有限自动机课程考试内容与形式都比较固定，每张卷子共有七个大题，依次考察以下内容，周老师说以下皆为必考：

- 构造不同分类的文法（一道大题）
- 构造较难的文法（一道大题）
- 构造正则表达式（一道大题）
- 构造 DFA　（一道大题）
- 构造 PDA　（一道大题）
- 构造 TM　（两道大题，40 分这是试卷区分度的重要体现）
  - 单道机 15 分
  - 三道机 25 分

以上皆为个人猜想，仅供参考，本人不承担任何责任

文法相关

文法题的一般形式为给出语言的文字描述，要求写出文法的推导式组。有时也会结合文法的分类出题。文法的分类分为：

1. RG：regular grammar（正则或右线性文法，3 型文法）
2. CFG：context free grammar（上下文无关文法）
3. CSG：context sensitive grammar（上下文相关）
4. PSG：phrase structure grammar（短语文法，0 型文法）

四种文法的区别：

a) PSG：无任何限制，产生式长度左边可以大于右边
b) CSG：产生式左边必须小于等于右边，但产生式左边可以包含终结符
c) CFG：产生式左边不能包含终结符，右边则可以终结符和非终结符随意组合
d) RG：生成式的形式只能为 A→ωB，其中ω是非终结符，B 是非终结符

##：一旦产生式包含 A→ε，则无论哪种文法直接成为 0 型文法

下面以一道例题来进行讲解：

二．L= $\{a,b\}^*\{1\}\{a,b,c\}^+$，构造满足以下条件的文法 10 分

1) 是 RG
2) 是 CFG，但不是 RG
3) 是 CSG，但不是 CFG
4) 是 PSG，但不是 CSG

$(a+b)^*1(a+b+c)^+$，观察易得此正规式可分为三部分，直接推：

$S\to A1B$
$A\to aA|bA|\varepsilon$
$B\to aB|bB|cB|a|b|c$

显然，产生式中出现空串，直接成为0型文法，现在，添加S'，尝试消除空串。

(1) $S\to S'B$
$S'\to A1B|1B$
$A\to aA|bA|a|b$
$B\to aB|bB|cB|a|b|c$

这个文法满足了2型文法的条件，尝试通过添加变量的方式使之成为3型文法（每次仅出现一个终结符+非终结符的组合

$S\to aS'|bS'|1B$
$S'\to aS'|bS'|1B$
$B\to aB|bB|cB|a|b|c$

接下来，我们添加一条CSG，将2型文法转化为1型文法：观察可得B前面必含1，故原式可改写为下式：

$1B\to 1aB|1bB|1cB|1a|1b|1c$，其余产生式不变。

至此，该正规式的0、1、2、3型文法均顺利完成

构造文法的试题中则容易出现个数一样多，回文串，某个字母至少出现一次等情况，下面使用两个例题来说明这三种情况：

一、字母表为{a，b，c}，构造下列语言的文法（10 分）。

1）$\{x \mid x=x^T,\ x\in\Sigma^+\}$；

2）$\{w \mid w\in\Sigma^+$，且 w 中 a，b 的个数一样多$\}$。

无论哪种回文串，包括简单回文串，中间包一个ω的回文串，其本质都是枚举法，都是将字母表中的元素进行依次列举，下面完成第一小问和两种回文串的变种。

(1) $X=X^T$，显然是回文串. 遇到回文串先看字母表，只有a、b、c.

依次列举，得基本文法如下：

$S \to aSa \mid bSb \mid cSc \mid a \mid b \mid c \mid \varepsilon$ (空串不能省)

下面是回文串的两个变种，仍然枚举字符表

(2) xwx^T, $x,w \in \Sigma^+$ (3) xx^Tw, $x,w \in \Sigma^+$

(2) $S \to aSa \mid bSb \mid cSc \mid aWa \mid bWb \mid cWc$

$W \to aW \mid bW \mid cW \mid a \mid b \mid c$

注意到(2)中 xx^T 仍然是回文串，(3)中也同理

(3) 将题目中简单左右分块，对分块后的子部分仍然回文处理即可

$S \to AW$

$A \to aAa \mid bAb \mid cAc \mid aa \mid bb \mid cc$ (xx^T 必然生成偶数个回文串)

$W \to aW \mid bW \mid cW \mid a \mid b \mid c$

对称结构，仍然采用 aXa 的形式，X可根据需要更换字母，非对称结构则左右分块，然后对对称部分采用相同方法处理.

回文串显然是比较容易的题目，无序字母个数相等也是常考的类型，下面解决第（2）小问并系统梳理有序无序字母个数相等的问题。

若干个元素相同的情况

(1) 有序(使用上下文无关文法)

① $\{a^n b^n \mid n \geq 0\}$

$S \to aSb \mid ab \mid \varepsilon$

② $\{a^n b^n c^n \mid n \geq 1\}$ (使用上下文相关文法)

$S \to aSBC$

1) $S \to aBC$

2) $CB \to BC$

$aB \to ab$

$bB \to bb$

$bC \to bc$

$cC \to cc$

注意到三个有序元素仍然采取分块的方法处理，1)行是为第一个产生式收尾，2)行起到将(BC)(BC)…(BC)的排列交换的作用，后几个产生式则是根据：

B只有在紧挨在a或b的右边才能替换为b，C亦同理方可替换为c

(2) 无序情况

无序情况看似复杂，然而基本思想仍然是枚举。

① 两字母a、b个数相等。(CFG)

$S \to aSbS \mid bSaS \mid \varepsilon$

② a恰好比b多一个：

使用E表示产生相同个数的a、b所有串(即①问)

$E \to aEbE \mid bEaE \mid \varepsilon$

∵ a比b多一个(扩展E和扩展S)

∴ $S \to aE \mid Ea \mid bSS \mid SbS \mid SSb$

③ 三字母a、b、c个数相等

三字母及以上相等的问题中须使用CSG. 仍然采用分块法和交换的原则。

$$S \to ABC \qquad A \to aS \mid a \qquad B \to bS \mid b \qquad C \to cS \mid c$$

$$\left.\begin{array}{l} AB \to BA \\ BC \to CB \\ AC \to CA \end{array}\right\} ※$$

总结

1. 每一字母所代表的非终结符都可以生成原串S ($A \to aS \mid a$) 或者变为单个非终结符

2. 非终结符的次序可以任意交换，且枚举出所有情况(见※)

使用以上理论解决此真题

1) $\{w \mid w \in \{a, b, c, d\}^+$，且w中a，b，d的个数一样多$\}$。

$S \to ABCD \qquad AB \to BA$

$A \to aS \mid a \qquad AC \to CA$

$B \to bS \mid b \qquad AD \to DA$

$C \to cS \mid c \qquad BC \to CB$

$D \to dS \mid d \qquad BD \to DB$

$CD \to DC$

其他类型的题目，只要熟练使用分块和交换的方法，基本上都可以轻松解决。

DFA 相关

FA（Finite Automachine）的本质是“有限”，有限就意味着你只能记住非常有限的东西，有限个状态是自动机能够记忆的唯一办法。在考题中，尽管输入串极长，但是并不需要记住整个输入，而只需要记住关键的信息，因此，解题的关键就是判断出哪些信息是关键信息，并据此设计自动机，这更像是一门 Art，而不是 Algorithm。下面以一道练习题来分析。

(7) $\{x|x \in \{0,1\}^*$ 且如果 x 以 1 结尾，则长度为偶数；如果 x 以 0 结尾，则长度为奇数$\}$

由题意我们可以分析串有四种情况：

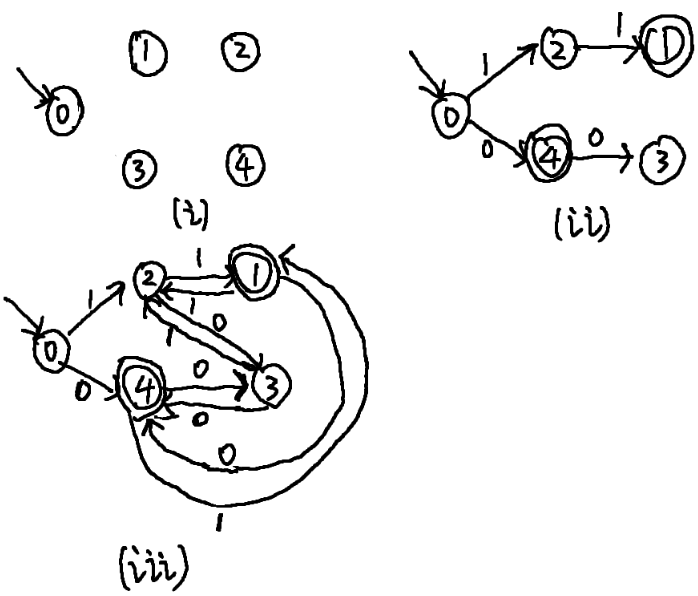
a) 尾数是 1，且长度为偶

b) 尾数是 1，且长度为奇

c) 尾数是 0，且长度为偶

d) 尾数是 0，且长度为奇

显然，a,d 为接收态。且串只能有四种状态。由于 x 属于星闭包，还要考虑接收为空串的情况（若输入存在空串则应考虑添加一个初始状态），因此初始共有 5 个状态（1-4 对应 a-d 四个状态，0 表示初态）。如下图 1。然后根据基本的输入 0,1,00,11 来进行状态之间的连接（见图 2），同时，因为每个状态都可能再接收一个 0 或 1 进入下一个状态，因此，根据之前的分析完成状态之间的连线，也就是最终答案（见图 3）。这个简单的例题应该可以加深对 DFA 的理解。

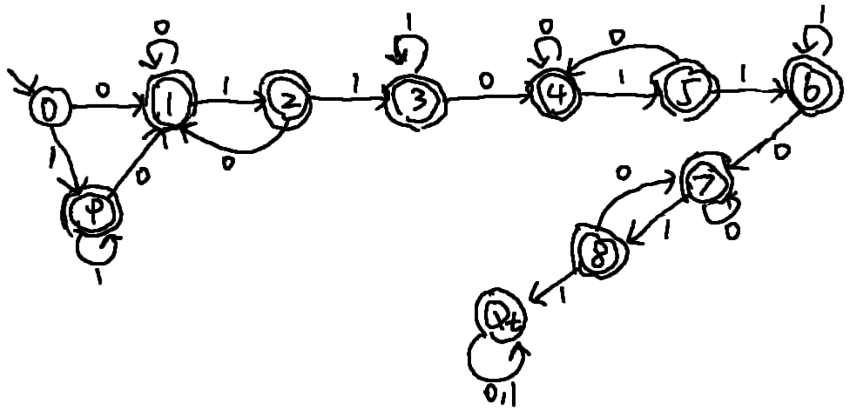


考试题中为了提升难度，状态数在 6-8 左右，太多的状态数会导致图过于复杂，不利于阅卷。下面以一道考试题为例进行讲解，这一部分还是比较简单的。

三、构造接收下列语言的 DFA。（15 分）

$\{w \mid w \in \{0,1\}^+$，且 w 中最多只能含二个 011 子串$\}$

根据题目可知，首先判断该 DFA 不接受空串，所以要添加一个初始状态。其次从最多包含两个 011 可以看出，先设计出接收三个 011 的自动机，然后对其他状态取反，问题得解。答案如下图所示。



此题难度不高，唯一要注意的就是初始状态不能接收，需要添加一个状态。DFA 不能停机，所以需要添加陷阱态 Qt。

DFA 还可能考到整除问题（可能性不大，时间急可不看），在此做简要讨论。

例　构造 DFA，接收{0，1}上的语言，该语言的每个字符串为二进制数时，代表的数字能被 5 整除。

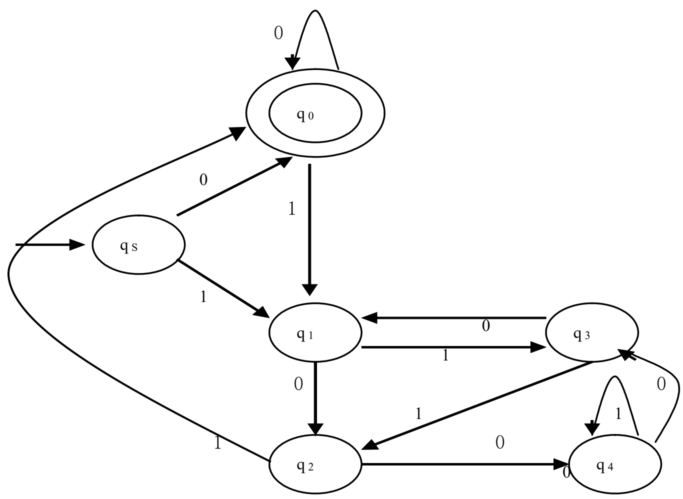
分析：任何整除问题都是余数问题，除以 5 则可能有 0,1,2,3,4 五种余数可能，因此为 5 种状态。另外由题意得不接收空串，所以另设一个开始状态 Qs。

[--以下内容为会员专属，非会员请自行看 PPT 理解--]

考虑两种情况：

a)　读入一个 0，则原数 x2，余数也 x2 然后 mod5.
　　因此，原余数为 0，新余数仍为 0。原余数为 1，新余数为 2。原余数为 2，新余数为 4。原余数为 3，新余数为 6 mod 5=1。原余数为 4，新余数为 8 mod 5=3。

b)　读入一个 1，则原数 x2+1，余数也 x2+1 然后 mod5.
　　因此，原余数为 0，新余数为 1。原余数为 1，新余数为 3。原余数为 2，新余数为 0。原余数为 3，新余数为 7 mod 5=2。原余数为 4，新余数仍为 9 mod 5=4。从下图答案中也可加深理解。



对于整除问题的推广，有以下结论：

不管读入 m 进制，需要整除 k，都采用十进制分析法。首先设立 k 个状态（0 to k-1）。每多读一个字符 c，就进入(当前余数状态 Xm+c.value) mod k 的状态，由此完成自动机即可。

NFA 相关（NFA 今年不考，不用看）

NFA 在下一个状态有若干种选择，也就是说可转换的后续状态是一个集合。因此不符合函数的性质。因此 NFA 对输入串的分析可以看做若干“过程”同时进行（人脑分析是为串行，若一条路径不接收则回溯），若这些子过程有一个被接受了，则原始串被接受。理解了这一点，NFA 的构造是要易于 DFA 的。另外，每一个 NFA，都有对应的 DFA 与之等价。处理补集类问题时，尤其要注意理解 NFA 的“回溯”过程，有可能接受不符合条件的结果。NFA 允许死机，故不必设立陷阱态。

NFA 常考计数+子串类问题，下面举例以加深理解。

三、构造识别下列语言的NFA。（15 分）

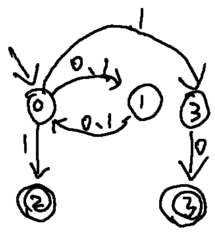
{w | w∈{0, 1}⁺；且如果w 以 10 结尾，则w 的长度为偶数；如果w 以 1 结尾，则w 的长度为奇数}。

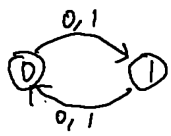
分析：

a) 不接收空串，设置开始状态 Qs

b) 奇偶状态要记忆，采取双循环模型。

c) 结尾 10 需增设不确定跳转来处理





双循环模型示意图

四、构造接收下列语言的NFA。（15 分）

$\{w \mid w\in\{0,1\}^*$；且若 w 以 111 结尾，则 w 的长度必须为奇数；若 w 以 0 结尾，则 w 的长度必须为偶数}

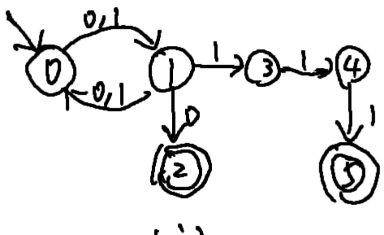
$\{w \mid w\in\{0,1\}^+$；且 w 必须包含 00 子串} $\{w \mid w\in\{0,1\}^+$；且 w 不包含 11 子串}

分析：

a) 第一题用双循环秒解

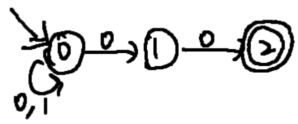
b) 包含子串类问题用 NFA 较易，不包含子串问题最好使用 DFA（补集法）。因为 NFA 优先选择接收的路径，所以补集法在 NFA 中要慎重使用。

答案：



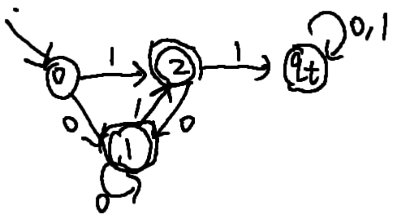
(i)

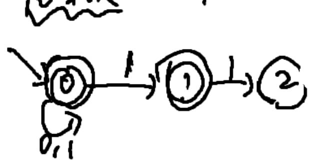
必含00子串



(ii)-①

不包含11子串(最好用DFA)，错误的NFA





(只要有一条接收的路径,NFA也会接收)

PDA 相关

下推自动机（PDA）结构上跟 NFA 相比增加了一个栈（无限容量），也就是下推栈。PDA 的能力与 CFG 等价。和 FA 类似，PDA 也分为确定的和不确定的。考试中对于 PDA 的确定性一般没有要求。PDA 是由若干条规则所定义的，规则之间的顺序不影响 PDA 的运行结果。此部分仅出一道大题，难度不高，应该满分。

考试时一般要求写上空栈接收的规则。但是也可以新添加 acc 和 ref 作为接收态和拒绝态。

PDA 的识别分为两种情况：

1. 终止状态接收：栈不为空，但输入串已读完，成为空串，添加 acc 态和 refuse 态。
2. 空栈接收：初态时的栈底符号 Z0 被弹出。一旦 Z0 被弹出，PDA 就不能工作了。

PDA 分为单态机和多态机，单态机和多态机在功能上是完全等价的。单态机是三元组，包括（读取内容，栈顶元素，栈动作），多态机相对于单态机添加了初态和终态，而单态机默认一个状态，故可以省略。注意，每次读取栈顶元素是破坏性读，即栈顶元素会被弹出，因此，压入一个新元素实际上需要先将原来的元素压进去。下面以一道例题讲解一下具体的区别：

接收语言

$L=\{\omega 2\omega^T \mid w\in\{0,1\}^*\}$

此题思路较易，读入 ω 串后入栈，读到 2 中心点后出栈并与读入的元素进行比较。

1. 多态机实现：read-match 模型，Q0 是 read 态，Q1 是 match 态

<Q0, 0, Z0, Q0, AZ0>

<Q0, 1, Z0, Q0, BZ0> //Q0 状态可以和 Q1 合并

<Q0, 2, Z0, QF, ε > //QF 是终态，此时 Z0 被弹出，多态机停机

<Q0, 0, A, Q0, AA>

<Q0， 0， B， Q0， AB>
<Q0， 1， A， Q0， BA>
<Q0， 1， B， Q0， BB>
<Q0， 2， A， Q1， A>
<Q0， 2， B， Q1， B>
<Q1， 0， A， Q1， ε>
<Q1， 1， B， Q1， ε>
<Q1， ε，Z0， QF， ε>
//QF 是终态，此时 Z0 被弹出，多态机停机，属于空栈接收

2. 单态机实现：需要先写出文法，还需要转换为 GNF，略复杂
   a) 产生 L 的上下文无关文法：
      S→2 ｜ 0S0 ｜ 1S1
   b) 将文法转化成 GNF
      S→2 ｜ 0SA ｜ 1SB
      A→0
      B→1
   c) 构造单态 PDA
      <0，S，SA> //S→0SA
      <1，S，SB> //S→1SB
      <2，S，ε> //S→2
      <0，A，ε> //A→0
      <1，B，ε> //B→1
      在产生式比较简单的情况下，可以使用这种方法。

接收语言

L={ω｜ w∈{a,b}*，且 b 的个数是 a 的两倍}

思路：

a) 读入一个 a，压入 AA
b) 读入一个 b，若栈顶为 A，则弹出 A，抵消。若栈顶为 B，则需要先欠着，压入一个 B
c) 为了消除欠下的 B，读入 a 时，若栈顶为 BB，则弹出 BB
d) 修改规则（a），添加栈顶为 A 的限制条件，还要考虑栈顶元素 Z0 的情况.

解 1：

<a， Z0， AAZ0>
<b， Z0， BZ0>
<a， A， AAA>
<a， BB， ε>//广义 PDA 的栈顶可以是一个串，与一般 PDA 的多条规则是等价的
<b， A， ε>
<b， B， BB>
<ε， Z0， ε>

以上都是确定 PDA 的例子，下面举一个不确定 PDA 的例子。

对于不确定的 PDA，如果存在一个可能的扫描过程使得串被接受，则称该串可被此 PDA 接受，这个定义类似于 NFA。

接收语言

$L=\{a^n b^m \mid 0<=n<=m<=2n\}$　一个 a 配一个或两个 b

思路：读入一个 a，既可以压入 A，也可以压入 AA，相同的状态有着不同的动作，也就是不确定性所在。此处构造一单态 PDA，多态可自行思考。

<a, Z0, AZ0>

<a, A, AA>

<a, A, AAA>

<b, A, ε>

<ε, Z0, ε>

真题解析

五、构造 PDAM 接受语言 $L=\{a^n b^{2m} a^n \mid n, m>=1\}$。（10 分）

思路：此题采用确定的多态机，则非常简单。

a) 读入 a，压栈，考虑 Z0
b) 读到 b，切换状态，注意 b 的数量是偶数，采用读一个弹一个的双状态法
c) 再读入 a，出栈，考虑 Z0

<Q0, a, Z0, Q0, AZ0>

<Q0, a, A, Q0, AA>

<Q0, b, A, Q1, BA>

<Q1, b, B, Q2, ε>

<Q2, b, A, Q1, BA>

<Q2, a, A, Q3, ε>

<Q3, a, A, Q3, ε>

<Q3, ε, Z0, Q3, ε>

TM 相关

图灵机是最强的计算设备模型。与 PDA 的最大区别是 TM 可以重写带上的符号，因而可以看做是拥有无限的存储。TM 拥有两个极其重要的性质：1.每个过程都是有穷的；2.过程由离散的步组成，且每一步都可机械执行。因此，TM 会不停的计算，当输入被接受时。如果未定义转换函数，TM 会卡死，将产生拒绝。否则，TM 会一直运行下去，永不停机。

考试对于 TM 的考察由一个单道机模型和多道机（三道）组成，单道机需要熟练掌握三个模型（见下文），多道机需要掌握二进制加减法模型。因为 TM 的构造已经趋近于真正的计算机汇编语言，所以必须要写注释，必须要写注释，不写注释会影响阅卷老

师的阅读，会丢掉很多的分!!! 下面通过例题介绍单道 TM 的计算模型。

II. 构造单道图灵机接收语言 $L = \{a^i b^j c^{i+j} | i, j > 0\}$

拿到 TM 题目以后，第一步应该是分析。观察题目发现这是一个加法题。显然，可以先检测 a+b+c+序列的合法性，然后读一个 a 转化一个 c，读一个 b 转化一个 c。a 和 b 除了顺序的区别是同质的。读写头的移动是具体分析的重点，也是边界情况分析的主线。具体的 TM 规则不会详细描述。

1. 检查 a+b+c+的格式串的正确性，此时读写头在 c 末尾处。
2. 左寻 a 或 b，找到一个就将 a 或 b 改成#，右寻 c，将 c 改成!。待所有的 c 转化完后，此时状态回到左寻 a 或 b 的状态，此时读到开始符。读到开始符说明 c 已经转化完毕，需要检查转换后的串的合法性。
3. 若串合法，则转换后串为#+!+的格式，检查该串是否合法即可。

III. 构造单道图灵机接收语言 $L = \{a^i b^j c^{i \times j} | i, j > 0\}$

基本思路：每转化一个 a，就在 c 序列中转换 i 个 b

1. 读写头到 a 序列末端，进入 del_a 态。
2. 将 a