编译原理

习题课 (1)

第二章 高级语言及其语法描述

- 程序语言的定义
- 高级语言的一般特性
- 程序语言的语法描述

上下文无关文法

- 一个上下文无关文法 G 是一个四元式
  $G=(V_T, V_N, S, P)$，其中
  - V_T： 终结符集合 (非空)
  - V_N： 非终结符集合 (非空)，且 $V_T \cap V_N=\varnothing$
  - S： 文法的开始符号， $S \in V_N$
  - P： 产生式集合 (有限)，每个产生式形式为
    - $P \rightarrow \alpha$， $P \in V_N$， $\alpha \in (V_T \cup V_N)^*$
  - 开始符 S 至少必须在某个产生式的左部出现一次

上下文无关文法

- 定义：称$\alpha A\beta$ 直接推出$\alpha\gamma\beta$，即

$$\alpha A\beta \Rightarrow \alpha\gamma\beta$$

  仅当 $A \rightarrow \gamma$ 是一个产生式，

  且α，$\beta \in (V_T \cup V_N)^*$。

- 如果$\alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_n$，则我们称这个序列是从$\alpha_1$到$\alpha_n$的一个推导。若存在一个从$\alpha_1$到$\alpha_n$的推导，则称$\alpha_1$可以推导出$\alpha_n$

上下文无关文法

- 定义：假定 G 是一个文法， S 是它的开始符号。如果 $S \overset{*}{\Rightarrow} \alpha$， 则α称是一个句型。
- 仅含终结符号的句型是一个句子。
- 文法 G 所产生的句子的全体是一个语言，将它记为 L(G) 。

$$L(G) = \{\alpha \mid S \overset{+}{\Rightarrow} \alpha, \ \alpha \in V_T^*\}$$

如何写出产生特定语言的文法

- 分析语言的特点
- 分解成层次结构
- 根据结构写出文法

P36-7 写一个文法，使其语言是奇数集，且每个奇数不以 0 开头。

非 0 开头数字串	奇数数字

- G(S):

  $S \rightarrow O \mid A\ O$

  $O \rightarrow 1 \mid 3 \mid 5 \mid 7 \mid 9$

  $N \rightarrow O \mid 2 \mid 4 \mid 6 \mid 8$

  $D \rightarrow 0 \mid N$

  $A \rightarrow A\ D \mid N$

P36-11. 给出下面语言的相应文法

- $L_1=\{a^n b^n c^i \mid n \geq 1$ □ $i \geq 0\}$
- □□□ G(S):

  $S \rightarrow A\ C$

  $A \rightarrow a\ A\ b \mid ab$

  $C \rightarrow c\ C \mid \varepsilon$

- $L_4=\{1^n 0^m 1^m 0^n \mid n$ □ $m \geq 0\}$
- □□□ G(S):

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

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

语法树与二义性 (ambiguity)

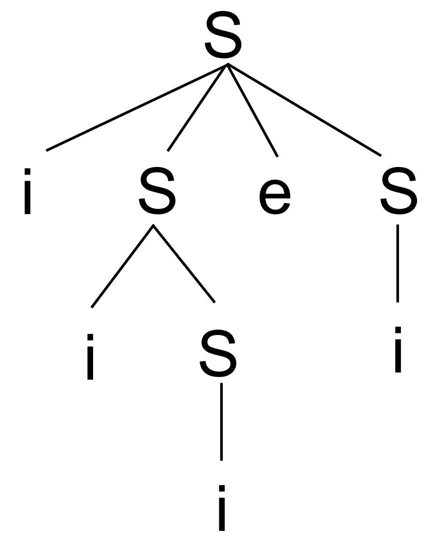
- 定义：如果一个文法存在某个句子对应两颗不同的语法树，则说这个文法是二义的
- 语言的二义性：一个语言是二义性的，如果对它不存在无二义性的文法
- 二义性问题是不可判定问题，即不存在一个算法，它能在有限步骤内，确切地判定一个文法是否是二义的
- 可以找到一组无二义文法的充分条件

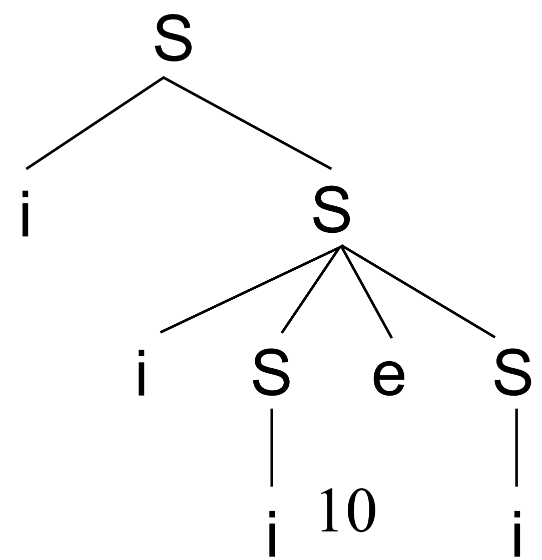
P36-9. 证明下面的文法是二义的：

S→iSeS | iS | i

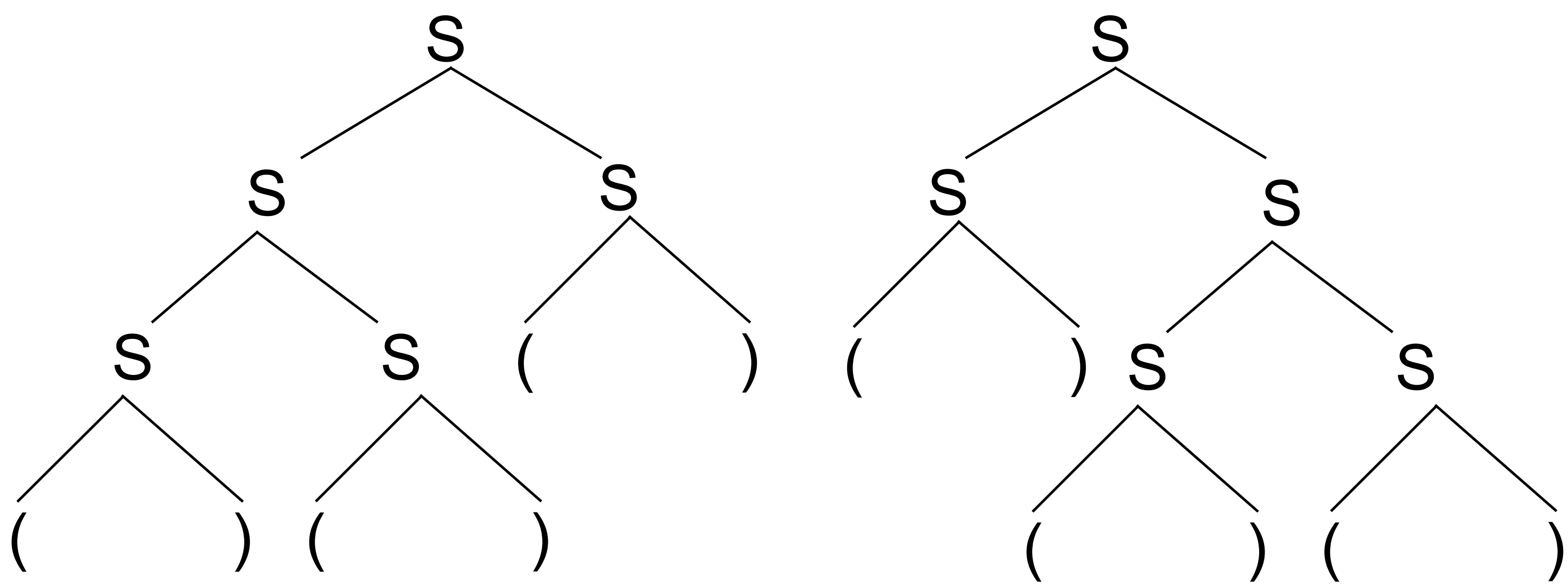
- 思路：找出一个句子有两棵语法树（两个最左推导、两个最右推导）
- 前提：分析文法的特点，找出歧义产生的源头
  - 考虑句型 iiSeS
- 解答：

  句子 iiiei 有两个语法树





P36-10. 把下面文法改写为无二义的： $S \rightarrow SS \mid (S) \mid (\ )$



- 考虑句型 SSS
- G(S):

  $S \rightarrow S\ T \mid T$

  $T \rightarrow (S) \mid (\ )$

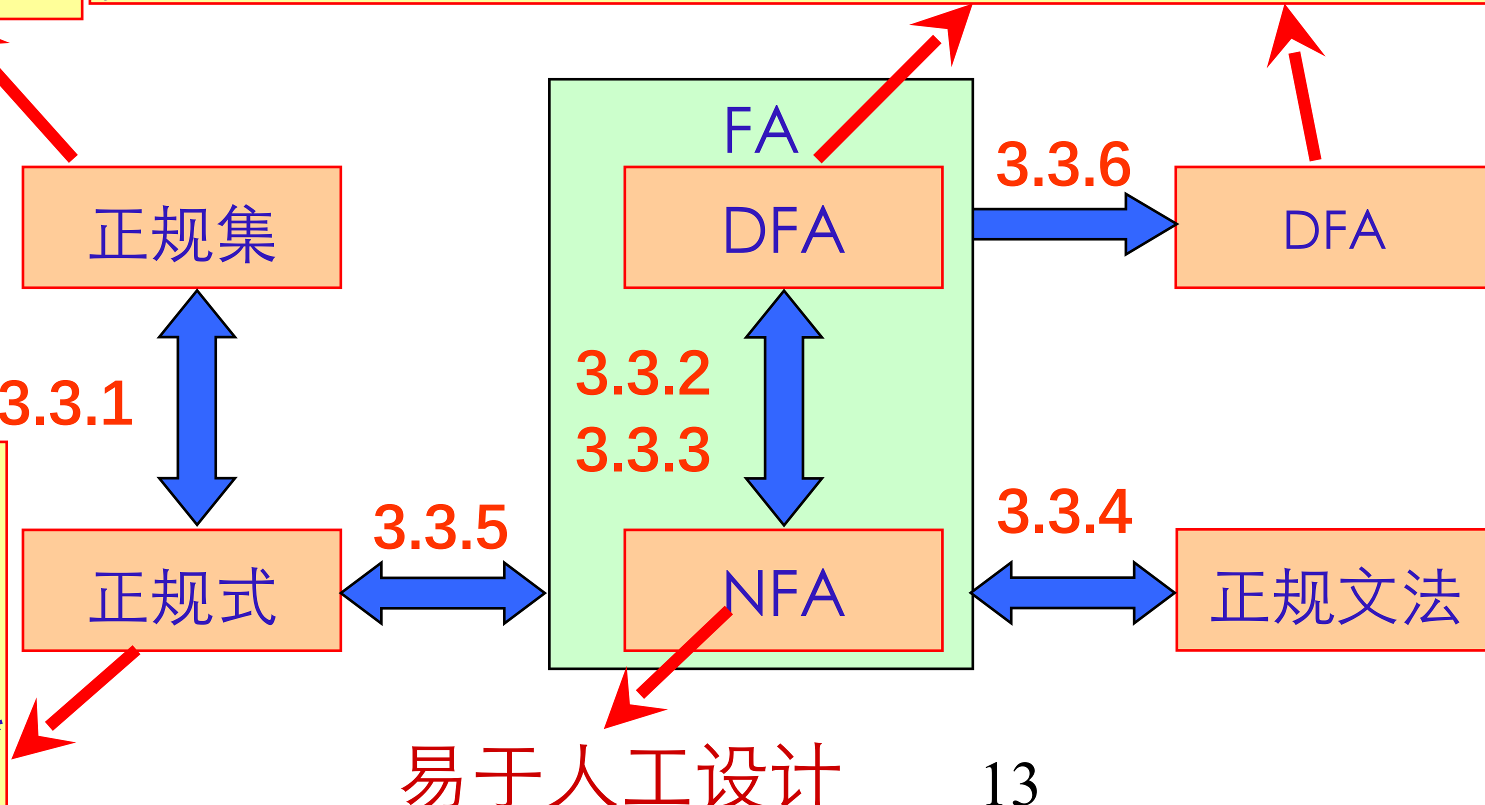
第三章　词法分析

- 对于词法分析器的要求
- 词法分析器的设计
- 正规表达式与有限自动机
- 词法分析器的自动产生 --LEX

关系图

```
curState = 初态
GetChar();
while( stateTrans[curState][ch] 有定义 ){
    // 存在后继状态，读入、拼接
    Concat();
    // 转换入下一状态，读入下一字符
   curState= stateTrans[curState][ch];
    if cur_state 是终态 then  返回 strToken 中的单
    GetChar( );
}
```

DIM,IF, DO,STOP,END
number, name, age
125, 2169
…



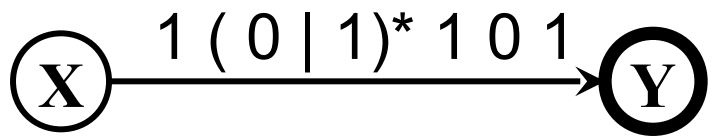
DIM
IF
DO
STOP
END
letter(letter|digit)*
digit(digit)*

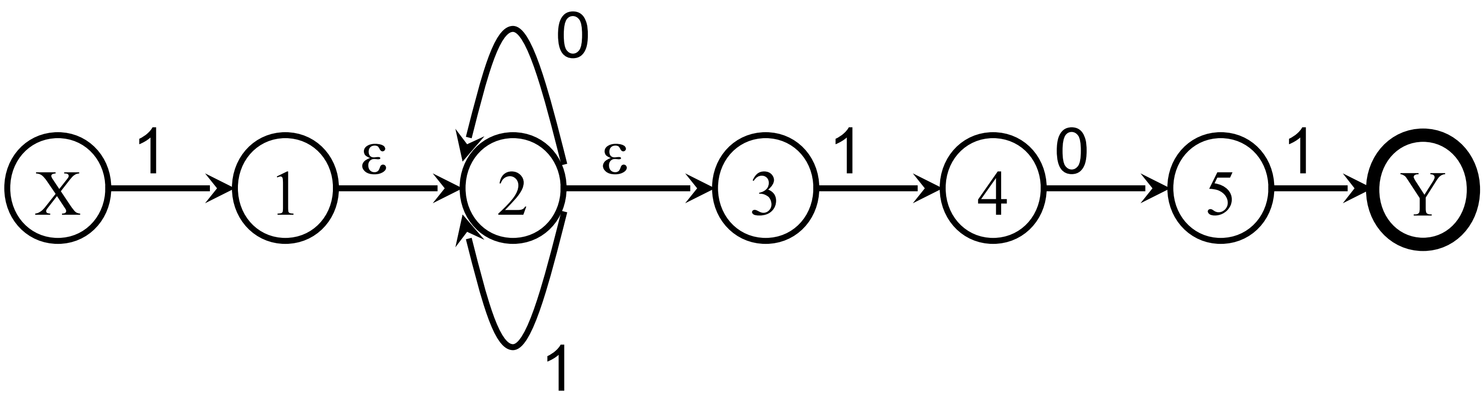
要点

- 几个转换算法
  - 正规式 ⇔ NFA
  - NFA ⇒ DFA
  - DFA 化简算法

P64-7. 构造下列正规式相应的 DFA
1(0 | 1)*101

- 思路：正规式⇒ NFA ⇒DFA





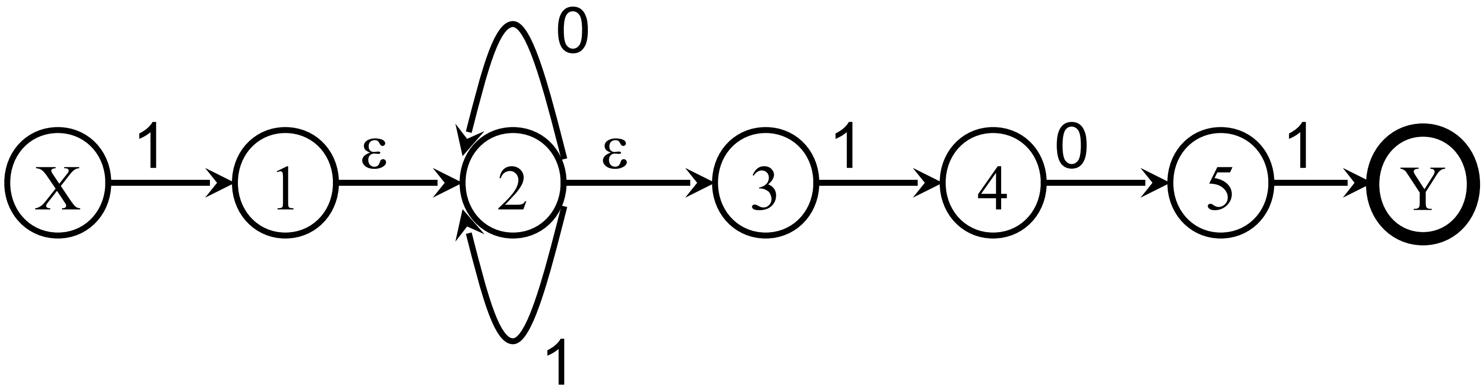
确定化的过程

- 不失一般性，设字母表只包含两个 *a* 和 *b*，我们构造一张表：

I	I_a	I_b
$\varepsilon-Closure(\{X\})$	{…}	{…}
{…}	{…}	{…}
{…}	{…}	{…}

- 首先，置第 1 行第 1 列为 ε-closure({X}) 求出这一列的 I_a，I_b；
- 然后，检查这两个 I_a，I_b，看它们是否已在表中的第一列中出现，把未曾出现的填入后面的空行的第 1 列上，求出每行第 2，3 列上的集合 …
- 重复上述过程，直到所有第 2，3 列子集全部出现在第一列为止

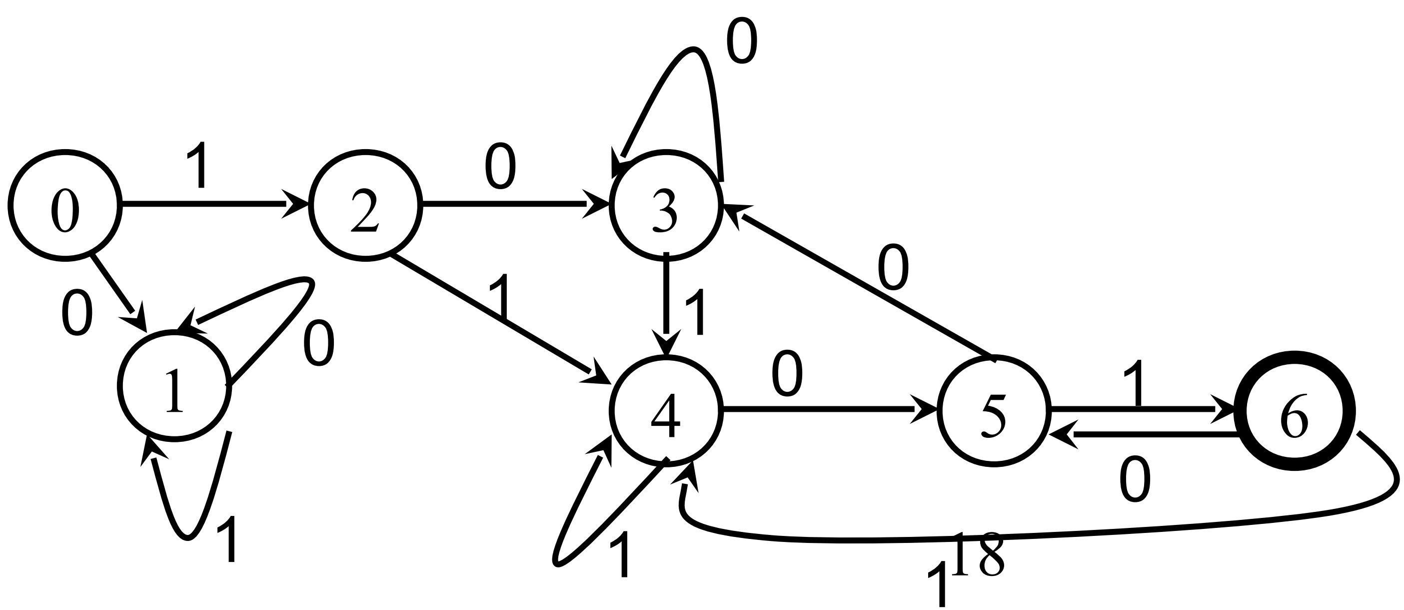
确定化



	0	1
{X}	φ	{1,2,3}
φ	φ	φ
{1,2,3}	{2,3}	{2,3,4}
{2,3}	{2,3}	{2,3,4}
{2,3,4}	{2,3,5}	{2,3,4}
{2,3,5}	{2,3}	{2,3,4,Y}
{2,3,4,Y}	{2,3,5}	{2,3,4,}

确定化

	0	1
{X}	ϕ	{1,2,3}
ϕ	ϕ	ϕ
{1,2,3}	{2,3}	{2,3,4}
{2,3}	{2,3}	{2,3,4}
{2,3,4}	{2,3,5}	{2,3,4}
{2,3,5}	{2,3}	{2,3,4,Y}
{2,3,4,Y}	{2,3,5}	{2,3,4,}



最小化：对状态集进行划分

- 首先，把 S 划分为终态和非终态两个子集，形成基本划分Π。
- 假定到某个时候，Π已含 m 个子集，记为 $\Pi=\{I^{(1)}, I^{(2)}, \ldots, I^{(m)}\}$，检查Π中的每个子集看是否能进一步划分：
  - 对某个 $I^{(i)}$，令 $I^{(i)}=\{s_1,s_2, \ldots,s_k\}$，若存在一个输入字符 a 使得 $I_a^{(i)}$ 不会包含在现行Π的某个子集 $I^{(j)}$ 中，则至少应把 $I^{(i)}$ 分为两个部分。

最小化

$$\{0,1,2,3,4,5\},\{6\}$$

$$\{0,1,2,3,4,5\}_0=\{1,3,5\} \quad \{0,1,2,3,4,5\}_1=\{1,2,4,6\}$$

$$\{0,1,2,3,4\},\{5\},\{6\}$$

$$\{0,1,2,3,4\}_0=\{1,3,5\}$$

$$\{0,1,2,3\},\{4\},\{5\},\{6\}$$

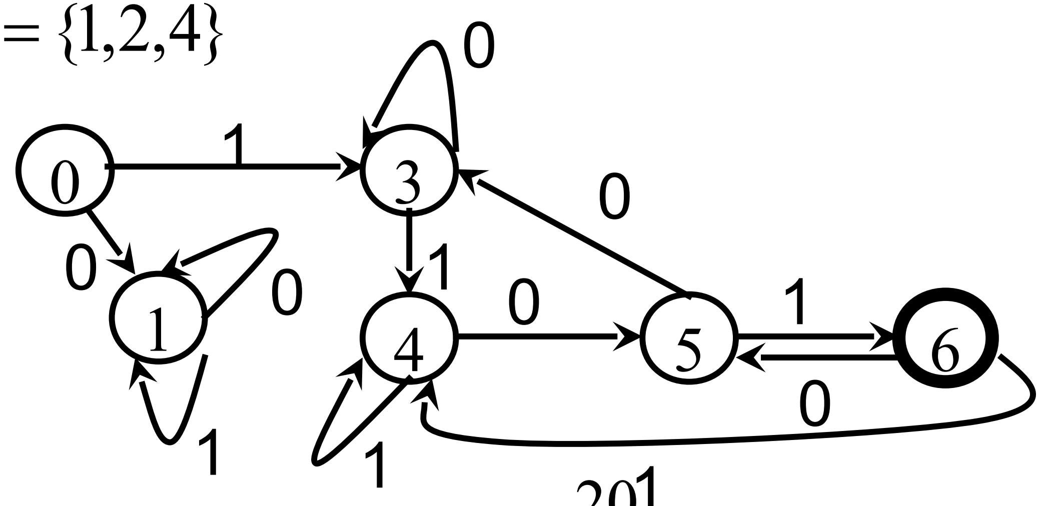
$$\{0,1,2,3\}_0=\{1,3\} \quad \{0,1,2,3\}_1=\{1,2,4\}$$

$$\{0,1\},\{2,3\}\ \{4\},\{5\},\{6\}$$

$$\{0,1\}_0=\{1\} \quad \{0,1\}_1=\{1,2\}$$

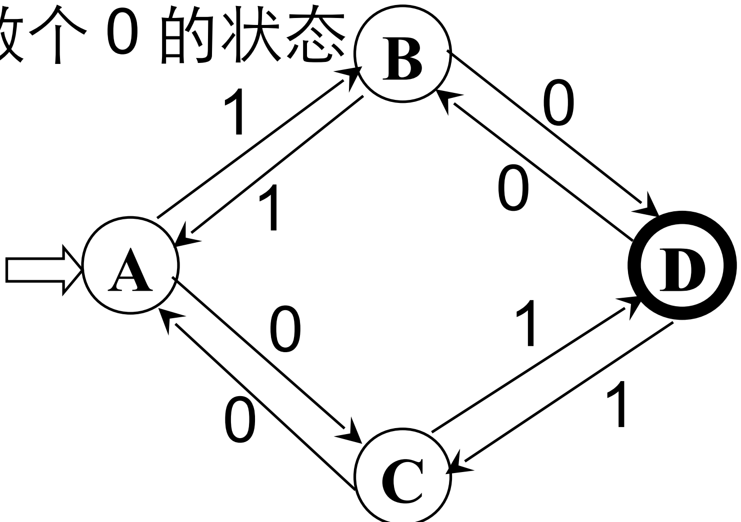
$$\{2,3\}_0=\{3\} \quad \{2,3\}_1=\{4\}$$

$$\{0\},\{1\},\{2,3\},\{4\},\{5\},\{6\}$$



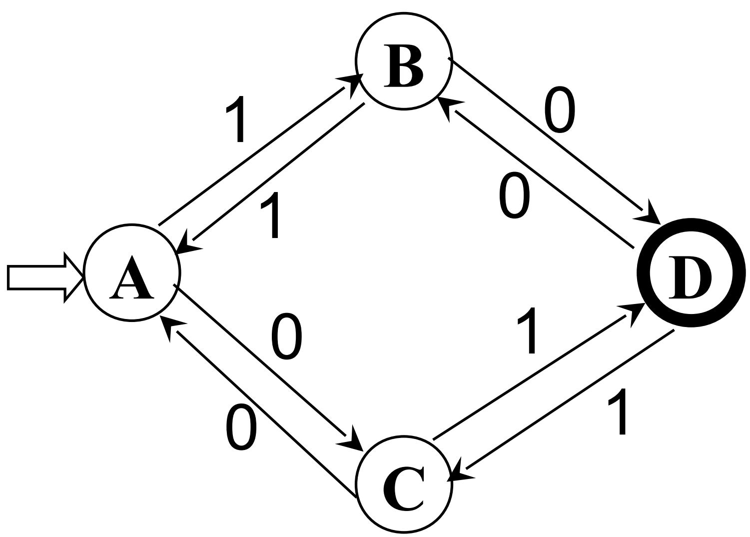
给出下面正规表达式：包含奇数个 1 和奇数个 0 的二进制数串

- 先设计 NFA
  - A： 识别了偶数个 1 和偶数个 0 的状态
  - B： 识别了奇数个 1 和偶数个 0 的状态
  - C： 识别了偶数个 1 和奇数个 0 的状态
  - D： 识别了奇数个 1 和奇数个 0 的状态
- 将 NFA 转换成正规式



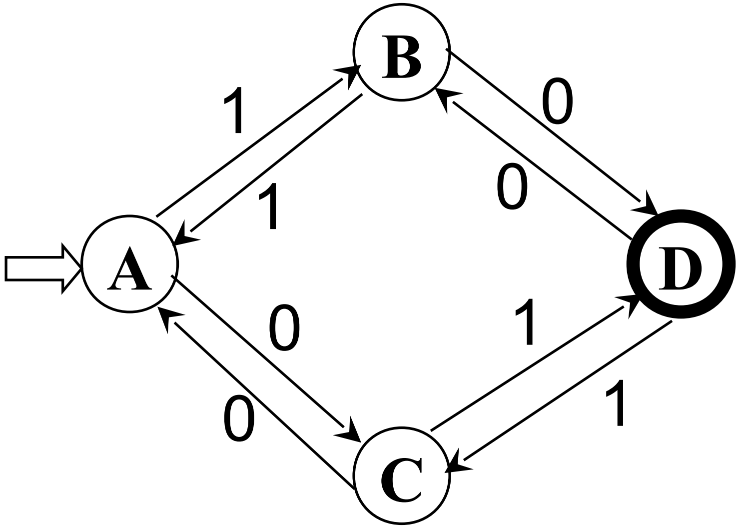
给出下面正规表达式：包含奇数个 1 和奇数个 0 的二进制数串

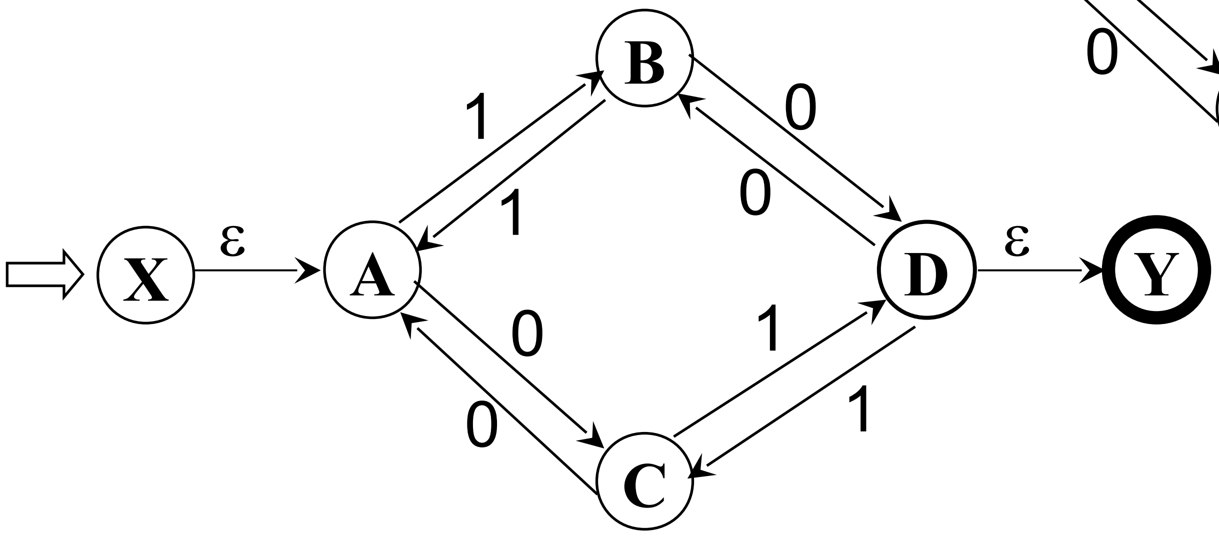
- 先设计 NFA
- 将 NFA 转换成正规式



给出下面正规表达式：包含奇数个 1 和奇数个 0 的二进制数串

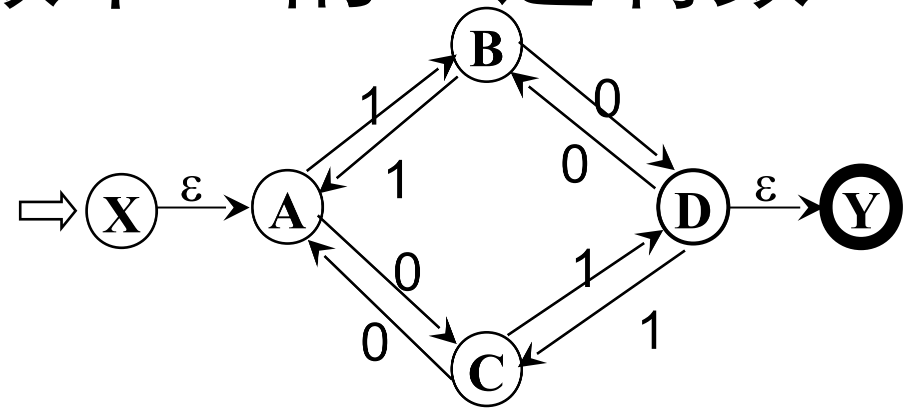
- 先设计 NFA
- 将 NFA 转换成正规式

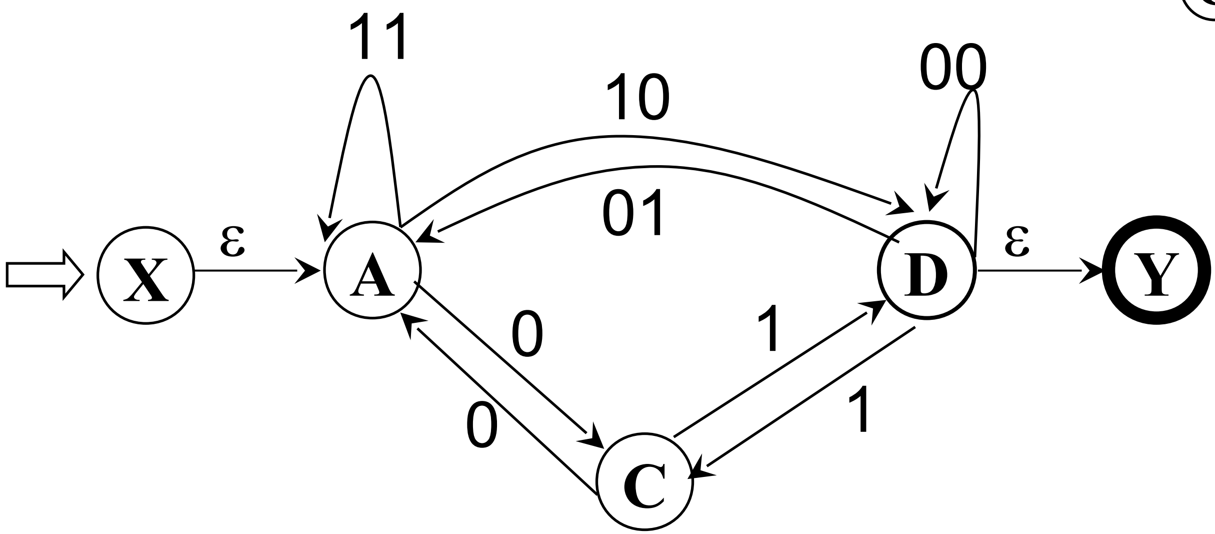




给出下面正规表达式:
包含奇数个 1 和奇数个 0 的二进制数串

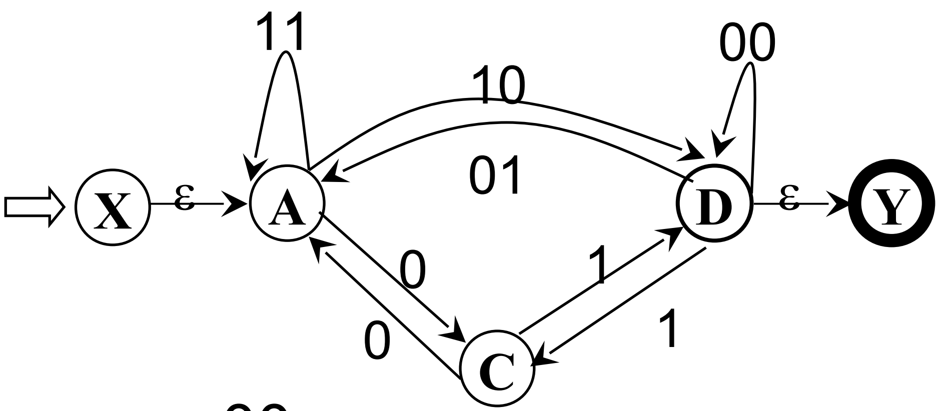
- 先设计 NFA
- 将 NFA 转换成正规式

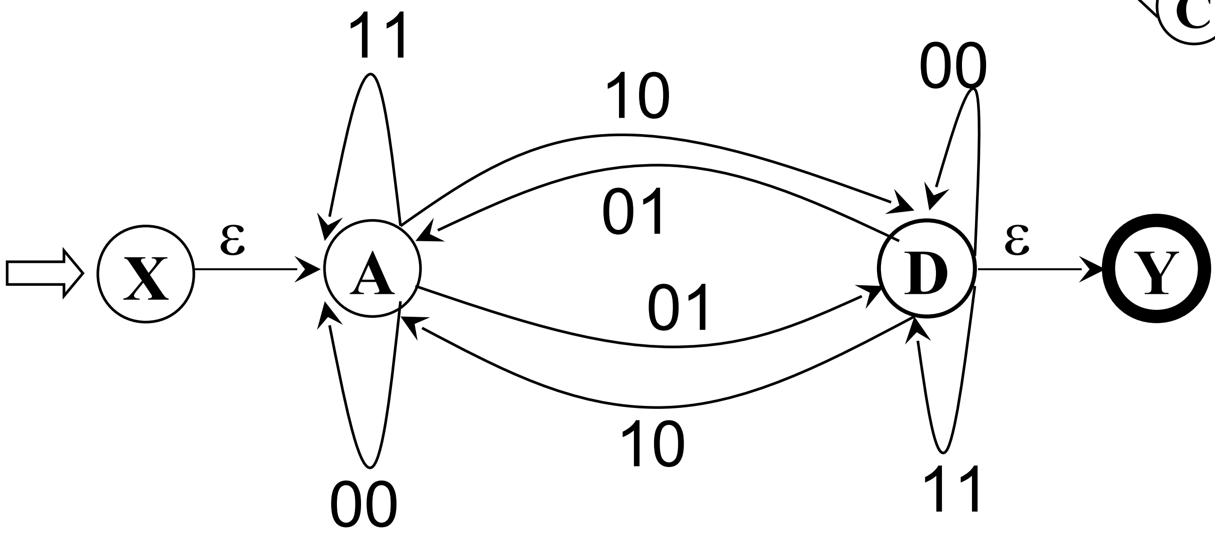




给出下面正规表达式:
包含奇数个 1 和奇数个 0 的二进制数串

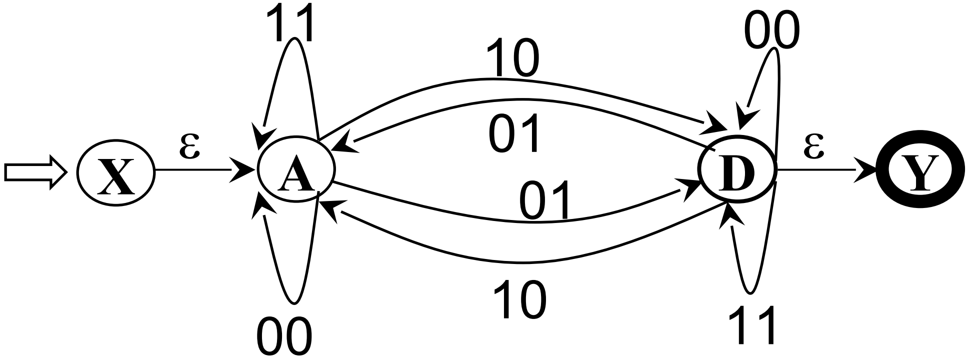
- 先设计 NFA
- 将 NFA 转换成正规式

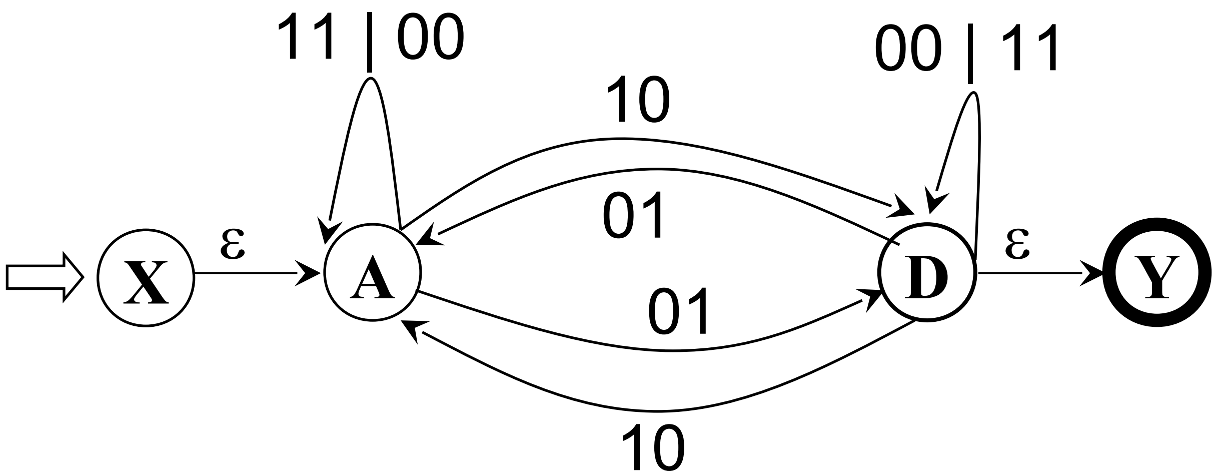




给出下面正规表达式:
包含奇数个 1 和奇数个 0 的二进制数串

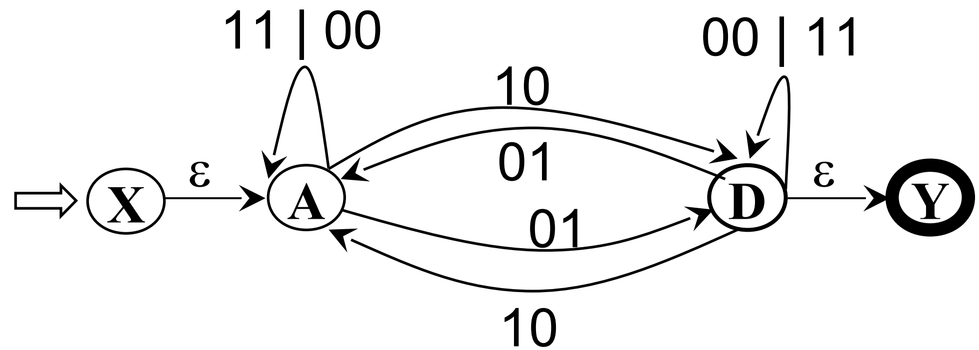
- 先设计 NFA
- 将 NFA 转换成正规式

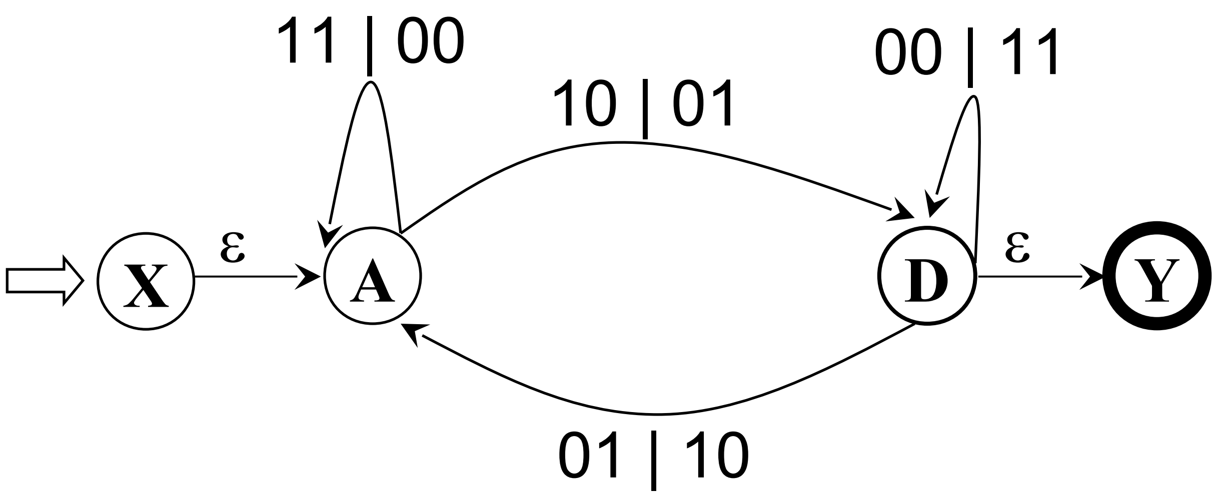




给出下面正规表达式：
包含奇数个 1 和奇数个 0 的二进制数串

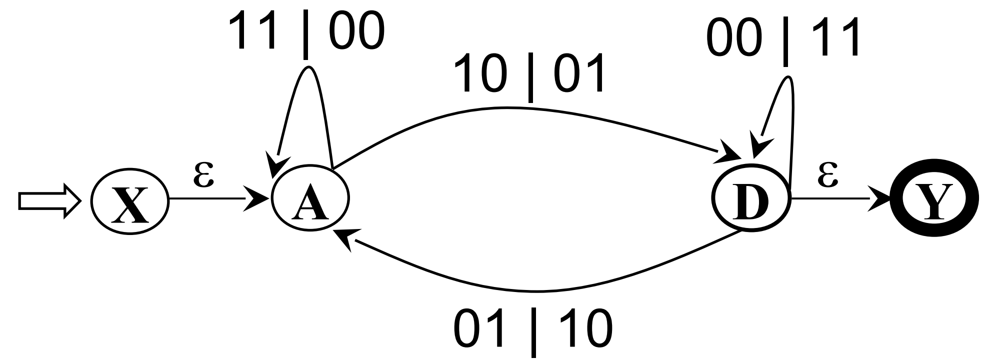
- 先设计 NFA
- 将 NFA 转换成正规式

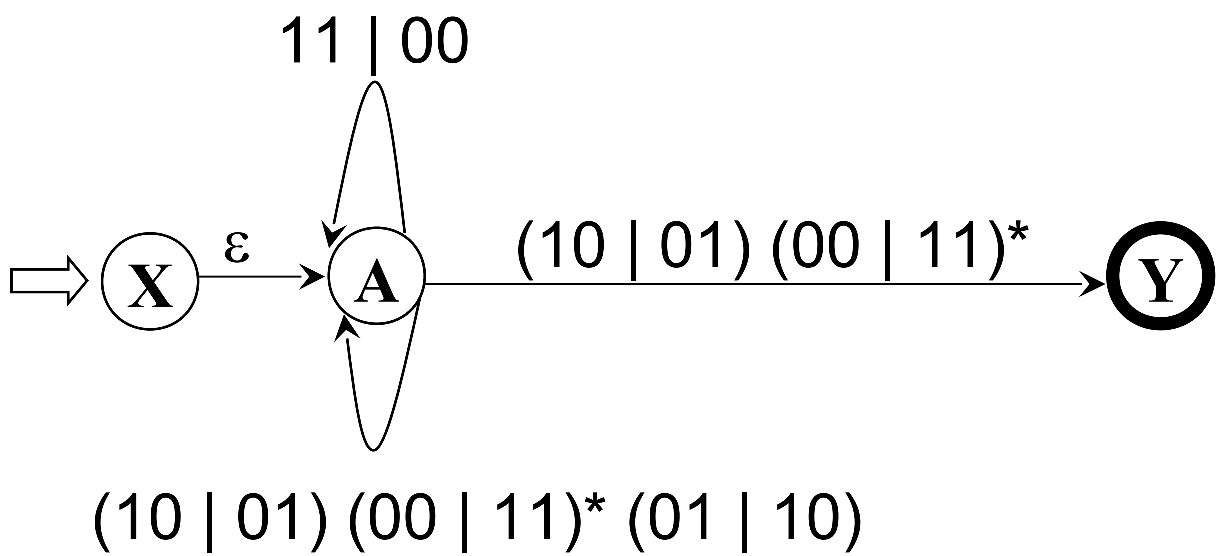




给出下面正规表达式:
包含奇数个 1 和奇数个 0 的二进制数串

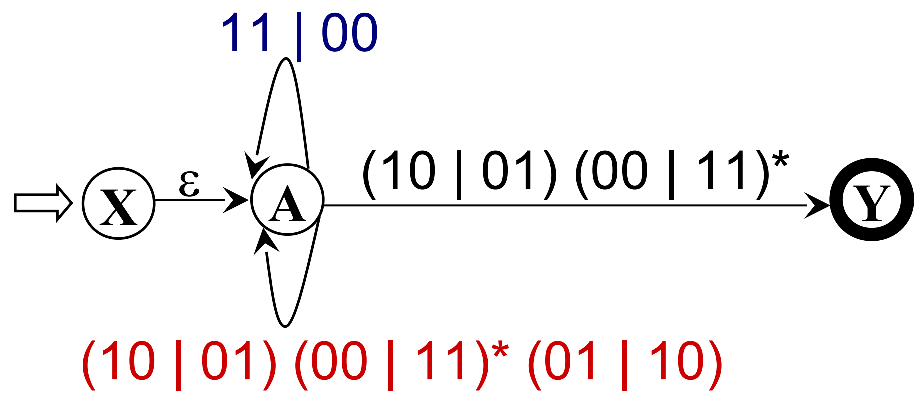
- 先设计 NFA
- 将 NFA 转换成正规式



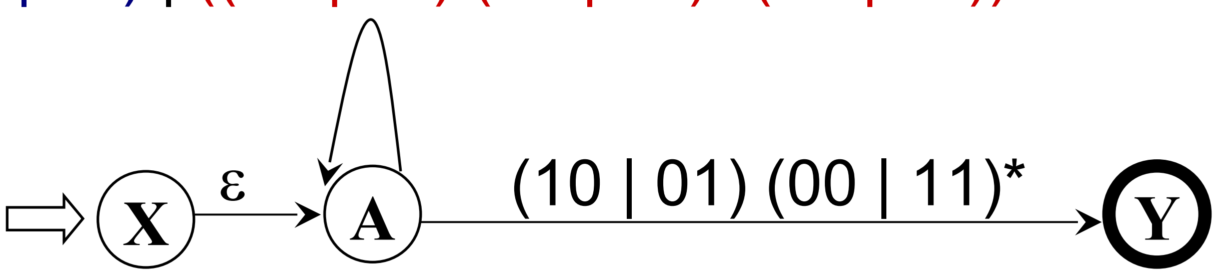


给出下面正规表达式:
包含奇数个 1 和奇数个 0 的二进制数串

- 先设计 NFA
- 将 NFA 转换成正规式

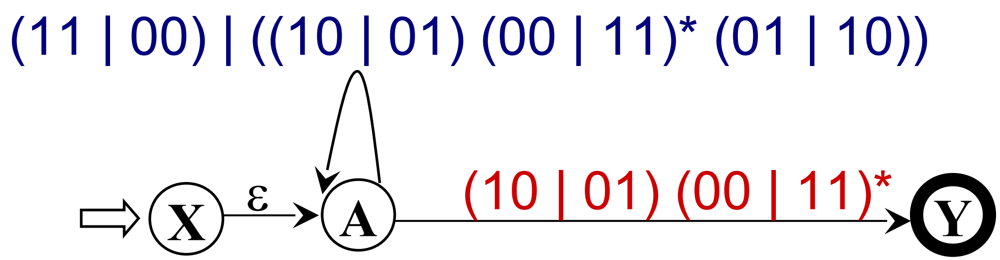


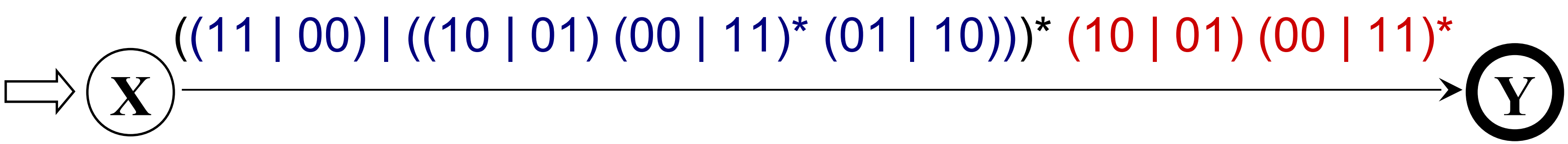
(11 | 00) | ((10 | 01) (00 | 11)* (01 | 10))



给出下面正规表达式:
包含奇数个 1 和奇数个 0 的二进制数串

- 先设计 NFA
- 将 NFA 转换成正规式

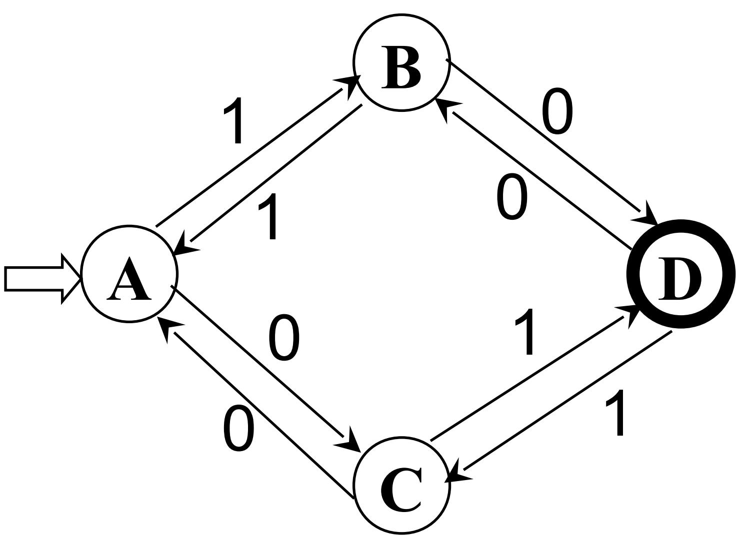




给出下面正规表达式：包含奇数个 1 和奇数个 0 的二进制数串

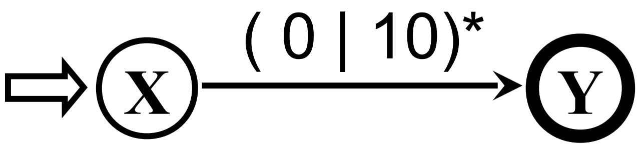
- 先设计 NFA
- 将 NFA 转换成正规式

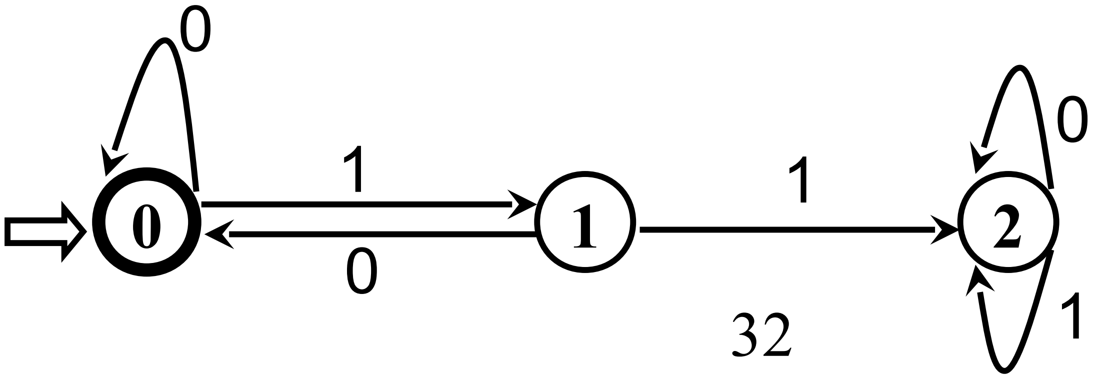
((11 | 00) | ((10 | 01) (00 | 11)* (01 | 10)))* (10 | 01) (00 | 11)*



P65-14. 构造一个 DFA， 它接受Σ = {0,1} 上所有满足如下条件的字符串：每个 1 都有 0 直接跟在右边。

- 思路
  - 分析语言特点
    - 0$\boxed{10}$00$\boxed{10}\boxed{10}$000$\boxed{10}$
  - 写出正规式
    - (0 | 10)*
  - 正规式 ⇒ NFA ⇒ DFA





小结

- 文法与语言
- 正规式 vs. NFA vs. DFA