0型文法：$P \in \alpha \to \beta$，$\alpha \in V^+$ 并至少一个非终结符。

1型 $P \in \alpha \to \beta$，除 $S \to \varepsilon$ 外，均 $|\beta| \geq |\alpha|$，if $S \to \varepsilon$，S不出现在右部

1' $P \in \alpha \to \beta$，除 $S \to \varepsilon$ 外，有 $\alpha A \beta \to \alpha \gamma \beta$，$A \in V_N$，$\gamma \in V^+$

$\alpha, \beta \in V^*$

2型 $A \to \beta$，$A \in V_N$，$\beta \in V^*$

3型 $A \to aB$，$A \to \alpha$，$A, B \in V_N$，$\alpha \in V_T^*$

$L(G) = (V_N, V_T, P, S)$

构造 ε 上下文无关文法

(1) $V_0 = \{A \mid A \in V_N \land A \xrightarrow{+} \varepsilon\}$

(2) 所有 if $B \to \alpha_0 B_1 \alpha_1 B_2 \cdots B_k \alpha_k$ 中 $B_i \in V_0$，将 B_i 用 ε 和 B_i 替代

(3) 删除 ε 产生式

(4) if $S \to \varepsilon$，删除并 + $S' \to \varepsilon \mid S$

DFA NFA $M=(S,\Sigma,f,s_0,Z)$

NFA→DFA $M=(S,\Sigma,f,S_0,Z) \Rightarrow M'=(Q,\Sigma,g,I_0,F)$

(1) 取 $I_0=S_0$

(2) while Q中有状态 $I_i=\{s_0,\dots,s_j\}$，且 $f(I_i,a)=\bigcup_{k=0}^{j} f(s_k,a)=I_t$

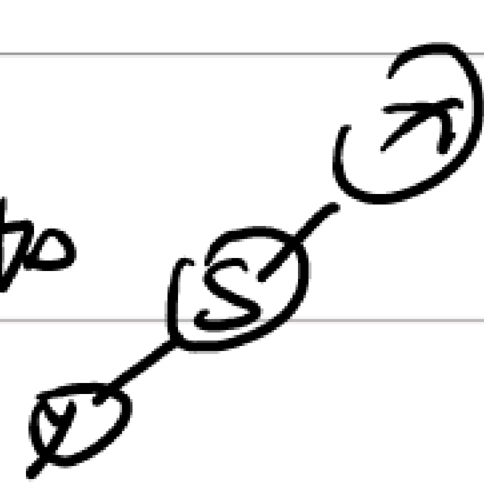
$I_t\notin Q$，I_t加入Q

(3) 取 $F=\{I \mid I\in Q，且 I\cap Z\neq\phi\}$

NFA化简.

(1) 划分 $\Pi_0=\{I_0^1,I_0^2\}$ I_0^1非终态集. I_0^2 终态集.

(2) while $f(I_k^i,a)$ 不全落在Π_k中，一分为二.

FA→Re 起点加 

Re→FA (1) 先拆得 NFA

(2) NFA 带ε的确定化 → DFA.

集合都换为 I → ε-closure(I)

$Rg \to FA$ 右线性文法

$G=(V_N, V_T, P, S) \to M=(Q, \Sigma, f, q_0, Z)$

(1) 新增一个终态 T

$Q = V_N \cup \{T\}$

(2) if P中 $S \to \varepsilon$，则 $Z=\{S, T\}$

(3) if $A_1 \to aA_2$，f加入 $f(A_1, a)=A_2$

if $A_1 \to a$，f加入 $f(A_1, a)=T$.

(4) all $f(T, a)=\phi$.

得 NFA，确定化得 DFA

自上而下语法分析

PDA: $M=(Q,\Sigma,H,\delta,q_0,Z_0,F)$

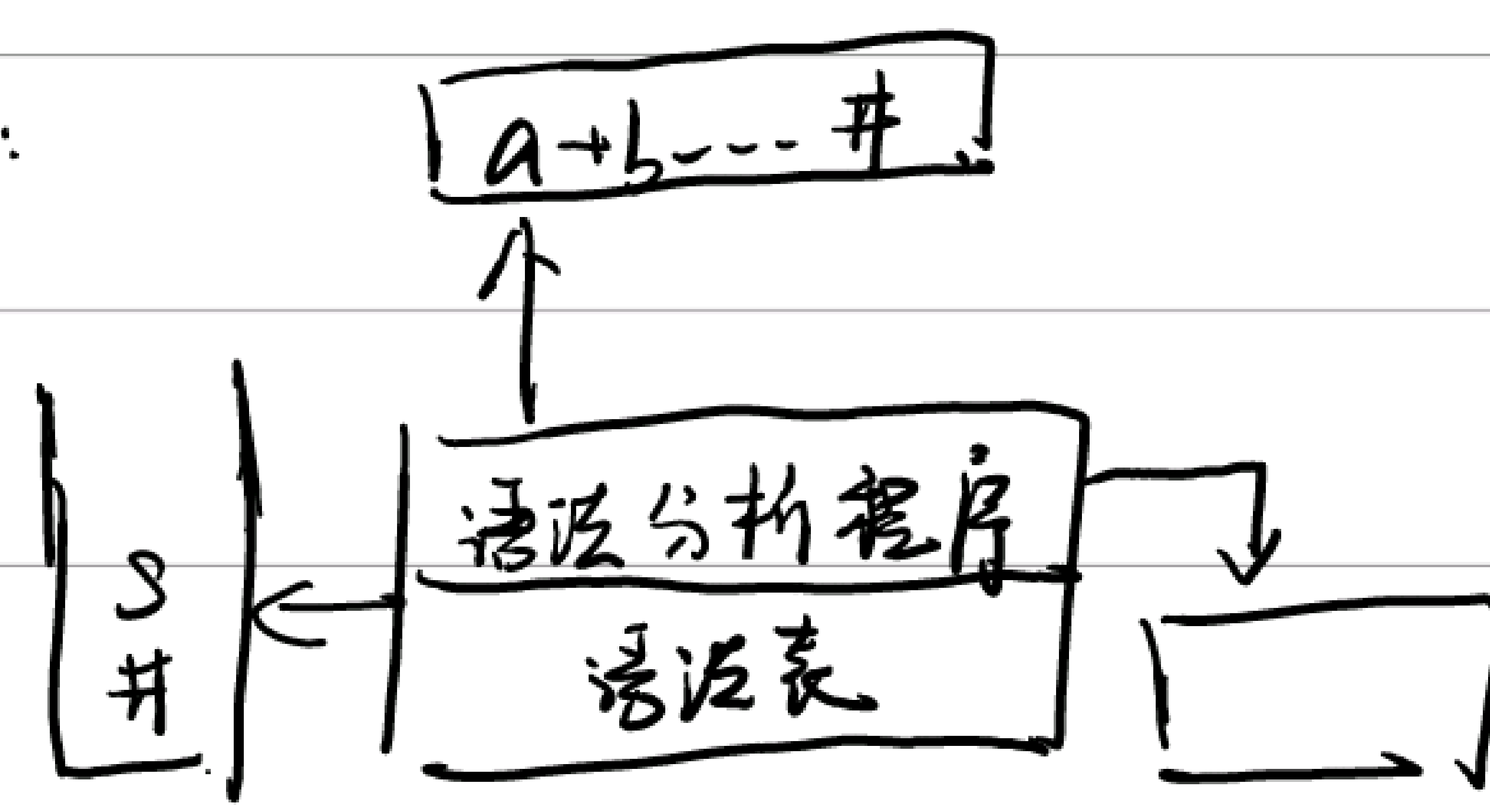
CFG $G=(V_N,\Sigma,P,S) \rightarrow$ PDA $M=(Q,\Sigma,H,\delta,q_0,Z_0,F)$

$Q=F=\{q_0\}$ 1个状态

$\delta(q,\varepsilon,A)=(q,w)$ 推导

$\delta(q,a,a)=(q,\varepsilon)$ 匹配

语法分析：



$X=$ stack.top

(1) $X\rightarrow a$，$\delta(q,\varepsilon,X)=(q,a)$，推导 并记下 $X\rightarrow a$ 编号

(2) $\delta(q,x,x)=(q,\varepsilon)$ 匹配 简单弹出

(3) $\delta(q,a,b)$ 匹配失败，回退上一次推导

问题：左递归，回溯

直接左递归 $P \to P\alpha | \beta$

$$\Rightarrow \left.\begin{array}{l} P \to \beta P' \\ P' \to \alpha P' | \varepsilon \end{array}\right\}$$

间接左递归 $\left\{\begin{array}{l} S \to Qc | c \\ Q \to Rb | b \\ R \to Sa | a \end{array}\right.$

(1) if 有 $P \to P$, $P \to \varepsilon$ then

(2) $V_N = \{P_1, P_2 \cdots P_n\}$

(3) for ($i=1$; $i \le n$; i++)

for ($j=1$; $j<i$; j++)

1. 把 $P_i \to P_j \gamma \longrightarrow P_i \to \delta_1 \gamma | \delta_2 \gamma | \cdots \delta_n \gamma$

2. 消除直接左递归

(4) 除去不可达 V_N

(5) 意义：对产生式 右部最左符号 $= V_N \wedge$ 序号 $>$ 左部 V_N，ignore

else 取代

finally 消除直接左递归

消除回溯：预测，提取公共左因子

First$(\alpha)=\{a \mid \alpha \overset{*}{\Rightarrow} a\cdots, a\in V_T\}$ 开头终结符或 ε

if $\alpha \overset{*}{\Rightarrow} \varepsilon$, $\varepsilon \in$ First(α)

Follow$(A)=\{a \mid S \overset{+}{\Rightarrow} \cdots Aa\cdots, a\in V_T\}$

if $S \overset{+}{\Rightarrow} \cdots A$，则 $\# \in$ Follow(A)

所有句型中在A之后的终结符或#

求 Follow(A)

(1) 对于 S. $\# \in$ Follow(S)

(2) 对于 $B\to\alpha A\beta$. First$(\beta)-\{\varepsilon\} \in$ Follow(A)

(3) 对于 $B\to\alpha A$ 或 $B\to\alpha A\beta \wedge \beta \overset{*}{\Rightarrow}\varepsilon$，把 Follow$(B) \in$ Follow(A)

构造LL(1)

(1) if $A\to\alpha$, $a\in$ First(α), $\delta(q,a,A)=(q,\alpha)$, $M[A,a]=A\to\alpha$

(2) if $A\to\alpha$, $\varepsilon\in$ First(α)，对于 Follow(A) 中 while b. 在 $M[A,b]=A\to\alpha$

(3) else $M[A,a]=$ error.

LL(1) $\Leftrightarrow$ 无二义性 $\Leftrightarrow$ 对于 $A \to \alpha \mid \beta$

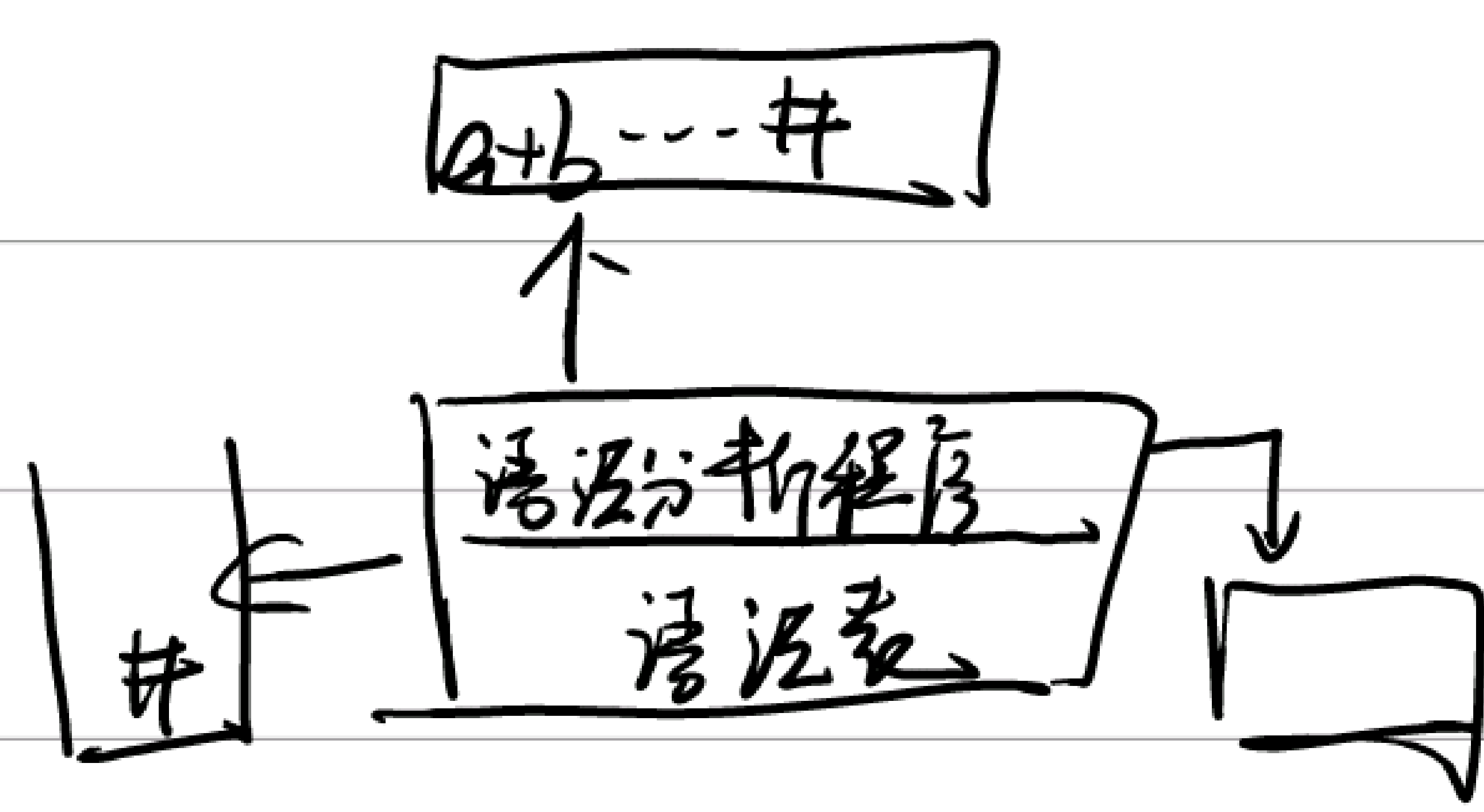
(1) $First(\alpha) \cap First(\beta) = \phi$

(2) if $\varepsilon \in First(\beta)$ then $First(\alpha) \cap Follow(A) = \phi$

有左递归和公共左因子的文法 X LL(1)

LL(1)文法无二义性 二义性文法绝不是LL(1)

优先分析法



简单优先分析法：

(1) if $P \to \cdots XY \cdots$ then $X \doteq Y$

(2) if $P \to \cdots XQ \cdots \wedge Q \overset{+}{\Rightarrow} Y \cdots$ then $X \lessdot Y$

(3) if $P \to \cdots QR \cdots \wedge Q \overset{+}{\Rightarrow} \cdots X \wedge Y \in First\ R$ then $X \gtrdot Y$

NOTE: 比较对象 $\lessdot$ $\gtrdot$ $\doteq$

算符优先分析法

NOTE：先将文法拓展 $S' \to \#S\#$　　#与#无关系

优先表构造

定义关系 (1) $a = b$，当且仅当G中含有 $P \to \cdots ab \cdots$ / $P \to \cdots aQb \cdots$

(2) $a < b$，$P \to \cdots aR \cdots$，$R \xrightarrow{+} b \cdots$ / $R \xrightarrow{+} Qb \cdots$

(3) $a > b$，$P \to \cdots Rb \cdots$，$R \xrightarrow{+} \cdots a$ / $R \xrightarrow{+} \cdots aQ$

定义 $FIRSTVT(P) = \{a \mid P \xrightarrow{+} a \cdots / P \xrightarrow{+} Qa \cdots, a \in V_T, P, Q \in V_N\}$

$LASTVT(P) = \{a \mid P \xrightarrow{+} \cdots a / P \xrightarrow{+} \cdots aQ, a \in V_T, P, Q \in V_N\}$

求 FIRSTVT

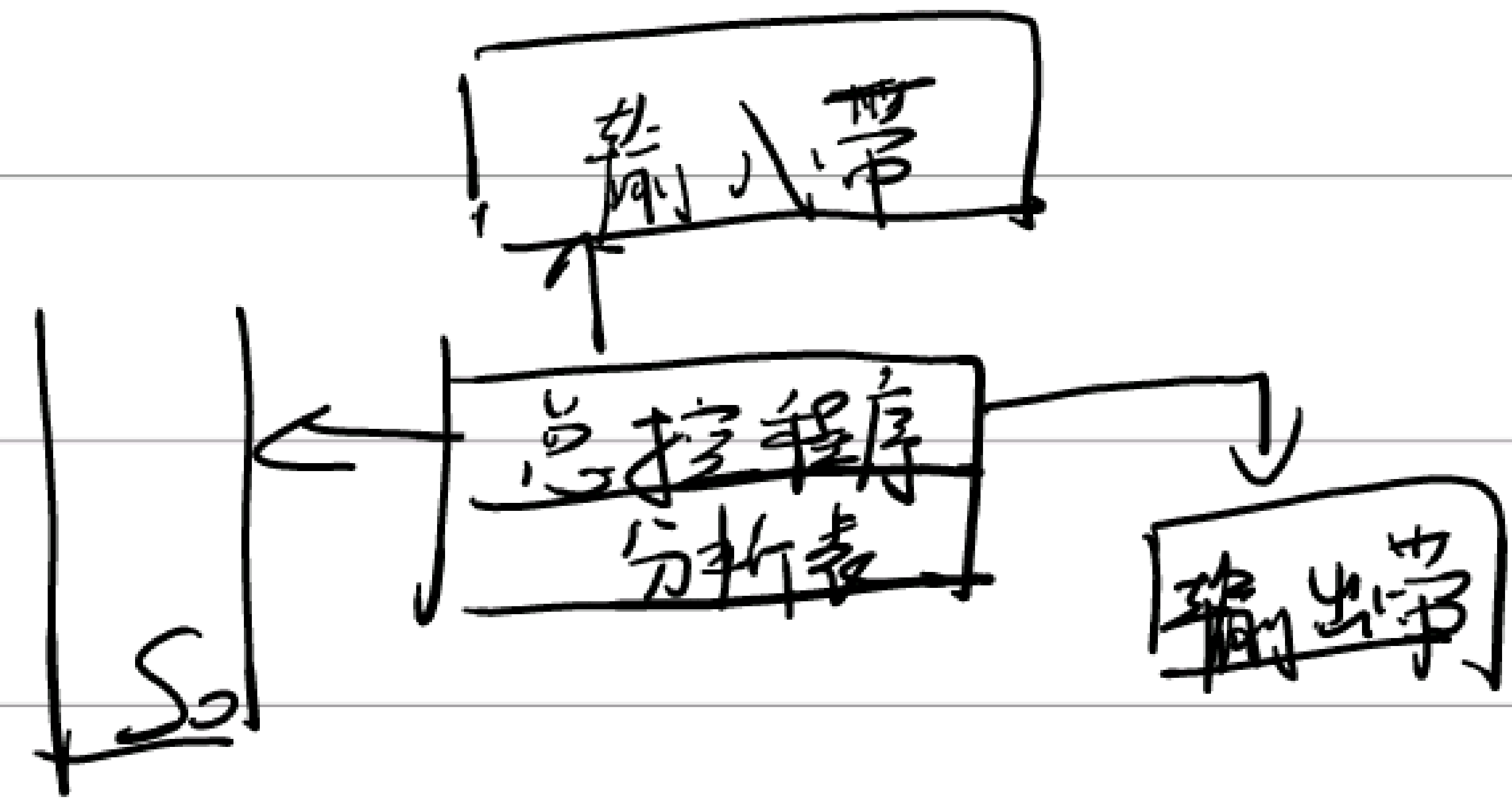
if $P \xrightarrow{+} a \cdots$ 或 $P \xrightarrow{+} Qa \cdots$ 则 $a \in FirstTV(P)$

素短语：至少有一个终结符，且没有更小的素短语。

NOTE：比较符号 $<$　$>$　$=$

比较对象：终结符 + '#'

LR分析器的工作



LR(0)项目 $A \to \cdot XYZ$

$A \to X \cdot YZ$

$A \to XY \cdot Z$

$A \to XYZ \cdot$ 归约项目

$A \to \varepsilon \Rightarrow A \to \cdot$

构造活前缀NFA

(1) 拓广文法 增加 $S' \to S$，$S' \to \cdot S$ 做初态项目

(2) $S' \to S \cdot$ 接受项目

(3) 移进，归约

(4) if $X \to \alpha \cdot A \beta$，画 ε 射向所有 $A \to \cdot \gamma$

LR(0) 项目集规范族构造

(1) 拓广文法

(2) CLOSURE(I)

(3) GO(I, X) => CLOSURE(J)

LR(0) 分析表构造

(1) $A \to \alpha \cdot a\beta \in I_k$. $GO(I_k, a) = I_j$，则 $ACTION[k, a] = S_j$.

对所有终结符a

(2) $A \to \alpha \cdot \in I_k$ 则 $ACTION[k, a) = r_j$，j 为 $A \to \alpha$ 编号

(3) $S' \to S \cdot \in I_k$ 则 $ACTION[k, \#] = acc$

(4) $GO(I_k, A) = I_j$. $GOTO[k, A] = j$.

(5) 其余 error

SLR

对 $I=\{X\to\delta\cdot b\beta,\ A\to\alpha\cdot,\ B\to\alpha\cdot\}$ 的 LR(0)，

if $Follow(A)\cap Follow(B)=\phi \wedge b\notin Follow(A),\ b\notin Follow(B)$

(1) if $a=b$ 移进

2) if $a\in Follow(A)$, $A\to\alpha$ 归约

(3) if $b\in Follow(B)$ $B\to\alpha$ 归约

同 LR(0)

1)

(2) 若 $A\to\alpha\cdot\in I_k$ $a\in Follow(A)$，则 $Action[k,a]=r_j$; j 为 $A\to\alpha$ 编号

(3)

(4)

(5)

LR(1)项目定义

$(A \to \alpha \cdot \beta, a) \quad a \in V_N$

定义

if 存在 $S' \overset{*}{\Rightarrow} \delta A w \to \delta \alpha \beta w$.

$a \in First(w)$

则项目有效.

LR(1) 项目集

CLOSURE(I)-I

(1) I

(2) if $(A \to \alpha \cdot B\beta, a) \in CLOSURE(I)$, $B \to \gamma$

对每个 $b \in First(\beta a)$, 加入 $(B \to \cdot \gamma, b)$

GO

$GO(I, x) = CLOSURE(J)$

$J = \{ (A \to \alpha x \cdot \beta, a) \mid (A \to \alpha \cdot x\beta, a) \in I \}$

构造 LR(1) 分析表

(1) $(A \to \alpha \cdot a\beta) \in I_k$, $GO(I_k, a) = I_j$, 则 $ACTION[k, a] = S_j$

(2) $(A \to \alpha \cdot, a) \in I_k$, 则 $ACTION[k, a] = r_j$ j 为 $A \to \alpha$ 编号

(3) $(S' \to S \cdot, \#) \in I_k$, $ACTION[k, \#] = acc$

(4) $GO(I_k, A) = I_j$, $GOTO[k, A] = j$

(5) error

LALR

在LR(1)基础上合并同心项目

(1) GOTO表合并，GO function与搜索符无关，不冲突

(2) ACTION合并.

① error 和 error 合并为 error

② 移进合并 如(1)

③ 归约合并

a. 同一产生式归约，无冲突

b. 不同产生式归约，因此 LALR 弱于LR(1)

④ 出错 移进合并 X

⑤ 移进 归约合并 X

⑥ 归约和出错合并为归约（人为）

算法 先把转换为同心集，之后一样 LR(1)

赋值翻译

$i := E$ GEN(:=, E.PLACE, —, ENTRY(i))

$E \to -E^{(1)}$ { T := NEWTEMP; GEN(@, $E^{(1)}$.PLACE, —, T), E.PLACE = T }

$E \to E^{(1)} * E^{(2)}$ { T = NEWTEMP;
GEN(*, $E^{(1)}$.PLACE, $E^{(2)}$.PLACE, T) E.PLACE = T }

$E \to E^{(1)} + E^{(2)}$ { T := NEWTEMP
GEN(+, $E^{(1)}$.PLACE, $E^{(2)}$.PLACE, T) E.PLACE = T }

$E \to (E^{(1)})$ E.PLACE = $E^{(1)}$.PLACE

$E \to i$ E.PLACE = ENTRY(i)

类型转换

GEN(op^i, I, J, T_1)

GEN(itr, T_1, —, T_2) 整转实

if 翻译

if $A \vee B < D$ then S_1 else S_2

(1) (jnz, A, -, (5))

(2) (j, -, -, (3))

(3) (j<, B, D, (5))

(4) (j, -, -, p+1)

(5) S_1

(p) (j, -, -, q)

(p+1) S_2

(q)

布尔表翻译

input	SYM	TC	FC	QUADRUPLE
A∧B∨¬C#	#	—	—	
∧B∨¬C#	# i	— —	— —	
∧B∨¬C#	# E	—(1)	—(2)	(jnz, A, —, ~~0~~ 3)
B∨¬C#	# E∧	—(1) —	—(2) —	(j, —, —, ~~0~~ 5)
B∨¬C#	# $E^{\wedge}$	— —	—(2)	
∨¬C#	# $E^{\wedge}$ i	— — —	—(2) —	
∨¬C#	# $E^{\wedge}$E	— —(3)	—(2)(4)	(3) (jnz, B, —, 0)
∨¬C#	# E	—(3)	—(4)	(4) (j, —, —, ~~0~~ 5)
¬C#	# E∨	—(3) —	—(4) —	
¬C#	# $E^{\vee}$	—(3)	— —	
C#	# $E^{\vee}$¬	—(3) —	— — —	
#	# $E^{\vee}$¬ i	—(3) — —	— — — —	
#	# $E^{\vee}$¬E	—(3) —(5)	— — —(6)	(5) (jnz, C, —, 0)
#	# $E^{\vee}$E	—(3)(6)	— —(5)	(6) (j, —, —, 0)
#	# E	—(6)	—(5)	

代码优化

(局部优化)

(1) $P_i := 3.14$

(2) $A := 2*P_i*(R+r)$

(3) $B := A$

(4) $B := 2*P_i*(R+r)*(R-r)$;

(1) $P_i := 3.14$ 合记变量

(2) $T_1 := 2*P_i$

删除公共子表达式

(3) $T_2 = R+r$

(4) $A = T_1 * T_2$

(5) $B := A$

(6) $T_3 = 2*P$

(7) $T_4 = R+r$

(8) $T_5 = T_3 * T_4$

(9) $T_6 = R-r$

(10) $B := T_5 * T_6$

变量传播与无用赋值删除.

循环优化

$j := 1$

for $i:=1$ to 100 do

  $A[i,j] := B[i,j]+2$

1) 循环代码外提

2) 降低运算强度.

3) 变换循环控制变量并删除其自增赋值式.

(1) $j := 1$
$i := 1$

B_1

if $i>100$ goto (15)

B_2

$T_1 := i*10$.

$T_2 := T_1 + j$

$T_3 := a - 11$

$T_4 := i*10$

$T_5 := T_4 + j$

$T_6 := b - 11$

$T_7 := T_6[T_5]$

$T_8 := T_7 + 2$

$T_3[T_2] := T_8$

$i := i+1$

goto (3)

(15)

DAG

(1) $T_0 := 3.14$

(2) $T_1 := 2 * T_0$

(3) $T_2 := R + r$

(4) $A := T_1 * T_2$

(5) $B := A$

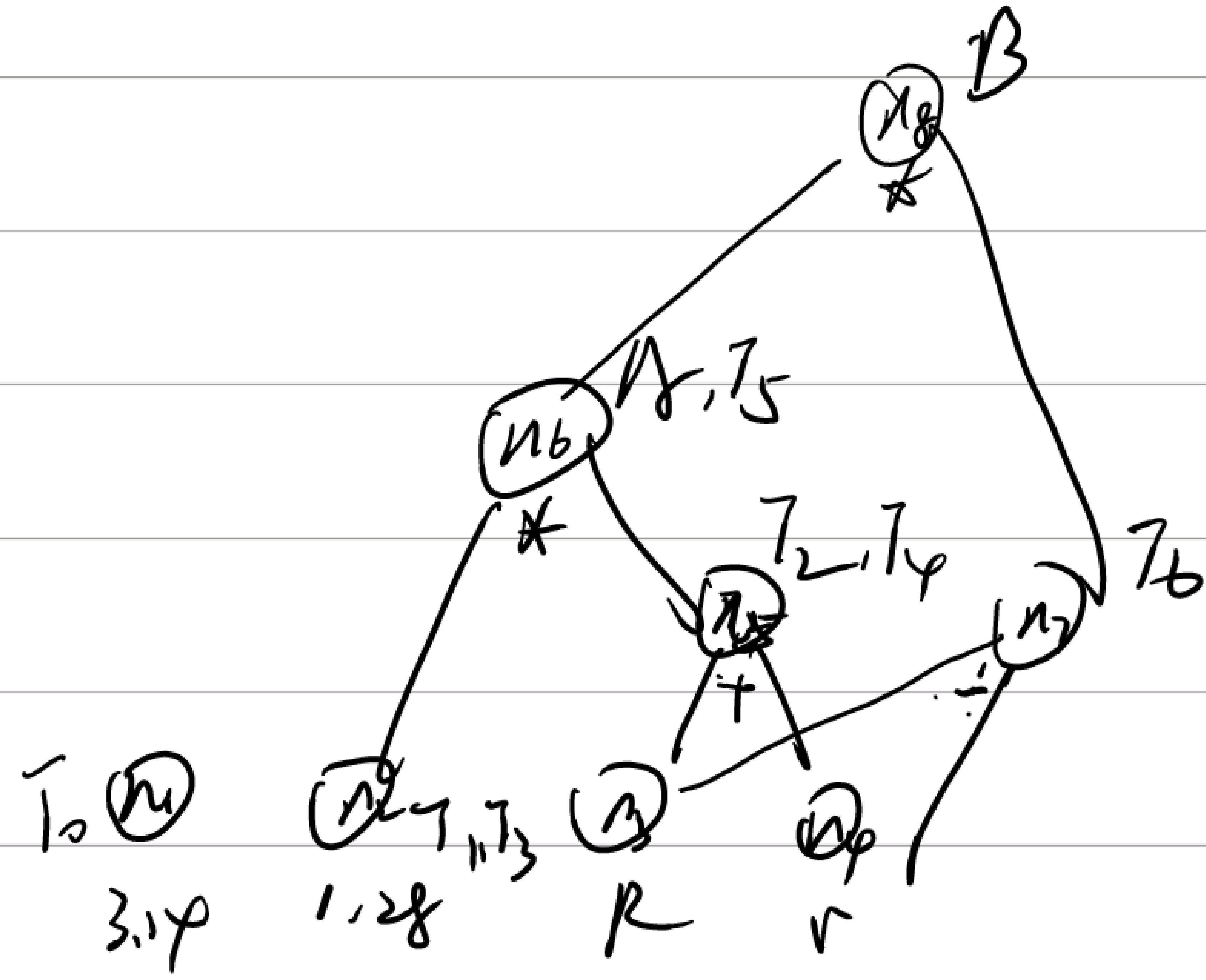
(6) $T_3 := 2 * T_0$

(7) $T_4 := R + r$

(8) $T_5 := T_3 * T_4$

(9) $T_6 := R - r$

(10) $B := T_5 * T_6$



优化作用 (1) 常数直接

(2) 重新赋值，删除之前

(3) 公共子表达式

还原四元式

(1) 若叶，无附标　　X

(2) 若叶，(n) A / B　　$A := B$

(3) 若中，$A := B \text{ op } C$　$A := \text{op } B$　$A := B[C]$ / if B rop C goto(S)

(4) 若中 无附，建 S_A，转(3)

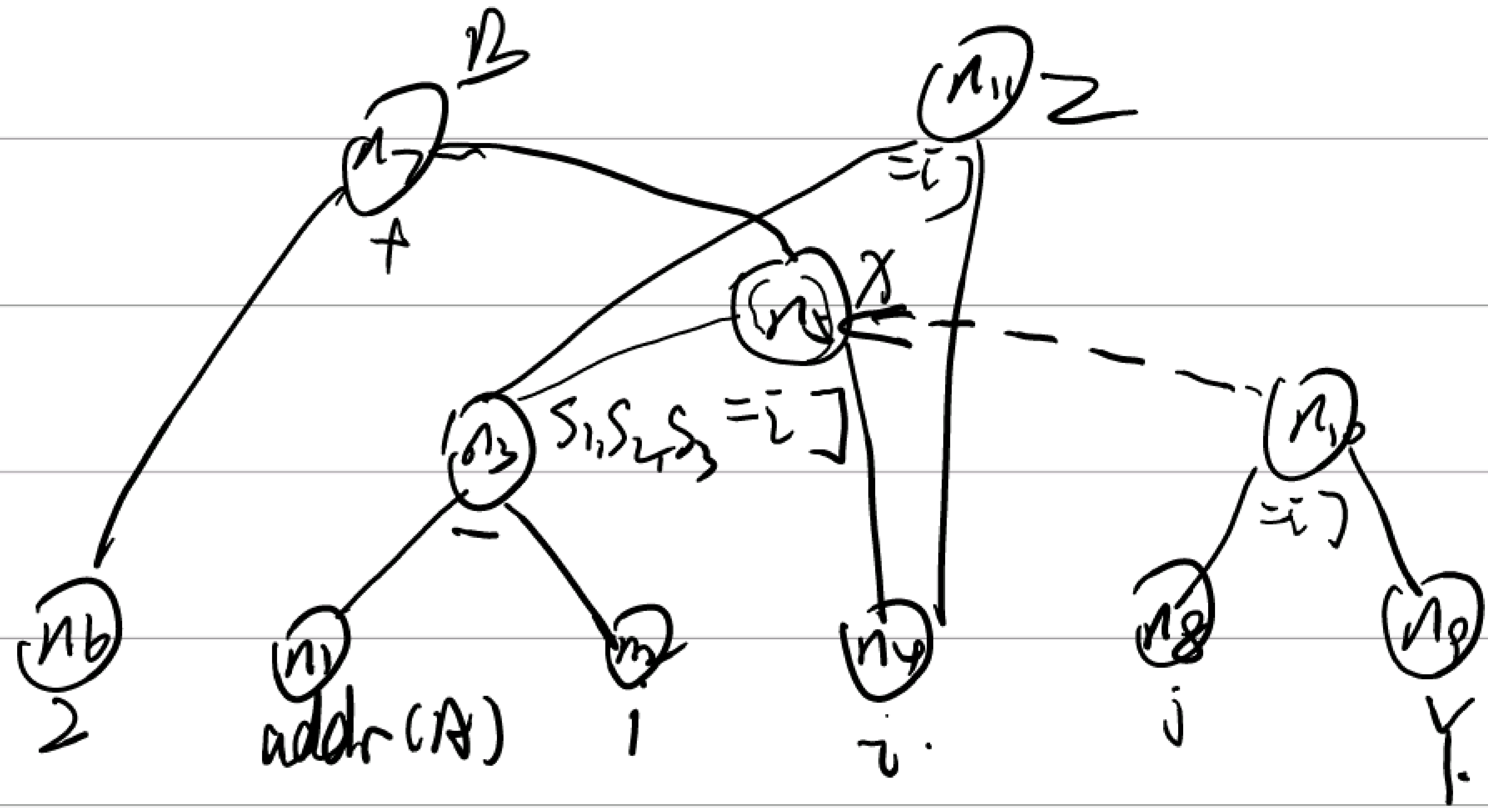
(5) 若多标，若叶 都 $A_i := B$，若 内除第一个标，其余 $A_i := B$

G -> DAG -> G. 做了一次局部优化

再 delete 不被引用的变量了

结点或标识符注编

$X := A[i]$	$S_1 := addr(A) - 1$
$B := X + 2$	$X := S_1[i]$
$A[j] := Y$	$B := X + 2$
$Z := A[i]$	$S_2 := addr(A) - 1$
	$S_2[j] := Y$
	$S_3 := addr(A) - 1$
	$Z := S_3[i]$



必经结点集

$D_n = \bigcap_{P_i \in P(n)} D(P_i) \cup \{n\}$ ，P_i 为 n 父结点、

n_i dom n_j ，n_i 为 n_j 必经结点集

E $a \to b$ ，b dom a ，则称 $a \to b$ 回边

查找循环

(1) 找出 $n \to d$ ，d dom n

(2) 则 循环 $L := \{n, d\}$

(3) 若 $n \neq d$ 且 n 的 father $n' \notin L$ ，则 $L := L \cup \{n'\}$

(4) 对 (3) 中 n 所有父做 (3)

到达定值数据流方程

ud 链：d点对A的定值能达到u点，d为u最近定值点、

在块中点值,kill其他 DAG中的附标

$$\begin{cases} OUT[B] = IN[B] - KILL[B] \cup GEN[B] \\ IN[B] = \bigcup_{P \in P[B]} OUT[B] \end{cases}$$

活跃变量

∃P开始的一条路径，A在定值前被引用或仅有引用

$$\begin{cases} L \cdot IN[B] = L \cdot OUT[B] - L \cdot DEF[B] \cup L \cdot USE[B] \\ L \cdot OUT[B] = \bigcup_{S \in S[B]} L \cdot IN[S] \end{cases}$$

启发式排序就是拓扑排序

JDBC

JSP 隐藏对象 EL表达式 WEB-INF/

Servlet xml配置文件web.xml

servlet生命周期

JSTL

重定向和转发区别

struts

request 作用域

session