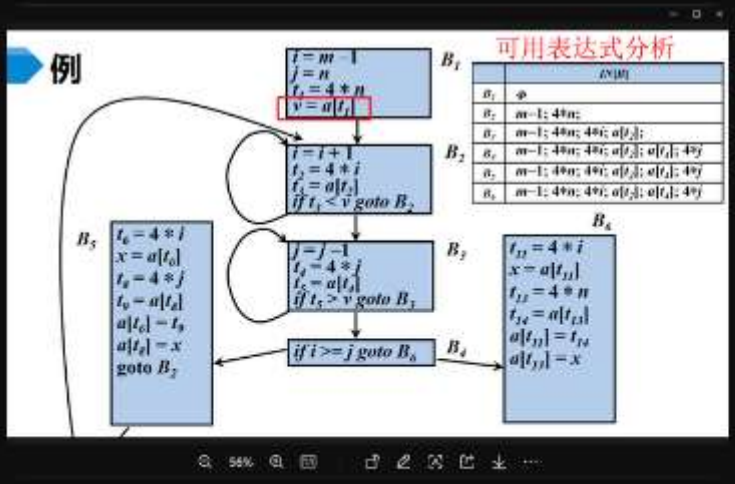
群聊的聊天记录

2024/05/31 20:17

TSUNDOKU

为啥可用表达式没a[t1]

TSUNDOKU



小五（21-计科-王继媛）

因为b5会重新赋值

小五（21-计科-王继媛）

就像DAG画图里面，有赋值，所有和a相关的都kill了

小五（21-计科-王继媛）

但是at2是在b5后面的

小五（21-计科-王继媛）

所以可以用

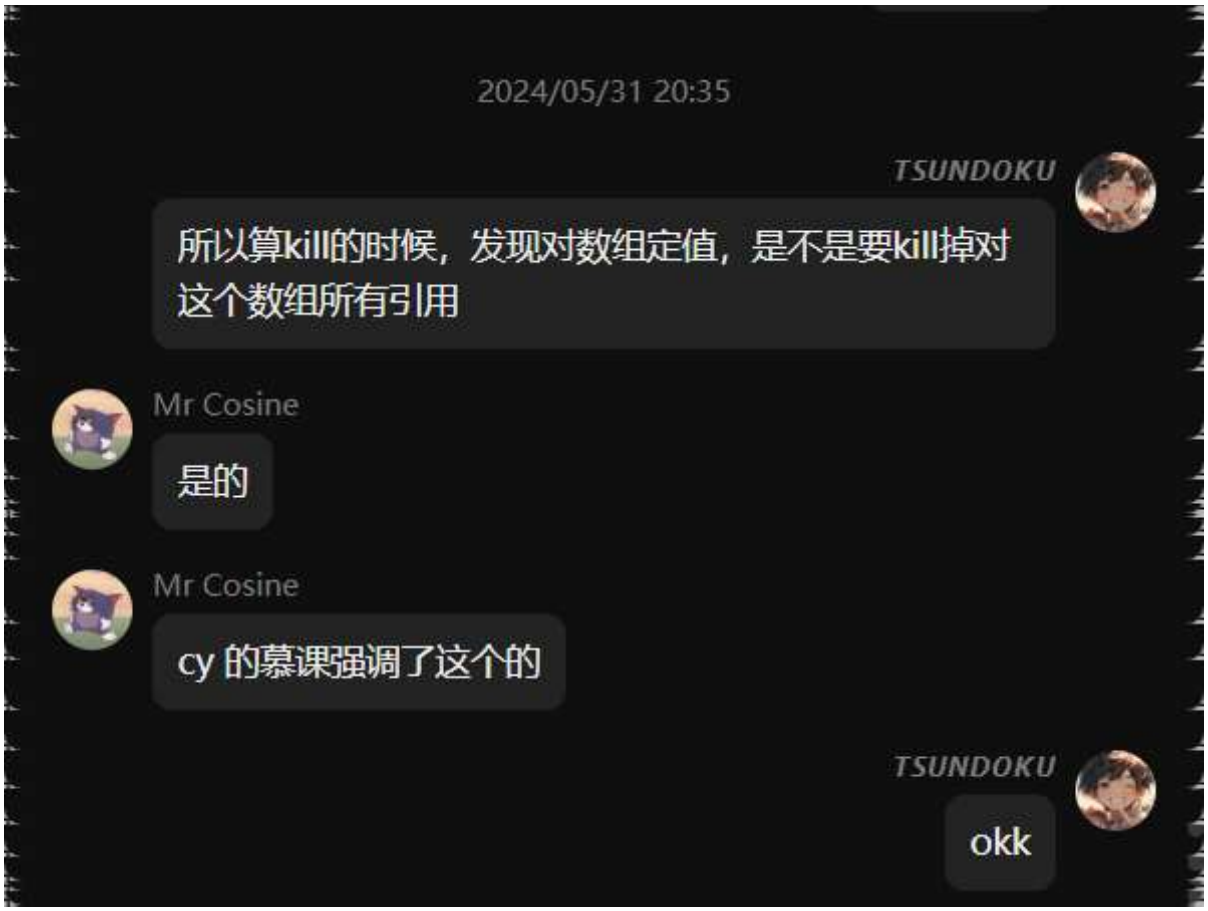
2024/05/31 20:25

TSUNDOKU

哦，对，看a

TSUNDOKU

不该看t1


2024/05/31 20:35
TSUNDOKU
所以算kill的时候，发现对数组定值，是不是要kill掉对这个数组所有引用
Mr Cosine
是的
Mr Cosine
cy 的慕课强调了这个的
TSUNDOKU
okk

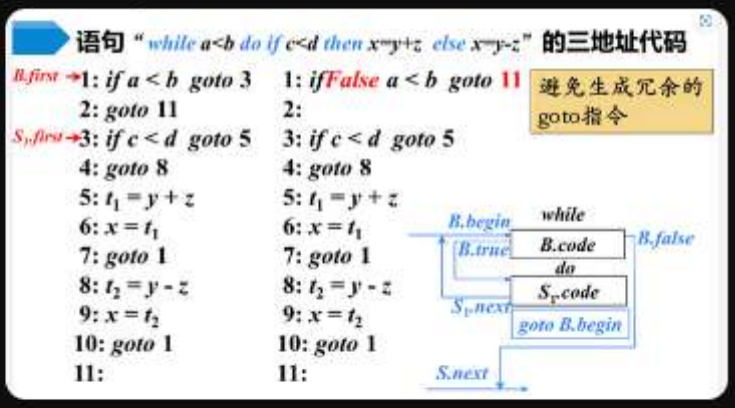
water boy和*TSUNDOKU*的聊天记录

2024/05/31 14:11

water boy（21-计科-原泽坤）

那如果要求画图的话S1.next要指向哪里啊

water boy（21-计科-原泽坤）



water boy（21-计科-原泽坤）

如果不优化的话

TSUNDOKU

这不写着吗

TSUNDOKU

B.begin

TSUNDOKU

只是改了SDT

water boy（21-计科-原泽坤）

那这感觉是优化了呀

water boy（21-计科-原泽坤）

不优化的话S1.next不应该指向goto吗

TSUNDOKU

那个goto语句是SDT生成的

TSUNDOKU

S1.next指向B.begin

water boy（21-计科-原泽坤）

嘶

TSUNDOKU

不是顺序执行下面的代码

TSUNDOKU

往上走

TSUNDOKU

故生成goto语句

TSUNDOKU

跳到上面代码

water boy（21-计科-原泽坤）

嘶我把这个例子想错了

water boy（21-计科-原泽坤）

我的本意是

water boy（21-计科-原泽坤）

如果S1中有个if else的话

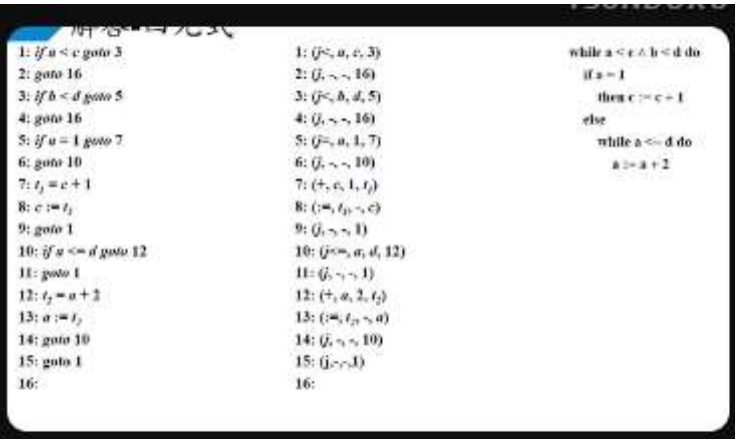
water boy（21-计科-原泽坤）

这个if else 内部的goto不是指向goto B.begin,而是指向B.begin

water boy（21-计科-原泽坤）

这个不属于goto优化对吗

TSUNDOKU



TSUNDOKU

这个?

water boy（21-计科-原泽坤）

啊差不多

water boy（21-计科-原泽坤）

9 行就是goto1

water boy（21-计科-原泽坤）

而不是goto15

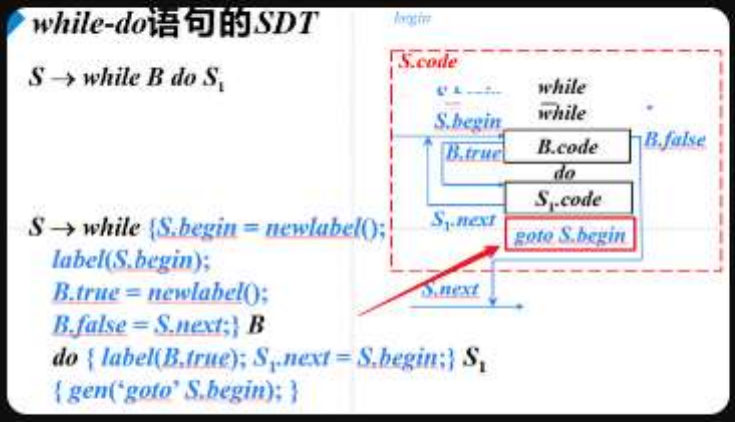
2024/05/31 14:21

TSUNDOKU

我想想

2024/05/31 14:31

TSUNDOKU



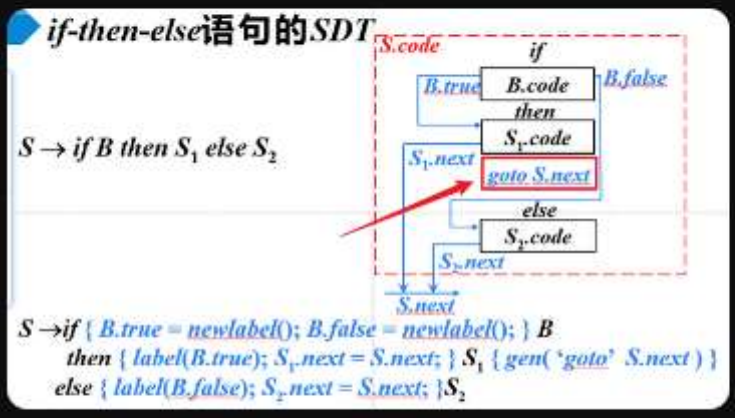
TSUNDOKU

这个对应15行代码

water boy（21-计科-原泽坤）

对

TSUNDOKU



TSUNDOKU

这个对应第9行

water boy（21-计科-原泽坤）

是的

water boy（21-计科-原泽坤）

这里S.next被设置成了while中的S.begin

TSUNDOKU

TSUNDOKU

这个图里的S.next相当于第一个图里的S1.next

water boy（21-计科-原泽坤）

所以里面第九行要直接指向1

TSUNDOKU

对

TSUNDOKU

他两相等

TSUNDOKU

都是一个值

water boy（21-计科-原泽坤）

所以写goto15是错的罢

water boy（21-计科-原泽坤）

我今天才发现问题

water boy（21-计科-原泽坤）

这个避免冗余goto是一定要做的对吧

2024/05/31 14:36

TSUNDOKU

正常来说应该按照SDT一步步搞

TSUNDOKU

有问题的其实都是人肉编译

TSUNDOKU

人肉编译尽量画个图

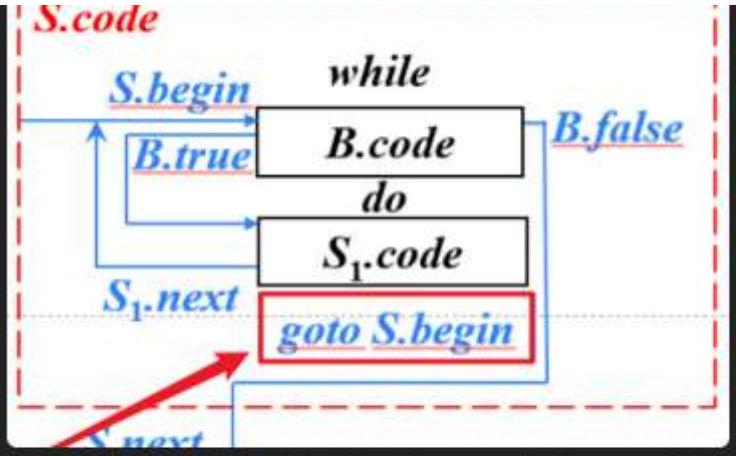
TSUNDOKU

知道goto到哪

water boy（21-计科-原泽坤）

如果单纯if while的话感觉可以直接人肉

TSUNDOKU



冗余的goto其实就相当于S1.next指向goto语句

water boy（21-计科-原泽坤）

是的

water boy（21-计科-原泽坤）

现在想想，这个似乎也不叫冗余goto吧

TSUNDOKU

对

water boy（21-计科-原泽坤）

比较goto S.begin还是会用到

TSUNDOKU

就是跳转的问题

TSUNDOKU

跳转的指令序号搞错了

TSUNDOKU和Shelly Shan的聊天记录

2024/03/29 22:25

TSUNDOKU

老师，实验的选做可以不做吗

2024/03/29 22:56

Shelly Shan

可以不做。

TSUNDOKU

不做不会影响分数吧

Shelly Shan

不影响

2024/05/20 21:27

TSUNDOKU

➢试将下面的语句翻译成四元式序列

（2）for i:=m step 2 until n do

if a<b then x:=x+1

TSUNDOKU

➢在SPOC讲义6.3节（控制语句的翻译）所示的SDT中添加处理下列控制流构造的翻译方案

➢(1) repeat语句： $S \rightarrow repeat\ S_1\ while\ B$

➢(2) for循环语句： $S \rightarrow for\ (S_1\ ;\ B\ ;\ S_2)\ S_3$

TSUNDOKU

老师，这种题用掌握吗

TSUNDOKU

这种语句上课好像没讲过

Shelly Shan

讲的是方法，要求会运用解决问题

2024/05/20 21:35

TSUNDOKU

那这种也得掌握是吗

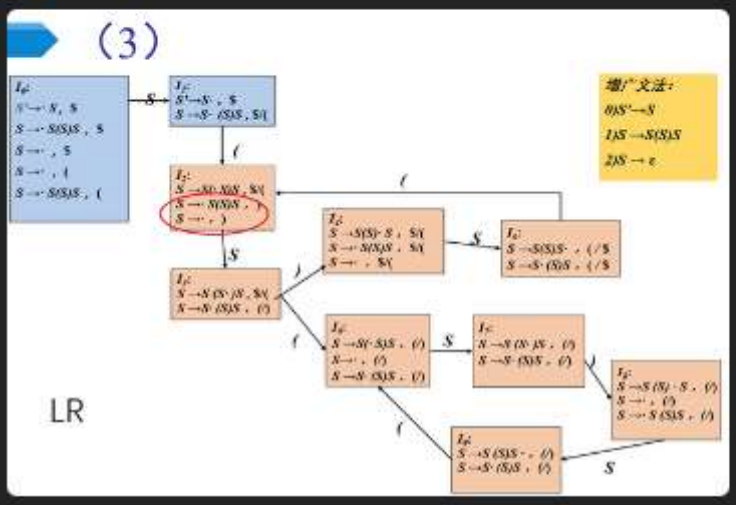
2024/05/21 09:02

Shelly Shan

是的

2024/05/26 13:26

TSUNDOKU



老师，这展望符是不是少了左括号

2024/05/26 13:32

Shelly Shan

是的

TSUNDOKU

这是一个习题集，学长写的答案，那个I7到I8，是不是也是少分析了左括号的情况

2024/05/26 13:39

Shelly Shan

I7是，现在的I8没有吧?

TSUNDOKU

是的

2024/05/27 13:58

TSUNDOKU

$S \rightarrow a \mid S + S \mid S\,S \mid S^* \mid (S)$

$S \rightarrow aS'$

$S \rightarrow (S)S'$

$S' \rightarrow +SS'$

$S' \rightarrow SS'$

$S' \rightarrow *S'$

$S' \rightarrow \varepsilon$

TSUNDOKU

老师，这样改写对吗

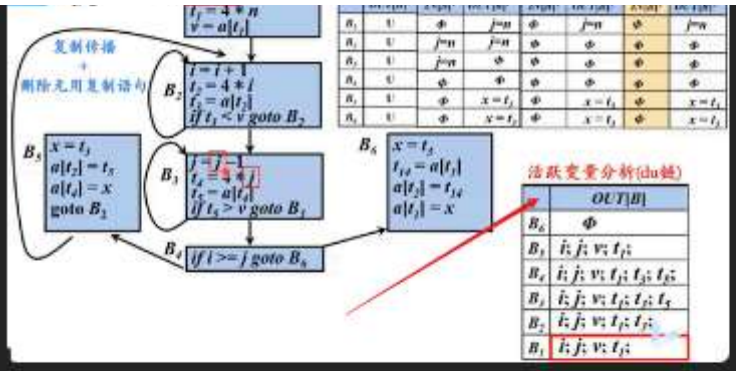
2024/05/27 14:37

Shelly Shan

对的

2024/05/27 16:33

TSUNDOKU



老师，这块指的是出口的活跃变量吗

TSUNDOKU

如果是，为啥会有v啊

TSUNDOKU

v没被引用过啊

TSUNDOKU

还是说if判断用了v也算

Shelly Shan

引用就算

TSUNDOKU

所以说if判断用了v也算是吗

Shelly Shan

对的

2024/05/31 11:45

TSUNDOKU

习题7.4

➢说明下面的文法

$$S \rightarrow S\,A \mid A$$

$$A \rightarrow a$$

是SLR(1)的，但不是LL(1)的

TSUNDOKU

老师，判断是不是LL（1）文法

TSUNDOKU

是不是先消除左递归再判断

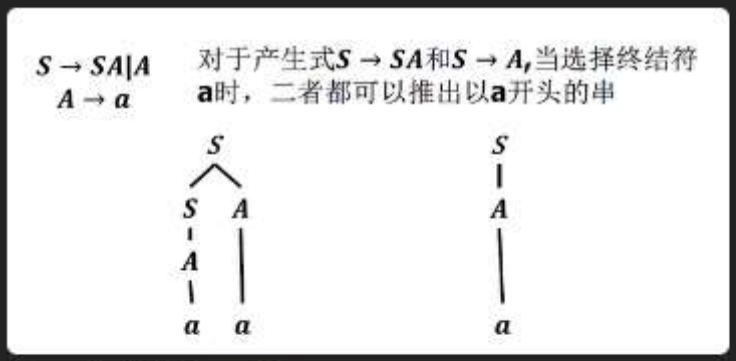
Shelly Shan

根据题意吧，此题应该不改造。

TSUNDOKU

直接找S的1，2产生式的select集判断，是吗

TSUNDOKU



这是之前学长写的

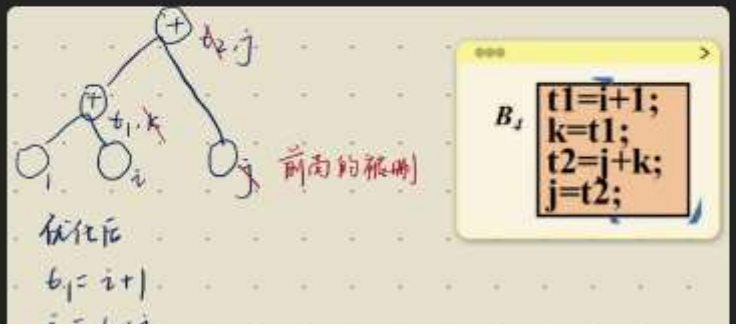
2024/05/31 12:30

Shelly Shan

是的

2024/05/31 18:35

TSUNDOKU



老师，这里i,j是活跃变量

TSUNDOKU

那个t1和k，我保留t1还是k有区别吗

TSUNDOKU

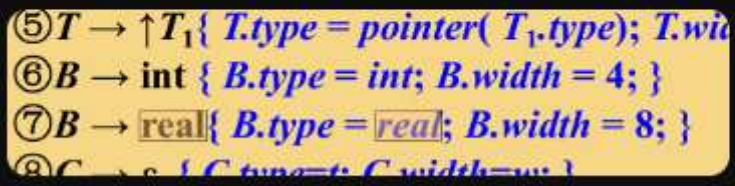
k=i+1
j=j+k

TSUNDOKU

像这个和图片里的

2024/05/31 21:25

TSUNDOKU



TSUNDOKU

老师，real算8个字节吗

2024/05/31 21:42

Shelly Shan

没有区别

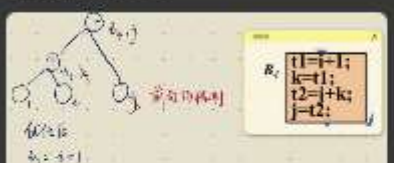
Shelly Shan

Shelly Shan

是的

TSUNDOKU

TSUNDOKU



那这个题呢

TSUNDOKU

t1和k，我保留t1还是k有区别吗

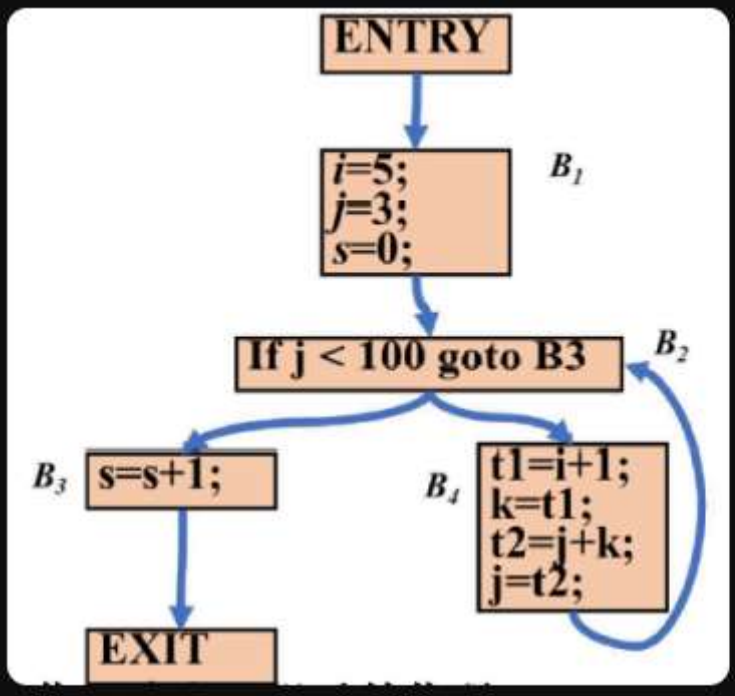
TSUNDOKU

> Shelly Shan
> 没有区别

哦，看到了

2024/06/01 08:18

TSUNDOKU



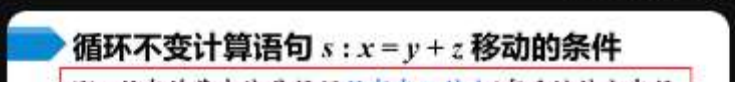
TSUNDOKU

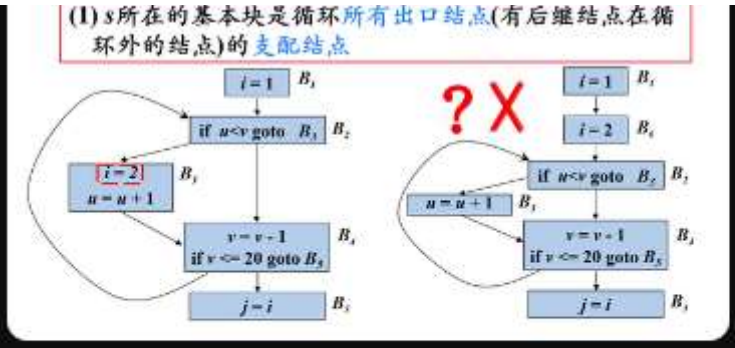
4.找出所有的循环不变计算；这些循环不变计算能不能提出循

TSUNDOKU

老师，2，4构成的循环中，循环不变计算能外提吗

TSUNDOKU





TSUNDOKU

感觉这个条件不满足

2024/06/01 10:17

Shelly Shan

不能外提

TSUNDOKU

TSUNDOKU

那老师

TSUNDOKU

这个为啥选D啊

TSUNDOKU

感觉声明没有生成中间代码

Shelly Shan

错在构造二字

TSUNDOKU

那该是啥，存入符号表?

2024/06/01 10:26

Shelly Shan

不总是构造

Shelly Shan

大多数时候只是添加条目

2024/06/01 12:42

TSUNDOKU

老师，a=1会不会跟e_kill有关，导致e_kill添加所有和a相关的表达式

2024/06/01 15:38

Shelly Shan

当然

TSUNDOKU

S→ { a1: L.side = left; }
LS' { a2: S.val= L.val+ S'.val; }
S'→ . { a3: L.side = right;}
L { a4: S'.val= L.val; }
S'→ ε { a5: S'.val= 0; }
L→ B { a6: L'.side=L.side; }
L' { a7: if L.side=left then L.val = B.val*2 $^{L'.length}$+ L'.val ;
if L.side=right then L.val = (B.val + L'.val)/2;
L.length = L'.length +1; }
L'→B { a8: L_1'.side=L'.side;}
L_1' { a9: if L'.side=left then L'.val= B.val*2 $^{L'.length}$+ L_1'.val ;
if L'.side=right then L'.val= (B.val + L_1'.val)/2;
L'.length= L_1'.length +1; }
L'→ ε { a10: L'.val= 0, L'.length=0; }
B→ 0 { a11: B.val=0; }
B→ 1 { a12: B.val=1; }

(2) 为了让(1)中 SDT 能在自顶向下分析中实现，请写出产生式 L→BL' 入栈时，与右部各记录相关联的语义动作的栈操作。(参考第 5 章 PPT73-76 页)。

答：

S→ { a1: L.side = left; }
LS' { a2: S.val= L.val+ S'.val; }

符号	属性	执行代码	动作
a1		stack[top-1].side = left; top=top-1;	

L	side	若选：L->BL'移入
		stack[top+3].side = stack[top].side;
		stack[top].side = stack[top].side;
		top=top+5;
			复制到{a6}和{a7}

L.syn	val, length	stack[top-3].val = stack[top].val; top=top-1;	复制到{a2}
S'			
S'.syn	val	stack[top-1].S'.val = stack[top].val; top=top-1;	复制到{a2}
a2	L.val, S'.val	stack[top-1].val; = stack[top].L.val+stack[top].s'.val; top=top-1;	

TSUNDOKU

s'那栏用不用，考虑top的变化

2024/06/01 15:44

Shelly Shan

不用

TSUNDOKU

TSUNDOKU

那老师

TSUNDOKU

这个C问题在哪呢

2024/06/01 15:50

Shelly Shan

B和C只能有一个对吧?

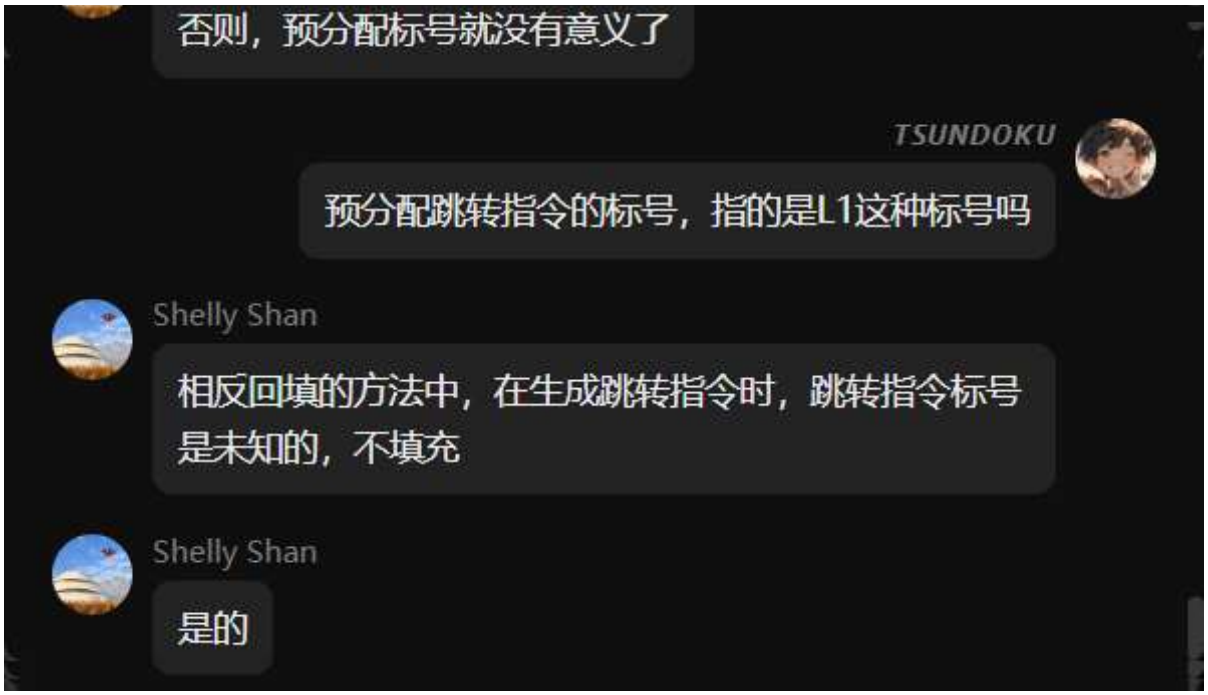
TSUNDOKU

嗯

Shelly Shan

预分配跳转指令的标号，在生成跳转指令时，当然就有标号了

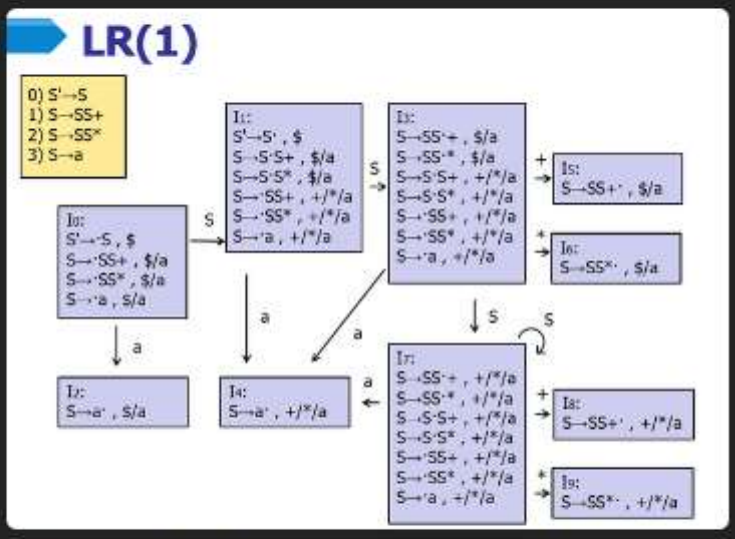
Shelly Shan


否则，预分配标号就没有意义了
TSUNDOKU
预分配跳转指令的标号，指的是L1这种标号吗
Shelly Shan
相反回填的方法中，在生成跳转指令时，跳转指令标号是未知的，不填充
Shelly Shan
是的

群聊的聊天记录

2024/05/23 13:24

白日梦想家



我想问下这道题，I0的展望符，那个a是咋来的呀

2024/05/23 13:34

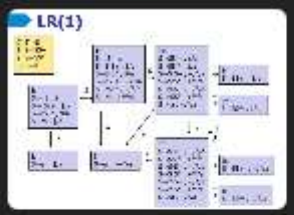
TSUNDOKU

S的first集吧

TSUNDOKU

白日梦想家

白日梦想家



图用的bei ta应该是空吧

图里的beita应该是空吧

白日梦想家

不应该是直接继承吗

求S->.SS+ 的等价项目的时候

beita是S+

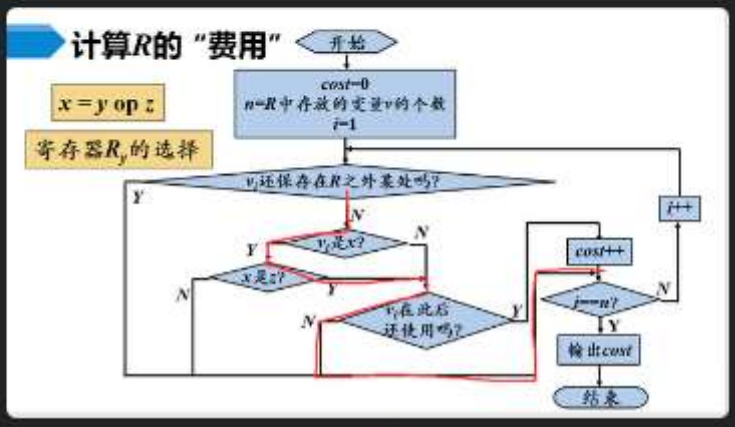
白日梦想家

> 白日梦想家
> 不应该是直接继承吗

哦我懂了，完事还得再求一次等价项

2024/05/24 22:40

一枝鹿



这条路想不明白，为啥这条路的条件下vi不需要被保存

一枝鹿

R唯一保存了vi，vi的值是x以及z，那计算op的时候要用到z啊，不保存vi，R的值把vi改了，用到z的时候找不到了啊

2024/05/25 06:40

君博Bo Y（21-物联网-君博）

> 一枝鹿
> R唯一保存了vi，vi的值是x以及z，那计算op的时候要用到...

确实，你说的对，这个图是有些小问题，我理解的是当vi的值是x以及x就是z的时候，直接cost++了，因为vi的值紧接着就会被用到

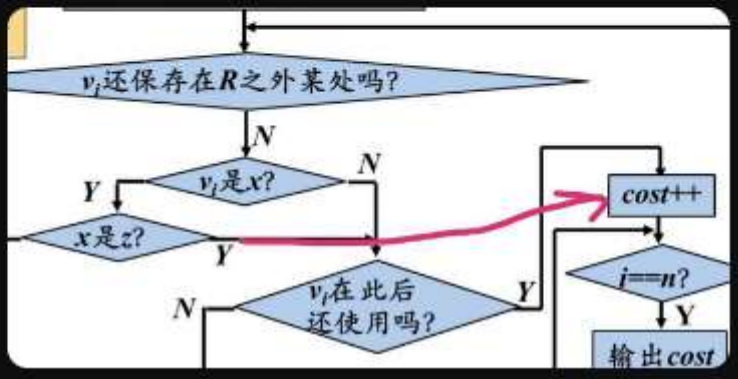
尹博Bo Y.（21-物联网-尹博）

所以这条路径下这个判断是冗余的

尹博Bo Y.（21-物联网-尹博）

流程图应该修改为：

尹博Bo Y.（21-物联网-尹博）



尹博Bo Y.（21-物联网-尹博）

（我课后就这个小点向陈老师问了一下，她也表示认同  ）

2024/05/25 18:09

甜筒喵  （人智2103602-周欣鹏）

想问一下，t1=y+1
param t1
t2=call f,1
t3=t2+2
x=t3
三地址码关于 x = f (y+l) + 2的翻译中

甜筒喵  （人智2103602-周欣鹏）

t2=call f,1中的1是啥意思

ERROR SEMI

参数个数

2024/05/28 17:24

甜筒喵🍦 (人智2103602-周欣鹏)

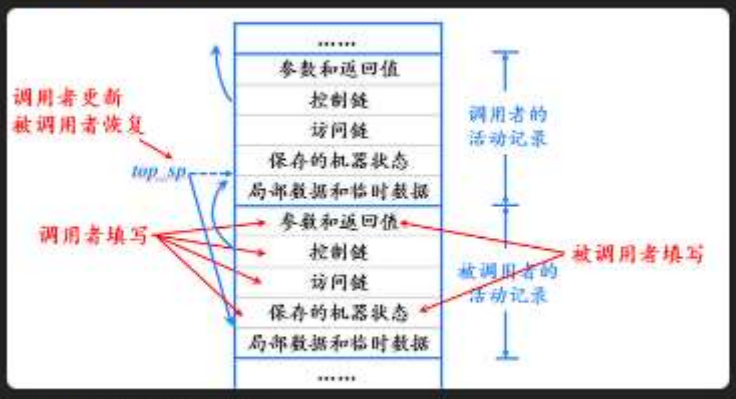
问一下，访问链是由被调用者填写还是由调用者填写

2024/05/28 18:40

TSUNDOKU 

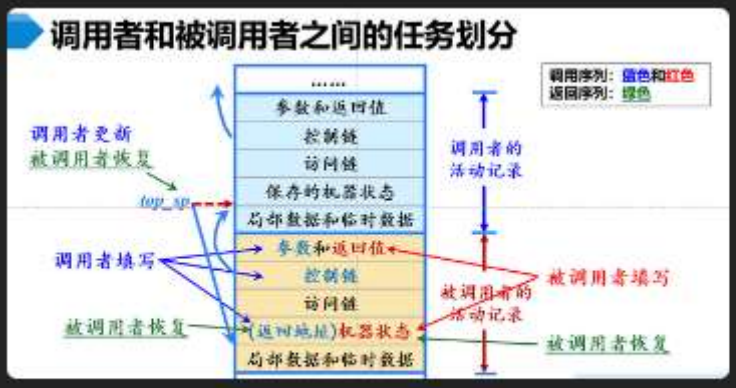
> 人智2103602-周欣鹏
> 问一下，访问链是由被调用者填写还是由调用者填写

@甜筒喵🍦



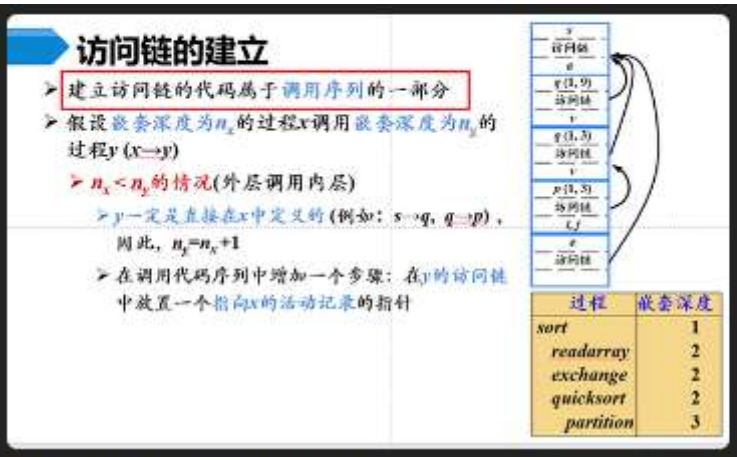
cyMOOC的课件

TSUNDOKU 



但sll的没写

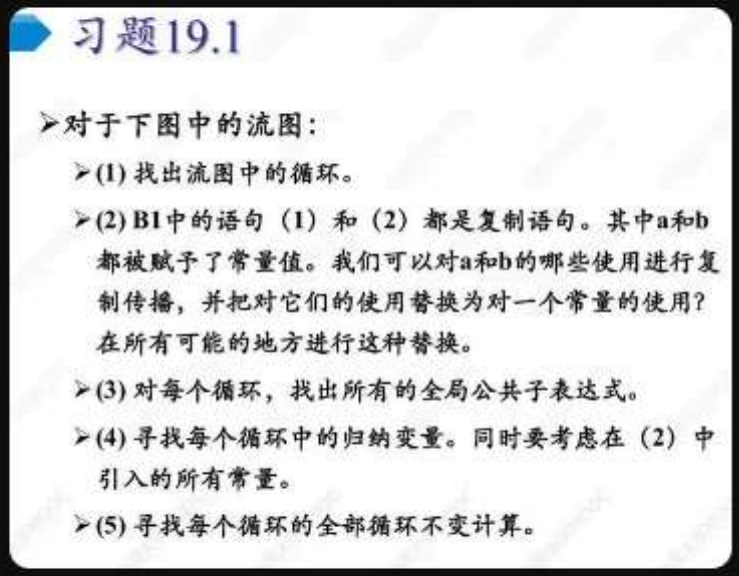
TSUNDOKU 



不过sll课件后面提到了这句

2024/05/28 21:53

新李



新李

对于循环的公共子表达式应该怎么定义？感觉按照公共子表达式的定义似乎是具体到出现的位置？

2024/05/28 22:04

一枝鹿

局部的公共子表达式在基本块内就可以解决，全局的公共子表达式可以用可用表达式分析，以基本快为单位，这时候其实跟循环没关系了，但循环相关的一些"约束"应该是在算可用表达式的时候就体现出来了

2024/05/28 22:10

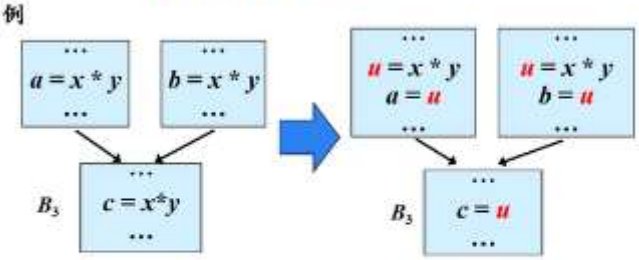
一枝鹿

➢如果表达式x op y先前已被计算过，并且从先前的计算到现在，x op y中变量的值没有改变，那么x op y的这次出现就称为公共子表达式(*common subexpression*)

① 删除全局公共子表达式

➢可用表达式的数据流问题可以帮助确定位于流图中p点的表达式是否为全局公共子表达式

➢例



全局公共子表达式删除算法

➢输入：带有可用表达式信息的流图

➢输出：修正后的流图

➢方法：

➢对于语句s：$z = x$ op y，如果x op y在s之前可用，那么执行如下步骤：

① 从s开始逆向搜索，但不穿过任何计算了x op y的块，找到所有离s最近的计算了x op y的语句

② 建立新的临时变量u

③ 把步骤①中找到的语句$w = x$ op y用下列语句代替：

$u = x$ op y

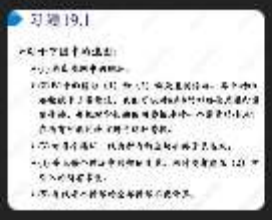
$w = u$

④ 用$z = u$替代s

涉及到定义的也就这三页ppt了，第一张是广义的，后两个是全局公共子表达式相关的

一枝鹿

新李


习题19.1

@新李 结合这个题来看，我感觉就是把这个循环里的公共子表达式找出来就行，不用在意有没有相关部分在循环外

新李

一枝鹿

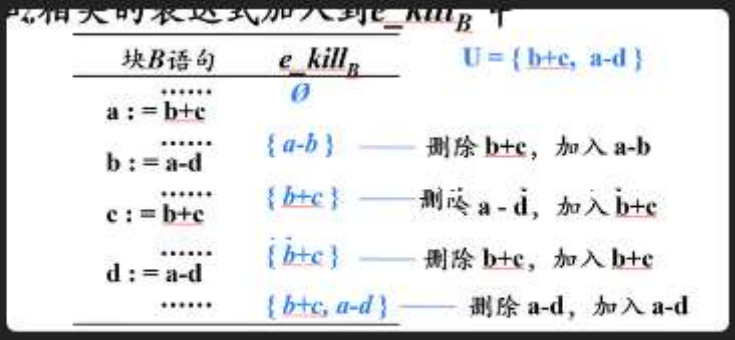
@新李 结合这个题来看，我感觉就是把这个循环里的公共...

@一枝鹿 嗯，我的意思是讲公共子表达式的时候要不要明确他在哪个位置出现

2024/05/28 22:17

一枝鹿

在算可用表达式分析的时候，ppt里只用了表达式，没有标出来位置，像这样



。但我觉得删除公共子表达式的时候，或许需要列出来删除的表达式出自哪一句

TSUNDOKU

sll练习题答案不是明确位置了吗

新李

比如a+b的话并不是在循环中的所有位置都是可用的，那它可以称为循环中的公共子表达式吗

一枝鹿

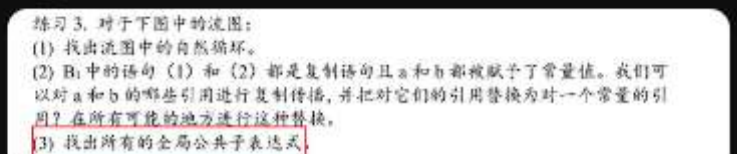
> TSUNDOKU
> sll练习题答案不是明确位置了吗

对，我正要找这个@TSUNDOKU

一枝鹿

很不幸，我找了sll的那道差不多的题，是这么问的

一枝鹿


练习 3. 对于下图中的流图：
(1) 找出流图中的自然循环。
(2) B_1 中的语句 (1) 和 (2) 都是复制语句且 a 和 b 都被赋予了常量值。我们可以对 a 和 b 的哪些引用进行复制传播，并把对它们的引用替换为对一个常量的引用？在所有可能的地方进行这种替换。
(3) 找出所有的全局公共子表达式

2024/05/29 10:25

甜筒喵🍦 (人智2103602-周欣鹏)

- 三地址语句
  - *call callee*
- 目标代码: 调用序列
  - *ADD SP, SP, #caller.recordsize*
  - *ST 0(SP), #here + 16*
  - *BR callee.codeArea*

甜筒喵🍦 (人智2103602-周欣鹏)

想问一下这个的16是怎么计算出来的呀

TSUNDOKU

ST BR 这两个指令#here+16 callee.codeArea这两个常量，每个算一个字，占4字节

TSUNDOKU

4*4=16

TSUNDOKU

0 (SP)是寄存器

TSUNDOKU

不占字节

2024/05/29 11:16

白日梦想家

```
PROGRAM p;
    VAR a,b,c,d,e: real;
    PROCEDURE a;
        VAR c,e,f,g: real;
        BEGIN
        ...
```

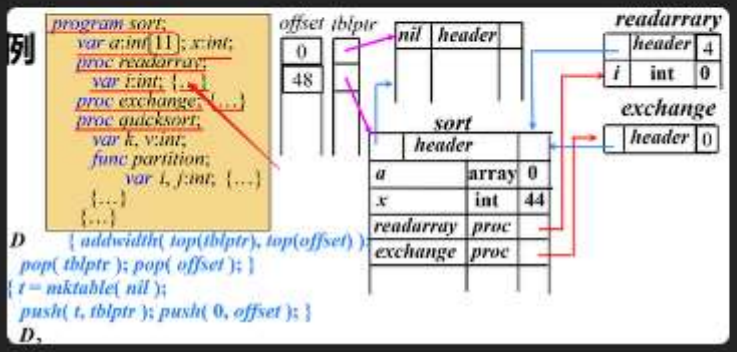
```
    c;
    ...
    END;
PROCEDURE b;
```

请问这个"c"要怎么往符号表里写呀，还是不需要管呢

TSUNDOKU

不用吧

TSUNDOKU



c相当于这里面的东西

RxR

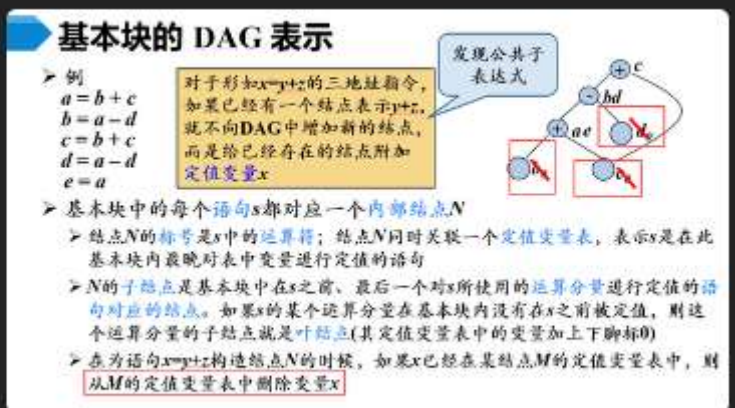
这里的C是引用不是声明吧

RxR

之前声明过了

2024/05/29 12:15

一枝鹿



问一下，这个删除操作，对后续DAG的应用有什么意义吗，这个删除操作对代码优化有帮助吗

2024/05/29 12:33

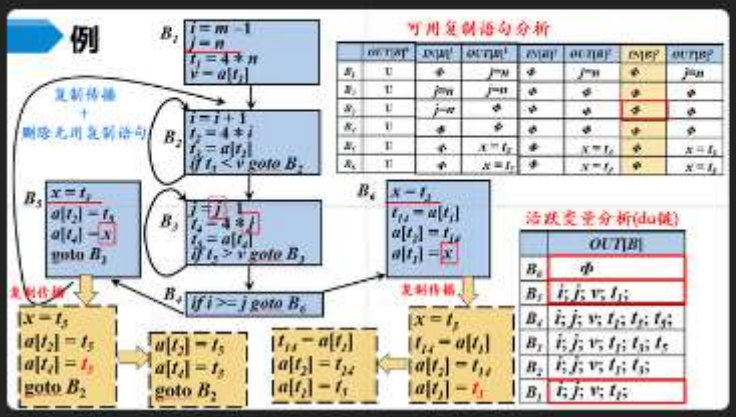
TSUNDOKU

感觉只是用来分析变量的最新值，对DAG删除无用代码和找局部公共子表达式好像没什么用

2024/05/29 16:00

请问哪里用到了活跃变量分析

TSUNDOKU



B5中的x,出了B5就不活跃了

TSUNDOKU

就把x=t3删了

2024/05/29 22:48

超级无敌暴龙战士（人智2103602-沈正冉）

$x=0;$

这种复制语句翻译成三地址代码还用不用添加临时变量了?

超级无敌暴龙战士（人智2103602-沈正冉）

t1=0;x=t1这种?

超级无敌暴龙战士（人智2103602-沈正冉）

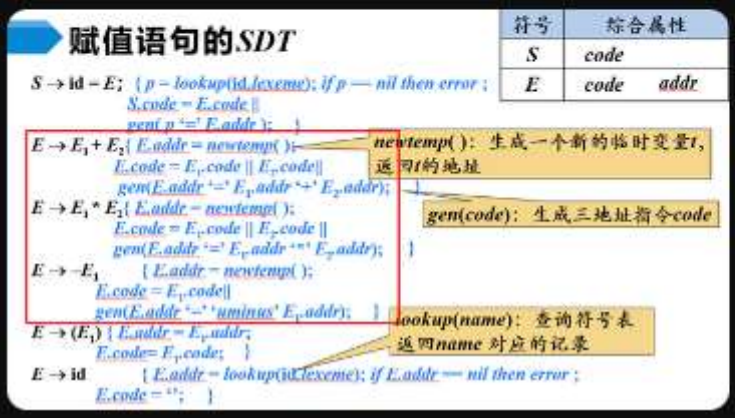
还是直接翻译成x=0;

2024/05/29 22:58

TSUNDOKU

直接就行吧

TSUNDOKU



TSUNDOKU

这几种生成临时变量了

2024/05/30 16:10

RxR

语法分析是先消除左递归还是先提取左公因子?

鸣剑

先消除左递归

鸣剑

按老师的说法，先消除左递归可能消掉左公因子，先消除左公因子可能产生左递归

2024/05/30 16:49

超级无敌暴龙战士（人智2103602-沈正冉）

为了实现数组引用翻译，除了数组的类型和数组的基地址，还需要符号表中数组的哪个信息？

A 数组各子类型的宽度

B 数组的维度

C 数组引用的下标

D 数组类型本身的宽度

请问这个为什么不选c？a[i]=base+i*w，A选项就是w，C选项就是i吧

一言为定

> 计科-琪铠
>
> 为了实现数组引用翻译，除了数组的类型和数组的基地址，还需要符号表中数组的哪个信息？
>
> A 数组各子类型的宽度
>
> B 数组的维度
>
> C 数组引用的下标
>
> D 数组类型本身的宽度

@计科-琪铠 符号表中信息?

2024/05/30 16:57

Wrecked.（人工智能-张智雄）

> 计科-琪铠
>
> 为了实现数组引用翻译，除了数组的类型和数组的基地址，还需要符号表中数组的哪个信息？
>
> A 数组各子类型的宽度
>
> B 数组的维度
>
> C 数组引用的下标
>
> D 数组类型本身的宽度

@计科-琪铠 i的信息不存在符号表中数组那块

2024/05/30 17:05

超级无敌暴龙战士（人智2103602-沈正冉）

> 张智雄
>
> @计科-琪铠 i的信息不存在符号表中数组那块

@张智雄 好的

超级无敌暴龙战士（人智2103602-沈正冉）

在自底向上的翻译中，假设栈顶已形成句柄：$A \rightarrow XYZ$，且A有继承属性$A.inh$,那么$A.inh$的位置在

A top

B top-1

C top-2

D top-3

这个题为什么选d啊

TSUNDOKU

答案解析

归约后，XYZ被弹出栈， A被压入栈，A将在现在X的位置top-2，A的继承属性一定在紧挨着A之下即top-3的位置

2024/05/30 17:11

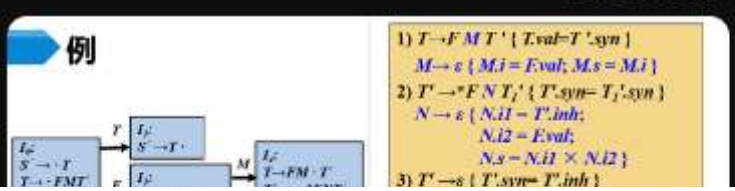
超级无敌暴龙战士（人智2103602-沈正冉）

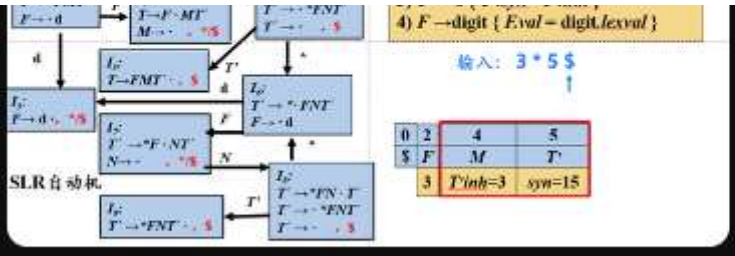
这里使用自底向上翻译，那对应S-SDD吧，S-SDD不是没有继承属性吗

TSUNDOKU

应该是改装

TSUNDOKU





TSUNDOKU

类似这种

TSUNDOKU

2024/05/31 21:11

Amadeus（21-生信-张恒铭）

> 新李
>
> 

@新李 有老哥写了这个的答案吗

2024/05/31 21:16

Wrecked.（人工智能-张智雄）

我猜c＝28，e＝e+f，a＝e+12

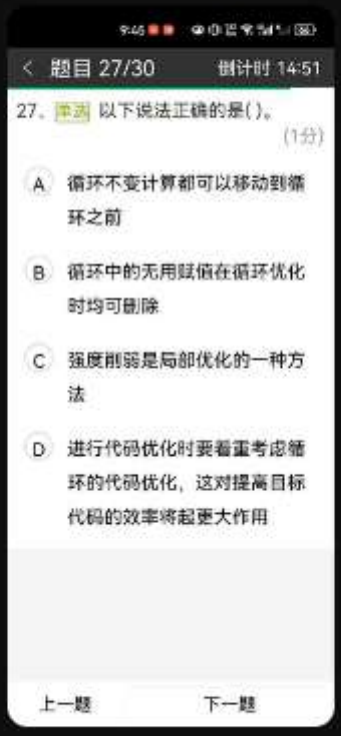
Amadeus（21-生信-张恒铭）

> 张智雄
>
> 我猜c＝28，e＝e+f，a＝e+12

@张智雄 我也这么觉得

2024/05/31 21:45

天马行空（人智2103602-张天一）



天马行空（人智2103602-张天一）

xdm选啥

一枝鹿

d

天马行空（人智2103602-张天一）

但ab哪错了

2024/05/31 21:50

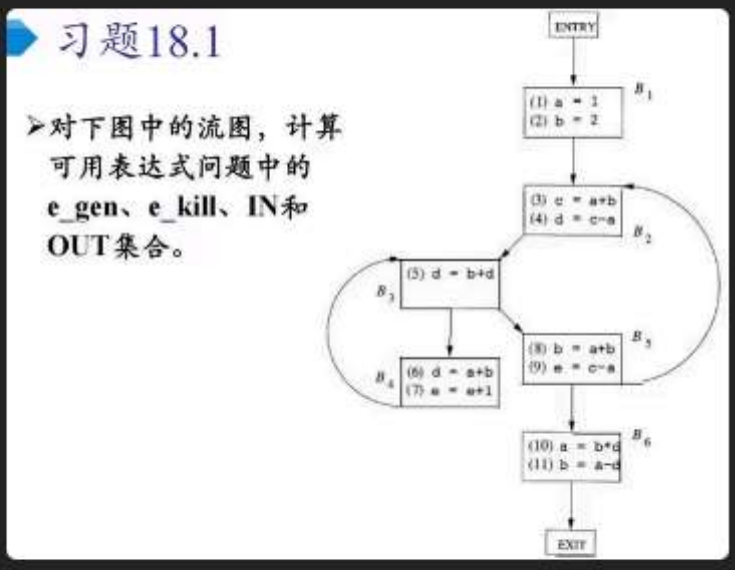
唯唯诺诺长生尊者（21-软工-刘思齐）

a 并不都可以

唯唯诺诺长生尊者（21-软工-刘思齐）

b 循环中无用 外面可能活跃

2024/06/01 11:37



请问一下在计算B1的e_gen时用不用加入a=1、b=1，看后面有个题涉及到复制传播

Wrecked. (人工智能-张智雄)

要吧

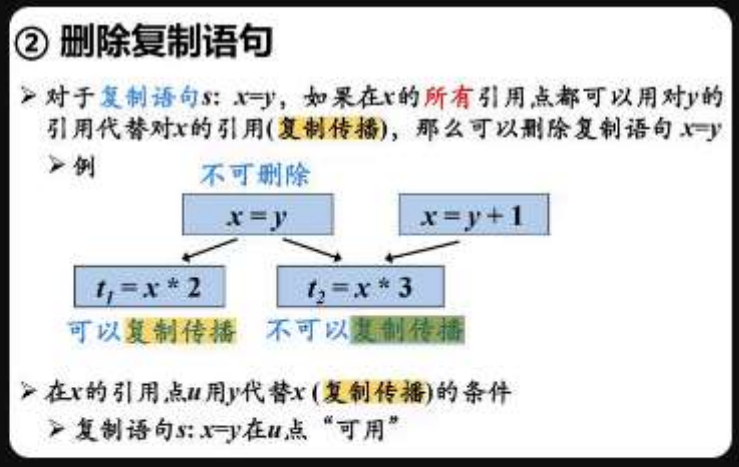
晴雨

复制也要加吗?

晴雨

我以为只加右边是表达式那种

绘星



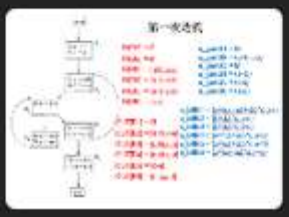
Wrecked. (人工智能-张智雄)

考虑复制转播得看有没有重定义吧

Wrecked.（人工智能-张智雄）

反正算kill的时候得算这俩

绘星

> RxR
>
> 

看这个答案里没有，感觉很奇怪

Wrecked.（人工智能-张智雄）

算gen应该算不算都行

2024/06/01 12:39

JHT213

> 绘星
>
> 

我又看了下概念，e_gen和e_kill的元素是表达式x op y。a=1，b=2是定值，和e_gen、e_kill没关系吧

JHT213

应该和kill有关

JHT213

答案给的e_killB1也确实不是空集

2024/06/01 12:45

绘星

🤔在复制传播时怎么说明可用呢?

e_kill不空我可以理解，因为对a和b重新定值且没有更

e_kill不空我可以理解，因为对a和b重新定值后没有再计算，杀死相应的表达式

lx

> 绘星
> 🤔在复制传播时怎么说明可用呢？ e_kill不空我可以理解...

感觉ppt这里不太严谨。复制传播那里的意思是把x =y这种也当作一个表达式去分析是不是可用的

2024/06/01 13:26

lx

我的想法是这快他会把x=y作为一个表达式。

欧克.

把x=y当做表达式?

lx

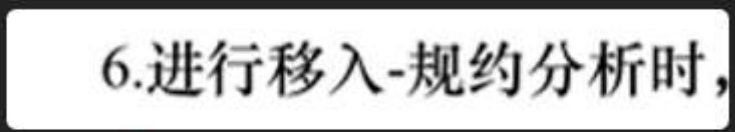
嗯，他不是说x等于y是可以用的。

lx

相当于前面的x op y是可用的。

2024/06/01 14:48

RxR



请问这个填什么

suteishi


LV2群主 TSUNDOKU
a[t2]=t5，t5=a[t2]，这里的a是常量，不涉及引用定值， t2都算做引用