

中科大 2012 年计算机考研复试笔试题

感谢王道论坛 “tangbohu222” 和 “海风林影” 版主回忆并分享!

离散数学

1. (4 分) 求与下述公式逻辑等值的前束合取范式:

$$(\forall x)(P(x, y) \vee (\forall y)R(y, z)) \rightarrow (\forall z)Q(x, z)$$

2. (8 分) 对于命题公式 P 、 Q 、 R , 证明以下等值关系:

$$(1) (P \rightarrow Q) \wedge (R \rightarrow Q) \Leftrightarrow (P \vee R) \rightarrow Q$$

$$(2) P \rightarrow (Q \rightarrow R) \Leftrightarrow Q \rightarrow (P \rightarrow R)$$

3. (8 分)

(1) 定义集合 $S=\{1, 2, 3, 4, 5\}$ 上的二元关系 $R_1=\{\langle i, j \rangle \mid j = i \text{ 或 } j = i / 2\}$, $R_2=\{\langle i, j \rangle \mid i < j\}$, 求合成关系 R_2R_1 和 R_1R_2 .

(2) 写出 R_2 的关系图和关系矩阵, 并指出其是否自反、反自反、对称、反对称、传递?

4. (8 分) 设 $\langle G, * \rangle$ 是群, H 是 G 的非空子集。

证明: 若任给 $a, b \in H$, 都满足 $a*b^{-1} \in H$, 则 $\langle H, * \rangle$ 是 $\langle G, * \rangle$ 的子群。

5. (6 分) 证明: 图 G 连通, 每个结点的度数均为偶数, 则对 G 的任意结点 v, w $w(G-v) \leq 1/2d(v)$ 。其中 $d(v)$ 表示节点 v 的度数, $w(G)$ 表示图 G 的连通分图的个数。

6. (6 分) 设 G 为连通图, 任给节点 $v \in V(G)$, 是否有 $w(G-v) \leq 2$? 任给边 $e \in E(G)$, 是否有 $w(G-e) \leq 2$? 请分别给出证明或反例。

编译原理

1. (8 分) 画出接受 $(a|ab)^*a^*$ 所描述语言的最简 DFA 的状态转化图。

2. (6 分) 试证明如下文法既不是 LL(1) 文法, 也不是 LR(1) 文法。

$$A \rightarrow aAa \mid a$$

3. (8 分) 有文法如下:

$$S \rightarrow (L) \quad S \rightarrow a \quad L \rightarrow L, S \quad L \rightarrow S$$

假设我们对输入句子中每一个 a 进行编号。方法如下: 将配对的括号看作一个“作用域”, 当进入一个作用域时, 接着“当前序号”进行连续编号; 当退出一个作用域时, 则恢复在进入该作用域时的那个“当前序号”。对于句子 $(a, (a, (a, a), (a)))$, 编号为 1 2 3 4 3。试写一个语法制导定义, 实现上序编号。

4. (6 分) C 语言函数 f 的定义如下:

```
Int f(int x, int *py, int **ppz) {  
    **ppz += 1;    *py += 2; x*=3; return x + *py + **ppz;  
}
```

若 c 为整型变量, 执行下面代码后 d 和 c 的值分别为多少, 为什么?

```
a = &b; b = &c; c = 5; d = f(c, b, a);
```

5. (6分) 一个C语言程序如下:

```
Void fun(struct{int x; double r;} *val){ }  
Main(){  
    Struct{int x; double r;} *val;  
    Fun (val, 1);  
}
```

改程序在 X86/Linux 机器下报告错误信息如下:

(省略, 若干个警告, 一个错误)

请解释为什么会报告上述的警告和错误。

6. (6分) 如果变量 i 和 j 都是 int 类型, k 是 char 类型, 请写出表达式 &i、&k 和表达式 &i-&j 的类型表达式。为帮助你回答问题, 下面给出程序作为提示

```
#include <stdio.h>  
Main()  
{  
    Int i, j;  
    Char k;  
    Printf ("%d %d\n", &i - &j, &i - &k);  
}
```

它编译是输出如下错误:

Type.c: in function 'main':

Type.c:6:error:invalid operands to binary -

如果将 k 的类型改成 int, 则程序编译通过且运行时输出 1 2。

解答思路: <http://www.cskaoan.com/thread-88561-1-1.html>

中科大 2011 年计算机考研复试笔试试题

感谢王道论坛“水东流”道友回忆并分享!

离散数学

- 1, 证明等价关系 (得说明什么是等价关系)。
- 2, 给一个条件, 证明 S 是 G 的子群。有陷阱, 得先说明 S 不为空 (没说扣一分)。
- 3, 证明图度的总和是偶数。
- 4, 克鲁斯卡尔算法。
- 5, 给出两个谓词表达式, 说明是否是重言式。
- 6, 给出两个命题, 证明。

编译原理

- 1, 证明所给 D F A 是最简 D F A。
- 2,
- 3, C 类型检查。
- 4, 考察 int a[0][4] 的存储分配。(这个题是习题集上的原题。)

可见, 复试所考察的内容都是很简单的一些基础知识, 虽然简单, 但还是要慎重复习。

因为基础的东西如果复习仓促的话就会留意不到。离散的我一个都没有复习到，基本上的靠之前的学习了。所以建议大家早早的开始复习。

中科大 2010 年计算机考研复试笔试题

感谢王道论坛 “zyd320” 和 “lowtec” 道友回忆并分享!

离散数学

1、A 且 $C \Rightarrow B$ 且 C，是否 $A \Rightarrow B$?

A 或 $C \Rightarrow B$ 或 C，是否 $A \Rightarrow B$?

2、所有羊吃草，所有死羊不吃草，是否死羊都不是羊?

3、实数集 R 中，“+”，“-”，“ $|x-y|$ ”，“min”是否可交换，可结合，有么元，有逆元?

4、证明 $N = \{0, 1, 2, 3, 4\}$ ，N 对 $+\text{mod}5$ 为群， $N - \{0\}$ 对 $\times \text{mod}5$ 为群

5、R 和 S 都是 A 上的自反、对称、传递关系，求 R 交 S 的自反、对称、传递关系的闭包

6、求一个图的邻接矩阵，及图中 a 点到 b 点长度为 4 的路径数目

7、一个连通图中所有定点的度数为偶数，证明删除一个点后的连通分量数目小于等于 $1/2 \deg(v)$ ，v 为顶点

另一回忆版本:

离散 8 道大题，1 小时。难度不是很大，但是时间很紧。

第一题：是一个等价命题的判断，不难。

第二题：所有羊都是吃草的，所有死羊都是不吃草的，所以死羊不是羊。

能得出这个结论吗，证明之!

第三题:

‘+’，‘-’， $|x-y|$ ，Min 在实数集内是不是可交换的，可结合的，是否有逆元，是否有么元

第四题：证明 $N = \{0, 1, 2, 3, 4\}$ ，N 对 $+\text{mod}5$ 是群， $N - \{0\}$ 对 $\times \text{mod}5$ 是群

第六题：R 和 S 都是 A 上的自反，对称，传递关系，求 R 交 S 的自反，对称，传递闭包是什么

第七题：给一个 5 个节点的图，画出方向矩阵，然后问从 A 到 B 长度为 4 的路径的条数

第八题：一个连通图所有顶点的度为偶数，证明删除一个顶点后，连通分量数目小于等于 $\deg(v)$ ，v 是顶点

编译原理

也是 1 小时

第一题：写出生成由 a,b 组成，且不含 2 个连续的 a 的字符串的正规式，并画出最简的 DFA

第二题：简述 SLR，LR，LALR 所描述的语言集合的关系

第三题：给了 2 个函数 f(),g(), f()中调用了 g(), 画出程序运行时的 活动记录站的结构，以及控制链，假设返回值用寄存器存储

第四题：struct stu{ int n; long m;}val;

```
main()
{ struct stu{int n;long m;}val;
  f(val);
}
```

为什么 f(val)会提示类型不匹配

第五题: 题目记不清了, 就是 3 个指针指来指去的, 然后问你最后结果多少

中科大 2009 年计算机考研复试笔试题

感谢王道论坛 “chanllenger” 道友回忆并分享!

离散数学

- 1、约束变元和自由变元问题
- 2、用推理规则证明
- 3、搜集、包含关系、关系的性质(自反、对称、传递)
- 4、群、等价关系
- 5、图,和习题 8.1 第十题类似(西电第二版)

编译原理

- 1、求产生式 $(a|b)^*(aa|bb)$ 的 DFA
- 2、3、4、5 题均为历年真题

王道论坛