Приложение Б
(справочное)
Примеры решения CSP задач

Б.1. Раскрашивание карты

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Решение задачи о раскрашивании карты
%---------------------------------------------------------------------------------
% Карта представляется в виде списка [A::[B,D], B::[C,D], C::[D,E], D::[E]],
% элементами которого являются структуры, образованные оператором «::».
% Каждая структура состоит из имени региона и списка имен смежных регионов.
% Например, A::[B,D] означает, что регион A имеет границы с регионами B и D.
% В предикате раскрасить_карту(Карта,Список_цветов) список Карта обрабатывается
% в порядке следования элементов. При этом цвет очередного региона выбирается
% с помощью предиката select (удалить) из Списка_цветов, а цвета сопряженных
% регионов назначаются из списка оставшихся цветов. Назначенные цвета действуют в
% пределах всего списка Карта, и тем самым ограничивают области определения
% сопряженных переменных. Что приводит в конечном итоге к сокращению ветвей
% дерева  поиска
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%объявление инфиксного оператора
:- op(100,xfy,'::').

раскрасить_карту([Регион::Список_смежных_регионов|Ост_Регионы],Список_цветов):-
  присвоить_цвет(Регион,Список_цветов,Оставшиеся_цвета),
  присвоить_цвета_смежным_регионам(Список_смежных_регионов,Оставшиеся_цвета),
  раскрасить_карту(Ост_Регионы,Список_цветов).
раскрасить_карту([],_).

присвоить_цвет(Регион,Список_цветов,Оставшиеся_цвета):-
      select(Регион,Список_цветов,Оставшиеся_цвета).
присвоить_цвета_смежным_регионам(Список_смежных_регионов,Оставшиеся_цвета):-
      members(Список_смежных_регионов,Оставшиеся_цвета).

% members(L1,L2)проверяет вхождение эл-тов из списка L1 в список L2
% Если L1 не конкретизирован, а L2 конкретизирован, то members будет
% наоборот назначать значения элементам списка L1, выбирая их из L2
members([X|Xs],Ys):-member(X,Ys), members(Xs,Ys).
members([],Ys).

 /*--------------------------------------------------------------------------------
Пример вызова:
:-time(повторять(раскрасить_карту([A::[B,D], B::[C,D], C::[D,E], D::[E]],
                 [red, green, blue]),1000)).
Результат:
% 72,105 inferences, 0.031 CPU in 0.023 seconds (136% CPU, 2311043 Lips)
A = C, C = red,
B = E, E = green,
D = blue
--------------------------------------------------------------------------------*/
```

Б.2. Сравнение решений логической задачи

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Поиск решения логической задачи с ограничения типа Alldiff
%-----------------------------------------------------------------------------------------
%Три друга заняли 1-ое, 2-ое и 3-е места на чемпионате университета.
%Они имеют разные имена: Майкл,Ричард,Саймон.
%Они разных национальностей: американец,австралиец,израильтянин.
%Любят разные виды спорта: баскетбол,крикет,теннис.

%Ограничения:
%   1)Майкл предпочитает баскетбол и играет лучше, чем американец.
%   2)Саймон - израильтянин и играет лучше теннисиста.
%   3)Игрок в крикет занял первое место.
%Решение представляется в виде списка, упорядоченного в соответствии с местами.
%Каждый элемент списка - структура из трех компонент: Имя::Национальность::Спорт,
%где "::" инфиксный оператор, объявляемый в программе.
%Задача: Определить место, имя, национальность, вид спорта для каждого из друзей.
%-----------------------------------------------------------------------------------------
%Для сравнения вариантов решений следует оценить время выполнения целей:
% 1) :-time(повторять(решить1(X),1000)).
% 2) :-time(повторять(решить2(X),1000)).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%объявление инфиксного оператора
:- op(100,xfy,'::').

%Вариант 1: генерация решения , а потом проверка выполнения ограничений
решить1(X) :-генерировать_решение(X), проверить_ограничения(X).

%Вариант 2: проверка выполнения ограничений, а потом генерация решения
решить2(X) :-проверить_ограничения(X), генерировать_решение(X).

%формирование очередного кандидата в решения
%в виде списка [P1::N1::S1, P2::N2::S2, P3::N3::S3]
%Решение-кандидат формируется с помощью  перестановок
%имен P, национальностей N и видов спорта S. Всего  6*6*6=218 сочетаний
генерировать_решение([P1::N1::S1, P2::N2::S2, P3::N3::S3]):-
  перестановка([P1,P2,P3],[майкл,ричард,саймон]),            %перестановки имен
  перестановка([N1,N2,N3],[американец,австралиец,израильтянин]),  %национальностей
  перестановка([S1,S2,S3],[баскетбол,крикет,теннис]).            %видов спорта

%проверка выполнения ограничений
проверить_ограничения(Решение) :-
      Решение = [X1,X2,X3],
      member(майкл::_::баскетбол,Решение),                  %ограничение 1
      предшествует(майкл::_::_,_::американец::_,Решение),   %ограничение 1
      member(саймон::израильтянин::_,Решение),              %ограничение 2
      предшествует(саймон::_::_,_::_::теннис,Решение),      %ограничение 2
      X1 = _::_::крикет.                                    %ограничение 3

%========предикаты обработки списков========================================

%отношение предшествует(X,Y,L) верно, если  X следует в списке L  раньше  Y
предшествует(X,Y,[X|T]):- member(Y,T).
предшествует(X,Y,[_|T]):- предшествует(X,Y,T).

%перестановка элементов списка
%в начале находим перестановку L1 для хвоста списка L, а
%затем выполняем вставку головы списка H в произвольную позицию L1
перестановка([],[]).
перестановка([X|L],P):-перестановка(L,L1),вставить(X,L1,P).

%вставка элемента X в список L1
%реализована через удаление X из рез. списка L2
вставить(X,L1,L2):-удалить(X,L2,L1).

%удаление элемента списка: удалить(X,L,L1), где L1- это L без X
```

```
удалить(X,[X|T],T).
удалить(X,[H|T],[H|T1]):-удалить(X,T,T1).

%==============вспомогательный предикат=======================================
% цикл повторения выполнения Цели заданное число раз (N)
повторять(Цель,1):-Цель.
повторять(Цель,N):-
      not(not(Цель)),                %стирание предыдущих подстановок
      M is N-1,повторять(Цель,M).
/*-------------------------------------------------------------------------------
 Примеры вызовов:

:-time(повторять(решить1(X),1000)).
% 1,179,051 inferences, 0.312 CPU in 0.324 seconds (96% CPU, 3778985 Lips)
X = [саймон::израильтянин::крикет,
     майкл::австралиец::баскетбол,
     ричард::американец::теннис]

:- time(повторять(решить2(X),1000)).
% 110,051 inferences, 0.047 CPU in 0.049 seconds (96% CPU, 2351502 Lips)
X = [саймон::израильтянин::крикет,
     майкл::австралиец::баскетбол,
     ричард::американец::теннис]

Второй вариант выполняется примерно в 6 раз быстрее!
 -------------------------------------------------------------------------------*/
```

Б.3. Решение задачи «Загадка Эйнштейна»

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Загадка Эйнштейна — известная логическая задача,
% по легенде созданная Альбертом Эйнштейном в годы его детства
%-------------------------------------------------------------------------------
%  На улице стоят пять домов.
%  Каждый из пяти домов окрашен в свой цвет,
%  а их жители — люди разных национальностей,
%  владеют разными животными, пьют разные напитки
%  и курят разные марки американских сигарет.
%-------------------------------------------------------------------------------
%  Ограничения:
%  1. Англичанин живёт в красном доме;
%  2. У испанца есть собака;
%  3. В зелёном доме пьют кофе;
%  4. Русский пьёт чай;
%  5. Зелёный дом стоит сразу справа от белого дома;
%  6. Тот, кто курит Old Gold, разводит улиток;
%  7. В жёлтом доме курят Kools;
%  8. В центральном доме пьют молоко;
%  9. Норвежец живёт в первом доме;
%  10. Сосед того, кто курит Chesterfield, держит лису;
%  11. В доме по соседству с тем, в котором держат лошадь, курят Kools;
%  12. Тот, кто курит Lucky Strike, пьёт апельсиновый сок;
%  13. Японец курит Parliament;
%  14. Норвежец живёт рядом с синим домом.
%  Вопрос:
%  Кто пьёт воду? Кто держит зебру?
%-------------------------------------------------------------------------------
%  Будем представлять решение в виде списка, состоящего из 5 подсписков
%  (по количеству домов): Реш =[[1,N1,C1,P1,D1,S1],…,[5,N5,C5,P5,D5,S5]],
%  где цифра обозначает номер дома, N - национальность,  C- цвет дома,
%  P - животное, D - напиток, S - сигареты
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


эйнштейн:-
      эйнштейн(Решение),
      вывод_табл(Решение).

эйнштейн(Реш) :-
      Реш =[[1,N1,C1,P1,D1,S1],
            [2,N2,C2,P2,D2,S2],
            [3,N3,C3,P3,D3,S3],
            [4,N4,C4,P4,D4,S4],
            [5,N5,C5,P5,D5,S5]],

% 1. Англичанин живет в красном доме
  member([_,англичанин,красный,_,_,_], Реш),
% 2. У испанца есть собака
  member([_,испанец,_,собака,_,_], Реш),
% 3. В зеленом доме пьют кофе
  member([_,_,зеленый,_,кофе,_], Реш),
% 4. Русский пьет чай
  member([_,русский,_,_,чай,_], Реш),
% 5. Зеленый дом стоит сразу справа от белого дома
  member([БД,_,белый,_,_,_], Реш), member([ЗД,_,зеленый,_,_,_], Реш),
  ЗД =:= БД + 1,
% 6. Тот, кто курит Old Gold, разводит улиток
  member([_,_,_,улитки,_,old_gold], Реш),
% 7. В желтом доме курят Kools
  member([_,_,желтый,_,_,kools], Реш),
% 8. В центральном доме пьют молоко
  member([3,_,_,_,молоко,_], Реш),
% 9. Норвежец живет в первом доме
  member([1,норвежец,_,_,_,_], Реш),
% 10.Сосед того, кто курит Chesterfield, держит лису
  member([СД,_,_,_,_,chesterfield], Реш), member([ЛД,_,_,лиса,_,_], Реш),
```

```
  (ЛД =:= СД + 1; ЛД =:= СД - 1),
% 11.В доме по соседству с тем, в котором держат лошадь, курят Kools
  member([КД,_,_,_,_,kools], Реш), member([ЛоД,_,_,лошадь,_,_], Реш),
  (ЛоД =:= КД + 1; ЛоД =:= КД - 1),
% 12.Тот, кто курит Lucky Strike, пьёт апельсиновый сок
  member([_,_,_,_,сок,luckystrike], Реш),
% 13.Японец курит Parliament
  member([_,японец,_,_,_,parliament], Реш),
% 14.Норвежец живёт рядом с синим домом
  member([НД,норвежец,_,_,_,_], Реш), member([СД,_,синий,_,_,_], Реш),
  (НД =:= СД + 1; НД =:= СД - 1),
перестановка([норвежец,русский,англичанин,японец,испанец],[N1,N2,N3,N4,N5]),
перестановка([желтый,синий,красный,зеленый,белый],[C1,C2,C3,C4,C5]),
перестановка([лиса,лошадь,улитки,зебра,собака],[P1,P2,P3,P4,P5]),
перестановка([сок,чай,молоко,кофе,вода],[D1,D2,D3,D4,D5]),
перестановка([kools,old_gold,chesterfield,luckystrike,parliament],[S1,S2,S3,S4,S5]).

% вывод решения в форме таблицы
вывод_табл([A,B,C,D,E]) :-
  H=['N','Национальн.','Цвет дома','Животное','Напиток','Сигареты'],
  write('+--+------------+----------+----------+----------+-------------+'),nl,
  writef('|%2L|%12L|%10L|%10L|%10L|%13L|',H),nl,
  write('+--+------------+----------+----------+----------+-------------+'),nl,
  writef('|%2L|%12L|%10L|%10L|%10L|%13L|',A),nl,
  writef('|%2L|%12L|%10L|%10L|%10L|%13L|',B),nl,
  writef('|%2L|%12L|%10L|%10L|%10L|%13L|',C),nl,
  writef('|%2L|%12L|%10L|%10L|%10L|%13L|',D),nl,
  writef('|%2L|%12L|%10L|%10L|%10L|%13L|',E),nl,
  write('+--+------------+----------+----------+----------+-------------+'),nl.

/* ----------------Решение-----------------------------------------------------
Вызов:
  :-эйнштейн.
Результат:
+--+------------+----------+----------+----------+-------------+
|N |Национальн. |Цвет дома |Животное  |Напиток   |Сигареты     |
+--+------------+----------+----------+----------+-------------+
|1 |норвежец    |желтый    |лиса      |вода      |kools        |
|2 |русский     |синий     |лошадь    |чай       |chesterfield |
|3 |англичанин  |красный   |улитки    |молоко    |old_gold     |
|4 |испанец     |белый     |собака    |сок       |luckystrike  |
|5 |японец      |зеленый   |зебра     |кофе      |parliament   |
+--+------------+----------+----------+----------+-------------+

-------------------------------------------------------------------------------*/
```