

1

编译是对()。

- - A.**
机器语言的执行
- - B.**
汇编语言的翻译
- - C.**
高级语言的翻译
- - D.**
高级语言程序的解释执行

C

2

用高级语言编写的程序经编译后产生的程序叫()。

- - A.**
源程序
- - B.**
目标程序
- - C.**
连接程序

•

•

D.
解释程序

•

B

3

()不是编译程序的组成部分。

•

A.
词法分析程序

•

•

B.
代码生成程序

•

•

C.
设备管理程序

•

•

D.
语法分析程序

•

C

4

源程序是句子的集合，()可以较好地反映句子的结构。

- **A.**
线性表

•

- **B.**
树

•

- **C.**
完全图

•

- **D.**
堆栈

•

B

5

编译程序是一种()。

•

- A.**
汇编程序

•

- **B.**
翻译程序

•

- **C.**
解释程序

•

•

D.

目标程序

•

B

6

按逻辑上划分，编译程序第三步工作是()。

•

A.

语义分析

•

•

B.

词法分析

•

•

C.

语法分析

•

•

D.

代码生成

•

A

7

编译程序中语法分析器接收以()为单位的输入。

- - A.**
单词
-
-
- B.**
表达式
-
-
- C.**
产生式
-
-
- D.**
句子
-

A

8

编译过程中，语法分析器的任务就是()。

- - A.**
分析单词是怎样构成的
-
-
- B.**
分析单词串是如何构成语句和声明的
-
-
- C.**
分析语句和声明是如何构成程序的

•

•

D.

分析程序的结构

•

B

9

语法分析时所依据的是()

•

A.

语法规则

•

•

B.

词法规则

•

•

C.

语义规则

•

•

D.

等价变换规则

•

A

1

文法： $G: S \rightarrow xSx \mid y$ 所识别的语言是()。

- **A.**
 xyx
-
- **B.**
 $(xyx)^*$
-
- **C.**
 x^*yx^*
-
- **D.**
 $x^nyx^n (n \geq 0)$

D

2

给定文法 $A \rightarrow bA \mid ca$ ，为该文法句子的是()。

- **A.**
bba
-
- **B.**
cab
-
- **C.**

bca

•

D.
cba

C

3

设有文法 $G[S]: S \rightarrow S1 | S0 | Sa | Sc | a | b | c$ ，下列符号串中是该文法的句子有()。

•

A.
ab0

•

•

B.
a0b01

•

•

C.
a0b0a

•

•

D.
bc10

•

D

4

文法 G 产生的()的全体是该文法描述的语言。

•

A.
句型

•

•

B.
终结符集

•

•

C.
非终结符集

•

•

D.
句子

•

D

5

若文法G定义的语言是无限集，则文法必然是()。

•

A.
递归的

•

•

B.
上下文无关的

•

•

C.
二义性的

•

-
- **D.**
无二义性的
-

A
6

乔姆斯基(Chomsky)把文法分为四种类型，即0型、1型、2型、3型。其中3型文法是()。

-
- **A.**
非限制文法
-
-
- **B.**
正则文法
-
-
- **C.**
上下文有关文法
-
-
- **D.**
上下文无关文法
-

B
7

一个上下文无关文法G包括四个组成部分，它们是一组非终结符号，一组终结符号，一个开始符号，以及一组()。

- **A.**
句子
- **B.**
产生式
- **C.**
单词
- **D.**
句型

B
8

若一个文法是递归的，则它所产生的语言的句子()。

- **A.**
是无穷多个
-
- **B.**
是有穷多个
-
- **C.**
是可枚举的
-

D.
个数是常量

A
9

给定文法 $A \rightarrow bA \mid cc$ ，则符号串①cc ②bcbc ③bcbcc ④bccbcc ⑤bbbcc中，是该文法句子的是()。

A.
①

B.
③④⑤

C.
②④

D.
①⑤

D
10

文法 $E \rightarrow E+E \mid E * E \mid i$ 的句子 $i * i + i * i$ 有()棵不同的语法树。

A.

1

•

B.
3

•

C.
5

•

D.
7

C

11

文法 $S \rightarrow aaS \mid abc$ 定义的语言是()。

•

A.

$\{a^{2k}bc \mid k > 0\}$

•

B.

$\{a^kbc \mid k > 0\}$

•

C.

$\{a^{2k-1}bc \mid k > 0\}$

•

D.

$\{a^k a^k bc \mid k > 0\}$

C

12

文法G: $S \rightarrow xSx \mid xS \mid y$ 所识别的语言是 \emptyset 。

•

A.

$$x^m y x^n (m \geq n \geq 0)$$

•

B.

$$(xyx)^*$$

•

•

C.

$$x^n y x^n (n \geq 0)$$

•

D.

$$x^* y x^*$$

A

13

由文法的开始符号出发经过若干步（包括0步）推导产生的文法符号序列称为（）。

•

A.

语言

-
- **B.**
句型
-
-
- **C.**
句子
-
-
- **D.**
句柄
-

B

14

下列符号串不可以由符号集 $S = \{a, b\}$ 上的正闭包运算产生的是()。

-
- **A.**
 ϵ
-
- **B.**
a
-
- **C.**
aa
-
- **D.**
ab

A

15

文法G: $S \rightarrow x xS \mid y$ 所识别的语言是()。

•

A.

xy^*

•

B.

$(xy)^*$

•

C.

xx^*yx

•

D.

$(xx)^*y$

D

16

文法G: $S \rightarrow xS \mid y$ 所识别的语言是()。

•

A.

xy^*

•

B.

$(xy)^*$

•

C.

$$xx^*yx$$

•

D.

$$x^*y$$

D

1

词法分析器的输出结果是()。

•

A.

单词自身值

•

•

B.

单词在符号表中的位置

•

•

C.

单词的种别编码

•

•

D.

单词的种别编码和自身值

•

D

2

词法分析器不能()。

- - A.**
识别出数值常量
-
- - B.**
过滤源程序中的注释
-
- - C.**
扫描源程序并识别记号
-
- - D.**
发现括号不匹配
-

D

3

()这样一些语言，它们能被确定的有穷自动机识别，但不能用正则表达式表示。

- - A.**
存在
-
- - B.**
不存在
-

-
- **C.**
无法判定是否存在
-
-
- **D.**
以上答案都不对
-

B

4

两个有穷自动机等价是指它们的()。

-
- **A.**
状态数相等
-
-
- **B.**
有向弧数相等
-
-
- **C.**
所识别的语言相等
-
-
- **D.**
状态数和有向弧数相等
-

C

5

词法分析器用于识别()。

- - A.
句子
-
- - B.
产生式
-
- - C.
单词
-
- - D.
句型
-

C
6

正则表达式 R_1 和 R_2 等价是指()。

- - A.
 R_1 和 R_2 都是定义在一个字母表上的正则表达式
-
- - B.
 R_1 和 R_2 使用的运算符相同

-
- C.
 R_1 和 R_2 代表同一正则集
-
-
- D.
 R_1 和 R_2 代表不同正则集

C
7

已知文法 $G[S]: S \rightarrow A1, A \rightarrow A1|S0|0$, 与 G 等价的正规式是()。

-
- A.
 $0(0|1)^*$
-
-
- B.
 $1^*|0^*1$
-
-
- C.
 $0(1|10)^*1$
-
- D.
 $1(10|01)^*0$
-

C

8

与 $(a|b)^*(a|b)$ 等价的正规式是()。

•

A.

$a^*|b^*$

•

•

B.

$(ab)^*(a|b)$

•

•

C.

$(a|b)(a|b)^*$

•

•

D.

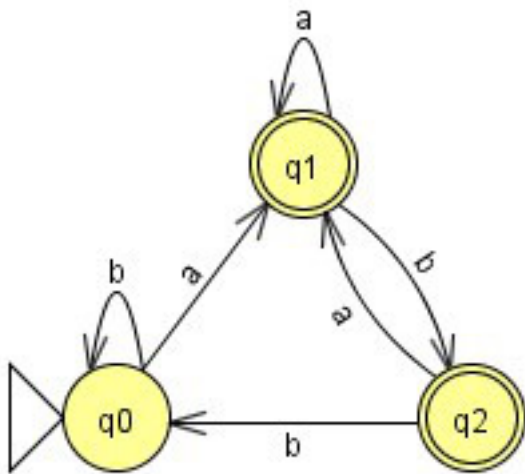
$(a|b)^*$

•

C

9

如图所示自动机M，请问下列哪个字符串不是M所能识别的()。



- **A.**
bbaa
-
- **B.**
abba
-
- **C.**
abab
-
- **D.**
aabb
-

D

10

有限状态自动机能识别()。

- **A.**
上下文无关语言

-
-
- **B.**
上下文有关语言
-
- **C.**
正规语言
-
- **D.**
0 型文法定义的语言

C
11

()不是DFA的成分。

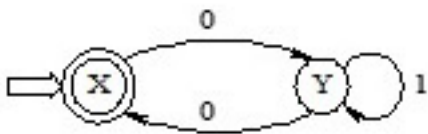
-
- **A.**
有穷字母表
-
-
- **B.**
多个初始状态的集合
-
-
- **C.**
多个终态的集合
-

D.
转换函数

B

12

DFA M(见图)接受的字集为()。



A.
以0开头的二进制数组成的集合

B.
以0结尾的二进制数组成的集合

C.
含奇数个0的二进制数组成的集合

D.
含偶数个0的二进制数组成的集合

D

13

同正规式 $(a|b)^+$ 等价的正规式是 ()。

•

A.

$(a|b)^*$

•

•

B.

$(a|b)(a|b)^*$

•

•

C.

$(ab)^*(ab)$

•

•

D.

$(a|b)|(a|b)^*$

B

14

称有限自动机 A_1 和 A_2 等价是指 ()。

•

A.

A_1 和 A_2 都是定义在一个字母表上的有限自动机

•

•

B.

A_1 和 A_2 状态数和有向边数相等

-
-
- C.
 A_1 和 A_2 状态数或有向边数相等
-
-
- D.
 A_1 和 A_2 所能识别的字符串集合相等
-

D

15

同正规式 $(a|b)^*$ 等价的正规式为()。

- A.
 $(a|b)^+$
-
- B.
 $a^*|b^*$
-
-
- C.
 $(ab)^*$
-
-
- D.
 $(a^*|b^*)^+$

D

16

词法分析器的加工对象是（ ）。

•

A.

中间代码

•

•

B.

单词

•

•

C.

源程序

•

•

D.

元程序

•

C

17

如果一个正规式所代表的集合是无穷的，则它必含有的运算是()。

•

A.

接运算“ \cdot ”

•

-
- **B.**
或运算“|”
-
-
- **C.**
闭包运算“*”
-
-
- **D.**
括号“（”和“）”
-

C

18

同正规式 a^*b^* 等价的文法是()。

-
- **A.**
 $G1: S \rightarrow aS | bS | \epsilon$
-
-
- **B.**
 $G2: S \rightarrow aSb | \epsilon$
-
-
- **C.**
 $G3: S \rightarrow aS | Sb | \epsilon$
-
-
- **D.**
 $G4: S \rightarrow abS | \epsilon$

•

C

19

一个正规式只能对应一个确定的有限状态自动机。

•

A.

•

B.

B

20

一个正规语言可能对应多个正规文法。

•

A.

•

B.

A

1

如果文法G是无二义的，则它的任何句子 α ()。

•

A.

最左推导和最右推导对应的语法树必定相同

•

•

B.

最左推导和最右推导对应的语法树可能不同

•

•

C.

最左推导和最右推导必定相同

•

•

D.

可能存在两个不同的最左推导，但它们对应的语法树相同

•

A

2

采用自上而下分析，不必()。

•

A.

消除回溯

•

•

B.

消除左递归

•

•

C.

消除右递归

•

-
- **D.**
提取公共左因子
-

C

3

识别上下文无关语言的自动机是()。

-
- **A.**
下推自动机
-
-
- **B.**
NFA
-
-
- **C.**
DFA
-
-
- **D.**
图灵机
-

A

4

()文法不是LL(1)的。

-
- **A.**
递归

•

•

B.

右递归

•

•

C.

2型

•

•

D.

含有公共左因子的

•

D

5

已知文法G是无二义的，则对G的任意句型 α ()。

•

A.

最左推导和最右推导对应的语法树必定相同

•

•

B.

最左推导和最右推导对应的语法树可能相同

•

•

C.

最左推导和最右推导必定相同

•

-
- **D.**
可能存在两个不同的最左推导，但他们对应的语法树相同
-

A

6

在自上而下的语法分析中，应从()开始分析。

-
- **A.**
句型
-
-
- **B.**
句子
-
-
- **C.**
文法开始符号
-
-
- **D.**
句柄
-

C

7

一个文法G，若()，则称它是LL（1）文法。

-
- **A.**
G中不含左递归

•

•

B.

G无二义性

•

•

C.

G的LL（1）分析表中不含多重定义的条目

•

•

D.

G中产生式不含左公因子

•

C

8

语法分析器的输入是（）。

•

A.

Token序列

•

•

B.

源程序

•

•

C.

目标程序

•

-
- **D.**
符号表
-

A
9

在递归子程序方法中，若文法存在左递归，则会使分析过程产生()。

-
- **A.**
回溯
-
-
- **B.**
非法调用
-
-
- **C.**
有限次调用
-
-
- **D.**
无限循环
-

D
10

LL (1) 分析法中“1”的含义是在输入串中查看一个输入符号，其目的是 () 。

-
- **A.**
确定最左推导

•

•

B.

确定句柄

•

•

C.

确定使用哪一个产生式进行展开

•

•

D.

确定是否推导

•

C

1

在语法分析处理中，FIRST集合、FOLLOW集合均是()。

•

A.

非终结符集

•

•

B.

终结符集

•

•

C.

字母表

-

-

- **D.**
状态集

-

B

2

在编译过程中，如果遇到错误应该()。

-

- **A.**
把错误理解成局部的错误

-

-

- **B.**
对错误在局部范围内进行纠正，继续向下分析

-

-

- **C.**
当发现错误时，跳过错误所在的语法单位继续分析下去

-

-

- **D.**
当发现错误时立即停止编译，待用户改正错误后再继续编译

-

C

3

已知文法G[S]:

$S \rightarrow eT|RT \quad T \rightarrow DR|\epsilon \quad R \rightarrow dR|\epsilon \quad D \rightarrow a|bd$

求 $FIRST(S) = ()$ 。

•

A.

$\{e\}$

•

B.

$\{e, d, a, b\}$

•

C.

$\{e, d\}$

•

D.

$\{e, d, a, b, \epsilon\}$

D

4

已知文法 $G[S]$:

$S \rightarrow eT|RT \quad T \rightarrow DR|\epsilon \quad R \rightarrow dR|\epsilon \quad D \rightarrow a|bd$

求 $FOLLOW(D) = ()$ 。

•

A.

$\{d, e\}$

•

B.

$\{d, \epsilon\}$

•

C.

$\{d, \$\}$

- D.
{a,d}

C

5

FIRST集中可以含有 ϵ 。

- A.

- B.

A

6

FOLLOW集中可以含有 ϵ 。

- A.

- B.

B

7

SELECT集中可以含有 ϵ 。

- A.

- B.

B

1

若 a 为终结符，则 $A \rightarrow \alpha \cdot a\beta$ 为()项目。

- - A.**
归约
-
- - B.**
移进
-
- - C.**
接受
-
- - D.**
待约
-

B

2

一个句型中的()称为该句型的句柄。

- - A.**
最左直接短语

- **B.**
最右直接短语
- **C.**
终结符
- **D.**
非终结符

A

3

在自底向上的语法分析方法中，分析的关键是()。

- **A.**
寻找句柄
-
- **B.**
寻找句型
-
- **C.**
消除递归
-
- **D.**
选择候选式
-

A

4

在自顶向下的语法分析方法中，分析的关键是()。

- - A.
寻找句柄
-
- - B.
寻找句型
-
- - C.
消除递归
-
- - D.
选择候选式
-

D

5

若B为非终结符，则 $A \rightarrow a \cdot Bb$ 为()。

- - A.
移进项目
-
- - B.
归约项目

-
-
- **C.**
接受项目
-
-
- **D.**
待约项目
-

D
6

在规范归约中，用()来刻画可归约串。

-
- **A.**
直接短语
-
-
- **B.**
句柄
-
-
- **C.**
最左素短语
-
-
- **D.**
素短语
-

B 7

下列动作中，不是自下而上分析动作的是()。

- - A.
移进
-
- - B.
展开
-
- - C.
接受
-
- - D.
报错
-

B 8

下列动作中，不是自上而下分析动作的是()。

- - A.
匹配
-
- - B.
展开

-
-
- **C.**
移进
-
-
- **D.**
报错
-

C
9

若B为非终结符，则 $A \rightarrow \alpha \cdot$ 为()项目。

-
- **A.**
归约
-
-
- **B.**
移进
-
-
- **C.**
接受
-
-
- **D.**
待约
-

A

10

设有文法G[T]:

$T \rightarrow T * F | F$

$F \rightarrow F \uparrow P | P$

$P \rightarrow (T) | a$

该文法句型 $T * P \uparrow (T * F)$ 的句柄是下列符号串()。

•

A.

$(T * F)$

•

•

B.

$T * F$

•

•

C.

P

•

•

D.

$P \uparrow (T * F)$

•

C

11

LR分析表中的转移表（goto）是以（ ）作为列标题的。

•

A.

终结符

•

•

B.

非终结符

•

•

C.

终结符或非终结符

•

•

D.

表示状态的整型数

•

B

12

LR分析表中的动作表（action）是以（ ）作为列标题的。

•

A.

终结符

•

•

B.

非终结符

•

•

C.

终结符或非终结符

•

-
- **D.**
终结符和结束符\$
-

D

13

设有文法 $G[T]$:

$T \rightarrow T * F | F$

$F \rightarrow F \uparrow P | P$

$P \rightarrow (T) | a$

该文法句型 $T * F \uparrow (T * F)$ 的句柄是下列符号串 () 。

-
- **A.**
(T * F)
-
- **B.**
T * F
-
- **C.**
P
-
- **D.**
 $P \uparrow (T * F)$

B

14

下列项目中为可归约项目的是 () 。

-
- **A.**
 $E' \rightarrow \cdot E$

•

•

B.

$L \rightarrow \cdot$

•

•

C.

$L \rightarrow \cdot L$

•

•

D.

$F \rightarrow L^* \cdot F$

•

B

15

同正规式 $(a|b)^+$ 等价的正规式是()。

•

A.

$(a|b)^*$

•

B.

$(a|b)(a|b)^*$

•

C.

$(ab)^*(ab)$

-
- **D.**
 $(a|b)|(a|b)^*$

B

16

LR分析器的核心部分是一张分析表，该表由()组成。

-
- **A.**
ACTION表
-
- **B.**
GOTO表
-
-
- **C.**
预测分析表
-
-
- **D.**
ACTION表和GOTO表
-

D

1

一个LR(1)文法合并同心集后若不是LALR(1)文法()

-
- **A.**
则可能存在移进/归约冲突

•

•

B.

则可能存在归约/归约冲突

•

•

C.

则可能存在移进/归约冲突和归约/归约冲突

•

•

D.

以上说法都不对

•

B

2

若状态 k 含有项目“ $A \rightarrow \alpha \cdot$ ”，且仅当输入符号 $a \in FOLLOW(A)$ 时，才用规则“ $A \rightarrow \alpha$ ”归约的语法分析方法是()。

•

A.

LALR分析法

•

•

B.

R(0)分析法

•

•

C.

LR(1)分析法

•

•

D.
SLR(1)分析法

•

D

3

LR(1)文法都是()。

•

A.
无二义性且无左递归

•

•

B.
可能有二义性但无左递归

•

•

C.
无二义性但可能是左递归

•

•

D.
可以既有二义性又有左递归

•

C

4

同心集合并可能会产生新的()冲突。

-
- **A.**
二义
-
-
- **B.**
移进/移进
-
-
- **C.**
移进/归约
-
-
- **D.**
归约/归约
-

D

5

就文法的描述能力来说，有()。

-
- **A.**
 $\text{SLR}(1) \subset \text{LR}(0)$
-
- **B.**
 $\text{LR}(1) \subset \text{LR}(0)$
-
-
- **C.**
 $\text{SLR}(1) \subset \text{LR}(1)$

•

•

D.

无二义文法 \subset LR(1)

•

C

6

在LR(0)的Action表中，如果某行中存在标记为“ T_j ”的栏，则()。

•

A.

该行必定填满“ T_j ”

•

•

B.

该行未必填满“ T_j ”

•

•

C.

其他行也有“ T_j ”

•

D.

goto表中也有“ T_j ”

A

7

若状态k含有项目“ $A \rightarrow \alpha \cdot$ ”，对任意非终结符a，都用规则“ $A \rightarrow \alpha$ ”归约的语法分析方法是()。

-
- **A.**
LALR分析法
-
- **B.**
LR(0)分析法
-
- **C.**
LR(1)分析法
-
- **D.**
SLR(1)分析法
-

B

8

在SLR(1)的Action表中，如果某行中存在标记为“ r_j ”的栏，则()。

-
- **A.**
该行必定填满“ r_j ”
-
- **B.**
该行未必填满“ r_j ”
-

-
- **C.**
其他行也有“ T_j ”
-
-
- **D.**
goto表中也有“ T_j ”
-

B

1

文法G[S]及其语法制导翻译定义如下：

产生式	语义动作
$S' \rightarrow S$	print(S.num)
$S \rightarrow (L)$	S.num = L.num + 1
$S \rightarrow a$	S.num = 0
$L \rightarrow L(1), S$	L.num = L(1).num + S.num
$L \rightarrow S$	L.num = S.num

若输入为(a,(a)), 且采用自底向上的分析方法，则输出为()。

-
- **A.**
0
-
- **B.**
1
-
- **C.**
2
-
- **D.**
4

C

2

有文法G及其语法制导翻译如下所示(语义规则中的*和+分别是常规意义下的算术运算符):

$E \rightarrow E(1) \wedge T \{E.val = E(1).val * T.val\}$

$E \rightarrow T \{E.val = T.val\}$

$T \rightarrow T(1) \# n \{T.val = T(1).val + n.val\}$

$T \rightarrow n \{T.val = n.val\}$

则分析句子 $3 \wedge 3 \# 4$ 其值为()。

•

A.
10

•

B.
21

•

C.
14

•

D.
24

B

3

有一语法指导定义如下:

$S \rightarrow bAb \quad \text{print "1"}$

$A \rightarrow (B \quad \text{print "2"}$

$A \rightarrow a \quad \text{print "3"}$

$B \rightarrow aA) \quad \text{print "4"}$

若输入序列为 $b(a(a(aa)))b$ ，且采用自底向上的分析方法，则输出序列为()。

•

A.
32224441

•

B.
34242421

•

C.
12424243

•

D.
34442212

B

4

有一语法指导定义如下，其中+表示符号连接运算：

$S \rightarrow B$ print B.vers

$B \rightarrow a$ B.vers=a

$B \rightarrow b$ B.vers=b

$B \rightarrow Ba$ B.vers=a+B.vers

$B \rightarrow Bb$ B.vers=b+B.vers

若输入序列为abab，且采用自底向上的分析方法，则输出序列为()。

•

A.
aabb

•

B.
abab

•

C.

bbaa

- **D.**
baba

D

5

使用()可以定义一个程序的意义。

- **A.**
语义规则
-
- **B.**
词法规则
-
- **C.**
产生规则
-
- **D.**
词法规则
-

A

6

以下说法正确的是()。

-
- **A.**
语义规则中的属性有两种：综合属性与继承属性
-
-
- **B.**
终结符只有继承属性，它由词法分析器提供
-
-
- **C.**
非终结符可以有综合属性，但不能有继承属性
-
-
- **D.**
属性值在分析过程中可以进行计算，但不能传递
-

A

1

关于将L-SDD转换为SDT的规则，以下选项中，正确的是()。

-
- **A.**
将计算某个非终结符号A的继承属性的动作放在产生式的最后
-
-
- **B.**
将计算一个产生式左部符号的继承属性的动作放置在产生式的最后
-
- **C.**

将计算某个非终结符号A的继承属性的动作插入到产生式右部中紧靠在A的本次出现之前的位置上

•

•

D.

将每个语义动作都放在产生式的最后

•

C

2

以下说法不正确的是()。

•

A.

如果一个S-SDD的基本文法可以使用LR分析技术，那么它的SDT可以在LL语法分析过程中实现

•

•

B.

如果一个S-SDD的基本文法可以使用LR分析技术，那么它的SDT可以在LR语法分析过程中实现

•

•

C.

如果一个L-SDD的基本文法可以使用LL分析技术，那么它的SDT可以在LL语法分析过程中实现

•

•

D.

如果一个L-SDD的基本文法可以使用LL分析技术，那么它的SDT可以在LR语法分析过程中实现

•

A

3

以下说法不正确的是()。

•

A.

使用语法制导翻译方案的编译程序能同时进行语法分析和语义分析

•

B.

语法制导翻译方案(*SDT*)是在产生式右部中嵌入了程序片段(称为语义动作)的*CFG*

•

C.

SDD可以看作是SDT的具体实施方案

•

D.

将一个S-SDD转换为SDT的方法是：将每个语义动作都放在产生式的最后

•

C

4

在非递归的预测分析过程中进行翻译，以下说法不正确的是()。

•

A.

要想在非递归的预测分析过程中进行翻译，需要扩展语法分析栈

•

•

B.

非终结符A的继承属性和综合属性的计算时机不同

•

•

C.

将非终结符A的继承属性和综合属性存放在不同的记录中

•

•

D.

综合属性在A出现之前就可以计算

•

D

5

在非递归的预测分析过程中进行翻译，以下说法不正确的是()。

•

A.

要想在非递归的预测分析过程中进行翻译，需要扩展语法分析栈

•

•

B.

综合记录用于存放非终结符综合属性值

•

•

C.

动作记录，用来存放指向将被执行的语义动作代码的指针

•

- **D.**
综合属性存放在A本身的记录中

D
6

在非递归的预测分析过程中进行翻译，以下说法不正确的是()。

- **A.**
分析栈中的每一个记录都对应着一段执行代码
- **B.**
综合记录出栈时，要将综合属性值复制给后面特定的语义动作
- **C.**
变量展开时(即变量本身的记录出栈时)，如果其含有继承属性，则要将继承属性值复制给后面特定的语义动作
- **D.**
继承属性在A的儿子们都分析完毕之后才能计算

D

1

在递归的预测分析过程中进行翻译，以下说法不正确的是()。

- **A.**
可以将一个递归的预测分析器扩展为一个翻译器

•

B.

在语法分析器中，每个非终结符A对应一个过程，在做语义分析时，要将过程扩展成一个函数

•

C.

以继承属性作为函数的参数，以综合属性作为函数的返回值

•

D.

以综合属性作为函数的参数，以继承属性作为函数的返回值

D
2

在递归的预测分析过程中进行翻译，以下说法不正确的是()。

•

A.

在语法分析器中，每个非终结符A对应一个过程，在做语义分析时，要将过程扩展成一个函数

•

B.

对出现在A产生式右部中的每个文法符号的每个属性都设置一个局部变量

•

C.

如果非终结符含有继承属性，需要将函数调用的返回值赋给相应的局部变量

•

D.

对于产生式右部的每个动作，将其代码复制到语法分析器，并把对属性

的引用改为对相应变量的引用

C

3

以下说法不正确的是()。

- - A.
语法制导翻译方案只限自底向上的分析方法
- - B.
给定一个以LL文法为基础的L-SDD，可以修改这个文法，并在LR语法分析过程中计算这个新文法之上的SDD
- - C.
对于这个内嵌的语义动作，向文法中引入一个标记非终结符M来替换它
- - D.
每个标记非终结符M对应着一个空产生式 $M \rightarrow \epsilon$ ，该产生式对应着一段语义子程序，它的任务就是完成M所替换的那个语义动作要完成的工作

A

4

给定一个以LL文法为基础的L-属性定义，可以修改这个文法，并在LR语法分析过程中计算这个新文法之上的SDD。

- - A.
- - B.

A

5

在各个非终结符之前放置语义动作来计算它的继承属性，并在产生式后端放置语义动作计算综合属性。

•

A.

•

B.

A

6

在各个非终结符之前放置语义动作来计算它的综合属性，并在产生式后端放置语义动作计算继承属性。

•

A.

•

B.

B

7

对每个内嵌的语义动作，向文法中引入一个标记非终结符来替换它。每个这样的位置都有一个不同的标记，并且对于任意一个标记M都有一个产生式 $M \rightarrow \epsilon$ 。

•

A.

•

B.

A

1

中间代码生成时所依据的是()。

- - A.
语法规则
- - B.
词法规则
- - C.
语义规则
- - D.
等价变换规则

C

2

在编译程序中与中间代码生成无关的是()。

- - A.
便于目标代码的优化
- - B.
便于存储空间的组织
- - C.

便于编译程序的移植

- **D.**
便于目标代码的移植

D

3

以下说法不正确的是()。

- **A.**
对于声明语句，语义分析的主要任务就是收集标识符的类型等属性信息，为每一个名字分配一个相对地址

- **B.**
从变量类型可以知道该变量在运行时刻需要的内存数量。在编译时刻，可以使用这些数量为每一个名字分配一个相对地址

- **C.**
名字的类型和相对地址信息保存在相应的符号表条目中

- **D.**
对声明的处理要构造符号表，但不产生中间代码

D

4

以下说法不正确的是()。

- **A.**

类型自身也有结构，用类型表达式来表示这种结构

- - B.**
基本类型不是类型表达式
- - C.**
类型名也是类型表达式
- - D.**
将类型构造符作用于类型表达式可以构成新的类型表达式

B

5

数组元素的地址计算与数组的存储方式有关。

- - A.**
- - B.**

A

6

在程序中标识符的出现仅为使用性的。

- - A.**
- - B.**

B
7

在编译阶段只对可执行语句进行翻译。

- A.
- B.

B
8

在程序中标识符的出现仅为定义性的。

- A.
- B.

B

1

有文法G及其语法制导翻译如下所示(语义规则中的*和+分别是常规意义下的算术运算符):

$$E \rightarrow E^{(1)} \wedge T \{ E.val = E^{(1)}.val * T.val \}$$

$$E \rightarrow T \{ E.val = T.val \}$$

$$T \rightarrow T^{(1)} \# n \{ T.val = T^{(1)}.val + n.val \}$$

$$T \rightarrow n \{ T.val = n.val \}$$

则分析句子 $1 \wedge 2 \wedge 3 \# 4$ 其值为()。

- **A.**
10

- **B.**
34

- **C.**
14

- **D.**
54

C
2

用()可以把 $a:=b+c$ 翻译成四元式序列。

- **A.**
语法规则

- **B.**
词法规则

- **C.**
语义规则

- **D.**
等价变换规则

C 3

有文法G及其语法制导翻译如下所示(语义规则中的*和+分别是常规意义下的算术运算符):

$$E \rightarrow E^{(1)} \wedge T \{ E.val = E^{(1)}.val * T.val \}$$

$$E \rightarrow T \{ E.val = T.val \}$$

$$T \rightarrow T^{(1)} \# n \{ T.val = T^{(1)}.val + n.val \}$$

$$T \rightarrow n \{ T.val = n.val \}$$

则分析句子 $2 \wedge 3 \# 4$ 其值为()。

- A.
10
- B.
21
- C.
14
- D.
24

C 4

以下说法不正确的是()。

- A.
赋值语句翻译的主要任务是生成对表达式求值的三地址码

•

B.

在增量翻译方法中，gen()函数不仅要构造出一个新的三地址指令，还要将它添加到至今为止已生成的指令序列之后

•

C.

如果一个赋值语句中涉及到数组元素，那么将该语句翻译成三地址码时要解决的主要问题是确定数组元素的存放地址，也就是数组元素的寻址

•

D.

数组元素的地址计算与数组的存储方式无关

D

5

数组元素的地址由两部分构成，一部分是基地址，另一部分是偏移量。

•

A.

•

B.

A

6

基地址通过查符号表即可获得。

•

A.

•

B.

A

7

数组元素的偏移地址的计算只取决于数组的下标。

•

A.

•

B.

B

8

设计数组引用的SDT的关键问题是：如何将地址计算公式和数组引用的文法关联起来。

•

A.

•

B.

A

1

关于布尔表达式的叙述，不正确的是()。

•

A.

布尔常量是布尔表达式

•

B.

布尔常量不是布尔表达式

-
- **C.**
关系表达式是布尔表达式
-
- **D.**
将括号和逻辑运算符作用于布尔表达式得到一个新的布尔表达式

B

2

以下说法不正确的是()。

- **A.**
为布尔表达式和控制流语句生成目标代码时，关键问题之一是确定跳转指令的目标标号
- **B.**
在生成跳转指令时，就可以确定目标标号
- **C.**
在生成跳转指令时，目标标号还不能确定
- **D.**
可以将标号的地址作为继承属性传递到生成相关跳转指令的地方，但是这样的做法需要再进行一趟处理，将标号和具体地址绑定起来

B

3

令非终结符B表示布尔表达式，为B设置两个继承属性true和false。B.true是一个地址，该地址中存放了当B为真时控制流转向的指令的标号；B.false也是一个地址，该地址中存放了当B为假时控制流转向的指令的标号，则下列SDT中不正确的是()。

- - A.

$$B \rightarrow \text{true} \{ \text{gen}(\text{'goto' } B.\text{true}); \}$$
- - B.

$$B \rightarrow \text{false} \{ \text{gen}(\text{'goto' } B.\text{false}); \}$$
- - C.

$$B \rightarrow (\{ B_1.\text{true} = B.\text{true}; B_1.\text{false} = B.\text{false}; \} B_1)$$
- - D.

$$B \rightarrow \text{not} \{ B_1.\text{true} = B.\text{true}; B_1.\text{false} = B.\text{false}; \} B_1$$

D

4

令非终结符B表示布尔表达式，为B设置两个继承属性true和false。B.true是一个地址，该地址中存放了当B为真时控制流转向的指令的标号；B.false也是一个地址，该地址中存放了当B为假时控制流转向的指令的标号，则下列SDT中不正确的是()。

- - A.

$$B \rightarrow \text{true} \{ \text{gen}(\text{'goto' } B.\text{true}); \}$$
- - B.

$$B \rightarrow \text{false} \{ \text{gen}(\text{'goto' } B.\text{false}); \}$$

- C.

$$B \rightarrow (B_1 \{ B_1.true = B.true; B_1.false = B.false; \})$$
- D.

$$B \rightarrow not \{ B_1.true = B.false; B_1.false = B.true; \} B_1$$

C

5

在分支和循环中会用到条件式，而用作条件式的通常是布尔表达式。

- A.
- B.

A

6

在控制流语句的翻译中，布尔表达式B被翻译成由跳转指令构成的跳转代码。

- A.
- B.

A

7

逻辑运算符&&、|| 和 ! 会出现在代码中。

- A.

•

B.

B

1

令非终结符B表示布尔表达式。在布尔表达式的回填技术中，为非终结符B设置两个综合属性：

B.truelist：指向一个包含跳转指令的列表，这些指令最终获得的目标标号就是当B为真时控制流应该转向的指令的标号；B.falselist：指向一个包含跳转指令的列表，这些指令最终获得的目标标号就是当B为假时控制流应该转向的指令的标号。则下列SDT中不正确的是()。

•

A.

$B \rightarrow E_1 \text{relop} E_2$

{

$B.\text{truelist} = \text{makelist}(\text{nextquad});$

$B.\text{falselist} = \text{makelist}(\text{nextquad} + 1);$

$\text{gen}(\text{'goto_'});$

}

•

B.

$B \rightarrow \text{true}$

{

$B.\text{truelist} = \text{makelist}(\text{nextquad});$

$\text{gen}(\text{'goto_'});$

}

- - C.

$$B \rightarrow false$$

```

{
    B.falselist=makelist(nextquad);
    gen('goto_');
}
          
```
 - D.

$$B \rightarrow (B_1)$$

```

{
    B.truelist = B1.truelist;
    B.falselist = B1.falselist;
}
          
```

A

2

在下面的语句中，()不需要回填技术。

- - A.
 赋值语句
- - B.
 goto语句
- - C.
 条件语句

- **D.**
循环语句

A
3

四元式之间的联系是通过()实现的。

- **A.**
指示器
- **B.**
临时变量
- **C.**
符号表
- **D.**
程序变量

B
4

四元式表示法的优点为 ()。

- **A.**
不便于优化处理，但便于表的更动
- **B.**
不便于优化处理，但节省存储空间

- - C.
便于优化处理，也便于表的更动
- - D.
便于表的更动，也节省存储空间

C

5

在回填技术中，生成一个跳转指令时，暂时不指定该跳转指令的目标标号。

- - A.
- - B.

A

6

在回填技术中，同一个列表list中的跳转指令具有相同的目标标号。

- - A.
- - B.

A

7

在回填技术中，同一个列表list中的跳转指令可能具有不同的目标标号。

- A.
- B.

B

8

在回填技术中，等到能够确定正确的目标标号时，才去填充指令的目标标号。

- A.
- B.

A

1

在目标代码生成阶段，符号表用于()。

- A.
目标代码生成
- B.
语义检查
- C.
语法检查
- D.

地址分配

D

2

PASCAL语言中过程声明的局部变量地址分配在()。

- - A.
调用者的数据区中
- - B.
被调用者的数据区中
- - C.
主程序的数据区中
- - D.
公共数据区中

B

3

编译方法中，动态存储分配的含义是()。

- - A.
在编译阶段为源程序中的量进行分配
- - B.
在编译阶段为源程序中的量进行分配，运行时可动态调整

- - C.**
在运行阶段为源程序中的量进行分配
- - D.**
都不正确

C

4

运行阶段的存储组织与管理的目的是()。

- - A.**
提高编译程序的运行速度
- - B.**
为运行阶段的存储分配做准备及提高目标程序的运行速度
- - C.**
优化运行空间的管理
- - D.**
节省内存空间

B

5

以下说法正确的是()。

- - A.**
对于数据空间的存贮分配，FORTRAN采用动态贮存分配策略

- **B.**
对于数据空间的存贮分配，C语言仅采用栈式贮存分配策略
- **C.**
动态存储分配是指在编译阶段对源程序中的量进行分配，以使目标代码在运行时加快运行速度
- **D.**
如果两个临时变量的作用域不相交，则可以将它们分配在同一单元中

D

6

以下说法正确的是()。

- **A.**
编译程序除解决源程序中用户定义的量在运行时刻的存储组织与分配问题之外，还应完成为临时变量和参与运算的寄存器组织好存储空间的任务
- **B.**
由于C语言的函数允许递归调用，因此对C语言中的所有变量的单元分配一律采用动态分配方式
- **C.**
动态数组的存储空间在编译时即可完全确定
- **D.**

“运算符与运算对象类型不符”属于语法错误

A

7

以下说法正确的是()。

•

A.

符号表由词法分析程序建立，由语法分析程序使用

•

B.

符号表的内容在词法分析阶段填入并在以后各个阶段得到使用

•

C.

对一般的程序设计语言而言，其编译程序的符号表应包含哪些内容及何时填入这些信息不能一概而论

•

D.

“运算符与运算对象类型不符”属于语法错误

C

1

优化可生成 () 的目标代码。

•

A.

运行时间较短

•

B.

占用存储空间较小

- - C.**
运行时间短但占用内存空间大
- - D.**
运行时间短且占用存储空间小

D

2

基本块内的优化为 ()。

- - A.**
代码外提，删除归纳变量
- - B.**
删除多余运算，删除无用赋值
- - C.**
强度削弱，代码外提
- - D.**
循环展开，循环合并

B

3

对一个基本块来说， ()是正确的。

- - A.**
只有一个入口语句和一个出口语句
- - B.**
有一个入口语句和多个出口语句
- - C.**
有多个入口语句和一个出口语句
- - D.**
有多个入口语句和多个出口语句

A

4

在程序流图中，我们称具有下述性质()。

- - A.**
它们是非连通的且只有一个入口结点
- - B.**
它们是强连通的但有多多个入口结点
- - C.**
它们是非连通的但有多多个入口结点
- - D.**
它们是强连通的且只有一个入口结点

D

5

中间代码的优化依赖于具体的计算机。

- A.
- B.

B

6

代码优化应以等价变换为基础，既不改变程序的运行结果，又能使生成的目标代码更有效。

- A.
- B.

A

7

一个程序可用一个流图来表示。

- A.
- B.

A

8

优化工作只能在中间代码这一层次上进行。

•

A.

•

B.

B

1

数据流分析的主要应用不包括()。

•

A.

到达-定值分析

•

B.

活跃变量分析

•

C.

可用表达式分析

•

D.

自然循环分析

D

2

以下说法不正确的是()。

- - A.
大部分全局优化是基于数据流分析技术实现的
- - B.
数据流分析是一组用来获取程序执行路径上的数据流信息的技术
- - C.
在每一种数据流分析应用中，都会把每个程序点和一个数据流值关联起来
- - D.
一个语句之前和之后的数据流值不受该语句的语义的约束

D
3

以下说法不正确的是()。

- - A.
一个赋值语句之前和之后的数据流值的关系被称为传递函数
- - B.
传递函数有两种风格，一种是信息沿执行路径前向传播，称为前向数据流问题；另一种是信息沿执行路径逆向传播,称为逆向数据流问题
- - C.
在逆向数据流问题中，一个语句s的传递函数以语句前的数据流值作为输入，以语句后的数据流值作为输出

•
D.

设基本块B由语句 s_1, s_2, \dots, s_n 顺序组成, 则 s_i 输出的数据流值与 s_{i+1} 输入的数据流值相同

C

4

以下说法不正确的是()。

•
A.

变量x的定值是 (可能) 将一个值赋给x的语句

•
B.

假设存在一条从紧跟在x的定值d后面的点到达某一程序点p的路径, 如果在此路径上有对变量x的其它定值d', 则称定值d被定值d' “杀死”了

•
C.

如果存在一条从紧跟在x的定值d后面的点到达某一程序点p的路径, 则称定值d到达程序点p

•
D.

直观地讲, 如果某个变量x的一个定值d到达点p, 在点p处使用的x的值可能就是由d最后赋予的

C

5

以下说法不正确的是()。

•
A.

基本块B的传递函数 f_B 可以通过将基本块中各语句的传递函数组合起来

得到

•

B.

如果循环中含有赋值 $x=y+z$ ，而 y 和 z 所有可能的定值都在循环外面(包括 y 或 z 是常数的特殊情况)，那么 $y+z$ 就是循环不变计算

•

C.

定值 $d: u = v + w$ “生成”了一个对变量 u 的定值 d ，并“杀死”了程序中其它对 u 的定值

•

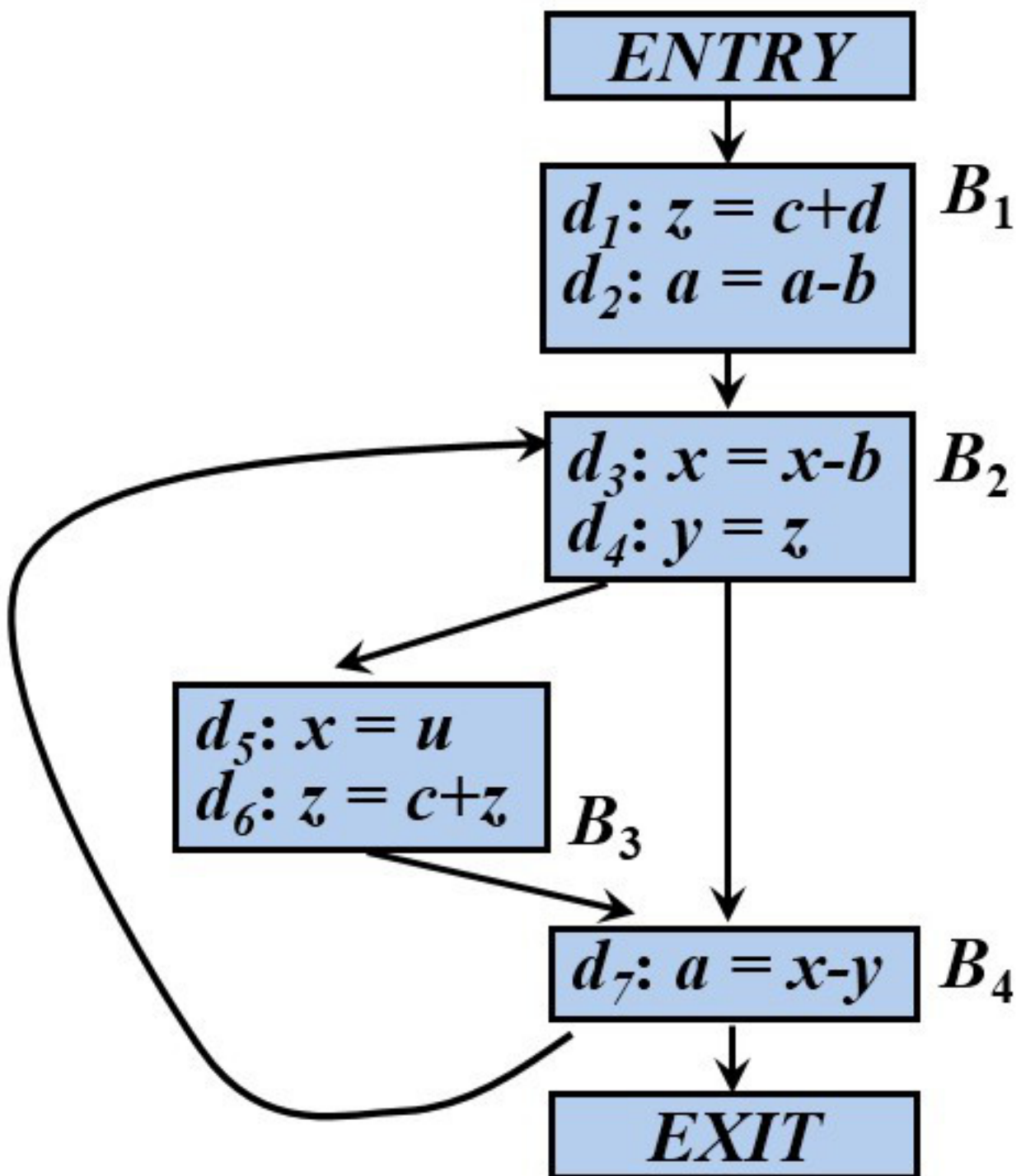
D.

对于到达定值问题，一个基本块 B 的IN值等于它的所有前驱基本块 P 的OUT值的交集

D

6

下图中，可以到达基本块B3的入口处的定值集合是()。



- A.
 $\{d_1, d_2, d_3, d_4, d_6, d_7\}$
-
- B.
 $\{d_3, d_4, d_6, d_7\}$

•

•

C.

$$\{d_1, d_2, d_3, d_4\}$$

•

•

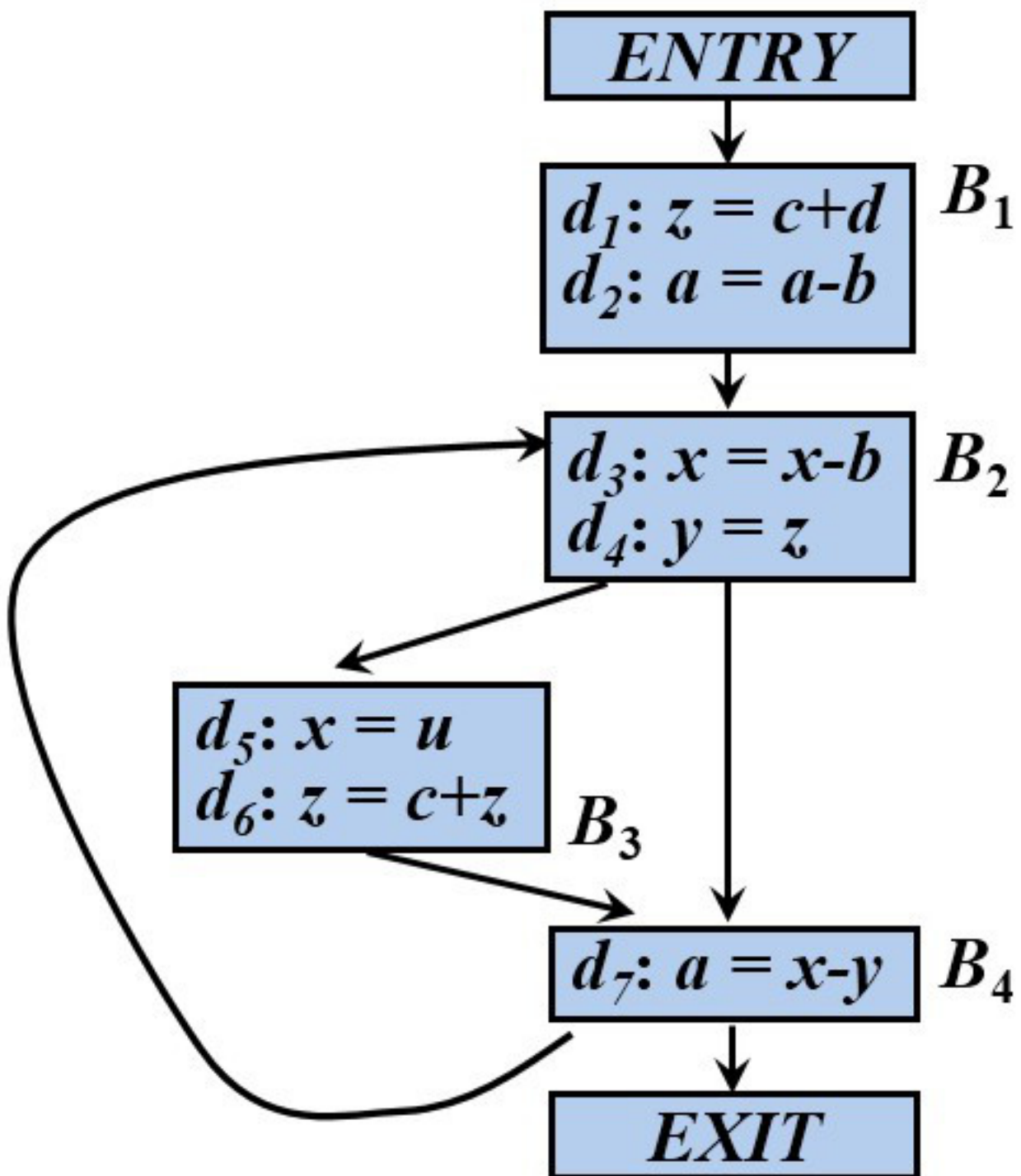
D.

$$\{d_3, d_4\}$$

A

7

关于下图的叙述中，错误的是()。



•

A.

$$gen_{B1} = \{d_1, d_2\} \quad kill_{B1} = \{d_6, d_7\}$$

•

B.

$$gen_{B2} = \{d_3, d_4\} \quad kill_{B2} = \{d_6, d_7\}$$

-
- C.
 $gen_{B3} = \{d_5, d_6\} \quad kill_{B3} = \{d_1, d_3\}$
-
- D.
 $gen_{B4} = \{d_7\} \quad kill_{B4} = \{d_2\}$

B

8

以下说法不正确的是()。

- A.
引用-定值链(ud-chains)是一个列表，对于变量的每一次引用，到达该引用的所有定值都在该列表中
- B.
如果块B中变量a的引用之前有a的定值，那么只有a的最后一次定值会在该引用的ud链中
- C.
如果块B中变量a的引用之前没有a的定值，那么a的这次引用的ud链就是IN[B]中a的定值的集合
- D.
如果块B中变量a的引用之前没有a的定值，那么只有a的最后一次定值会在该引用的ud链中

D

1

以下属于逆向数据流问题的是()。

- - A.
到达-定值分析
- - B.
活跃变量分析
- - C.
可用表达式分析
- - D.
寻找支配结点

B

2

以下说法不正确的是()。

- - A.
对于变量x和程序点p，如果在流图中沿着从p开始的某条路径会引用变量x在p点的值，则称变量x在点p是活跃的，否则称变量x在点p不活跃
- - B.
变量在某点活跃，表示变量在该点的值在以后会被使用

•
C.

判断在基本块出口之后，变量的值是否还被引用的这种判断工作称为活跃变量分析

•
D.

对于活跃变量问题，一个基本块B的OUT值等于它的所有后继基本块S的IN值的交集

D

3

以下说法不正确的是()。

•
A.

活跃变量信息可用于检测循环不变计算

•
B.

无用赋值：如果x在点p的定值在基本块内所有后继点都不被引用，且在基本块出口之后又是不活跃的，那么x在点p的定值就是无用的

•
C.

活跃变量信息可用于为基本块分配寄存器

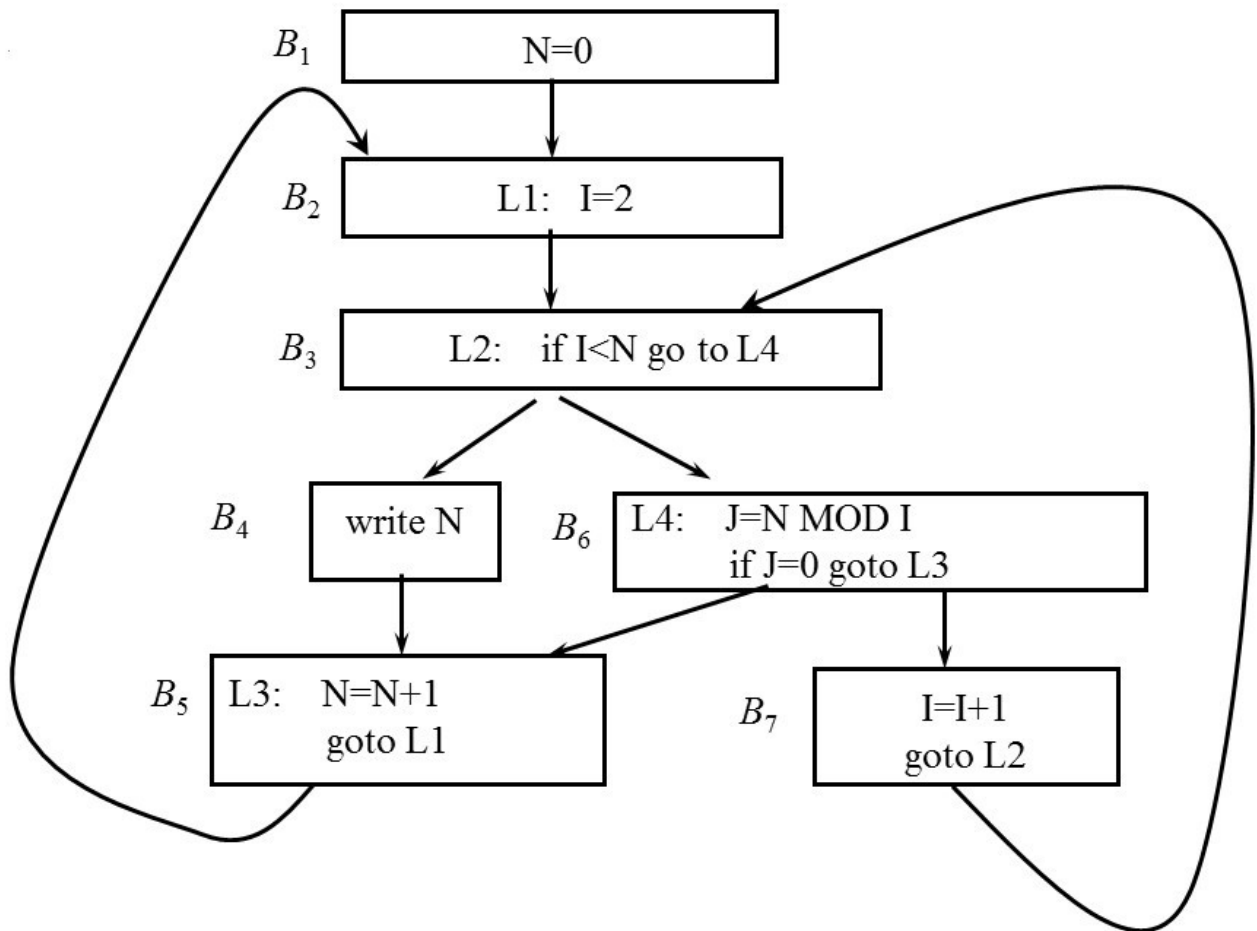
•
D.

如果所有寄存器都被占用，并且还需要申请一个寄存器，则应该考虑使用已经存放了死亡值的寄存器，因为这个值不需要保存到内存

A

4

下图中，基本块 B_1 的出口处的活跃变量集合是()。



- A. $\{N\}$
- B. $\{I\}$
- C. $\{N, I\}$
- D. 空集

A 5

以下说法不正确的是()。

•

A.

设变量 x 有一个定值 d ，该定值所有能够到达的引用 u 的集合称为 x 在 d 处的定值-引用链，简称 du 链

•

B.

如果在求解活跃变量数据流方程中的 $OUT[B]$ 时，将 $OUT[B]$ 表示成从基本块 B 的末尾处能够到达的引用的集合，那么，可以直接利用这些信息计算基本块 B 中每个变量 x 在其定值处的 du 链

•

C.

如果基本块 B 中 x 的定值 d 之后有 x 的第一个定值 d' ，则 d 和 d' 之间 x 的所有引用构成 d 的 du 链

•

D.

如果基本块 B 中 x 的定值 d 之后有 x 的新的定值，则 B 中 d 之后 x 的所有引用以及 $OUT[B]$ 中 x 的所有引用构成 d 的 du 链

D 6

以下说法不正确的是()。

•

A.

如果从流图的首节点到达程序点 p 的每条路径都对表达式 $x \text{ op } y$ 进行计算，那么表达式 $x \text{ op } y$ 在点 p 是可用的

- - B.**
在x的引用点u可以用y代替x的条件是：复制语句 $x = y$ 在引用点u处可用
- - C.**
在x的引用点u可以用y代替x的条件是：从流图的首节点到达u的每条路径都存在复制语句 $x = y$ ，并且从最后一条复制语句 $x = y$ 到点u之间没有再次对x或y定值
- - D.**
如果一个值在基本块结尾处是死的就不必在结尾处保存这个值

A

1

删除公共子表达式需要用到以下哪种数据流分析技术()。

- - A.**
到达定值分析
- - B.**
活跃变量分析
- - C.**
可用表达式分析
- - D.**
以上都不正确

C

2

复制传播需要用到以下哪种数据流分析技术()。

- - A.
到达定值分析
- - B.
活跃变量分析
- - C.
可用表达式分析
- - D.
以上都不正确

C

3

删除无用代码需要用到以下哪种数据流分析技术()。

- - A.
到达定值分析
- - B.
活跃变量分析

- - C.**
可用表达式分析
- - D.**
以上都不正确

B

4

识别循环不变计算需要用到以下哪种数据流分析技术()。

- - A.**
到达定值分析
- - B.**
活跃变量分析
- - C.**
可用表达式分析
- - D.**
以上都不正确

A

5

关于支配结点的二元关系，下列叙述中不正确的是()。

- - A.**
满足自反性

- **B.**
满足传递性
- **C.**
满足反对称性
- **D.**
满足对称性

D
6

下列优化方法()不是针对循优化进行的。

- **A.**
强度削弱
- **B.**
删除归纳变量
- **C.**
删除多余运算
- **D.**
代码外提

C

7

程序流图中，若 m 是 n 的必经节点，并且存在有向边 $n \rightarrow m$ ，则称 $n \rightarrow m$ 是流图中的()

- - A.
一个循环
- - B.
一条回边
- - C.
一条基本边
- - D.
一个基本通图

B

8

程序流图中，具有性质()的节点序列为一个自然循环。

- - A.
它们是非连通的且只有一个入口节点
- - B.
它们是强连通的但有多多个入口节点
- - C.
它们是非连通的但有多多个入口节点

- - D.**
它们是强连通的且只有一个入口节点

D

9

采用DAG图不可以实现的优化有()。

- - A.**
合并已知量
- - B.**
删除公共子表达式
- - C.**
删除无用赋值
- - D.**
删除归纳变量

D

10

()属于局部优化。

- - A.**
代码外提

- - B.**
删除多余运算
- - C.**
强度削弱
- - D.**
删除归纳变量

B

11

以下说法正确的是()。

- - A.**
进行代码优化时要着重考虑循环的代码优化，这对提高目标代码的效率将起更大作用
- - B.**
强度削弱是局部优化的一种方法
- - C.**
循环不变计算都可以移动到循环之前
- - D.**
循环中的无用赋值在循环优化时均可删除

A

12

以下说法不正确的是()。

- - A.
为了找出程序中的循环，就需要对程序中的控制流程进行分析
- - B.
在循环中强度削弱仅对乘法运算实行
- - C.
仅考虑一个基本块，不能确定一个赋值是否真是无用的
- - D.
在中间代码优化中循环上的优化主要有不变表达式外提和削减运算强度

B

1

在目标代码生成阶段，符号表用于()。

- - A.
目标代码生成
- - B.
语义检查
- - C.
语法检查

- **D.**
地址分配

D
2

经编译得到的目标程序是()。

- **A.**
机器语言程序或汇编语言程序
- **B.**
四元式序列
- **C.**
三元式序列
- **D.**
二元式序列

A
3

()不可能是目标代码。

- **A.**
中间代码

- - B.**
汇编代码
- - C.**
绝对指令代码
- - D.**
可重定位指令代码

A

4

以下说法不正确的是()。

- - A.**
源程序到目标程序的变换是等价变换，即两者结构不同，但语义是一致的
- - B.**
源程序和目标程序是等价关系
- - C.**
目标代码生成时，需要考虑目标计算机的指令系统
- - D.**
窥孔优化是在中间代码层次上进行的优化工作

D

5

所有编译程序都有目标代码生成阶段。

- A.
- B.

A

6

代码生成器的设计要着重考虑目标代码的质量问题。

- A.
- B.

A

7

目标代码生成时，无需考虑目标计算机的系统结构。

- A.
- B.