МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА — Российский технологический университет»

Институт информационных технологий (ИТ)
Кафедра математического обеспечения и стандартизации
информационных технологий (МОСИТ)

ОТЧЁТ

по комплексу лабораторных работ

по дисциплине

«Системно-программные основы искусственного интеллекта»

Выполнил студент группы	ИКМО-05-18	Карих Д.С.
Принял	к.т.н., доцент	Смольянинова В.А.
Выполнено	«__» _____ 2018 г.	
Зачтено	«__» _____ 2018 г.	

МОСКВА 2018

Лабораторная работа 1

Задание 1. Составьте программу **Родственные отношения**, которая кроме родственных отношений parent (родитель) и ancestor (предок) программа должна содержать одно или несколько из следующих отношений:

brother (брат); sister (сестра); grand-father (дедушка); grand-mother (бабушка); uncle (дядя).

```
domains
  s = symbol
predicates
  nondeterm parent(s,s)
  female(s)
  male(s)
  nondeterm mother(s,s)
  nondeterm father(s,s)
  nondeterm ancestor(s,s)
  child(s,s)
  % ---- %
  nondeterm brother(s,s)
  nondeterm sister(s,s)
  nondeterm grandFather(s,s)
  nondeterm grandMother(s,s)
  nondeterm uncle(s,s)
clauses
  parent(pam,bob).
  parent(tom,bob).
  parent(tom,liz).
  parent(liz,jack).
  parent(bob,ann).
  parent(bob,pat).
  parent(pat,jim).
  female(pam).
  female(liz).
  female(ann).
  female(pat).
  male(tom).
  male(bob).
  male(jim).
child(Y,X):-
  parent(X,Y).
mother(X,Y):-
  parent(X,Y),
  female(X).
father(X,Y):-
  parent(X,Y),
  male(X).
ancestor(X,Z):-
  parent(X,Z).
ancestor(X,Z):-
  parent(X,Y),
  ancestor(Y,Z).
% ---- %
brother(X,Y):-
  parent(Z,X),
  parent(Z,Y),
  male(X).
sister(X,Y):-
  parent(Z,X),
  parent(Z,Y),
  female(X).
grandFather(Y,X):-
  parent(Z,X),
  father(Y,Z).
grandMother(Y,X):-
  parent(Z,X),
  mother(Y,Z).
uncle(Y,X):-
  parent(Z,X),
  brother(Y,Z).
```

Результаты выполнения программы для проверочных целевых утверждений приведены в таблице 1.

Целевое утверждение	Результат
brother(bob,liz).	yes
sister(liz,bob).	yes
grandFather(bob,jim).	yes
grandMother(pam,pat).	yes
uncle(bob,jack).	yes

Таблица 1: Результаты выполнения программы

Задание 2. Составьте программу, используя отношения likes ("нравится") и can_buy ("может купить").

```
domains
  s = symbol
predicates
  person(s)
  product(s)
  can_afford(s,s)
  likes(s,s)
  nondeterm can_buy(s,s)
clauses
  person(alice).
  person(bob).
  product(phone).
  product(laptop).
  can_afford(alice,phone).
  can_afford(bob,phone).
  can_afford(bob,laptop).
  likes(alice,laptop).
  likes(bob,phone).
  can_buy(X,Y):-
    person(X),
    product(Y),
    can_afford(X,Y),
    likes(X,Y).
```

Результаты выполнения программы для проверочных целевых утверждений приведены в таблице 2.

Целевое утверждение	Результат
can_buy(bob,phone).	yes
can_buy(alice,laptop).	no
can_buy(bob,alice).	no

Таблица 2: Результаты выполнения программы

Задание 3. Составьте собственную программу, состоящую из фактов и правил. Проверьте ее работу.

```
domains
  s = symbol
predicates
  nondeterm os(s)
  nondeterm arch(s)
  nondeterm runs_on(s,s)
  nondeterm device(s)
  nondeterm based_on(s,s)
  nondeterm supports(s,s)
clauses
  os(linux).
  os(android).
  os(windows).
  os(asusWRT).
  arch(x86).
  arch(arm).
  arch(mips).
  runs_on(linux,x86).
  runs_on(linux,arm).
  runs_on(linux,mips).
  runs_on(android,x86).
  runs_on(android,arm).
  runs_on(windows,x86).
  runs_on(asusWRT,mips).
  device(raspberrypi).
  device(oneplusone).
  device(laptop).
  device(router).
  based_on(rasbperry_pi,arm).
  based_on(oneplusone,arm).
  based_on(laptop,x86).
  based_on(router,mips).
  supports(X,Y):-
    device(X),
    os(Y),
    based_on(X,Z),
    runs_on(Y,Z).
```

Результаты выполнения программы для проверочных целевых утверждений приведены в таблице 3.

Целевое утверждение	Результат
supports(laptop,X).	linux,android,windows
supports(X,asuswrt).	router
runs_on(X,mips).	linux,asuswrt

Таблица 3: Результаты выполнения программы

Лабораторная работа 2

Задание. Напишите программу моделирования нескольких действий в соответствии с собственной предметной областью.

```
domains
  State = connected; installed; loaded; delivered; chosen; assembled;
          ready; configured; fixed; applied; none

  Part = case(State); motherboard(State); socket(State); cooler(State);
         powerSupply(State); ssd(State); videocard(State); cpu(State);
         thermalGrease(State); ram(State)
  Parts = parts(State, Part, Part, Part, Part,
                Part, Part, Part, Part, Part, Part)

  Peripheral = keyboard(State); mouse(State); monitor(State)
  Peripherals = peripherals(State, Peripheral, Peripheral, Peripheral)

  OS = temporaryOS(State); permanentOS(State)
  Software = os(OS, OS)
  Order = order(State)
  BootDrive = bootDrive(State)

  PC = pc(State, State, State, Parts, Peripherals,
          Order, BootDrive, Software)

  Action = assemble(string); install(string); deliver(string);
           choose(string); connect(string); attach(string); load(string)
  Action_List = Action*

predicates
  nondeterm do(PC, Action, PC)
  nondeterm actions(PC, Action_List)

clauses
  % choose parts and peripherals
  do(pc(_, _, _,
        parts(none, _, _, _, _, _, _, _, _, _, _),
        peripherals(none, _, _, _),
        _, _, _),
     choose("parts and peripherals"),
     pc(none, none, none,
```

```
      parts(chosen, case(none), motherboard(none), socket(none),
            cooler(none), powerSupply(none), ssd(none),
            videocard(none), cpu(none), thermalGrease(none),
            ram(none)),
      peripherals(chosen, keyboard(none), mouse(none), monitor(none)),
      order(none), bootDrive(none),
      os(temporaryOS(none), permanentOS(none)))).

% install RAM
do(pc(A, none, none,
      parts(chosen, B, C, D, E, F, G, H, I, J, ram(none)),
      K, L, M, N),
   install("RAM"),
   pc(A, none, none,
      parts(chosen, B, C, D, E, F, G, H, I, J, ram(installed)),
      K, L, M, N)).

% install CPU
do(pc(none, none, none,
      parts(chosen, A, motherboard(none), socket(none),
            cooler(none), B, C, D, cpu(none), thermalGrease(none), E),
      F, G, H, I),
   install("CPU"),
   pc(none, none, none,
      parts(chosen, A, motherboard(none), socket(fixed),
            cooler(installed), B, C, D, cpu(installed),
            thermalGrease(applied), E),
      F, G, H, I)).

% assemble case
do(pc(none, none, none,
      parts(chosen, case(none), motherboard(none), socket(fixed),
            cooler(installed), powerSupply(none), A, B,
            cpu(installed), thermalGrease(applied), H),
      I, J, K, L),
   assemble("case"),
   pc(assembled, none, none,
      parts(chosen, case(assembled), motherboard(installed),
            socket(fixed), cooler(installed), powerSupply(connected),
            A, B, cpu(installed), thermalGrease(applied), H),
      I, J, K, L)).
```

```
% install SSD
do(pc(assembled, none, none,
      parts(chosen, case(assembled), motherboard(installed), A, B,
            powerSupply(connected), ssd(none), C, D, E, F),
      G, H, I, J),
   connect("ssd"),
   pc(assembled, none, none,
      parts(chosen, case(assembled), motherboard(installed), A, B,
            powerSupply(connected), ssd(connected), C, D, E, F),
      G, H, I, J)).

% install videocard
do(pc(assembled, none, none,
      parts(chosen, case(assembled), motherboard(installed), A, B,
            C, D, videocard(none), E, F, G),
      H, I, J, K),
   connect("videocard"),
   pc(assembled, none, none,
      parts(chosen, case(assembled), motherboard(installed), A, B,
            C, D, videocard(installed), E, F, G),
      H, I, J, K)).

% deliver
do(pc(assembled, none, none,
      parts(chosen, case(assembled), A, B, C, D, ssd(connected),
            videocard(installed), E, F, ram(installed)),
      H, order(none), I, J),
   deliver("order"),
   pc(assembled, none, none,
      parts(chosen, case(assembled), A, B, C, D, ssd(connected),
            videocard(installed), E, F, ram(installed)),
      H, order(delivered), I, J)).

% connect peripherals
do(pc(assembled, none, none, A,
      peripherals(chosen, keyboard(none), mouse(none), monitor(none)),
      order(delivered), bootDrive(none), B),
   attach("peripherals"),
   pc(assembled, ready, none, A,
      peripherals(chosen, keyboard(connected), mouse(connected),
```

```
                    monitor(connected)),
          order(delivered), bootDrive(connected), B)).

  % load temporaryOS OS
  do(pc(assembled, ready, A, B, C, D, bootDrive(connected),
        os(temporaryOS(none), permanentOS(none))),
     load("Temporary OS"),
     pc(assembled, ready, A, B, C, D, bootDrive(connected),
        os(temporaryOS(loaded), permanentOS(none)))).

  % install permanentOS OS
  do(pc(assembled, ready, none, A, B, C, bootDrive(connected),
        os(temporaryOS(loaded), permanentOS(none))),
     install("Permanent OS"),
     pc(assembled, ready, configured, A, B, C, bootDrive(connected),
        os(temporaryOS(loaded), permanentOS(installed)))).

  actions(pc(assembled, ready, configured,
             parts(chosen, case(assembled), motherboard(installed),
                   socket(fixed), cooler(installed),
                   powerSupply(connected), ssd(connected),
                   videocard(installed), cpu(installed),
                   thermalGrease(applied), ram(installed)),
             peripherals(chosen, keyboard(connected), mouse(connected),
                         monitor(connected)),
             order(delivered), bootDrive(connected),
             os(temporaryOS(loaded), permanentOS(installed))),
           []).

  actions(S, [H|T]):-
    do(S,H,S1),
    actions(S1,T).

goal
  actions(pc(none, none, none,
             parts(none, case(none), motherboard(none), socket(none),
                   cooler(none), powerSupply(none), ssd(none),
                   videocard(none), cpu(none), thermalGrease(none),
                   ram(none)),
             peripherals(none, keyboard(none), mouse(none),
                         monitor(none)),
```

```
            order(none), bootDrive(none),
            os(temporaryOS(none), permanentOS(none))),
        X).
```

Некоторые результаты выполнения программы приведены ниже и представляют собой последовательность операций, необходимых для решения задачи.

```
X=[choose("parts and peripherals"), install("RAM"), install("CPU"),
   assemble("case"), connect("ssd"), connect("videocard"),
   deliver("order"), attach("peripherals"), load("Temporary OS"),
   install("Permanent OS")]
X=[choose("parts and peripherals"), install("RAM"), install("CPU"),
   assemble("case"), connect("videocard"), connect("ssd"),
   deliver("order"), attach("peripherals"), load("Temporary OS"),
   install("Permanent OS")]
X=[choose("parts and peripherals"), install("CPU"), install("RAM"),
   assemble("case"), connect("ssd"), connect("videocard"),
   deliver("order"), attach("peripherals"), load("Temporary OS"),
   install("Permanent OS")]
X=[choose("parts and peripherals"), install("CPU"), install("RAM"),
   assemble("case"), connect("videocard"), connect("ssd"),
   deliver("order"), attach("peripherals"), load("Temporary OS"),
   install("Permanent OS")]
```

Лабораторная работа 3

Вариант 3. Определение произведения элементов списка.

```
domains
  i = integer
  i_list = i*
predicates
  mul_list(i_list, i)
clauses
  mul_list([],1).
  mul_list([H|T],Res):-
    mul_list(T,PartRes),
    Res = H * PartRes.
```

Результаты выполнения программы для проверочных целевых утверждений приведены в таблице 4.

Целевое утверждение	Результат
mul_list([2,3,4,5,6], X).	720
mul_list([0,1,2,3], X).	0
mul_list([], X).	1

Таблица 4: Результаты выполнения программы

Вариант 8. Определить отношение перевод(Список1, Список2) для перевода списка чисел от 0 до 9 в список соответствующих слов.

```
domains
  i = integer
  s = symbol
  i_list = i*
  s_list = s*
predicates
  nondeterm translate(i_list,s_list)
clauses
  translate([],[]).
  translate([0],[zero]):-!.
  translate([1],[one]):-!.
  translate([2],[two]):-!.
  translate([3],[three]):-!.
  translate([4],[four]):-!.
  translate([5],[five]):-!.
  translate([6],[six]):-!.
  translate([7],[seven]):-!.
  translate([8],[eight]):-!.
  translate([9],[nine]):-!.
  translate([H|T],Res):-
    translate([H], [H_T]),
    translate(T, T_T),
    Res = [H_T | T_T].
```

Результаты выполнения программы для проверочных целевых утверждений приведены в таблице 5.

Целевое утверждение	Результат
translate([ ], X).	[ ]
translate([0], X).	[zero]
translate([4,2,2,1], X).	[four,two,two,one]

Таблица 5: Результаты выполнения программы

Вариант 13. Используя отношение conc, напишите цель, соответствующую вычеркиванию трех последних элементов списка L. Результат - новый список L1. Указание: L - сцепление L1 и трех элементного списка.

```
domains
  i = integer
  i_list = i*
predicates
  nondeterm conc(i_list,i_list,i_list).
clauses
  conc([], L, L).
  conc([H|T], L, [H|Res]):-
    conc(T,L,Res).
goal
  L = ...,
  conc(L1, [_,_,_], L).
```

Результаты выполнения программы для различных значений L приведены в таблице 6.

L	L1
[0,0,0,0,1,1,1]	[0,0,0,0]
[0,0]	Нет решения
[1,2,3,4,5]	[1,2]

Таблица 6: Результаты выполнения программы