习题课
(1-3章)

1、复习

2、习题讲解

第一章

概论

(介绍名词术语、了解编译系统的结构和编译过程)

1.2 编译过程

所谓编译过程是指将高级语言程序翻译为等价的目标程序的过程。

习惯上是将编译过程划分为5个基本阶段:

词法分析

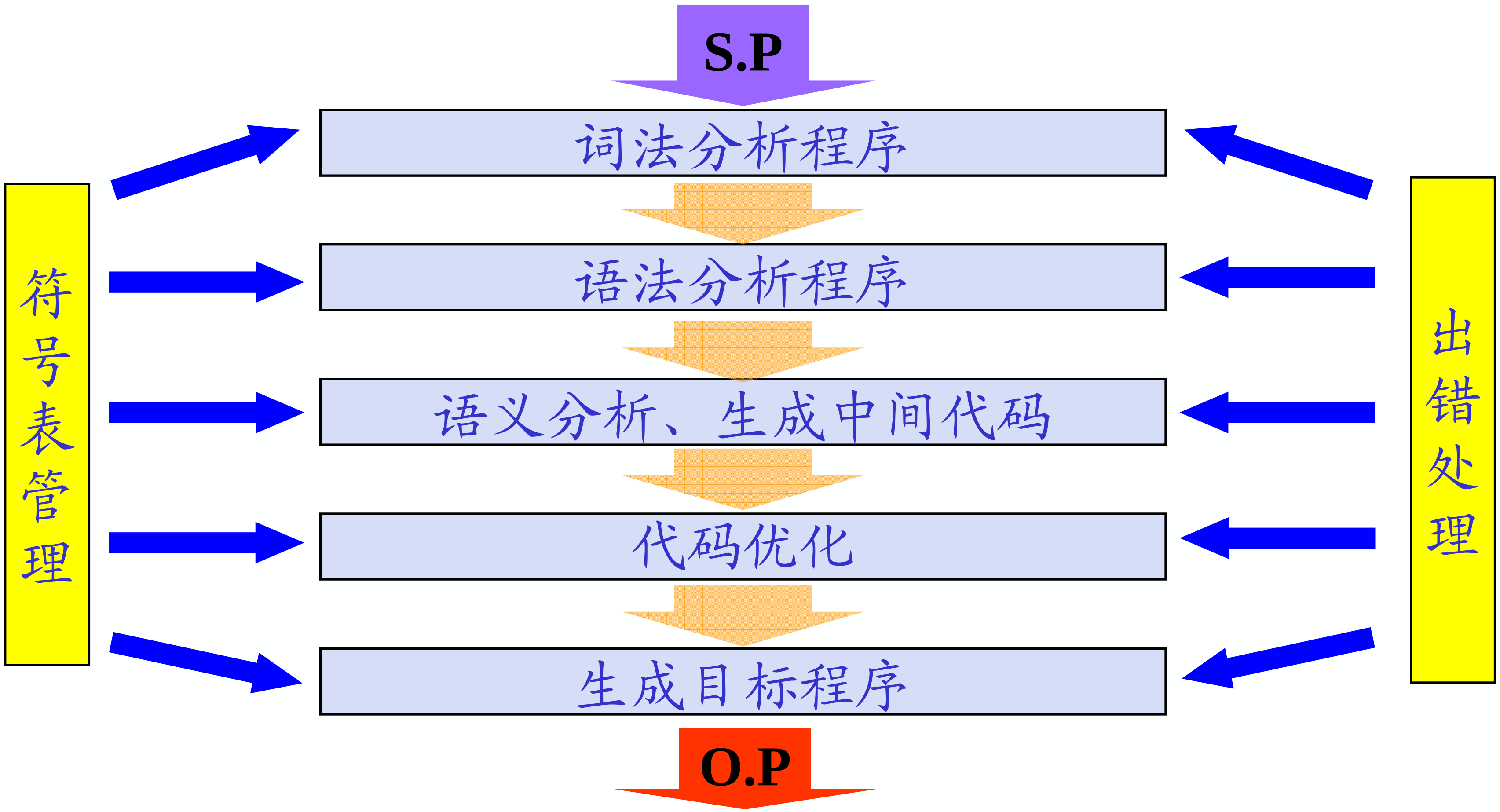
语法分析

语义分析、生成中间代码

代码优化

生成目标程序

典型的编译程序具有7个逻辑部分



第二章

- 掌握符号串和符号串集合的运算、文法和语言的定义
- 几个重要概念：递归、短语、简单短语和句柄、语法树、文法的二义性、文法的实用限制等。
- 掌握文法的表示：BNF、扩充的BNF范式、语法图。
- 了解文法和语言的分类

第三章: 词法分析

3.1 词法分析的功能

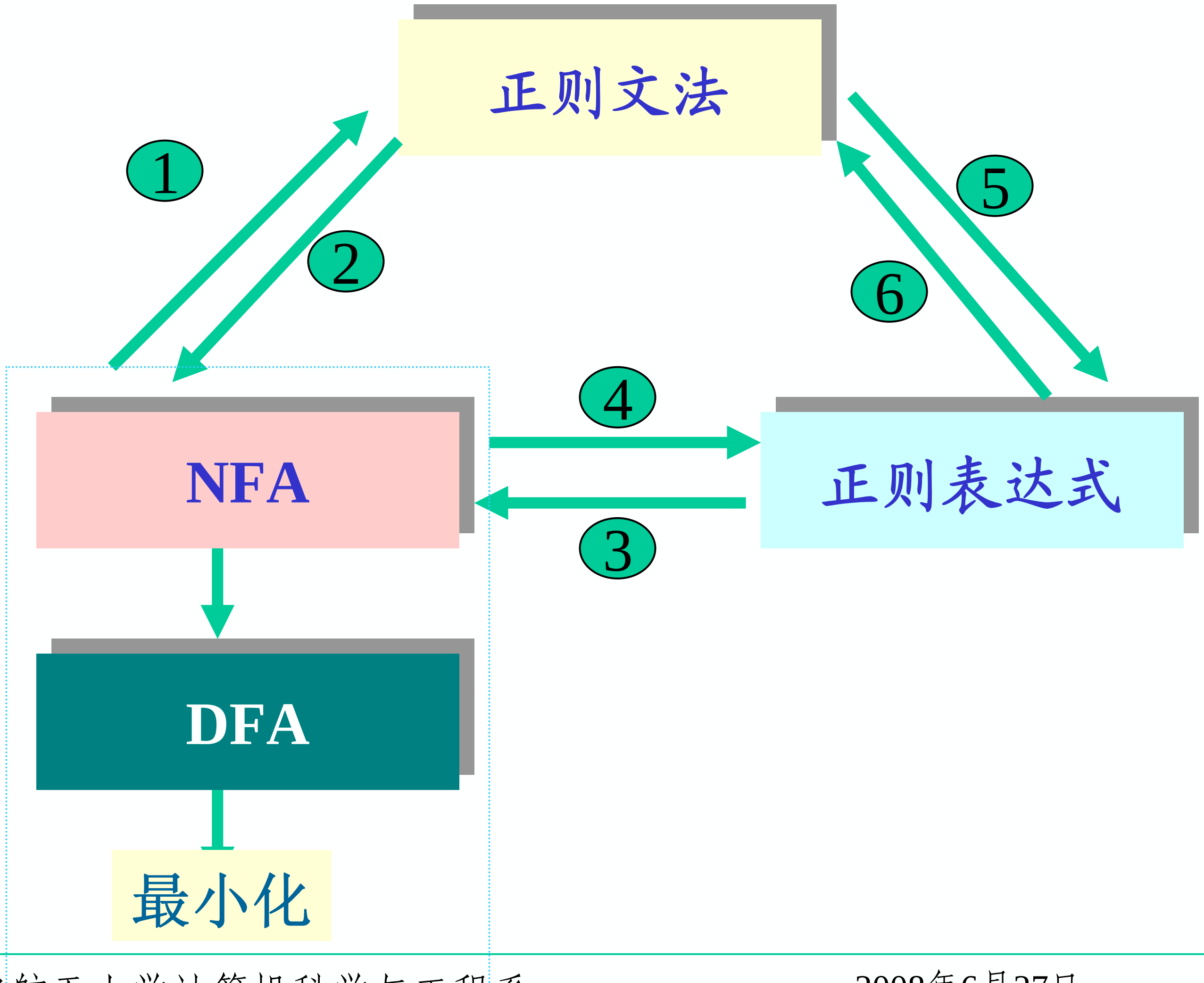
3.2 词法分析程序的设计与实现

– 状态图

3.3 词法分析程序的自动生成

– 有穷自动机、LEX

补充



习题1－3章

第一章

- 2．典型的编译程序可划分为哪几个主要的逻辑部分？各部分的主要功能是什么？

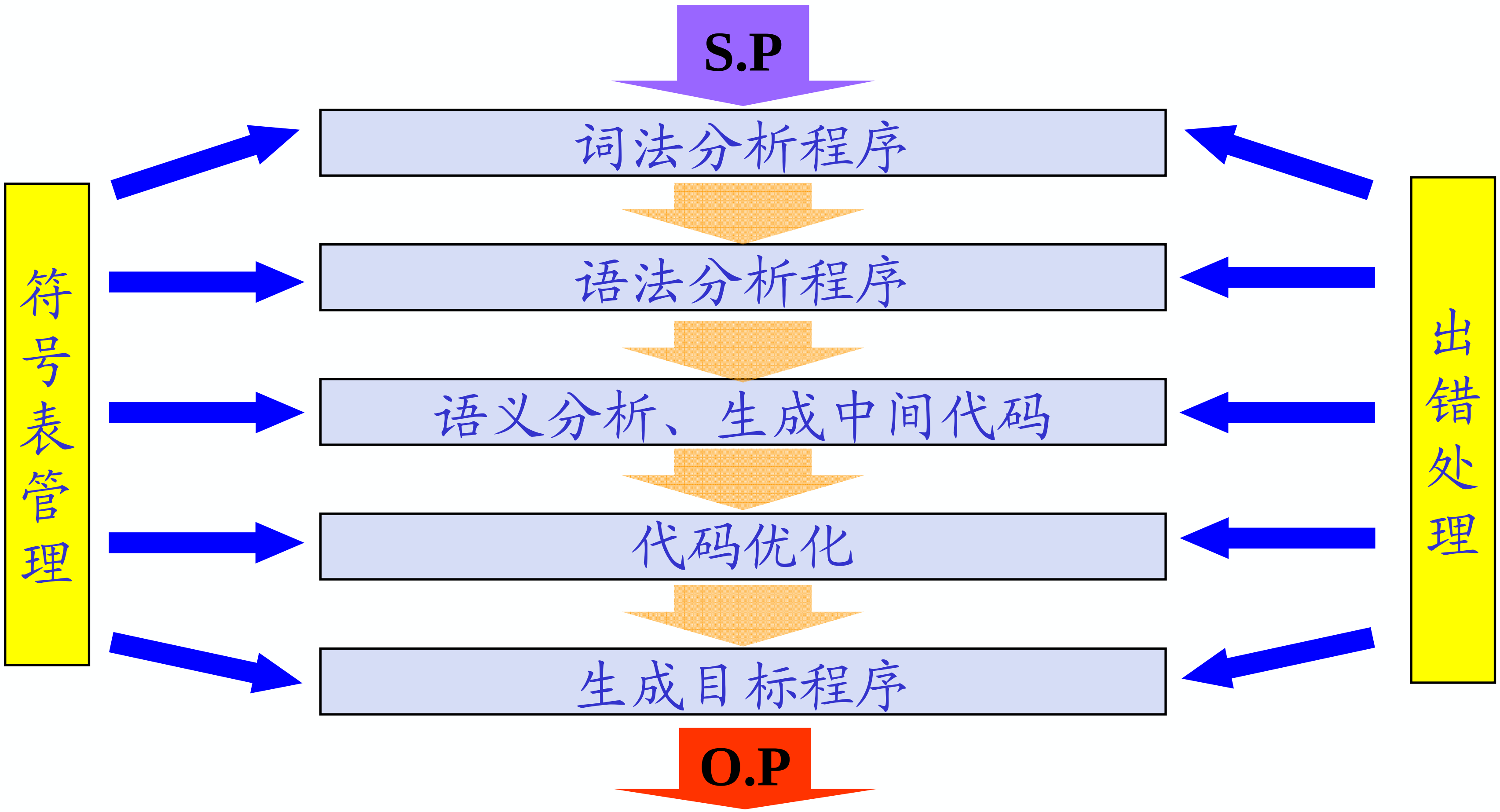
1.2 编译过程

所谓编译过程是指将高级语言程序翻译为等价的目标程序的过程。

习惯上是将编译过程划分为5个基本阶段:

- 词法分析
- 语法分析
- 语义分析、生成中间代码
- 代码优化
- 生成目标程序

典型的编译程序具有7个逻辑部分



P19：4. 试证明：$A^{+}=AA^{*}=A^{*}A$

证：∵ $A^{*}=A^{0}\cup A^{+}$，$A^{+}=A^{1}\cup A^{2}\cup\ldots\cup A^{n}\cup\ldots$

得：$A^{*}=A^{0}\cup A^{1}\cup A^{2}\cup\ldots\cup A^{n}\cup\ldots$

∴ $AA^{*}=A\,(A^{0}\cup A^{1}\cup A^{2}\cup\ldots\cup A^{n}\cup\ldots)$

$=AA^{0}\cup AA^{1}\cup AA^{2}\cup\ldots\cup A\;A^{n}\cup\ldots$

$=A\cup A^{2}\cup A^{3}\cup A^{n+1}\cup\ldots$

$=A^{+}$

同理可得:

$A^{*}A=(A^{0}\cup A^{1}\cup A^{2}\cup\ldots\cup A^{n}\cup\ldots)\,A$

$=A^{0}A\cup A^{1}A\cup A^{2}A\cup\ldots\cup A^{n}A\cup\ldots$

$=A\cup A^{2}\cup A^{3}\cup A^{n+1}\cup\ldots$

$=A^{+}$

因此：$A^{+}=AA^{*}=A^{*}A$

P26:1.设G[〈标识符〉]的规则是：

〈标识符〉::=a|b|c|

〈标识符〉a|〈标识符〉c|

〈标识符〉0|〈标识符〉1

试写出V_T和V_N，

并对下列符号串a,ab0,a0c01,0a,11,aaa给出可能的一些推导。

解：V_T ={a, b, c, 0, 1}， V_N ={〈标识符〉}

(1) 不能推导出ab0,11,0a

(2)〈标识符〉=>a

(3)〈标识符〉=>〈标识符〉1

=>〈标识符〉01

=>〈标识符〉c01

=>〈标识符〉0c01

=> a0c01

(4)〈标识符〉=>〈标识符〉a

=>〈标识符〉aa

=>aaa

P26： 2．写一文法，其语言是偶整数的集合

常见错误：

1.终结符集中没有奇数。

2.如下定义：<偶整数> => <数字串> <偶数字>，

<数字串> => <数字> | <数字串> <数字>

<数字串>不能=> ε 。

3.忽略负偶数。

作法一：<偶整数>::=2×<整数>

<整数>::=<数字串><数字>

<数字串>::=<数字>

<数字>::=0|1|2|3|4|5|6|7|8|9

作法二：z=偶整数

G(z)=0|2|2Z|2(Z+1)|2(Z-1)

或：G(Z)=0|2|Z+2|Z-2

解：G[<偶整数>]：

<偶整数>::= <符号> <偶数字>| <符号><数字串><偶数字>

<符号> ::= + | — | ε

<数字串>::= <数字串><数字>|<数字>

<数字> ::= <偶数字>| 1 | 3 | 5 | 7 | 9

<偶数字> ::=0 | 2 | 4 | 6 | 8

3. 写一文法，使其语言是偶整数的集合，但不允许有以0 开头的偶整数。

Compiler

- 4. 设文法G的规则是：

〈A〉 ::=b<A>| cc

试证明：cc, bcc, bbcc, bbbcc∈L[G]

证：（1）〈A〉 =>cc

（2）〈A〉 =>b〈A〉 =>bcc

（3）〈A〉 =>b〈A〉 =>bb〈A〉 =>bbcc

（4）〈A〉 =>b〈A〉 =>bb〈A〉 =>bbb〈A〉 =>bbbcc

又∵cc, bcc, bbcc, bbbcc∈Vt*

∴由语言定义，cc, bcc, bbcc, bbbcc∈L[G]

5 试对如下语言构造相应文法：

（1）{ a(b^n)a | n=0,1,2,3,……},其中左右圆括号为终结符。

（2）{（a^n）（b^n）| n=1,2,3,……}

解：（1）文法[G〈S〉]：

S::= a(B)a

B::= bB | ε

(2) 文法[G〈S〉]：--错了，两个n不等

S ::= (A)(B)

A::= aA|a

B::= bB|b

S ::= (B)

B::= aBb|a)(b

7. 对文法G_3[〈表达式〉]

〈表达式〉::=〈项〉|〈表达式〉+〈项〉|〈表达式〉-〈项〉

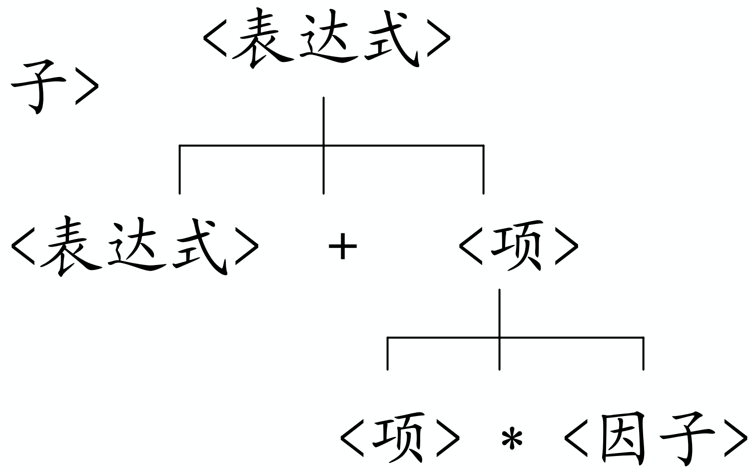
〈项〉::=〈因子〉|〈项〉*〈因子〉|〈项〉/〈因子〉

〈因子〉::=(〈表达式〉)| i

列出句型〈表达式〉+〈项〉*〈因子〉的所有短语和简单短语。

<表达式> => <表达式> + <项>

=> <表达式> + <项> * <因子>



短语有：

〈表达式〉+〈项〉*〈因子〉和〈项〉*〈因子〉

简单短语是： 〈项〉*〈因子〉

8 文法V::= aaV | bc的语言是什么？

- 解：L（G[V]）= $\{a^{2n}bc \mid n=0,1,2,\ldots\ldots\}$

$$V \Rightarrow aaV \Rightarrow aaaaV \Rightarrow \ldots. \Rightarrow a^{2n}bc \quad (n \geq 1)$$

$$V \Rightarrow bc \quad (n=0)$$

P33:

5. 已知文法G[E]:　E::=ET+ | T

T::=TF* |F

F::=FP↑ |P

P::=(E) | i

有句型TF*PP↑+,

问此句型的短语，简单短语，和句柄是什么？

常见错误：

1.认为下列是短语：TF*PP↑、PP↑+、+

2.认为TF*PP↑+不是短语。

3.认为PP↑是简单短语。

4.不能正确识别句柄

解：此句型的短语有：TF*PP↑+，TF*，PP↑，P

简单短语有：TF*，P

句柄是：TF*

8. 证明下面的文法G是二义的:

S::= iSeS | iS | i

证: 由文法可知iiiei是该文法的句子,

又由文法可知iiiei有两棵不同的语法树。

所以该文法是二义性文法。

P39

3．试构造一个从文法中删除**无用符号**的算法。

多余规则：（1）在推导文法的所有句子中，始终用不到的规则。即该规则的左部非终结符不出现在任何句型中。--不可达

（2）在推导句子的过程中，一旦使用了该规则，将推不出任何终结符号串。即该规则中含有推不出任何终结符号串的非终结符。--不活动

若还有U::=a，则此规则并非多余

例如给定G[Z]，若其中关于U

U::=xUy

该规则是多余规则。

若某文法中无有害规则或多余规则，则称该文法是压缩过的。

判断U是否为无用符号

条件

- 检查文法中每一条规则左部的每个非终结符U是否满足下述两个条件:
  - 条件1: $Z \overset{*}{\Rightarrow} xUy$, $x,y \in V^*$,Z为识别符号

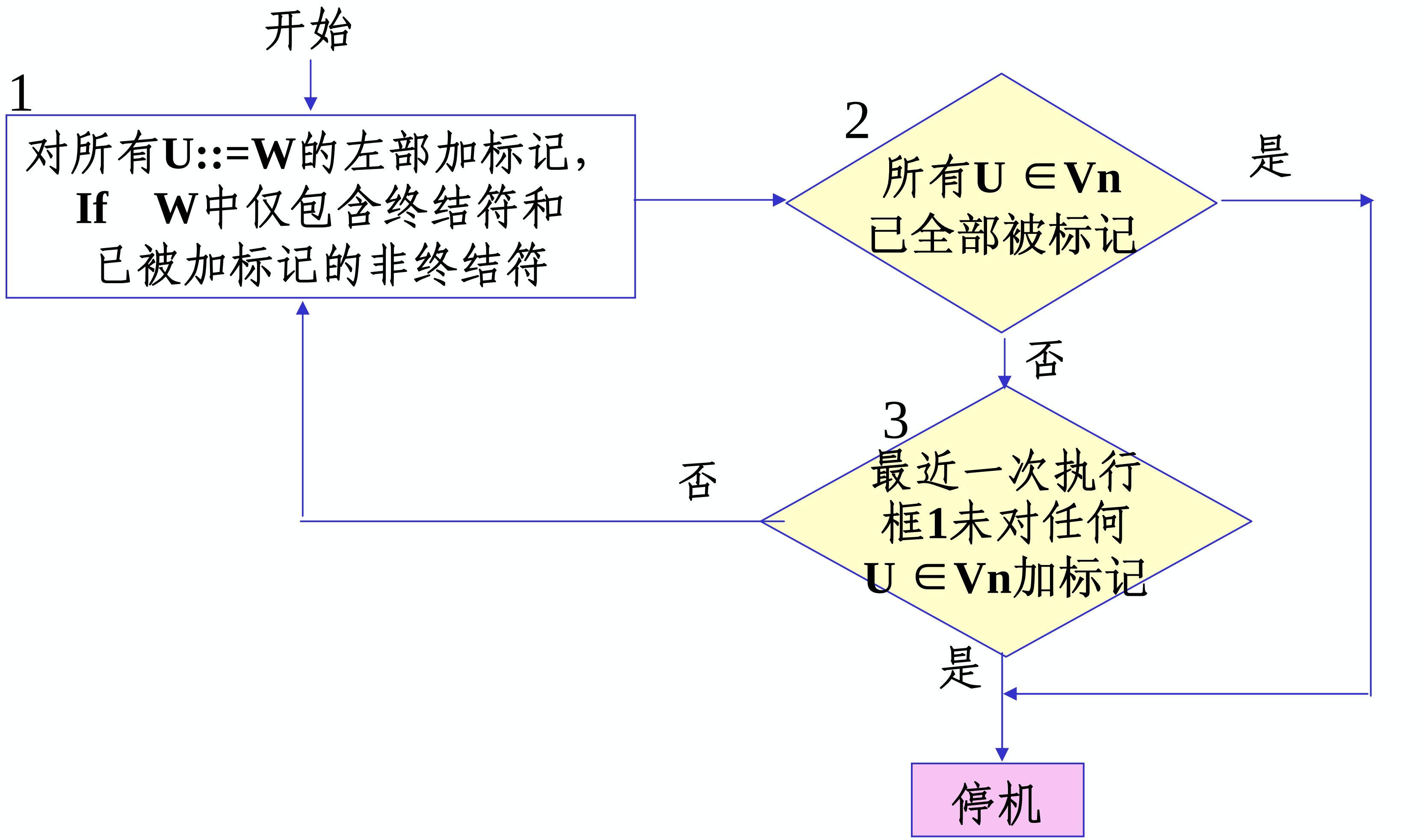
    （即：要求U必须出现在某个句型中）
  - 条件2：$U \overset{+}{\Rightarrow} t$, $t \in V_t{}^*$

    （即：能从U推出终结符号串）

判断条件2的算法

- 算法思路：对满足条件的符号作标记，最后没有标记的符号为无用符号。
- 加标记的方法如下:
  - 对于形如U::=u ($u \in V_t$ *)的规则，左部U自然满足条件2，因此对U标记。
  - 当一规则U::=W的右部W中包含的非终结符全被加标记时，即满足条件2时，左部U也满足条件2，对U加标记。

流程图



开始
1
对所有U::=W的左部加标记，
If W中仅包含终结符和
已被加标记的非终结符
2
所有U ∈Vn
已全部被标记
是
否
3
最近一次执行
框1未对任何
U ∈Vn加标记
否
是
停机

举例

- 例：对于文法

第一遍

第二遍

第三遍

* Z ::= Be

* A ::= Ae　　* A ::= e

B ::= Ce　　* B ::=Af

C ::= Cf

* D ::= f

所以：**B ::= Ce 和C ::= Cf 是多余**

也多余，未查出。不满足条件1，不是条件2

练习

- 写一文法，使其语言是偶整数的集合，但不允许有以0 开头的偶整数。

P67.1. 画出下述文法的状态图

〈Z〉::=〈B〉e

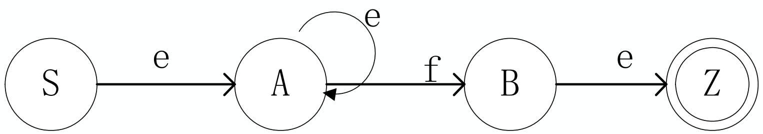
〈B〉::=〈A〉f

〈A〉::= e |〈A〉e

使用该状态图检查下列句子是否是该文法的合法句子

f，eeff，eefe

解:



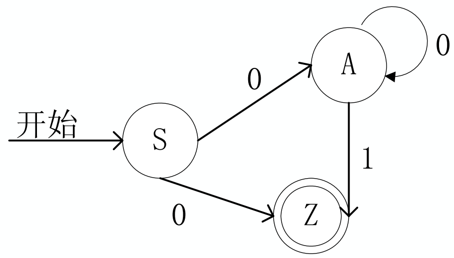
f，eeff不是该文法的合法句子，eefe是该文法的合法句子

2、有下列状态图，其中S为初态，Z为终态。

（1） 写出相应的正则文法：

（2） 写出该文法的V，V_n和V_t；

（3） 该文法确定的语言是什么？



解：（1） $Z \rightarrow A1 \mid 0$　　$A \rightarrow A0 \mid 0$

（2） V={A, Z, 0, 1}　　Vn={A, Z}　　Vt={0, 1}

（3） L（G[S]）= {0或0^n1, $n \geqslant 1$}

L（G[S]）= {0|00*1}

- 5．令A，B，C是任意正则表达式，证明以下关系成立:

  $A|A=A$

  $(A^*)^*= A^*$

  $A^*= \varepsilon \mid AA^*$

  $(AB)^*A = A(BA)^*$

  $(A|B)^* = (A^*B^*)^*= (A^*|B^*)^*$

- 证明：

(1) $A \mid A = \{x \mid x \in L(A)$ 或 $x \in L(A)\} = \{x \mid x \in L(A)\} = A$

(2) $(A^*)^* = (A^*)^0 \cup (A^*)^1 \cup (A^*)^2 \cup \ldots \cup (A^*)^n$

$= \varepsilon \cup (A^0 \cup A^1 \cup A2 \cup \ldots \cup An) \cup (A1 \ldots)$

$= \varepsilon \cup A0 \cup A^1 \cup A2 \cup \ldots \cup An$

$= A^*$

(3) ε ｜AA*所表示的语言是:

{ ε }∪LA·LA*

　＝LA0∪LA（LA0∪LA1∪LA2∪…）

＝LA0∪LA1∪LA2∪…＝LA*

故 ε ｜AA*＝A*

(4)

(LALB)*LA=({ ε }∪L(A)LB∪LALBLALB∪LALBLALBLALB∪…) LA

=LA∪LALBLA∪LALBLALBLA∪LALBLALBLALB∪LA…

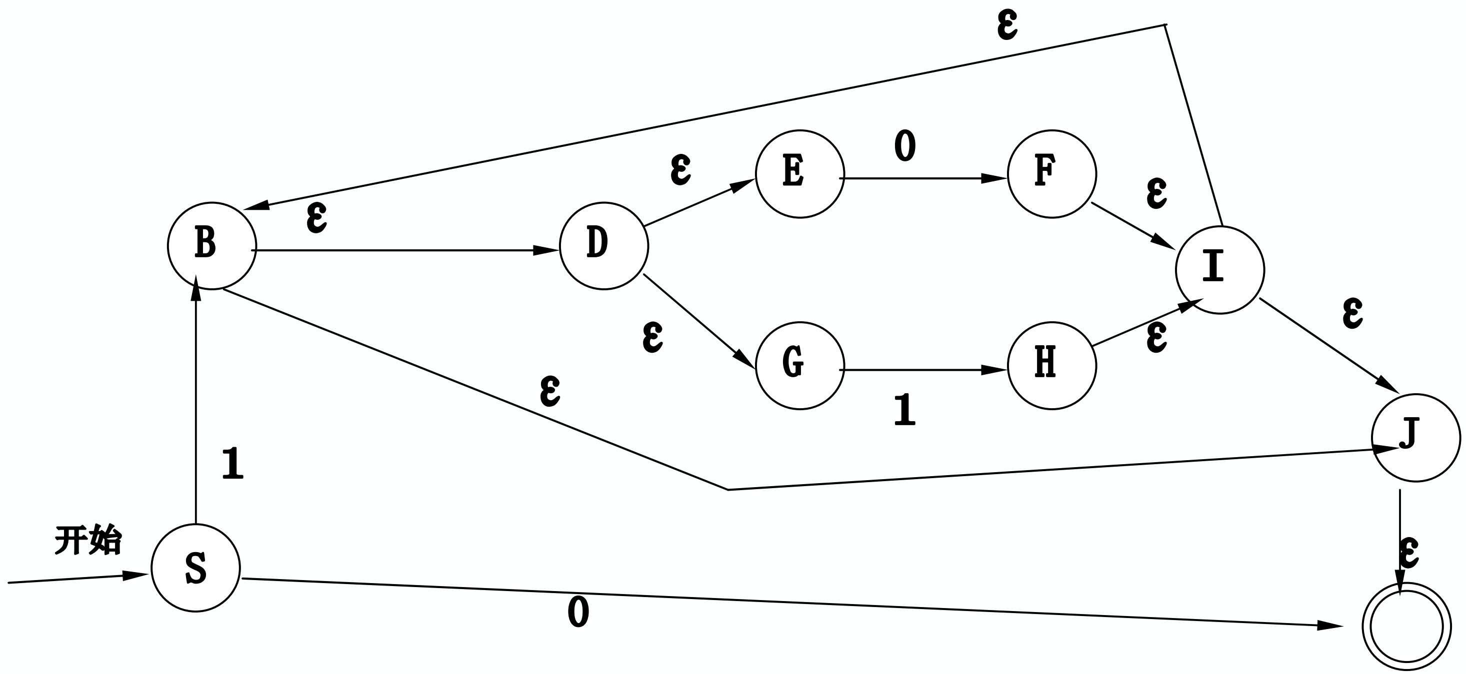
- = LA∪({ ε }∪LBLA∪LBLALBLA∪…)
- = LA(LBLA)
- ∴(AB)*A=A(BA)*

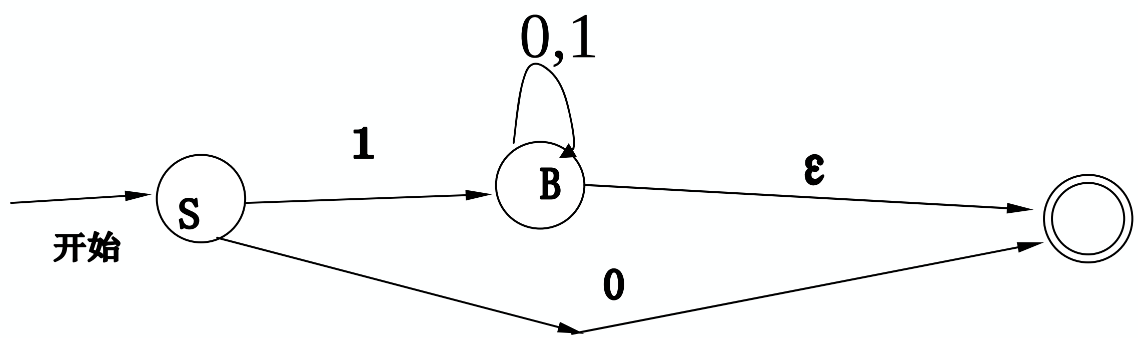
- (5)三个表达式所描述的语言都是LALB中任意组合
- ∴（A|B）*=(A*B*)=(A*|B*)*

6、构造下列正则表达式相应的DFA

- （1） 1（0|1）*|0
- （2） 1（1010*|1（010）*1）*0

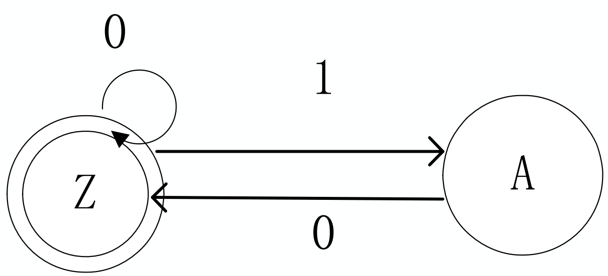
- （1）与1（0|1）*|0对应的NFA为：





8 把图3.24的（a）和（b）分别确定化

10、构造一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。



P80习题

a) 部分同学对不能正确的设计语法分析程序（不会设计递归程序）。

b) 有些人在求 first 集 和 follow 集时出错较多。

c) 在学习递归向下分析法时，不少人习惯于把文法最后归结为一个很长的右部，例如：S ::= (c{c})|dc{c}e, 实际上写成 如 S::= A|dAe ; A::= cA|c 的形式更易于写出分析程序，也更合理。

- P87习题
  a) 对LL(1), SLR(1),算符优先文法的定义理解不清。很多同学做题的时候根本不考虑文法是否满足这几种文法（LL(1), SLR(1),算符优先文法）的要求，以及是否需要改写文法，便开始做题了。
  b) 几种文法（如LL(1), SLR(1),算符优先）的判断应该归纳一下，有些人在判断时条件不充分就得出结果。
  c) 对于构造LL(1)分析表的步骤很模糊，很多同学写出来的分析表不够规范。

- P88页证明题，证明的人很少。证明思路仍然有问题，建议此题讲解一下。
- 在涉及到有关是否为LL(1)文法的证明时，约1/4的同学不知道正确证明的方式。

- P100、P108、P115习题
  a) 短语、素短语概念不清。
  b) 部分同学不会设计SLR(1)分析器。
  c) 活前缀概念不清，很多同学不能正确求出活前缀的有效项目集。

第五章

2、试分别构造一个符号串翻译文法，它将由一般中缀表达式文法所定义的中缀表达式翻译成波兰前缀表达式和波兰后缀表达式。

解：构造的波兰前缀表达式为：

E->@+E+T
T->F
E->T
F->@ii
T->@*T*F
F->(E)

构造的波兰后缀表达式为：

E->E+T@+
T->F
E->T
F->i@i
T->T*F@*
F->(E)

3、构造一符号串翻译文法，它将接受由0和1组成的任意输入符号串，并产生下面的输出符号串：

a) 输入符号串倒置。

b) 空符号串。

c) 输入符号串本身。

d) 符号串0^n1^m。

a) S->0S@0
S->1S@1
S->ε

b) S->T@ε
T->1T|0T
T->0|1|ε

E->1@1E
E->ε

d) S->0@0S
S->1S@1
S->ε

h

- P138习题
  a) 第2题很多同学没做对。

  p185习题
  第六题：很多同学不清楚栈式符号表的由那些部分组成。