
状态等价性并运用状态化简最小化DFA

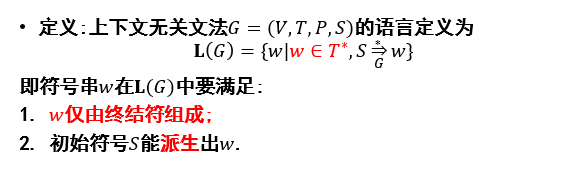
1. 上下文无关分析法



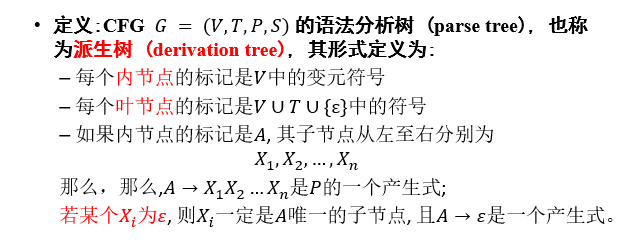


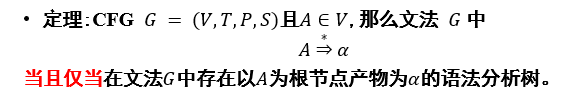
要求会写规约和派生过程。





语法分析树：





证明由归纳法证明。

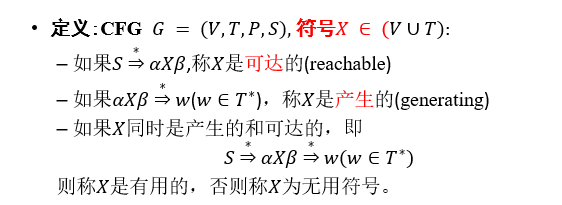
歧义性：针对文法，某个句型可由不同的语法树得到。

固有歧义性：针对语言，所有描述该语言的文法均有歧义。

文法的化简与范式：

方法：

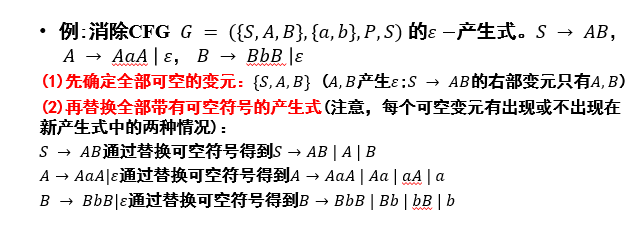
1. 消除无用符号



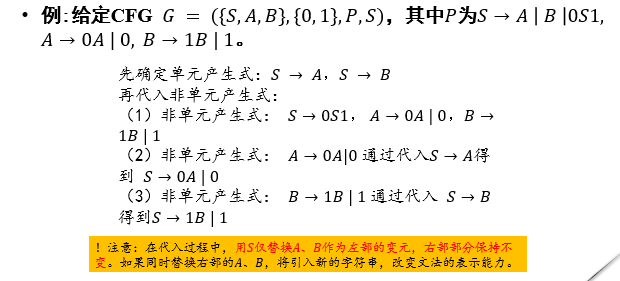
删除所有无用符号：连同表达式一起删除。

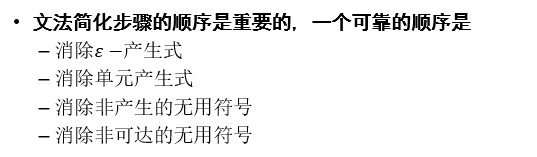
注意先找**“非产生的”**，再找**“非可达的”，即从下往上找**。

1. 消除产生式：



1. 消除单元产生式：

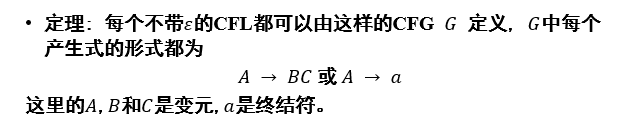


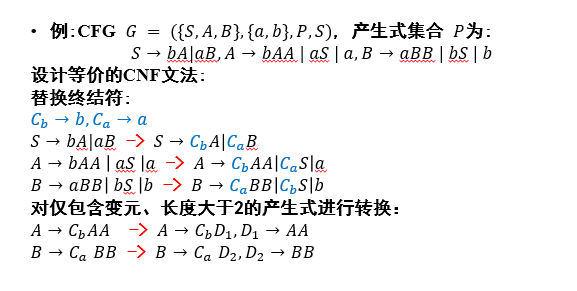


文法范式：

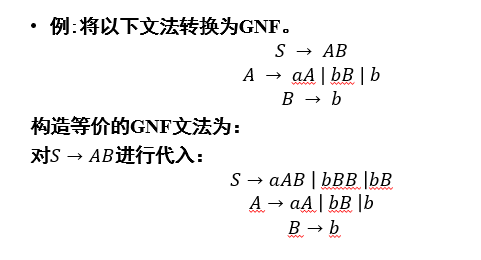
乔姆斯基范式（CNF）和格雷巴赫范式（GNF）

CNF:



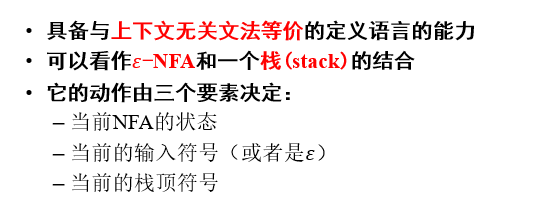


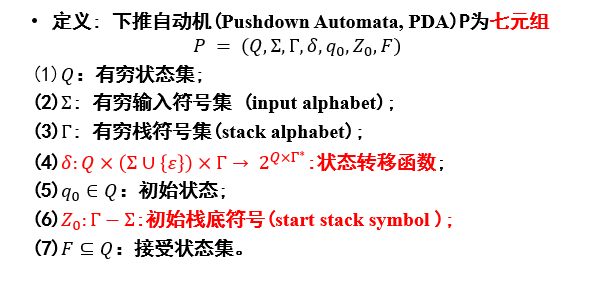
GNF:

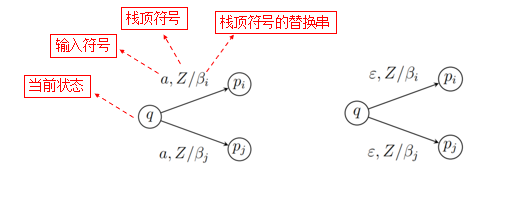


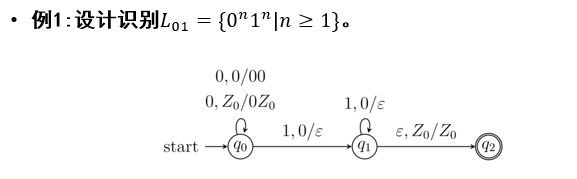
1. 下推自动机

下推自动机特点：

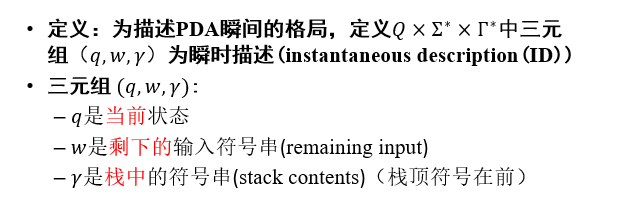


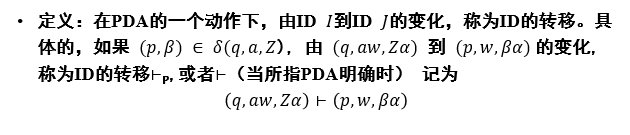




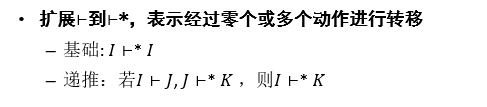


定义ID：



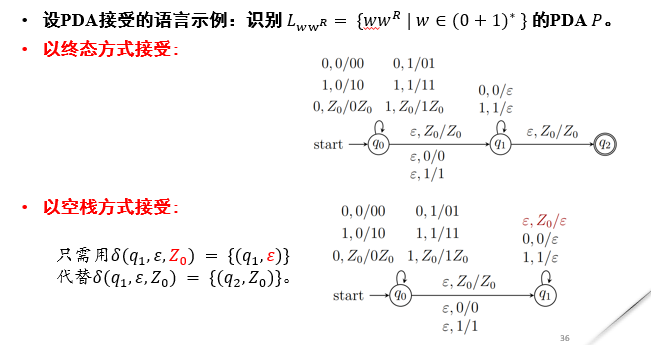


其实就是具体描述了的其中一种情况。



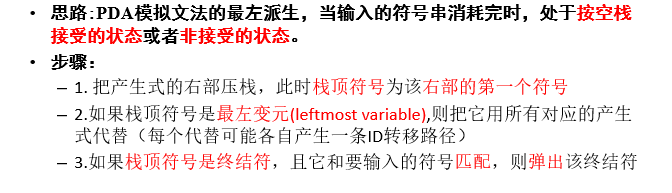
下推自动机接受的语言：

1. 以终态方式接受
2. 以空栈方式接受

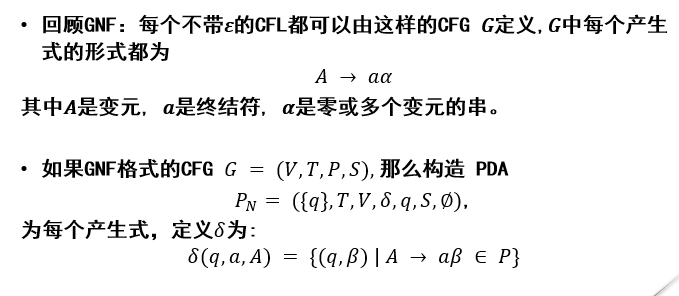


1. 两种方式的等价性

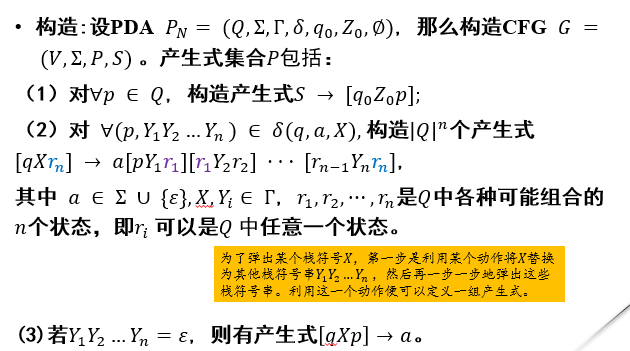
:



特别的，有：

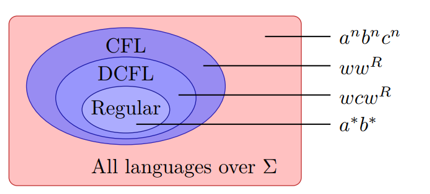


：

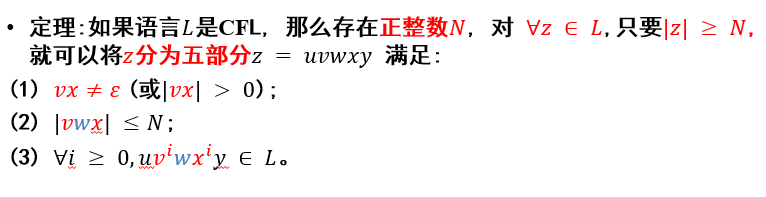


确定的下推自动机(DPDA)：接受确定的上下文无关语言(DCFL)

主要是接受非固有歧义的CFL



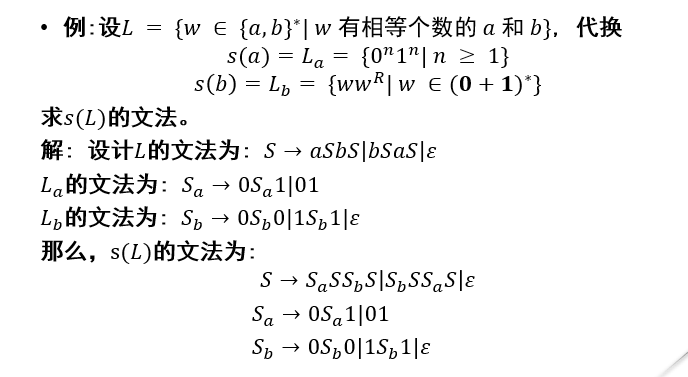
1. 上下文无关语言性质
2. 上下文无关语法的泵引理



证明L不是CFL。

1. 上下文无关语言的封闭性
   1. 代换封闭

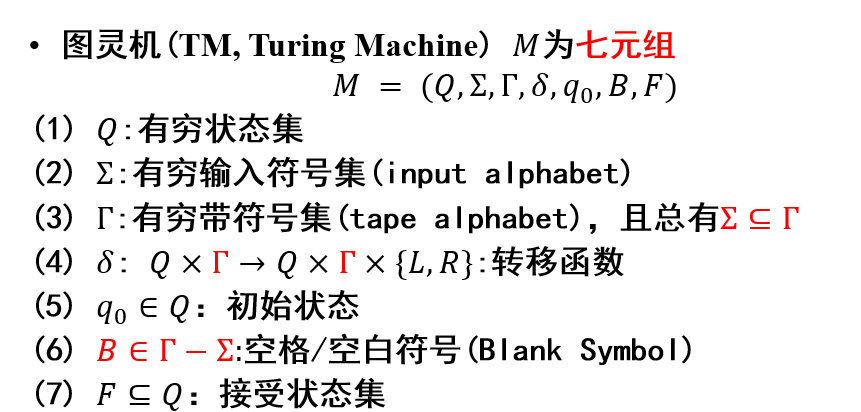
代换是指**中的一个字符在下为上的一个语言。**

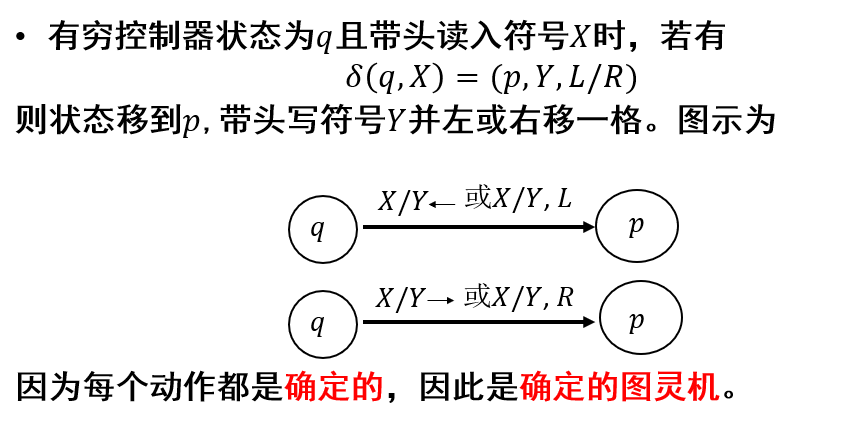


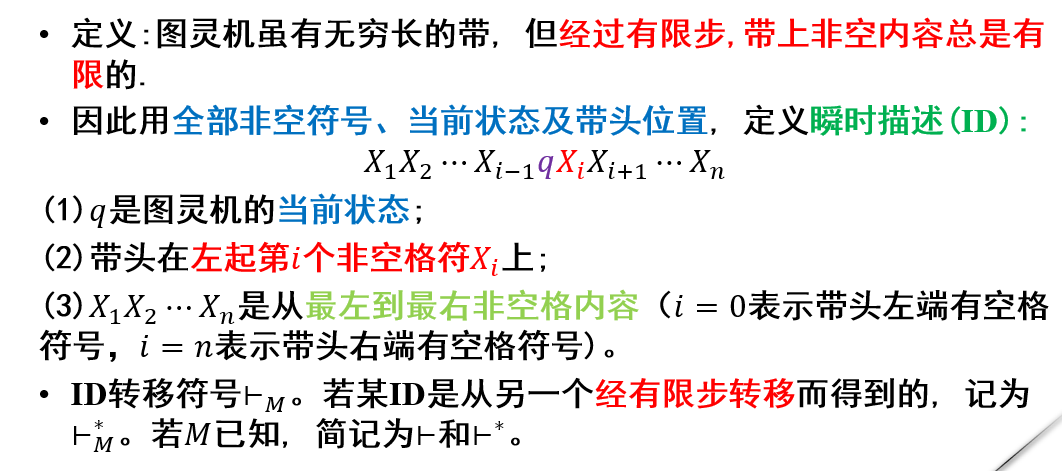
* 1. 交、补不封闭

但CFL和正则语言RL的交是CFL，可由此证明L不是CFL。

1. 上下文无关语言的判定性质
2. 乔姆斯基文法体系
3. 图灵机







1. 可能出现的题型和必须掌握的方法
2. 构造指定功能的DFA
3. 构造指定功能的NFA
4. 写出指定的正则表达式
5. 由泵引理推L不是正则语言
6. 由正则语言封闭性判断L不是正则语言
7. 最小化自动机
8. 会写上下文无关文法中的派生和规约过程
9. 上下文无关文法化简
10. 上下文无关文法化简为CNF和GNF
11. 写出下推自动机ID的变化过程
12. 由泵引理推L不是上下文无关语言
13. 由上下文无关语言封闭性推L不是上下文无关语言
14. 设计图灵机（整数运算？）
15. 图灵机ID描述