СОФИЙСКИ УНИВЕРСИТЕТ "СВ. КЛИМЕНТ ОХРИДСКИ"
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

ДЪРЖАВЕН ИЗПИТ
ЗА ПОЛУЧАВАНЕ НА ОКС "БАКАЛАВЪР ПО ИНФОРМАТИКА", 27-28.03.2004 г.

ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)

Задача 1

1) **{2т.}** Да се построи минимален детерминиран краен автомат, еквивалентен на автомата:
$A = < \{q_0, q_1, ..., q_6\}, \{0,1\}, q_0, \delta, \{q_6\} >$

δ :

q	0	1
q_0	$\varnothing$	$\{q_0, q_3, q_6\}$
q_1	$\{q_2, q_6\}$	$\{q_5\}$
q_2	$\{q_2, q_6\}$	$\{q_1\}$
q_3	$\{q_3\}$	$\{q_3, q_4, q_6\}$
q_4	$\varnothing$	$\varnothing$
q_5	$\varnothing$	$\{q_1\}$
q_6	$\varnothing$	$\varnothing$

2) **{3 т.}** Функцията $f(x_1,x_2,...,x_n)$ е симетрична, ако $f(x_1,x_2,...,x_n)=f(x_{i_1},x_{i_2},...,x_{i_n})$ за всяка пермутация $(i_1,i_2,...,i_n)$ на числата от 1 до n. Да се провери пълно ли е множеството $\{f(\tilde{x}^n), g(x,y,z)=x.\bar{y}\vee x.\bar{z}\vee \bar{y}.\bar{z}\}$, ако $f(x_1,x_2,...,x_n)$ е симетрична функция, съществено зависеща от n=2k (k≥1) променливи.

3) **{2 т.}** Да се намери мощността на единичното множество на функцията $f(\tilde{x}^n)\oplus g(\tilde{x}^n)$, ако са известни дължините l_1 и l_2 на съвършените дизюнктивни нормални форми съответно на функциите $f(\tilde{x}^n)\to g(\tilde{x}^n)$ и $g(\tilde{x}^n)\to f(\tilde{x}^n)$.

4) **{2 т.}** Да се построи краен автомат, разпознаващ езика, описан с регулярния израз:
$0^*(10^*(00)^+ + 1)^+$
Да се построи също и автоматна граматика, която го поражда.

5) **{2 т.}** Каква е дължината на съвършената дизюнктивна нормална форма на следната функция:
$f(\tilde{x}^n) = (x_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2) \to x_3 ... x_n, n \geq 3$
   a) $2^{n-1} - 2$
   b) $2^{n-1} + 2$
   c) $2^{n-1} + 1$
   d) $2^n - n + 1$

6) **{4 т.}** Да се състави командна процедура на езика на командния интерпретатор bash за Linux, която прочита от стандартния вход десетично число. В случай, че общият брой на файловете в текущата директория надвишава въведената стойност, съответна част от тези файлове да бъде прехвърлена в директорията, името на която е зададено като параметър на процедурата в командния ред, по такъв начин, че броят на останалите в директорията файлове да стане равен на въведеното число.

7) **{3 т.}** Като параметри на командна процедура са зададени десетични числа. Да се състави командна процедура на езика на командния интерпретатор bash за Linux, която извежда на стандартния изход:
   - онези от параметрите, които са идентификатори на стартирани в системата процеси;
   - броя на останалите параметри.

8) **{3 т.}** Даден е следният фрагмент от програма:

```
int fd, i;
fd = creat ( “new_file”, 0777 );
close ( 1 );
i = dup (fd);
write ( i, “EXAMPLE\n”, sizeof(“EXAMPLE \n” ) );
write ( 1, “EXAMPLE \n”, sizeof(“EXAMPLE \n” ) );
```

Като резултат от изпълнението на дадената поредица от оператори:

a) на терминала ще се изведе низът “EXAMPLE” и във файла “new_file” ще се запише низът “EXAMPLE”
b) във файла “new_file” ще се запише два пъти низът “EXAMPLE”
c) на терминала ще се изведе два пъти низът “EXAMPLE”

9) **{3 т.}** Даден е следният фрагмент от програма:

```
int  pid, k=0, status;
if ( (pid  = fork() ) = = 0 ) k++;
else  { wait( &status); --k ; }
++k;
printf( “ Stoinostta na k = %d;”, k );
```

Като резултат от изпълнението на дадената поредица от оператори на стандартния изход ще се изведе:

a) Stoinostta na k =2; Stoinostta na k =0;
b) Stoinostta na k =2;
c) Stoinostta na k =0;
d) Stoinostta na k =0; Stoinostta na k =2;

10) **{1 т.}** Какво е предназначението на следната функция на езика Scheme:

```
(define (f l1 l2)
  (cond ((null? l2) l1)
        ((member (car l2) l1) (f l1 (cdr l2)))
        (else (cons (car l2) (f l1 (cdr l2))))))
```

a) намира сечението l1∩l2
b) намира обединението l1∪l2
c) намира разликата l1 \ l2
d) намира разликата l2 \ l1

11) **{2 т.}** Нека е дадено (т.е. оценено от интерпретатора на Scheme) следното множество от дефиниции:

```
(define (function l)
  (if (null? l)
      '()
      (cons (car l) (function (delete-all (car l) (cdr l))))))
(define (delete-all s l)
  (cond ((null? l) '())
        ((equal? (car l) s) (delete-all s (cdr l)))
        (else (cons (car l) (delete-all s (cdr l)))) ))
```

Напишете оценката на следния израз:

(function '(a s d f c a f e d a)) → ……

12) **{2 т.}** Попълнете липсващите изрази в дефиницията на функцията list-numb-aux при условие, че функцията list-numb връща като резултат списък от поредните номера на тези елементи на списъка l, които съвпадат със символния атом a, например (list-numb '(x y a b c a d f a) 'a) → (3 6 9).

```
(define (list-numb l a)
  (define (list-numb-aux l a n)
    (cond ((null? l) '())
          ((eq? (car l) a) (cons n ……..))
          (else (list-numb-aux …….. a ……))))
  (list-numb-aux l a 1))
```

13) **{3 т.}** Даден е ориентиран граф, чиито възли са именувани с уникални идентификатори. Графът е описан със средствата на езика Scheme като асоциативен списък, в който ключове са имената на възлите, а асоциираната с даден ключ стойност е списък от имената на преките наследници на съответния възел, например:

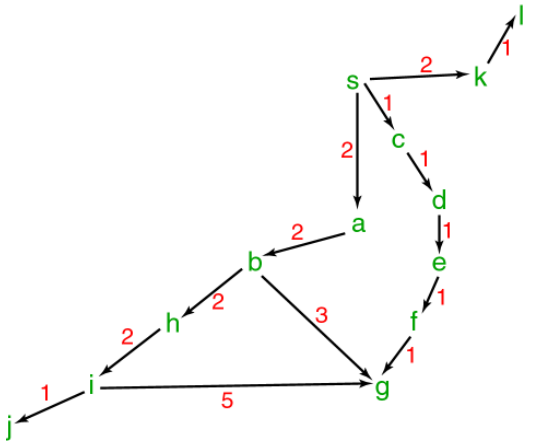
(define graph '((a b c) (b d e) (c f g h) (d i)))

Попълнете липсващите изрази в дефинициите на функциите depth-bound и extend-b при условие, че функцията depth-bound-search връща като резултат път в графа от възела start до възела goal, като реализира алгоритъма за търсене в дълбочина до определено ниво depth.

```
(define (depth-bound-search start goal depth)
  (reverse (depth-bound (list (list start)) goal depth)))
(define (depth-bound paths goal depth)
  (cond ((null? paths) '())
        ((eq? (car (car paths)) goal) (car paths))
        (else (depth-bound (append ……..……..…….. ……..……) goal depth))))
(define (extend-b path depth)
  (if ……
      '()
      (map (lambda (x) (cons x path)) (succs (car path)))))
(define (succs node)
  (cdr (assq node graph)))
```

14) **{2 т.}** Даден е ориентиран граф, представен на фигурата и описан чрез поредица от факти на Prolog от вида arc(<Node1>,<Node2>,<Cost>), всеки от които означава, че в графа съществува дъга с начало <Node1>, край <Node2> и дължина (цена) <Cost>.

Ако се търси път в графа от върха “s” до върха “g”, попълнете следната таблица:



Стратегия за търсене	Намерен път	Брой обходени възли	Посочете дали намереният път е оптимален (най-къс). Обосновете твърдението си
Depth-first search			
Breadth-first search			

15) **{3 т.}** Даден е ориентираният граф от предишната задача. Дадена е също така поредица от факти на Prolog от вида h(<Node>,<Cost>), дефиниращи евристичната функция, с помощта на която се пресмята приближена стойност <Cost> на разстоянието от върха <Node> до върха “g”:

h(a,2). h(b,3). h(c,4). h(d,3).
h(e,2). h(f,1). h(g,0). h(h,4).
h(i,5). h(j,6). h(k,5). h(l,6). h(s,4).

Ако се търси път в графа от върха “s” до върха “g”, попълнете следната таблица:

Стратегия за търсене	Намерен път	Брой обходени възли	Посочете дали намереният път е оптимален (най-къс). Обосновете твърдението си
Best-first search			

Beam search (широчина на лъча BeamSize = 2)			
A* search			

16) **{2 т.}** Базата от правила и работната памет (базата от факти) на една система, основана на правила, са описани със средствата на езика Prolog както следва:

```
% Дефиниции на използваните оператори
:- op(900,fx,if).
:- op(890,xfx,then).
:- op(880,xfy,or).
:- op(870,xfy,and).
% Правила
if has_hair(X) or gives_milk(X) then mammal(X).
if can_fly(X) then bird(X).
if can_swim(X) and has_gills(X) then fish(X).
if fish(X) then can_not_speak(X).
if bird(X) and can_swim(X) then swan(X).
if mammal(X) and has_brown_color(X) then bear(X).
% Факти (работна памет)
fact(can_fly(gogo)).
fact(can_fly(pepo)).
fact(can_swim(bobo)).
fact(can_swim(gogo)).
fact(has_brown_color(teddy)).
fact(has_gills(gogo)).
fact(gives_milk(teddy)).
```

Предполага се, че интерпретаторът на правилата извършва прав извод (извод-напред, управляван от данните), като на всяка стъпка от работния си цикъл записва заключението на избраното за изпълнение правило като нов факт в края на работната памет. Какво ще съдържа работната памет след завършване на работата на интерпретатора на правилата, ако на всяка стъпка от работния си цикъл той избира и изпълнява това правило от конфликтното множество, чието условие се удовлетворява от най-скоро записан в работната памет факт?

17) **{2 т.}** Допишете дефиницията на отношението is_true(<Goal>,<List>), ако то играе ролята на интерпретатор на правилата, осъществяващ обратен извод (извод-назад, управляван от целите) в система, основана на правила, в която базата от правила и работната памет са представени със средствата на езика Prolog по същия начин, както в предишната задача. Интерпретаторът проверява дали целта <Goal> е изводима и в случай на успех свързва <List> със списъка от използваните при извода факти и правила.

```
is_true(P,[fact(P)]):-
    fact(P), !.
is_true(P,[rule(if Condition then P)|L]):-
    if Condition then P,
    ..... .
is_true(P and Q,L):-
    is_true(P,L1),
    …….. ,
    …….. .
is_true(P or Q,L):-
    ……. ;
    …… .
```

18) **{1 т.}** Какъв ще бъде резултатът от изпълнението на следния програмен фрагмент?

```
int S[2] = {0, 0} ;
for (int i = 1 ; i <= 10 ; i++ )  S[ i % 2 ] += i ;
printf ( " %d ", S[ 0 ] – S[ 1 ] ) ;
```

a) 0 b) 5 c) 10 d) 15

19) **{1 т.}** В следващия програмен текст допишете липсващите 4 константи по такъв начин, че дефинираната целочислена функция да реализира съответствието от таблицата:

month	1	2	3	4	5	6	7	8	9	10	11	12
days	31	28	31	30	31	30	31	31	30	31	30	31

```
int  days  ( int month )
        {  int result       =  …….. ;
           const  int d28 =  …….. ;
           const  int d30 =  …….. ;
           const  int d31 =  …….. ;
           switch ( month )
                  {  case 2:                                result += d28;
                     case 4: case 6: case 9: case 11:       result += d30;
                     default :                              result += d31;
                  }
          return (result) ;
        }
```

20) **{2 т.}** Следващата функция трябва да извежда стринговия си аргумент в обратен ред, т.е.

```
//     print_back ( "abcdefgh" ) ;        трябва да изведе     hgfedcba
void print_back ( char * s )
     {  if  ( *s )   {   print_back ( s++ );
                         printf ( "%c ", *s );
                     }
     }
```

Подчертайте грешния оператор и напишете отдясно верния му вариант.

21) **{1 т.}** Какъв ще бъде резултатът от изпълненнието на следващия програмен фрагмент? Напишете извежданите стойности в съответните полета от коментарната част.

```
int w=3;
int  F( int x, int *y ) { x++; (*y)++; w++; return( x+*y+w); }
void main( )
   { int u=1, v=2 ;     printf ( "u=%d   v=%d   w=%d \n )", u, v, w );          //   u=1     v=2     w=3
     w = F(u,&v);       printf ( "u=%d   v=%d   w=%d \n )", u, v, w );          //   u=.....  v=.....  w=.....
   }
```

22) **{2 т.}** В израза (x = - y) ? (- x / y) : y променливите x и y са от тип int.
Кой от следващите изрази е еквивалентен на посочения?

a) (x = - y) b) (- x / y) c) y d) !! (x = - y)
e) !! (- x / y) f) !! y g) нито един от изброените h) посоченият израз е грешен

23) **{2 т.}** Обяснете каква задача решава и какъв резултат извежда следната програма:

```
#include <stdio.h>
int f(int m, int n)
{ int result;
  if (n==0)
     result = m;
  else if (n>m)
       result=f(n,m);
  else result=f(n,m%n);
  return result;
}
void main()
{ int m=18, n=24;
  printf("\nРезултатът = %d", f(m,n));
}
```

24) **{2 т.}** Функцията enqueue добавя елемент в последователна опашка. Попълнете липсващия оператор на мястото на многоточието в следния фрагмент от програма:

```
#define M 100
struct queue
{ int f; // номер на фиктивен елемент преди началото на опашката
  int r; // номер на елемента в края на опашката
  double q_array[M];
};
void enqueue(queue *q, double x)
{ if (q->r == M) q->r=1;
  else q->r++;
  if (q->r == q->f)
  { cout << "\nПрепълване\n";
    exit(1); }
    ……..
}
```

25) **{2 т.}** Функцията pop изключва елемент от свързан стек. Попълнете липсващите оператори на местата на двете многоточия в следния фрагмент от програма:

```
struct stack_el
{ int info;
  stack_el *link;
};
int pop (stack_el **t)
{ stack_el *p;
  int x;
  if (*t == NULL)
  { cout << "\nСтекът е празен\n";
    exit(1);}
  .....
  *t = p->link;
  .....
  delete p;
  return x;
}
```

26) **{2 т.}** Функцията add добавя нов елемент в свързан списък след k-тия елемент на списъка. Попълнете липсващия израз и липсващия оператор на местата на двете многоточия в следния фрагмент от програма:

```
struct student
{ long nomer;     // факултетен номер
  float sr_uspeh; // среден успех
  student *next;
};
// first – указател към началото на свързания списък
void add(student *first, int k)
{ student *ptr = first, *ptr1;
  while (.....)
    { ptr=ptr->next; i++;}
  if (ptr == NULL)
  { cout << "\nНяма " << k << "елемента в списъка\n";
    return; }
  if ((ptr1 = new student) == NULL)
  { cout << "\nНяма свободна памет.\n";
    exit(1); }
  ptr1->next = ptr->next;
  .....
  cout << "\nНомер:";
  cin >> ptr1->nomer;
  cout << "\nСреден успех:";
  cin >> ptr1->sr_uspeh;
}
```

27) **{2 т.}** Да се открие и коригира грешката в следния фрагмент от дефинирането на клас Array и на предефиниращата функция за равенство:

```
class Array {
public:
  . . .
  int operator==(const Array &) const;
  . . .
private:
  int *ptr; // указател към първия елемент на масива
  int size; // брой на елементите на масива
};
int Array::operator==(const Array &right) const
{ for(int i=0; i<size; i++)
    if (ptr[i] != right.ptr[i])
      return 0;
  return 1;
}
```

28) **{2 т.}** Да се открие, обясни и коригира грешката в следния фрагмент от дефинирането на клас String и предефиниращата функция на операцията за конкатенация:

```
class String {
public:
  . . .
  String &operator+=(const String &);
  . . .
private:
  char *sPtr; // указател към началото на низа
  int length;  // дължина на низа
};
```

```
String &String::operator+=(const String &right)
{ char *tempPtr = sPtr;
  length = right.length;
  sPtr = new char[length+1];
  assert(sPtr!=0);
  strcpy(sPtr, tempPtr);
  delete []tempPtr;
  return *this;
}
```

29) **{3 т.}** Нека φ е формула от първи ред, а $\psi = \forall X\, \varphi$. Кое от следните твърдения винаги е вярно?
a) Формулата φ е изпълнима точно тогава, когато ψ е изпълнима
b) $\varphi \leq \psi$
c) Във всяка едноелементна структура е вярна формулата $\forall X(\varphi \to \psi)$
d) φ е неизпълнима, ако и само ако ψ е неизпълнима

30) **{3 т.}** Нека
$\varphi = (\forall X(p(X) \to \neg\exists Y(p(Y) \vee q(X,Y))) \to (\exists Y\, p(Y) \& \forall X\, \exists Y\, q(X,Y))$,
където p и q са предикати, а X и Y са различни променливи. Коя от следните формули е пренексна нормална форма на φ:
a) $\forall X\exists Y\exists Z\exists W(\neg p(X) \vee \neg(p(Y) \vee q(X,Y)) \vee p(Z)\&q(W,Z))$
b) $\exists X\exists Y\exists Z\forall W(p(X)\&(p(Y) \vee q(X,Y)) \vee p(Z)\&q(W,Z))$
c) $\exists X\exists Y\exists Z\forall W\exists U(p(X)\&(p(Y) \vee q(X,Y)) \vee p(U)\&q(W,Z))$
d) $\exists X\exists Y\exists Z\forall W\exists U(p(X)\&(p(Y) \vee q(X,Y)) \vee p(Z)\&q(W,U))$

31) **{3 т.}** Нека P е следната програма на Prolog:
```
p(0).
p(X):- p(Y), X is Y+1.
q(X,0,Z):- !, fail.
q(X,Y,Z):- p(Z), Z*Y=<X, X<(Z+1)*Y, !.
```
При цел ?- q(20,7,Z). програмата извежда:
a) Z = 1
b) Z = 2
c) Z = 3
d) No

32) **{3 т.}** Нека P и Q са логически програми от един и същ език със съвпадащи най-малки ербранови модели. Посочете вярното твърдение:
a) P и Q съвпадат
b) Формулата P е логически еквивалентна на формулата Q
c) За всеки основен атом A е изпълнено: от P логически следва A, ако и само ако от Q логически следва A
d) M е ербранов модел на P точно тогава, когато M е ербранов модел на Q

33) **{2 т.}** Нека P е следната програма на Prolog:
```
p(X):- q(X).
q(X):- r(f(X)).
r(f(X)):- p(f(X)).
```
Нека M е най-малкият ербранов модел на P. Кое от следните твърдения е вярно:
a) $M = \{p(f^n(0)): n \in N\}$
b) $M = \{p(f^{2n}(0)): n \in N\}$
c) $M = \{p(f^{3n}(0)): n \in N\}$
d) $M = \varnothing$

34) **{2 т.}** Нека P и Q са непрекъснати условия. Кое от следните твърдения **не** е вярно:
a) Условието (P&Q) е непрекъснато
b) $\neg$P е непрекъснато
c) (P $\vee$ Q) е непрекъснато
d) Ако P не е непрекъснато, то и Q не е непрекъснато

35) **{2 т.}** За дадения оператор Г:
Г(f)(x,y) > if x=0 then 7 else f(x-1,f(x,y+1))+1
покажете кое от посочените свойства е изпълнено за най-малката му неподвижна точка $f_Г$.
a) $f_Г(x) = x+7$
b) $f_Г(x) = 7$
c) $f_Г$ е никъде недефинираната функция
d) $f_Г$ е дефинирана само при x=0

36) **{3 т.}** Нека R е рекурсивната програма в Nat:

F(X,1), *where*

F(X,Y) = *if* X = Y *then* 1 *else* F(X,Y+1)*(Y+1)

Кое от следните твърдения е в сила:

a) $\forall a(D_V(R)(a) > a!)$

b) $\forall a(D_V(R)(a) > 1)$

c) $\forall a(a \geq 1 \ \& \ !D_V(R)(a) \Rightarrow D_V(R)(a) > 1)$

d) $\forall a(a \geq 1 \ \& \ !D_V(R)(a) \Rightarrow D_V(R)(a) > a!)$

37) **{3 т.}** Нека R_1 и R_2 са следните рекурсивни програми в Nat:

R_1:

$F_1(X)$, *where*

$F_1(X)$ = *if* $X \leq 1$ *then* 1 *else* $F_1(X-1)+ F_1(X-2)$

R_2:

$F_2(X)$, *where*

$F_2(X)$ = *if* $X \leq 1$ *then* 1 *else* $2*F_2(X-2)$

Кое от следните твърдения е вярно:

a) $\forall a \forall b(!D_V(R_1)(a) \ \& \ !D_V(R_2)(b) \ \& \ a \geq b \Rightarrow D_V(R_1)(a) \geq D_V(R_2)(b))$

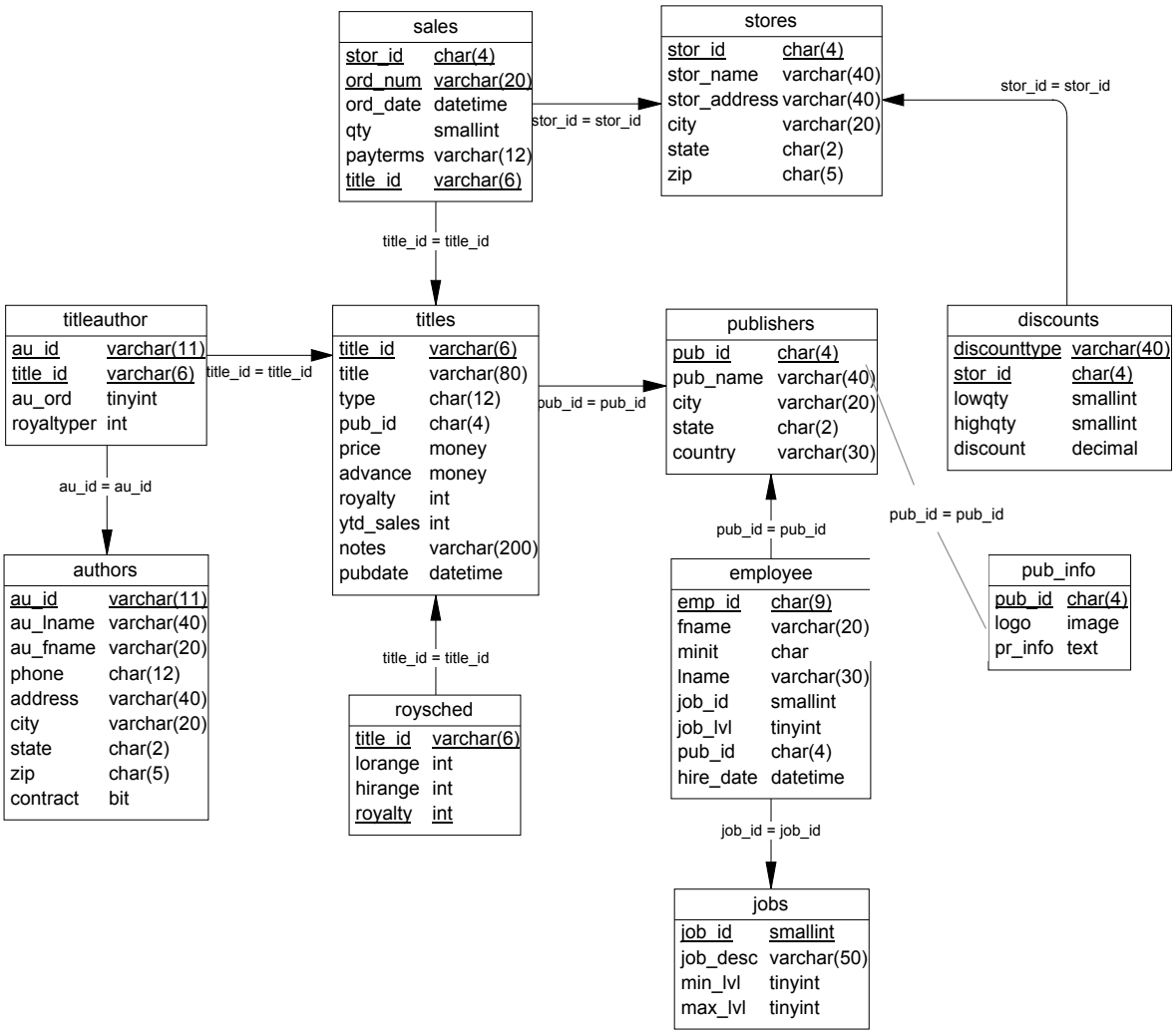
b) $\forall a \forall b(!D_V(R_1)(a) \ \& \ !D_V(R_2)(b) \ \& \ a \geq b \Rightarrow D_V(R_1)(a) = D_V(R_2)(b))$

c) $\forall a \forall b(!D_V(R_1)(a) \ \& \ !D_V(R_2)(b) \ \& \ a \geq b \Rightarrow D_V(R_1)(a) \leq D_V(R_2)(b))$

d) $\forall a(!D_V(R_1)(a) \ \& \ !D_V(R_2)(a) \Rightarrow D_V(R_1)(a) \leq D_V(R_2)(a))$

Задача 2

За нуждите на издателска къща е създадена база от данни pubs, която поддържа следната информация:
1. Автори (authors): име (au_fname), фамилия (au_lname), телефон, адрес, град, щат, вид на договора.
2. Книги (titles): заглавие, тип на книгата (бизнес, кулинария и др.), издателство, цена на отделен екземпляр, авансово плащане за всички бройки книги, процент от прихода, който ще се заплаща на автора (royalty), годишни продажби (ytd_sales), допълнителни бележки, дата на издаване.
Един автор може да издава една или повече книги.
Една книга може да има един или повече автори.
3. Издателства (publishers): име, град, щат, държава.
В издателствата работят служители.
4. М*агазини (stores*): име, адрес, град, щат, пощенски код.
5. *Продажби (sales)* в магазините: номер на магазин, номер на поръчка, дата на поръчката, брой книги, условия на плащане (с фактура или в брой), номер на книга.
6. *Намаления* (*discounts*) за първа поръчка, за голям обем на поръчки и за някои магазини.
7. *Служители (employee)*: име (fname), фамилия (lname), работа, кога е нает и в кое издателство работи.
8. *Работата (jobs)*, която извършва даден служител: описание на работата, минимално и максимално ниво на заплащане.



1) **{1 т.}** За базата от данни pubs кой от следните оператори ще изведе всички фамилии на автори от щат Калифорния (CA), започващи с Gr?
   a) SELECT au_lname, state
      FROM authors
      WHERE au_lname = 'Gr%';
   b) SELECT au_lname, state
      FROM authors
      WHERE au_lname LIKE 'Gr%';
   c) SELECT au_lname
      FROM authors
      WHERE au_lname LIKE 'Gr%' AND state='CA';
   d) SELECT au_lname, state
      FROM authors
      WHERE au_lname = 'Gr_' AND state='CA';

2) **{1 т.}** За базата от данни pubs кой от следните оператори ще изведе списък без повтарящи се елементи, който съдържа всички имена на служители?
   a) SELECT fname
      FROM employee
      WHERE fname='DISTINCT';
   b) SELECT DISTINCT fname
      FROM employee;
   c) SELECT ALL fname
      FROM employee;
   d) SELECT fname, lname
      FROM employee
      WHERE fname='UNIQUE';

3) **{2 т.}** За базата от данни pubs кой от следните оператори ще изведе средната годишна продажба на книги на автори от град Oakland?
   a) SELECT AVERAGE(titles.ytd_sales) AS AvgOfytd_sales
      FROM authors INNER JOIN (titles INNER JOIN titleauthor ON titles.title_id = titleauthor.title_id) ON authors.au_id = titleauthor.au_id
      HAVING city='Oakland';
   b) SELECT AVG(titles.ytd_sales) AS AvgOfytd_sales
      FROM authors INNER JOIN (titles INNER JOIN titleauthor ON titles.title_id = titleauthor.title_id) ON authors.au_id = titleauthor.au_id
      WHERE city='Oakland';
   c) SELECT AVG(titles.ytd_sales) AS AvgOfytd_sales
      FROM authors INNER JOIN (titles INNER JOIN titleauthor ON titles.title_id = titleauthor.title_id) ON authors.au_id = titleauthor.au_id
      HAVING city='Oakland';
   d) SELECT AVG(titles.ytd_sales) AS AvgOfytd_sales
      FROM authors INNER JOIN (titles INNER JOIN titleauthor ON titles.title_id = titleauthor.title_id) ON authors.au_id = titleauthor.au_id
      WHERE city IS 'Oakland';

4) **{2 т.}** За базата от данни pubs кой от следните оператори ще изведе имената на издателствата заедно с имената и фамилиите на всички автори, които са от един и същ град със съответното издателство?
   a) SELECT publishers.pub_name, authors.au_fname, authors.au_lname
      FROM authors INNER JOIN publishers ON authors.city = publishers.city;
   b) SELECT publishers.pub_name, authors.au_fname, authors.au_lname
      FROM authors LEFT OUTER JOIN publishers ON authors.city = publishers.city;
   c) SELECT publishers.pub_name, authors.au_fname, authors.au_lname
      FROM authors RIGHT OUTER JOIN publishers ON authors.city = publishers.city;
   d) SELECT publishers.pub_name, authors.au_fname, authors.au_lname
      FROM authors FULL OUTER JOIN publishers ON authors.city = publishers.city;