

Моля, освен условието публикувайте и решение (поне частично), ако имате възможност. Задал съм някои примерни категории. Ако смятате, че са некоректно дефинирани, моля коригирайте ги.

Категория “УП, СДА, ООП”

1. УСЛОВИЕ:

от Мирю П: КН, Юли 2009, задача 6:

<http://www.fmi.uni-sofia.bg/exams/drzhaven-izpit/temi-drzhaven-izpit/kompyutni-nauki/DI-KN-july-09.pdf>

РЕШЕНИЕ:

```
#include<stdio.h>
#include<string.h>
#include<iostream>
class Function
{
public:
    virtual int Calculate(int x);
};
class LinearFunction : public Function
{
private:
    int a,b;
public:
    LinearFunction(int a, int b)
    {
        this->a = a;
        this->b = b;
    }
    int Calculate(int x)
    {
        return a*x + b;
    }
};
class CharacteristicFunction : public Function
{
private:
    int setCount;
    int *set;
public:
    CharacteristicFunction(int aSet[], int aCount)
    {
        this->set = new int[aCount];
```

```

    for(int i=0; i<aCount; i++)
    {
        *(this->set + i) = aSet[i];
    }
    this->setCount = aCount;
}
int Calculate(int x)
{
    for(int i=0; i<this->setCount; i++)
    {
        int v = *(this->set + i);
        if(v == x)
        {
            return 1;
        }
    }
    return 0;
}
~CharacteristicFunction()
{
    delete [] this->set;
}
};

class Superposition : public Function
{
private:
    Function *inner, *outer;
public:
    Superposition(Function *inner, Function *outer)
    {
        this->inner = inner;
        this->outer = outer;
    }
    int Calculate(int x)
    {
        return outer->Calculate(inner->Calculate(x));
    }
    ~Superposition()
    {
        delete this->inner;
        delete this->outer;
    }
};

```

2. УСЛОВИЕ:

От Мирон П: КН, Юли 2009, Задача 5

<http://www.fmi.uni-sofia.bg/exams/drzhaven-izpit/temi-drzhaven-izpit/kompyutni-nauki/DI-KN-july-09.pdf>

РЕШЕНИЕ (Тествано е с компилатор :)):

1:

A::A()

A::g()

B::B()

2:

A::f()

B::g()

3:

B::g()

4:

A::A()

A::g()

A::=

5:

A::A()

A::g()

B::B()

A::=

6:

B::~~

A::~~

7:

B::~~

A::~~

A::~~

3. Задачи и решения на падали се задачи на КН и Инф –

<http://tinyurl.com/2v6z6r2> (файла за Програмиране)

Ако намерите грешки, пишете тук:

На задача 5 / КН / 15.07.2009: В 5: накрая има допълнително "A::~=".

4. УСЛОВИЕ (Информатика - 2003.03, Задача 1)

Каква ще бъде стойността на x след изпълнението на следния оператор:

```
int x = - 3 * 4 % - 6 / 5;
```

РЕШЕНИЕ

Миро. Д: x = 0

5. УСЛОВИЕ (Информатика - 2003.03, Задача 2)

Каква ще бъде стойността на x след изпълнението на следния фрагмент от програма:

```
int x = 2;
```

```
x *= 3+2;
```

РЕШЕНИЕ

Миро. Д: $x = 10$

6. УСЛОВИЕ (Информатика - 2003.03, Задача 3)

Каква ще бъде стойността на x и z след изпълнението на следния фрагмент от програма:

```
int x, y, z;  
x=y=1;  
z = x++ - 1;           // М.Д.:  $x = 2, z = 0$   
z+= - x++ + ++y;       // М.Д.:  $x = 3, z = 0$ 
```

РЕШЕНИЕ

Миро. Д: $x = 3, z = 0$

7. УСЛОВИЕ (Информатика - 2003.03, Задача 4)

Кой от операторите за цикли, означени с буквите от А до D, е еквивалентен на следния:

```
while ( U ) { if ( V ) continue;W; }
```

A) while (U) { W; if (V) break;}

B) while (U) { W; if (!V) break;}

C) while (U) if (!V) W;

D) while (U) { if (V) continue;
else break;
W; }

РЕШЕНИЕ

Миро. Д: C)

8. УСЛОВИЕ (Информатика - 2003.03, Задача 5)

Колко въпроса са необходими за оптималната стратегия за познаване на едно число между 1 и 1000, ако се разчита на верните отговори на въпроси, на които се отговаря само с "да/не"?

A) 1000 B) 999 C) 500 D) 32 E) 10

РЕШЕНИЕ

Миро. Д.: За търсене в наредено множество от елементи най-бързият алгоритъм, за който аз се сещам, е *двоичното търсене**. Максималният брой опити за познаване на набелязания елемент е $\log_2 T$, където T е броят на елементите в множеството.

$\log_2 1000 = 9,96 \sim 10$ (С най-много 10 опита ще се намери търсеното число)

* 7-ма лекция на Григоров за СДА -> стр.25 (задача за линейката - подобна на тази)

9. УСЛОВИЕ

РЕШЕНИЕ

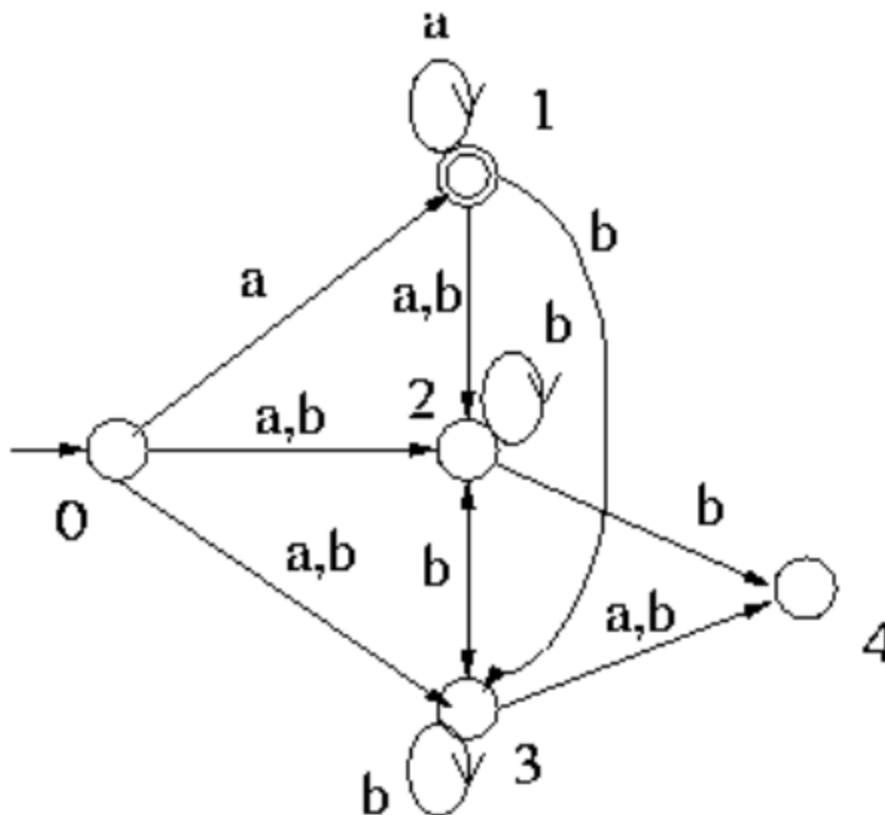
Категория "Дискретна математика"

1. УСЛОВИЕ:

<http://www.fmi.uni-sofia.bg/exams/drzhaven-izpit/temi-drzhaven-izpit/DI-kn>

от Васко, КН, юли 2009, задача 1 (15 точки)

Нека $A = (Q, \Sigma, \delta, s, F)$ е крайният недетерминиран автомат, представен чрез следния граф:



А) Да се напишат стойностите на символите Q , Σ , s , F и δ от дефиницията на A .

Б) Да се дефинира краен детерминиран автомат B , еквивалентен на автомата A . Обосновете отговора си.

РЕШЕНИЕ:

А) $Q = \{0, 1, 2, 3, 4\}$

$s = 0$

$F = \{1\}$

$\Sigma = \{a, b\}$

$\delta: Q \times \Sigma \rightarrow 2_Q$

$\delta(0, a) = \{1, 2, 3\}$

$\delta(0, b) = \{2, 3\}$

$\delta(1, a) = \{1, 2\}$

$\delta(1, b) = \{2, 3\}$

$\delta(2, a) = \langle \text{empty_set} \rangle$

$\delta(2, b) = \{2, 4\}$

$\delta(3, a) = \{4\}$

$\delta(3,b) = \{2,3,4\}$
 $\delta(4,a) = \delta(4,b) = \langle \text{empty_set} \rangle$

Б) Извинявам се за тъпанарския въпрос, но езикът на този автомат не е ли a^+ ? Тогава в детерминирания автомат съвсем спокойно могат да отпаднат състояния 2, 3 и 4, да не говорим че азбуката $\Sigma = \{a\}$ също би свършила работа и да си спестим цялостното прилагане на алгоритъма за детерминизация?

Какви разсъждения ме доведоха дотук - състояние 4 е такова, че $\delta^*(4,w) = \langle \text{empty_set} \rangle$ за всяко w . Следователно можем спокойно да го махнем, заедно с връзките към него и езикът на автомата си остава същият (важно е да се отбележи, че 4 не е крайно!). Оттам забелязваме, че $\delta^*(2,w) = \delta^*(3,w) = \{2,3\}$; т.е веднъж попаднем ли в състояние 2 или 3

ние не можем по никакъв начин да достигнем до крайно състояние (1). Следователно можем

да махнем 2 и 3 и връзките към тях като това няма да промени езикът, разпознаван от автомата.

Като оставим настрана това "решение":

Строим детерминиран автомат $A'(Q', \Sigma, s, F' \text{ и } \delta')$

$Q' \subseteq 2_q$; $Q' = \{q' : q' \in Q\}$

$F' \subseteq Q'$. $f^* \in F'$ е крайно, ако някое от $f^* \in F^*$ е крайно ($f^* \in F$)

$\delta'(q', w \in \Sigma) = \bigcup (q : \delta(q_0, w) = q, q_0 \in Q)$

Нека началните състояния са "необходени". (В случая такова е само 0)

Повтаряме следните стъпки до изчерпване на необходимите състояния

1) Избираме произволно необходимо състояние q . Добавяме го към A' .

2) Намираме $\delta(q,a)$ и $\delta(q,b)$. Означаваме ги като необходени

3) Означаваме q като обходено. Преминаваме на стъпка 1)

Спазваме следните правила:

- ако δ' не е дефинирана за някое състояние и буква, дефинираме я като $\delta'(q,w) = \text{epsilon}$ (празната дума)

- ако дадено състояние $q' \in A'$ съдържа крайно състояние от A - $q \in F$, то q' е крайно за A' ($q' \in F'$)

1) Тръгваме от началното състояние 0. Добавяме го към A' .

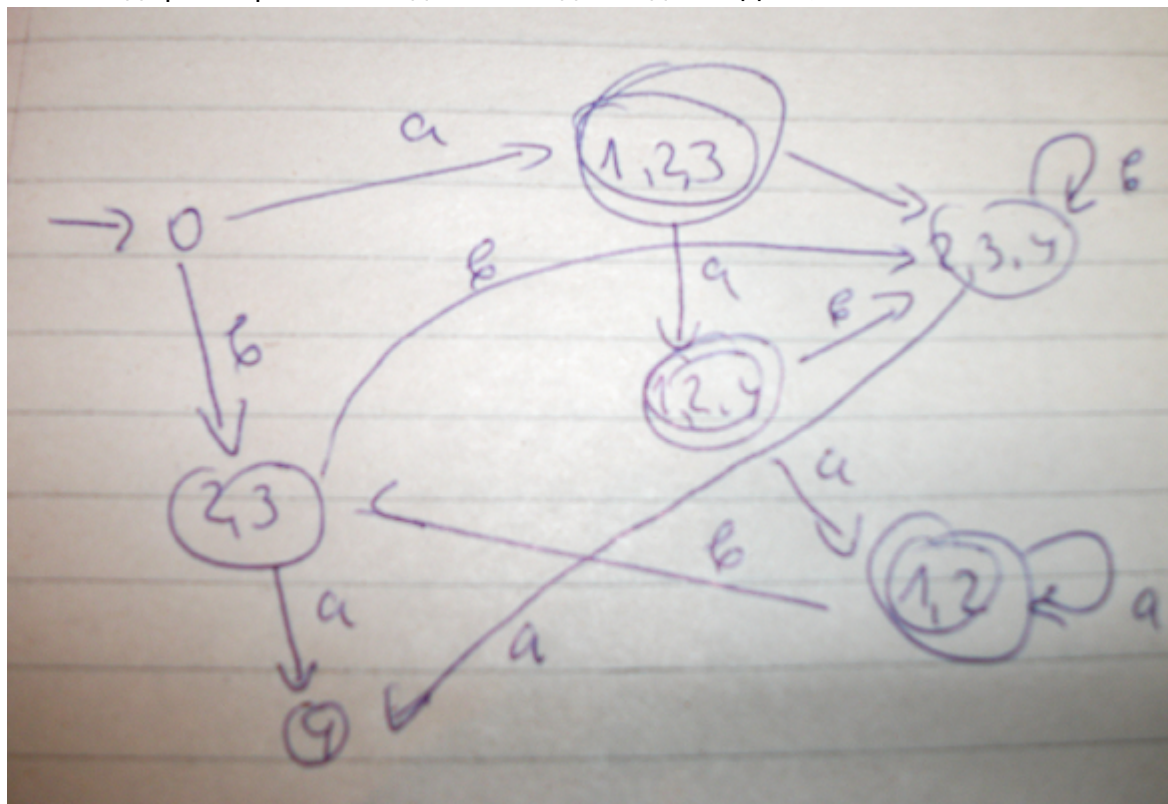
2) $\delta'(0,a) = \{1,2,3\} \Rightarrow$ Правим ново състояние $\{1,2,3\}$, означаваме го като крайно за A'

$\delta'(0,b) = \{2,3\} \Rightarrow$ Правим ново състояние $\{2,3\}$

3) Намираме $\delta'(\{1,2,3\},a) = \{1,2,4\} \Rightarrow$ Правим ново състояние $\{1,2,4\}$, означаваме го като крайно за A'

$\delta'(\{1,2,3\},b) = \{2,3,4\} \Rightarrow$ Правим ново състояние $\{2,3,4\}$

Така след краен брой стъпки достигаме до следния КДА В, еквивалентен на А:



Милена: Забележка: За да се минимизира един детерминиран автомат, той може и да е/ хубаво е/ по-нагледно става(при описването на таблиците с множествата) да е тотален. (От всяко състояние да се преминава към друго състояние с всяка буква от езика на автомата.)(**edited**)

(Не видях някъде да го пише изрично.)

И когато се търси автомат за допълнението на езика на даден автомат M , M трябва да е детерминиран и тотален.

2. УСЛОВИЕ (Държавен изпит на КН от 2008.07 - Задача 2)

05.07.2008г. ДИ ОКС „Бакалавър” по Компютърни науки, СУ-ФМИ фак. № стр. 2/11

Задача 2. (10 т.) Да се построи краен детерминиран автомат, еквивалентен на дадения краен недетерминиран автомат

$$A = \langle \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{a, b, c\}, q_0, \delta, \{q_0\} \rangle$$

с функция на преходите δ , дефинирана от следната таблица:

q	a	b	c
q_0	$\{q_0, q_1\}$	$\emptyset$	$\{q_3\}$
q_1	$\emptyset$	$\{q_2\}$	$\emptyset$
q_2	$\{q_0\}$	$\emptyset$	$\{q_2\}$
q_3	$\emptyset$	$\{q_5\}$	$\{q_4\}$
q_4	$\emptyset$	$\{q_1, q_2\}$	$\{q_1\}$
q_5	$\emptyset$	$\{q_1, q_2\}$	$\{q_1\}$
q_6	$\emptyset$	$\{q_1, q_2\}$	$\{q_1\}$

РЕШЕНИЕ:

	a		b	c
q ₀	q ₀	q ₁	q ₅	q ₃
q ₃		-		q ₄
q ₅				q ₇
q ₇			q ₅ , q ₈	q ₂
q ₀ q ₁	q ₀ q ₁		q ₂	q ₃
q ₂	q ₀		-	q ₂
q₄				
q ₅ , q ₆			q ₀	q ₄

3. УСЛОВИЕ:

Задача 3. (4 т.) Даден е крайният детерминиран автомат А:

$$A = \langle \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}, \{x, y, z\}, q_0, \delta, \{q_0, q_3, q_4, q_6, q_7\} \rangle$$

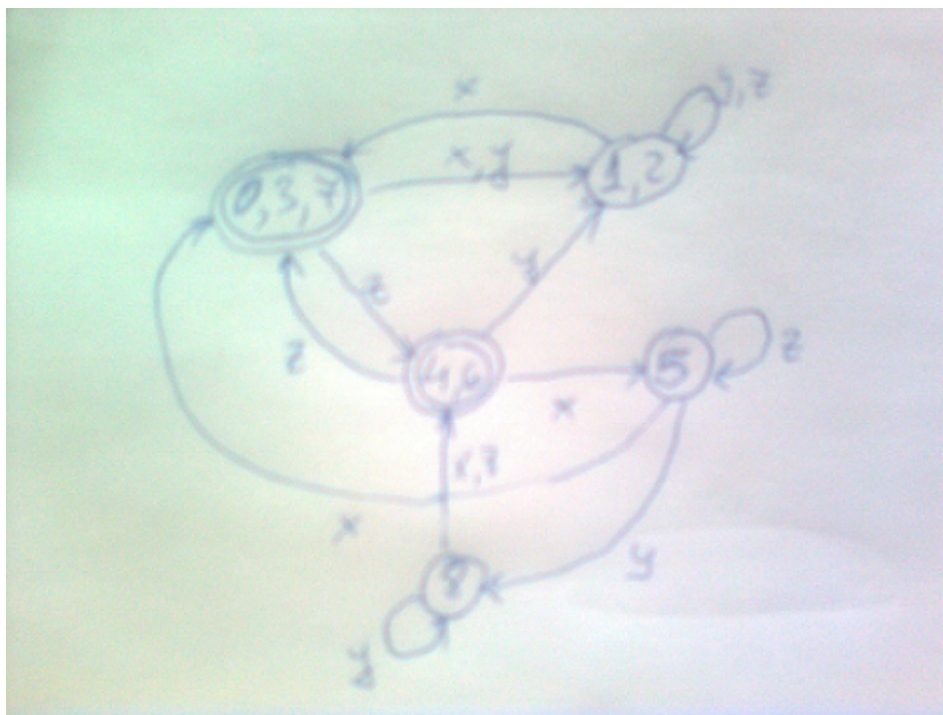
с функция на преходите δ , зададена чрез таблицата:

	x	y	z
q ₀	q ₁	q ₁	q ₄
q ₁	q ₀	q ₂	q ₂
q ₂	q ₃	q ₁	q ₁
q ₃	q ₁	q ₁	q ₄
q ₄	q ₅	q ₂	q ₃
q ₅	q ₇	q ₈	q ₅
q ₆	q ₅	q ₁	q ₇
q ₇	q ₂	q ₂	q ₆
q ₈	q ₆	q ₈	q ₆

Да се построи минимален детерминиран краен автомат, еквивалентен на дадения.

РЕШЕНИЕ

(от Васко)



Алгоритъм за минимизация на КДА: <http://fmi.wikidot.com/eai-exam#toc3>

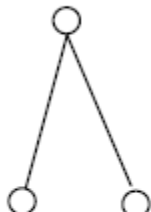
4.УСЛОВИЕ (Държавен изпит на КН от 2008.07 - Задача 1)

Задача 1. (10 т.) Дадена е рекурсивно дефинираната редица от коренови дървета.

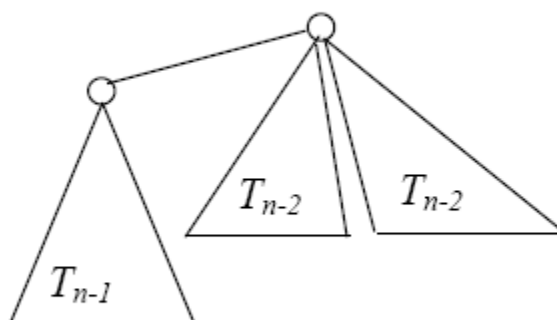
T_0 :



T_1 :



$T_n, n > 1$:



Намерете формула за броя на вътрешните върхове (нелиста) на дървото T_n .

РЕШЕНИЕ - Миро Д.:

Нека с $|T_n|$ означаваме броя на вътрешните върхове в дървото T_n . И нека приемем, че коренът също се брой за вътрешен връх. Тогава от така зададената рекурентна връзка, може да изведем следната формула:

$$(a) \quad |T_n| = |T_{n-1}| + 2 |T_{n-2}| - 1$$

$|T_0| = 1; \quad |T_1| = 1; \quad |T_2| = 2; \quad |T_3| = 3; \quad |T_4| = 6; \quad |T_5| = 11; \quad |T_6| = 22; \quad |T_7| = 43;$
 $|T_8| = 46; \dots$

(б) Да се представи $|T_n|$ единствено чрез променлива n ;

// **Наско:** До колкото пробвах, това трябва да е. Проверете все пак за грешки.

ЗАБЕЛЕЖКА: <http://dox.bg/files/dw?a=b7d8074c4f> - материал относно рекурентните отношения

$$T_0 = 1$$

$$T_1 = 1$$

$$T_n = T_{n-1} + 2T_{n-2} - 1$$

$$\text{Полагаме } P_n = T_n + C$$

$$P_n - C = P_{n-1} - C + P_{n-2} - 1 - 2C$$

$$P_n = P_{n-1} + 2P_{n-2} - 1 - 2C$$

$$\text{Избираме } C = -\frac{1}{2}$$

$$P_n = P_{n-1} + 2P_{n-2}$$

$$P_n = T_n - \frac{1}{2}, P_0 = \frac{1}{2}, P_1 = \frac{1}{2}$$

$$P_n = ax_1^n + bx_2^n$$

$$\text{Характеристично за } P : x^2 = x + 2$$

$$x^2 - x - 2 = 0$$

$$D = 1 + 4 \cdot 2 = 9$$

$$x_1 = 2, x_2 = -1$$

$$\begin{cases} a + b = \frac{1}{2} \\ 2a - b = \frac{1}{2} \end{cases}$$

$$2a - b = \frac{1}{2}$$

$$a = \frac{1}{3}, b = \frac{1}{6}$$

$$P_n = ax_1^n + bx_2^n = \frac{2^{n+1} + (-1)^n}{6}$$

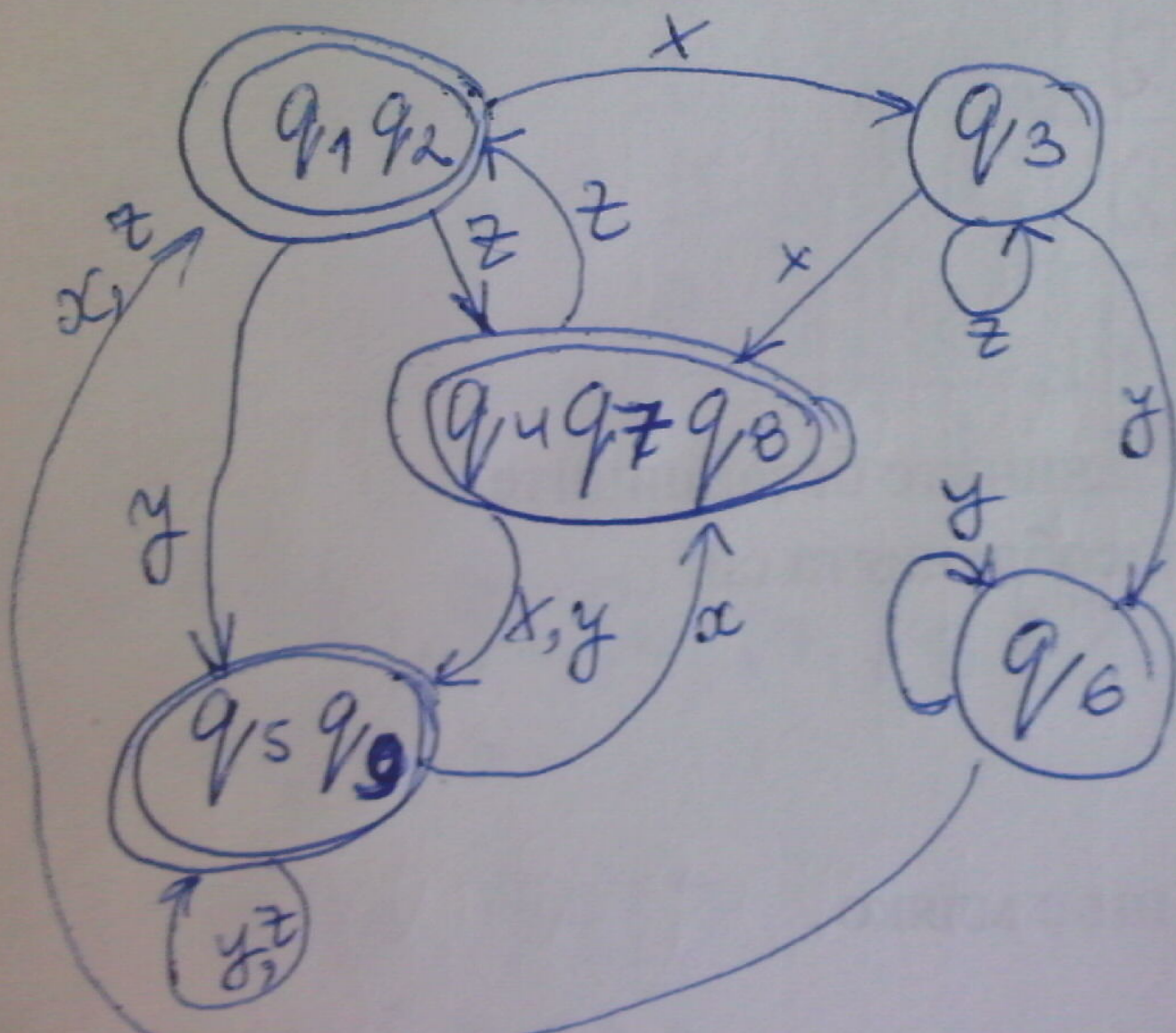
$$T_n = \frac{2^{n+1} + (-1)^n + 3}{6}$$

5.УСЛОВИЕ (КН, 2008.09)

Задача 1. (10 т) Да се построи минимален краен детерминиран автомат, еквивалентен на дадения краен детерминиран автомат $A = \langle \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}, \{x, y, z\}, q_1, \delta, \{q_1, q_2, q_4, q_7, q_8\} \rangle$ с функция на преходите δ :

q	x	y	z
q ₁	q ₃	q ₉	q ₇
q ₂	q ₃	q ₅	q ₄
q ₃	q ₄	q ₆	q ₃
q ₄	q ₉	q ₉	q ₂
q ₅	q ₈	q ₉	q ₉
q ₆	q ₂	q ₆	q ₂
q ₇	q ₅	q ₅	q ₁
q ₈	q ₅	q ₅	q ₁
q ₉	q ₇	q ₅	q ₅

РЕШЕНИЕ: (Katia)



6.УСЛОВИЕ (КН, 2009.09)

Задача 1. (15 точки)

А) Да се построи краен недетерминиран автомат, който разпознава езика, определен от регулярния израз R:

$$R = ((00) \cup (11))^*$$

Пример: Разпознава: ϵ , 00111100

Не разпознава: 0100, 0011100

Б) Да се построи минимален детерминиран автомат, еквивалентен на дадения недетерминиран автомат $K = (Q, \Sigma, \delta, s, F)$, където:

$$Q = \{s, p, q\}$$

$\Sigma = \{ a, b \}$

$F = \{ p \}$

$\delta(s, a) = \{ p \}, \delta(p, a) = \{ s, p, q \}, \delta(s, b) = \{ p, q \}, \delta(q, a) = \{ s \}$

РЕШЕНИЕ:

(Вики4)

А) Първо създаваме автомати, които разпознават $R1(00)$ и $R2(11)$.

$M(R1) = \{ Q1, \Sigma, s1, F1, \delta1 \}$, където:

$Q1 = \{ q0', q1', q2' \}$

$\Sigma = \{ 0, 1 \}$

$S = \{ q0' \}$

$F = \{ q2' \}$

δ

Q	0	1
q0'	q1'	$\emptyset$
q1'	q2'	$\emptyset$
q2'	$\emptyset$	$\emptyset$

$M(R2) = \{ Q2, \Sigma, s2, F2, \delta2 \}$, където:

$Q1 = \{ q0'', q1'', q2'' \}$

$\Sigma = \{ 0, 1 \}$

$S = \{ q0'' \}$

$F = \{ q2'' \}$

δ

Q	0	1
q0''	$\emptyset$	q1''
q1''	$\emptyset$	q2''
q2''	$\emptyset$	$\emptyset$

След това, за да създадем обединението създаваме ново състояние $q0$, което правим начално и правим преход от него към началните на $R1$ и $R2$ с празната дума.

И накрая, за да получим * правим преход от крайните състояния на $R1$ и $R2$ с празната дума до $q0$ и правим $q0$ крайно за да разпознава празната дума.

$M(R) = \{ Q, \Sigma, s, F, \delta \}$, където:

$Q = \{ q0, q0', q1', q2', q0'', q1'', q2'' \}$

$\Sigma = \{ 0, 1 \}$

$$S = \{q_0\}$$

$$F = \{q_0, q_2', q_2''\}$$

δ

Q	0	1	e
q_0	$\emptyset$	$\emptyset$	q_0', q_0''
q_0'	q_1'	$\emptyset$	q_0'
q_1'	q_2'	$\emptyset$	q_1'
q_2'	$\emptyset$	$\emptyset$	q_0
q_0''	$\emptyset$	q_1''	q_1''
q_1''	$\emptyset$	q_2''	q_2''
q_2''	$\emptyset$	$\emptyset$	q_0

Б)Имаме $K = \{Q, \Sigma, s, F, \delta\}$, където:

$$Q = \{s, p, q\}$$

$$\Sigma = \{a, b\}$$

$$S = \{s\}$$

$$F = \{p\}$$

δ

Q	a	b
s	p	p, q
p	s, p, q	$\emptyset$
q	s	$\emptyset$

Детерминираме и нека $K' = \{Q', \Sigma, s', F', \delta'\}$ е резултата, където:

$$Q' = \{q_1, q_2, q_3, q_4\}, \text{ където } q_1 = \{s\}, q_2 = \{p\}, q_3 = \{p, q\}, q_4 = \{s, p, q\}$$

$$\Sigma = \{a, b\}$$

$$S' = \{q_1\}$$

$$F' = \{q_2, q_3, q_4\}$$

δ'

Q	a	b

q1	q2	q3
q2	q4	∅
q3	q4	∅
q4	q4	q3

Минимизация:

В началото имаме два класа – крайни и некрайни състояния, както следва:

$A = \{q2, q3, q4\}$

$B = \{q1\} \Rightarrow B$ е клас на еквивалентност

Да разгледаме A :

A	A	b
q2	A	∅
q3	A	∅
q4	A	A

От таблицата се вижда, че A не е стабилен клас $\Rightarrow$ съдържа повече от един клас на еквивалентност.

$A = A1 \cup A2$, където $A1 = \{q2, q3\}$, а $A2 = \{q4\} \Rightarrow A2$ е стабилен $\Rightarrow$ е клас на еквивалентност.

Да разгледаме $A1$:

$A1$	A	b
q2	$A2$	∅
q3	$A2$	∅

Виждаме, че от двете състояния с δ' отиваме в едни и същи класове на еквивалентност $\Rightarrow$ че $A1$ е също клас на еквивалентност. Тъй като класът B е от един елемент няма как да бъде повлиян от разбиването на A , окончателно получаваме минималният автомат:

$K'' = \{Q'', \Sigma, s'', F'', \delta''\}$ където:

$Q'' = \{t1, t2, t3\}$, където $t1=B=\{q1\}$, $t2 =A2=\{q2, q3\}$, $t3=A1=\{q4\}$

$\Sigma = \{a, b\}$

$S'' = \{t1\}$

$F'' = \{t2, t3\}$

δ''

Q''	a	b
t1	t2	t2
t2	t3	$\emptyset$
t3	t3	t2

(///Картинки малко по-късно)

7.УСЛОВИЕ (Информатика - 2003.03)

Дадена е автоматната граматика $\Gamma = \langle N, T, S, P \rangle$, където:

$N = \{ S, A, B, C, D \}$

$T = \{ 0, 1 \}$

$S \in N$

$P = \{ S \rightarrow 0S \mid 1S \mid 0A \mid 1B, A \rightarrow 0A \mid 1A \mid 1C, B \rightarrow 0B \mid 1B \mid 0D, C \rightarrow 0, D \rightarrow 1 \}$

Постройте краен детерминиран автомат, който разпознава езика, породен от граматиката Γ .

РЕШЕНИЕ:

(Наско)

Недетерминиран:

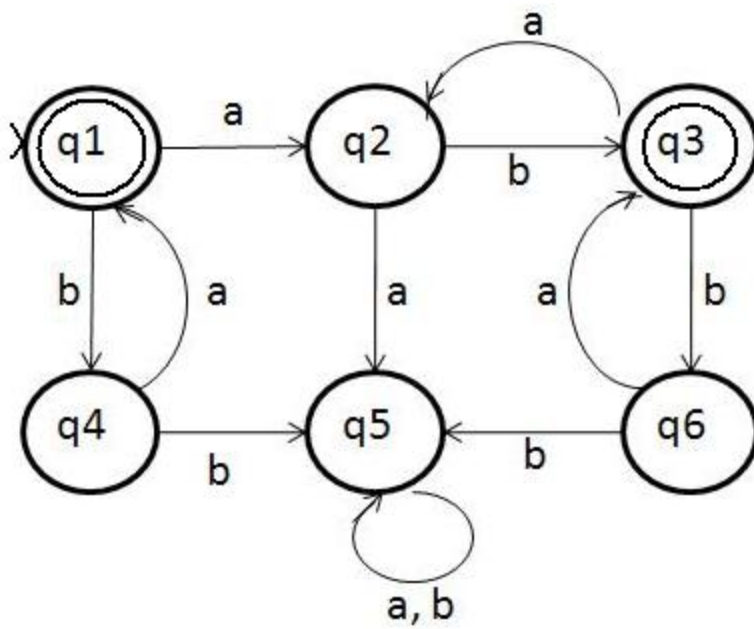
q	0	1
S	S, A	S, B
A	A	A, C
B	B, D	B
C	0	-
D	-	1
0	-	-
1	-	-

Начално: S,

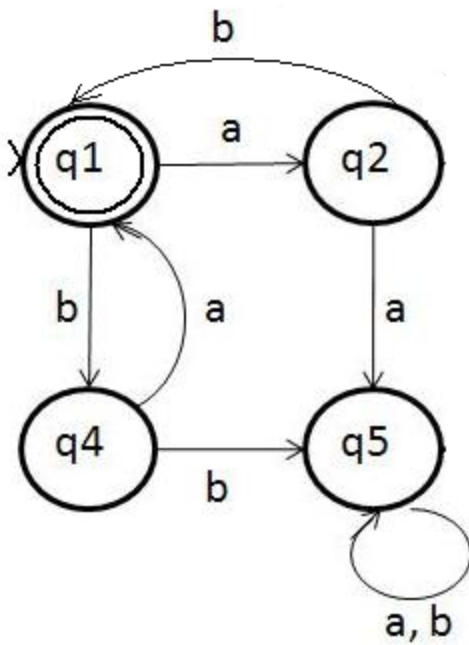
Крайни: 0, 1

8. УСЛОВИЕ

Минимизация на автомат :

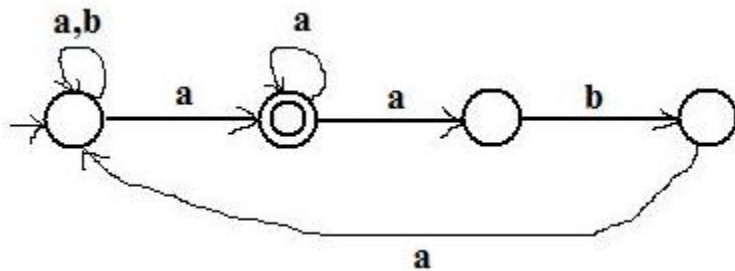


РЕШЕНИЕ:

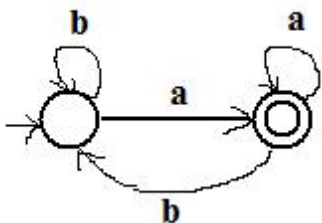


9. УСЛОВИЕ

Да се минимизира автомата:

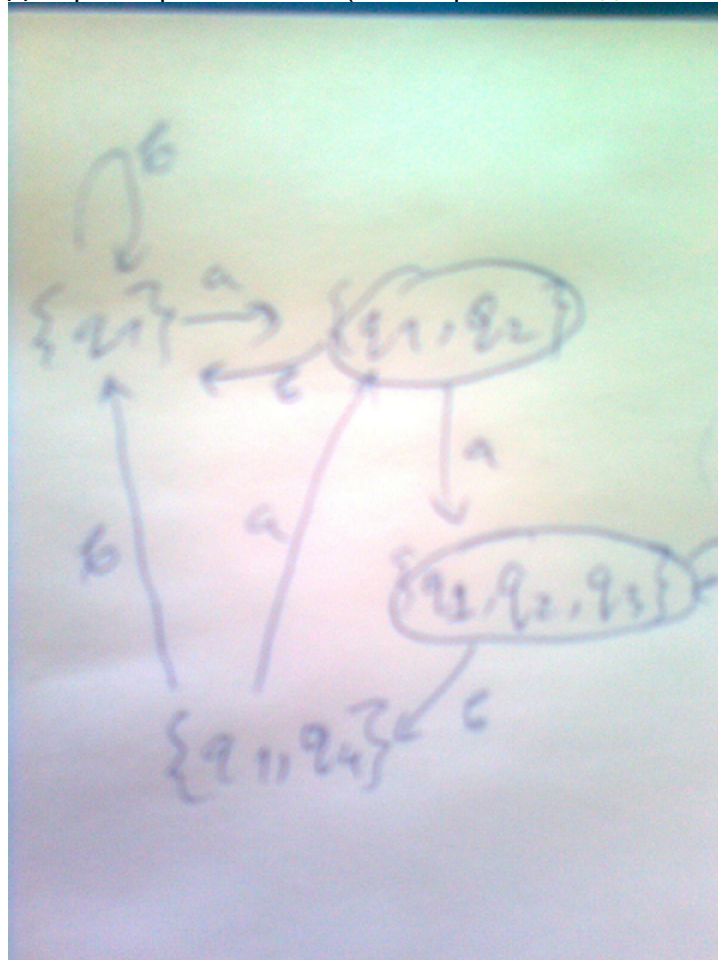


РЕШЕНИЕ:



(Vasko):

Детерминиран автомат: (това изрязаното вдясно е 'a')



Ако вземем 2та класа на еквивалентност $P1 = \{\{q1\}, \{q1, q4\}\}$ и $P2 = \{\{q1, q2\}, \{q1, q2, q3\}\}$ (разбиване на крайни и некрайни състояния) още на първата проверка се вижда, че по-фино разбиване не може да се получи => Правим минимален автомат с 2 състояния, $P1$ и $P2$ и гледаме с a и b къде отиват едно спрямо друго => gorenapisaniat avtomat. $P2$ е крайно, $P1$ - начално

Rosen: Аз нещо бях изключил, че 1 се върти в себе си при четене на Б и от там ми е дошла грешката.

Bob: pu d***a i az ne sum vidql che 1,2,3 s "b" vurti osven v 4 i v 1.. tei si e

Vasko: То по-добре тук да не видиш ама на изпита да видиш :)

Категория "Бази данни"

1. УСЛОВИЕ:

от Мирю П: КН, Юли 2009, задача 10:

<http://www.fmi.uni-sofia.bg/exams/drzhaven-izpit/temi-drzhaven-izpit/kompyutni-nauki/DI-KN-july-09.pdf>

РЕШЕНИЕ (примерно, тествано е с БД, може и по-кратко сигурно):

А)

```
SELECT DeptName, COUNT(*) as ProjCount, SUM(Salary)
FROM Project
JOIN Employee
    ON Project.RespEmp = Employee.EmpNo
JOIN Dept
    ON Project.DeptNo = Dept.DeptNo
GROUP BY DeptName
```

Б)

```
SELECT Wow.DeptName FROM
    (SELECT DeptName, COUNT(*) as ProjCount, SUM(Salary) as TotSalary
    FROM Project
    JOIN Employee
        ON Project.RespEmp = Employee.EmpNo
    JOIN Dept
        ON Project.DeptNo = Dept.DeptNo
    GROUP BY DeptName ) AS Wow
WHERE wow.TotSalary IN
    (SELECT A.TotSalary FROM
        (SELECT TOP 1 DeptName, COUNT(*) as ProjCount, SUM(Salary) as
        TotSalary
        FROM Project
        JOIN Employee
            ON Project.RespEmp = Employee.EmpNo
        JOIN Dept
            ON Project.DeptNo = Dept.DeptNo
        GROUP BY DeptName
        ORDER BY TotSalary DESC) AS A)
```

ВТОРО РЕШЕНИЕ (Мирю Д.)

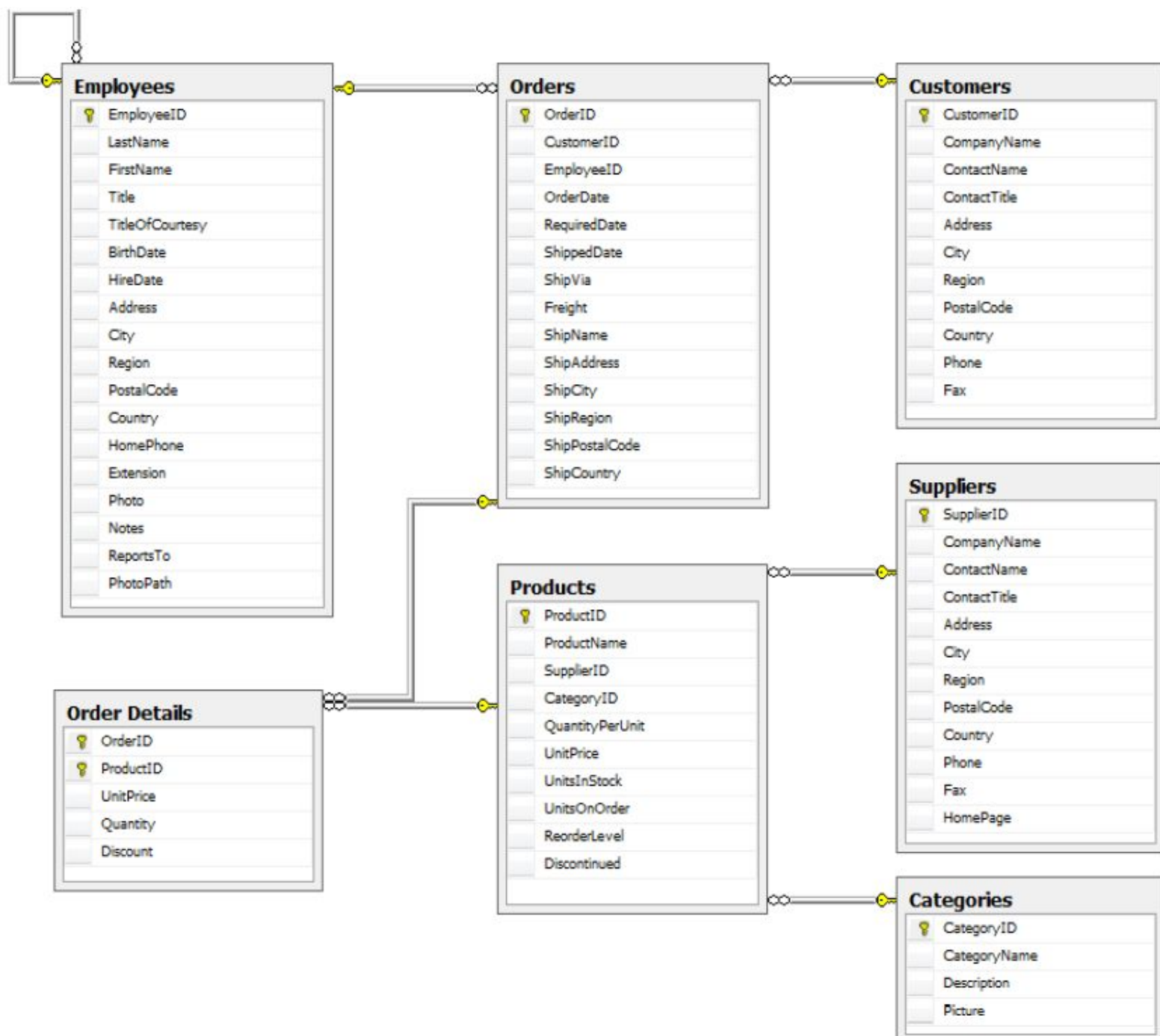
//Трето - всички отдели, чиято сумарна заплата на менажери е = максималната

```

SELECT d.DeptName
FROM Departments d
      JOIN
      // Отново се намират всички менажери на проекти
      (SELECT DISTINCT e.EmpNo      //distinct решава проблема на Наско
      FROM Employees e JOIN Projects p
      ON e.EmpNo = p.RespEmp) managers
      ON d.DeptNo = managers.DeptNo
// Връзваме с Employees, само за да вземем заплатата
      JOIN Employees e
      ON e.EmpNo = managers.EmpNo
GROUP BY e.WorkDept
// Отново се смята сумарната заплата на менижърите
HAVING
      SUM(e.Salary) =
      // Второ - макс. сумарна заплата на менажери, групирани по отдел
      (SELECT TOP 1 SUM(e.Salary) as TotSalary, e.WorkDept
      FROM
      // Първо - се намират всички менажери на проекти
      (SELECT DISTINCT e.EmpNo      //distinct решава проблема
      на Наско
      FROM Employees e JOIN Projects p
      ON e.EmpNo = p.RespEmp) managers
      JOIN Employees e
      ON e.EmpNo = managers.EmpNo
      GROUP BY e.WorkDept
      ORDER BY TotSalary)

```

2. УСЛОВИЕ



За всеки служител да се изведе неговото име, колко бройки е продал от продукти в една категория и името на съответната категория.

РЕШЕНИЕ

```

select e.FirstName+' '+e.LastName as [EMP_Name], SUM(q1.[Total Units]) as [Total Sum],
q1.CAT_Name
from Employees e

```

```

join (select o.EmployeeID, od.Quantity*p.UnitsOnOrder as [Total Units],
(select cat.CategoryName+
from Products pr
join Categories cat on pr.CategoryID=cat.CategoryID
where pr.ProductName=p.ProductName
group by cat.CategoryName) as [CAT_Name]
from Orders o
join [Order Details] od on od.OrderID=o.OrderID
join Products p on p.ProductID=od.ProductID) as q1 on e.EmployeeID=q1.EmployeeID
group by e.FirstName, e.LastName, q1.CAT_Name

```

=====

Лъчо: Моят вариант, не съм го тествал, за това плз Наско да пусне заявката на ум и да каже как е :-)

```

SELECT LastName, CategoryName, total_cnt FROM
(SELECT EmployeeID, CategoryID, SUM(Quantity) AS total_cnt FROM [Order Details] AS a1
JOIN Orders ON a1.OrderID = Orders.OrderID
JOIN Products ON Products.ProductID = a1.ProductID
GROUP BY EmployeeID, CategoryID
) AS a2
JOIN Employees ON Employees.EmployeeID = a2.EmployeeID
JOIN Categories ON Categories.CategoryID = a2.CategoryID

```

=====

Наско: Ето и едно решение от мен, че тези ми се струват много сложни.
Кажете, ако намерите проблеми.

```

select e.FirstName, e.LastName, c.CategoryName,
sum(od.Quantity) as count1, sum(od.Quantity*p.UnitsOnOrder) as count2
from [Order Details] od --N reda
JOIN Orders o ON od.OrderID = o.OrderID --Nx1 reda
JOIN Products p ON p.ProductID = od.ProductID --Nx1x1 reda
JOIN Categories c on c.CategoryID = p.CategoryID --Nx1x1x1 reda
JOIN Employees e on e.EmployeeID = o.EmployeeID --Nx1x1x1x1 reda
--vseki red ot od go ima po 1 pyt vuv "from" i mojem da sumirame v/u nego
group by e.EmployeeID, e.FirstName, e.LastName, c.CategoryID, c.CategoryName
--e.EmployeeID e kluch, taka che grupiraneto po e.FirstName, e.LastName ne vliqe na
razbivaneto po slujiteli
--analogichno c.CategoryID e kluch

```

3. Задачи и решения, на падали се задачи на КН и Инф –

<http://tinyurl.com/2v6z6r2> (файла за Бази)

Ако намерите грешки, пишете тук:

На последната задача отговора е: b)

4. Да се изведат имената на всички клиенти от таблицата Customers (същата е като

горната), които никога не са поръчвали продукт с име "Tofu":

```
SELECT c.CustomerName, COUNT(*) FROM Orders o, [Order Details] od, Products p,
Customers c
WHERE
  o.OrderID = od.OrderID
  AND od.ProductID = p.ProductID
  AND o.CustomerID = c.CustomerID
  AND p.ProductName LIKE '%[^Tofu]%'

HAVING COUNT(*) < 1
```

Категория "АСИ"

1. УСЛОВИЕ (Примерна от конспекта на СИ-2010)

Извличане и спецификация на изисквания към софтуерна система от примерно описание на такава.

РЕШЕНИЕ:

Категория "САРС"

1. УСЛОВИЕ: от курсовият си проект по ПИСС.

РЕШЕНИЕ:

2. УСЛОВИЕ (Примерна от конспекта на СИ-2010)

По зададени изисквания към софтуерната система да се построи модулна декомпозиция на системата.

РЕШЕНИЕ:

3. УСЛОВИЕ (Примерна от конспекта на СИ-2010)

По зададена модулна декомпозиция на системата и зададени изисквания да се построи дадена структура.

РЕШЕНИЕ:

Категория "УК"

1. УСЛОВИЕ (От лекциите по УК)

С какъв набор от входни данни за P и Q трябва да изследвате следния код по отношение на Code Coverage:

Read P


```

Read Q
IF P+Q > 100 THEN
    Print "Large"
ENDIF
IF P > 50 THEN
    Print " P Large"
ENDIF

```

РЕШЕНИЕ:

1 test for statement coverage & branch coverage, 2 for branch coverage

$P=60$ $Q=60$ ($P>50$; $P+Q>100$) $\rightarrow$ "Large", "P Large"

$P=40$ $Q=80$ ($P<50$; $P+Q > 100$) $\rightarrow$ "Large"

$P=80$ $Q=10$ ($P>50$; $P+Q < 100$) $\rightarrow$ "P Large"

2. УСЛОВИЕ (Примерна от конспекта на СИ-2010)

- Katia

Да се създадат тестови сценарии по зададена примерна софтуерна система:

Тестови сценарии за организация на функционалното тестване в софтуерна система контролира улично осветление с цел икономия на енергия

Входни данни:

N – краен брой точки (улични лампи)

SS – status season / днешен режим за съответния сезон (SS x интензитет, като SS е различна константа за всеки сезон)

L – естествена осветеност на областта на тестване (0~1)

F – function / +0.5x интензитет ако минава кола през съответната точка

-0.5x интензитет, ако не минва кола през съответната точка

k – константа, по която се умножава интензитета на дадена лампа, при положение, че съседната до нея не работи (=0.5)

Да приемем, че интензитета, който лампите са длъжни да предоставят, се измерва и пресмята по формулата: $X = L * F * SS * k$ за лампа (съответно за цялата улица интензитетът на всички лампи е $e = m * (L * F * SS * k) + n * (L * F * SS * k)$, $m+n=N$, m – брой лампи с неработещи съседи, n - брой лампи с работещи съседи)

Описание на тестовите сценарии:

При валидни данни:

- 1) $L=1$ (слънчев ден); $F=0.5$ (преминава кола); $SS=0.25$ (лято е, няма нужда от допълнително осветление); $k=0.5$ (съседната лампа не работи)

Резултат: Ако $X=1*0.5*0.25*0.5$, тогава системата е изправна и е пресметнала всички външни фактори правилно. При несъответствие на коефициентите се проверяват съответно дали има преминаваща кола(сензорите на движение), дали има неизправна лампа до дадената.

- 2) $L=1$ (слънчев ден); $F=0.1$ (не преминава кола); $SS=0.75$ (есен е, има нужда от допълнително осветление); $k=0.5$ (двете съседни лампи не работят)

Резултат: Ако $X=1*0.1*0.75*0.5$, тогава системата е изправна и е пресметнала всички външни фактори правилно. При несъответствие на коефициентите се проверяват съответно дали има преминаваща кола(сензорите на движение), дали взимайки датата в предвид, системата е установила правилно сезона(есен в този тестов сценарий), дали в дадения момент наистина е преминала кола и били ли са в изправност обкръжаващите лампи.

При невалидни данни*:

- 1) $L=0.2$ (мъглив ден); $F=0$ (не се отчита движение по улицата); $SS=0.75$ (есен е, има нужда от допълнително осветление); $k=0.5$ (двете съседни лампи не работят)

Резултат: Ако $X=0.2*0*0.75*0.5$, системата не отчита никакво движение по улицата, което е невъзможно, тоест има някаква неизправност в сензорите за отчитане на движение.

Решение: Изпращане на компетентен персонал, който да оправи уредите.

- 2) $L=0.2$ (мъглив ден ден); $F=0.5$ (преминава кола по улицата); $SS=0.67$ (зима е, има нужда от допълнително осветление); $k=0.1$ (съседните лампи работят)

Резултат: Ако $X=0.2*0.5*0.67*0.1$, системата не отчита никаквасветлина по улицата, тоест има някаква неизправност в сензорите за измерване на осветеността.(не е възможно осветеността да е 0, защото дори и през облачна нощ, луната хвърля някаква светлина по улицата)

Решение: Изпращане на компетентен персонал, който да оправи уредите.

- 3) $L=1$ (слънчев ден); $F=0.1$ (не преминава кола); $SS=0.75$ (есен е, има нужда от допълнително осветление); $k=0.5$ (двете съседни лампи не работят)

Резултат: Ако $X=1*0.1*0.75*0.5$, тогава систематаби трябвало да е изправна и да е пресметнала всички външни фактори правилно. Проблемът в този тестов сценарий е, че независимо, че сезонът е есен и би трябвало да се вземе под значение, че е по-тъмен сезон, даденият ден е много слънчев, тоест коефициентът SS трябва да се намали както за случай, че сезона е лято.

Решение: Оправят се грешки в имплементацията на системата, които трябва да съпоставят логически всички външни фактори и да взимат адекватно променените такива,

ако се налага(както в дадения пример отгоре)

*Системата е конструирана така, че всяка една от константите ѝ трябва да има положителна стойност, за да се каже, че тестовите данни са изправни. Нулевата стойност оказва точно кой компонент не работи изправно и трябва да бъде оправен.

РЕШЕНИЕ:

Тестов сценарий 2 с валидни данни:

$L=1$ (слънчев ден); $F=0.1$ (не преминава кола); $SS=0.75$ (есен е, има нужда от допълнително осветление); $k=0.5$ (двете съседни лампи не работят)

Обхват: N на брой улични лампи на дадена улица

Критерии за вход и изход: За вход – да има ненулеви данни; За изход – съобщение, което системата изпраща на набора от улични лампи за включване

Тестова техника: Функционално тестване(Black Box)

Използвана метрика: Defect Rate

Инструмент за тестване: IBM Rational Functional Tester(RFT) v7.0.1

Приложен стандарт: ISO 9001:2000

Резултат: Ако $X=1*0.1*0.75*0.5$, тогава системата е изправна и е пресметнала всички външни фактори правилно. При несъответствие на коефициентите се проверяват съответно дали има преминаваща кола(сензорите на движение), дали взимайки датата в предвид, системата е установила правилно сезона(есен в този тестов сценарий), дали в дадения момент наистина е преминала кола и били ли са в изправност обкръжаващите лампи.

Старт на тестов сценарий:

- Пускане на системата
- Проверка на коефициента за естествената осветеност
- Ако е 0, има грешка при измерването му от сензорите, тоест грешка във въведените данни → коригиране и отстраняване на грешката
- Ако е по-голям от 0, проверка дали времето наистина е такова, каквото той показва(сверяване), ако не е -> поправка на коефициента
- Проверка на коефициента, който показва дали има движение по улицата(справка с видеото от поставените камери в точния ден и час)
- Проверка на коефициента за състоянието на сезона

- Проверка на изправността на съседните лампи(малък тест,който показва дали работи или има проблем). Ако някоя от двете съседни лампи не работи, коефициентът k става равен на 0.5
- Всички входни данни са сверени и проверени -> проверка дали това е направено за всичките N на брой лампи на улицата
- При изправност на входни данни и улични лампи -> пуснаке на осветлението на улицата
*Нужни са два теста с валидни данни, с които се изчерпва промяната на възможните коефициенти

Базов тестов сценарий 3 с невалидни данни:

Входни данни: L ; F ; SS ; k

Цел: Да покаже неизправността на системата при неверни данни

Обхват: N на брой улични лампи на дадена улица

Критерии за вход и изход: За вход – да има ненулеви данни; За изход – съобщение за намерена грешка в невалидните данни

Тестова техника: Функционално тестване(Black Box)

Използвана метрика: Defect Rate

Инструмент за тестване: IBM Rational Functional Tester(RFT) v7.0.1

Приложен стандарт: ISO 9001:2000

Старт на тестов сценарий:

- Проверка на коефициента за естествената осветеност
- Ако е 0, има грешка при измерването му от сензорите, тоест грешка във във входните данни
- Намиране на решение за коригиране на грешката
- Изпълнение на взетото решение
- Започване на тестовият сценарий отново с наяална проверка за валидността на данните
- Проверка на коефициента, който показва дали има движение по улицата(справка с видеото от поставените камери в точния ден и час)
- Ако е 0, има грешка при измерването му от сензорите, тоест грешка във във входните данни
- Намиране на решение за коригиране на грешката
- Изпълнение на взетото решение
- Започване на тестовият сценарий отново с наяална проверка за валидността на данните
- Проверка на коефициента за състоянието на сезона
- Ако е 0, има грешка при измерването му от сензорите, тоест грешка във във входните данни

данни

- Намиране на решение за коригиране на грешката
- Изпълнение на взетото решение
- Започване на тестовият сценарий отново с наяална проверка за валидността на данните
- Проверка на изправността на съседните лампи(малък тест,който показва дали работи или има проблем). Ако някоя от двете съседни лампи не работи, коефициентът k става равен на 0.5
- Ако е 0, има грешка при измерването му от сензорите, тоест грешка във във входните данни

данни

- Намиране на решение за коригиране на грешката
- Изпълнение на взетото решение
- Започване на тестовият сценарий отново с наяална проверка за валидността на данните
- При изкарване на верен резултат въпреки греяни входни данни, започни изследване отначало

Кат

егория “XML”

1. УСЛОВИЕ: (Katia)

XML file + DTD file + XML schema.

XML файлът описва информация за хотели като име, адрес, стаи, а DTD файлът и XML schema-та валидират този XML.

РЕШЕНИЕ:

[XML file](#)

[XML schema](#)

XML file 1 - DTD

[DTD filey](#)