



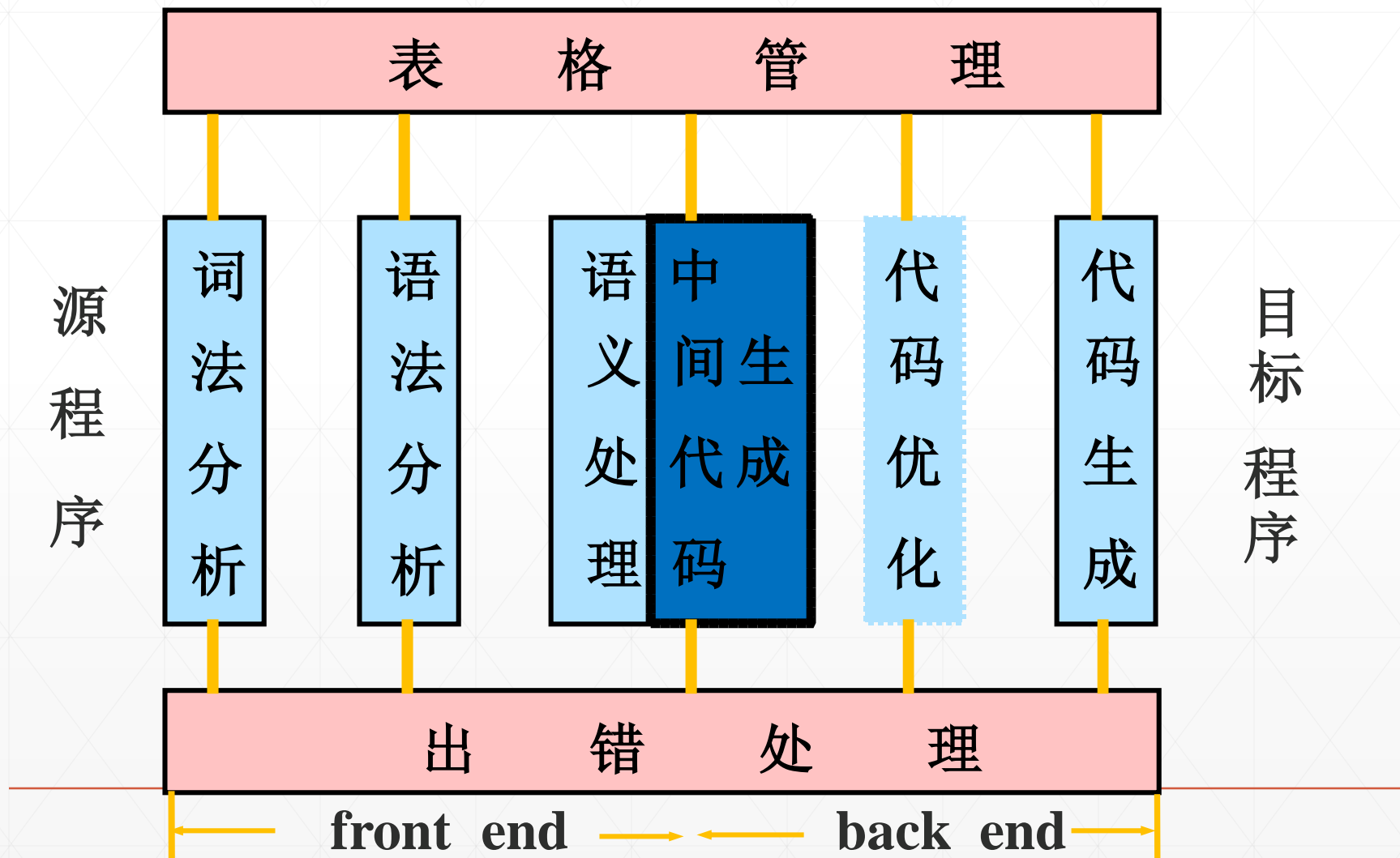
# 编译原理与设计：IR

北京理工大学 计算机学院

---



# 中间代码生成：概览





# 中间代码

- 介于源语言和目标代码之间一种代码。
- 引入中间代码的目的
  - 方便生成目标代码
  - 便于优化
  - 便于移植

**中间语言的设计与应用既要考虑从源语言到目标语言的翻译跨度又要考虑目标机的指令集特点。**

---



# 中间代码

中间语言 { 逆波兰式  
N元式(三元式、间接三元式四元式)  
图

---



# 中间代码：逆波兰式（中缀-后缀）

## ■ 形式定义

$$\underbrace{e_1 e_2 \dots e_n}_{\text{运算对象}} \theta \quad (n \geq 1)$$

运算对象      运算符

## ■ 例如：

$$a * b \quad \Rightarrow \quad ab^*$$

$$- a \quad \Rightarrow \quad a@ \quad (@ \text{表示单目 “-”})$$

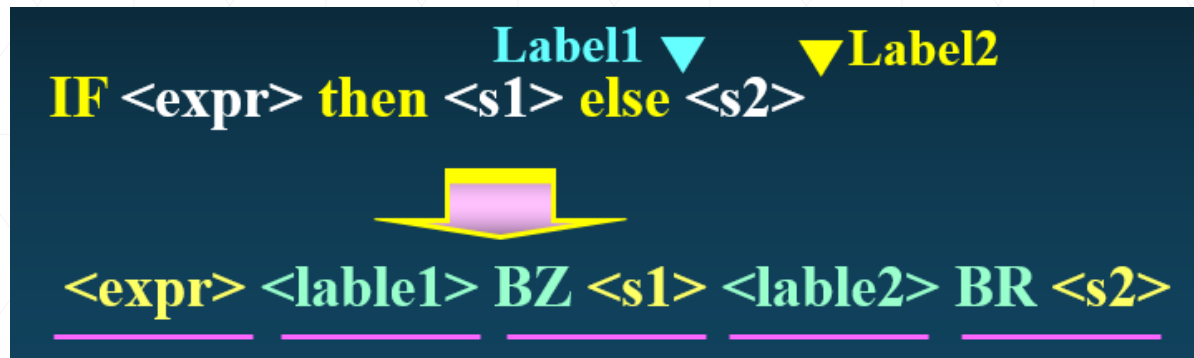
$$a + b * (c + d) * (e + f) \quad \Rightarrow \quad abcd + ^* ef + ^* +$$

---



# 中间代码：逆波兰式（中缀-后缀）

## ■ 高级语言语句表示



**BZ**、**BR**：引入的两个操作符。

**BZ**是二目操作符，如果<expr>的计算结果为假，则产生一个到<lable1>的转移。 <lable1>是<s2>的头一个符号。

**BR**则是一个单目操作符，它产生一个到<lable2>的转移。



## 中间代码：逆波兰式（中缀-后缀）

- 例：设有如下C程序片段

```
{  
    int k;  
    i=j=0;  
h:  k=100;  
    if (k>i+j) { k--; i++; }  
    else k=i*2-j*2;  
    goto h;  
}
```

---



# 中间代码：逆波兰式（中缀-后缀）

- 例：设有如下C程序片段

```
{  
    int k;  
    i=j=0;  
h:  k=100;  
    if (k>i+j) { k--; i++; }  
    else k=i*2-j*2;  
    goto h;  
}
```

```
(1)    block  
(2)    i j 0 ==  
(7)    h:  
(8)    k 100 =  
(11)   k i j +>(28) BZ  
(18)   k k 1 - = i i 1 + =(37) BR  
(29)   k i 2*j2*- =  
(38)   (7) BR  
(39)   blockend
```





# 中间代码：N元式

- N个域的记录结构

( D<sub>1</sub>, D<sub>2</sub>, D<sub>3</sub>, ... , D<sub>n</sub> )  
OP域                  操作对象域

- 常见N元式

三元式、间接三元式 ↔ 双地址机指令形式  
四元式 ↔ 三地址机指令形式



# 中间代码：三元式

- 三元式形式定义

**NO. ( OP, ARG<sub>1</sub>, [ARG<sub>2</sub>] )**

**NO.:**为产生的三元式的顺序编号;

**OP:**是操作符;

**ARG1**和**ARG2**为第一操作数和第二操作数。(也可以是前面某一个三元式的编号, 代表该三元式的计算结果被作为操作数)



# 中间代码：三元式

- 三元式形式定义

**NO. ( OP, ARG<sub>1</sub>, [ARG<sub>2</sub>] )**

- 例如，语句  $X=A+B*C$  的三元式表示为

NO.	OP	ARG1	ARG2
(1)	*	B	C
(2)	+	A	(1)
(3)	=	(2)	X



# 中间代码：三元式

- 例如

**if  $X > Y$  then  $Z = X$  else  $Z = Y + 1$**

- 对应的三元式表示为：

NO.	OP	ARG1	ARG2
(1)	—	X	Y
(2)	BZ	(1)	(5)
(3)	=	X	Z
(4)	BR		(7)
(5)	+	Y	1
(6)	=	(5)	Z
(7)			



## 中间代码：三元式

■ 例如

$$(a+b)^2 + (a+b)^3$$

■ 对应的三元式为：

①	<u>+, a, b</u>
②	↑, ①, 2
③	<u>+, a, b</u>
④	↑, ③, 3
⑤	+, ②, ④

其中：“↑”表示幂运算。



# 中间代码：间接三元式

## ■ 间接三元式形式定义

间接码表 + 三元式

$(a+b)^2 + (a+b)^3$  的间接三元式

间接码表		三元式	
控制三元式代码执行顺序 ↓	①	①	+, a, b
	②	②	↑, ①, 2
	①	③	↑, ①, 3
	③	④	+, ②, ③
	④		



# 中间代码：四元式

- 四元式形式定义

NO. ( OP, [ARG<sub>1</sub>], [ARG<sub>2</sub>], Result )

+	a	b	T <sub>1</sub>
↑	T <sub>1</sub>	2	T <sub>2</sub>
↑	T <sub>1</sub>	3	T <sub>3</sub>
+	T <sub>1</sub>	T <sub>2</sub>	T <sub>4</sub>

\*\* 注：T<sub>i</sub>是临时变量。



## 中间代码：图/树

一个三元式	↔	一棵子树
OP	↔	子树根
ARG1	↔	子树左叶节点
ARG2	↔	子树右叶节点



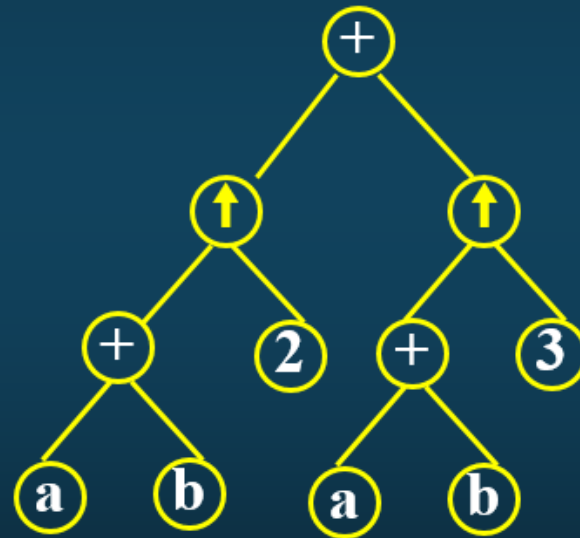


## 中间代码：图/树

$$(a+b)^2 + (a+b)^3$$

三元式

①	+, a, b
②	↑, ①, 2
③	↑, ①, 3
④	+, ②, ③





## 中间代码：几种表示间关系

