Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра теоретической и прикладной информатики

Лабораторная работа №2

по предмету

«Введение в искусственный интеллект и логическое программирование**»**

Факультет: прикладной математики и информатики

Группа: ПМИ-12

Бригада: 4

Студенты: Михайловский М.А.

Швадченко А.В

Преподаватель: Целебровская М. Ю.

Новосибирск, 2024

**1. Задание**

Разработать справочно-информационую систему авиакомпании.  
Система должна содержать базу данных со следующей информацией:  
номер рейса, пункт отправления, пункт назначения, тип самолета, время отправления, время прибытия, дни выполнения рейсов, цена билета.  
Система должна подбирать рейсы (с учетом стыковок рейсов в течение одних суток) по минимальной стоимости билетов.

**2.Листинг разработанной базы знаний на языке ПРОЛОГ**

predicates

nondeterm flight(symbol, symbol, symbol, symbol, real, real, integer)

nondeterm connection(symbol, symbol, integer, integer)

nondeterm connection2(symbol, symbol, integer, symbol, symbol, integer)

nondeterm cheapest\_flight(symbol, symbol, integer, integer)

nondeterm cheapest\_flight2(symbol, symbol, integer, integer)

nondeterm cheaper\_connection2(symbol, symbol, integer, integer, integer)

nondeterm cheaper\_connection(symbol, symbol, integer, integer, integer)

nondeterm day(symbol, integer)

clauses

flight(flight1, cityA, cityB, type1, 10.00, 12.12, 1000).

flight(flight2, cityB, cityC, type2, 11.10, 13.00, 1500).

flight(flight3, cityA, cityC, type3, 14.00, 23.00, 6000).

flight(flight4, cityA, cityB, type4, 15.00, 16.00, 2500).

flight(flight5, cityB, cityC, type5, 17.00, 18.00, 3000).

day(flight1, 1).

day(flight1, 2).

day(flight3, 2).

day(flight5, 2).

connection(X, Y, Cost, D) :-

flight(Fl, X, Y, \_, \_, \_, Cost), day(Fl, D).

connection2(X, Y, Cost, Flight1, Flight2,D1) :-

flight(Flight1, X, Z, \_, \_, Departure1, Cost1),

flight(Flight2, Z, Y, \_, Arrival1, \_, Cost2),

Departure1 < Arrival1,

day(Flight1, D1),

day(Flight2, D2), D1-D2=0,

connection(Z, Y, Cost2, \_),

Cost = Cost1 + Cost2 + 999.

connection2(X, Y, Cost, Flight1, Flight2,D1) :-

flight(Flight1, X, Z, \_, \_, Departure1, Cost1),

flight(Flight2, Z, Y, \_, Arrival1, \_, Cost2),

Departure1 < Arrival1,

day(Flight1, D1),

day(Flight2, D2), D1-D2=0,

connection2(Z, Y, Cost2,\_,\_,\_),

Cost = Cost1 + Cost2 + 999.

connection2(X, Y, Cost, Flight1, Flight2,D1) :-

flight(Flight1, X, Z, \_, \_, \_, Cost1),

flight(Flight2, Z, Y, \_, \_, \_, Cost2),

day(Flight1, D1),

day(Flight2, D2), abs(D1-D2)=1,

connection(Z, Y, Cost2, \_),

Cost = Cost1 + Cost2 + 999.

connection2(X, Y, Cost, Flight1, Flight2,D1) :-

flight(Flight1, X, Z, \_, \_, \_, Cost1),

flight(Flight2, Z, Y, \_, \_, \_, Cost2),

day(Flight1, D1),

day(Flight2, D2), abs(D1-D2)=1,

connection2(Z, Y, Cost2,\_,\_,\_),

Cost = Cost1 + Cost2 + 999.

cheaper\_connection2(X, Y, Cost, OtherCost, D) :-

connection2(X, Y, OtherCost, \_,\_, D),

OtherCost < Cost.

cheaper\_connection(X, Y, Cost, OtherCost, D) :-

connection(X, Y, OtherCost, D),

OtherCost < Cost.

cheapest\_flight(X, Y, Cost, D) :-

connection(X, Y, Cost,D),

not(cheaper\_connection(X, Y, Cost, \_, \_)).

cheapest\_flight2(X, Y, Cost, D) :-

connection2(X, Y, Cost,\_,\_,D),

not(cheaper\_connection2(X, Y, Cost, \_, \_)).

**3.Результаты выполнения**

|  |  |
| --- | --- |
| Запрос | Результат |
| cheapest\_flight2(cityA, cityC, Cost, D),connection2(cityA, cityC,C, F1, F2, D). |  |
| cheapest\_flight(cityA, cityB, Cost,D),flight(Fl, cityA, cityB, \_, \_, \_, Cost). |  |
| flight(Fl, cityA, cityB, \_, \_, \_, Cost). |  |
| cheapest\_flight2(cityA, cityC, C, D). |  |

**4.Описать декларативный смысл фактов и предложений разрабатываемой программы.**

* flight(flight1, cityA, cityB, type, time1, time2, cost).– предикат описывающий рейс: номер рейса(flight1), город отправления(cityA), город назначения(cityB), тип самолёта(type), время отправления (time1), время прибытия(time2), цена(cost).
* day(flight1, 1). - предикат описывающий дни, в которые летает рейс.

**5.Описать декларативный смысл целей,** **привести деревья поиска решения для поставленных целей.**

* connection(X, Y, Cost, D)- возвращает рейсы без пересадок, где X - город отправления, Y - город назначения, D – день, Cost – цена;
* connection2(X, Y, Cost, Flight1, Flight2,D1) - возвращает рейсы c пересадками, где X - город отправления, Y - город назначения, D1 – день вылета, Cost – цена, Flight1, Flight2-номера рейсов;
* cheaper\_connection2(X, Y, Cost, OtherCost, D) – ищет лучшую цену среди рейсов с пересадками, где X - город отправления, Y - город назначения, D – день вылета, Cost – цена, OtherCost-наименьшая цена перелёта;
* cheaper\_connection(X, Y, Cost, OtherCost, D) - ищет лучшую цену среди рейсов без пересадок, где X - город отправления, Y - город назначения, D – день вылета, Cost – цена, OtherCost-наименьшая цена перелёта;
* cheapest\_flight(X, Y, Cost, D) - возвращает рейс без пересадок с лучшей ценой, где X - город отправления, Y - город назначения, D – день вылета, Cost – цена;
* cheapest\_flight2(X, Y, Cost, D) - возвращает рейс с пересадками с лучшей ценой, где X - город отправления, Y - город назначения, D – день вылета, Cost – цена;

cheapest\_flight2(cityA, cityC, C, D).

connection2 (X, Y, Cost,\_,\_,D)

{ cityA/ cityB,}

{ cityB/ cityC,}

{ cityA/ cityB,}

{ cityB/ cityC,}

{ cityA/ cityC,}

connection2(cityA, cityC, Cost, Flight1, Flight2,D1)

False

flight(Flight1, X, Z, \_, \_, Departure1, Cost1),

X={ cityA}

No

Z={cityA}

flight(Flight1, cityA, Z, \_, \_, Departure1, Cost1),

Z={cityB}

{cityC}

flight(Flight1, cityA, cityB, \_, \_, Departure1, Cost1),

flight(Flight1, cityA, cityC, \_, \_, Departure1, Cost1),

flight(Flight2, Z, Y \_, \_, Departure1, Cost1),

Y={ cityC}

flight(Flight1, Z, cityC, \_, \_, Departure1, Cost1),

Z={cityB}

{cityA}

Z={cityC}

No

flight(Flight1, Z, cityC, \_, \_, Departure1, Cost1),

Z={cityB}

Z={cityC}

{cityA}

day(Flight1, D1)

No

Flight1 = flight1

Cost = 1500

Cost = 3000

flight(flight2, cityB, cityC, \_, \_, \_, Cost)

flight(flight5, cityB, cityC, \_, \_, \_, Cost)

Fl = flight2

flight5

No

Fl = flight1

flight3

flight4

Z={cityB}

flight(Fl, cityB, cityC, \_, \_, \_, Cost)

Y={ cityC}

flight(Fl, Z, Y, \_, \_, \_, Cost)

connection(Z, Y, Cost2, \_)

2-2=0

D1 = 2

D2 = 2

D1-D2=0

day(flight2, 2)

D2 = 2

day(flight2, D2)

Flight2 = flight2

day(Flight2, D2)

day(flight1, 1)

day(flight1, 1)

day(flight1, D1)

not(cheaper\_connection2(cityA, cityC, 3499, \_, \_))

not(cheaper\_connection2(cityA, cityC, 5499, \_, \_))

X=cityA, Y = cityC, Cost = 3499

X=cityA, Y = cityC, Cost = 5499

not(cheaper\_connection2(X, Y, Cost, \_, \_))

cheapest\_flight2(cityA, cityC, 3499, 2)

cheapest\_flight2(cityA, cityC, 5499, 2)

X=cityA, Y = cityC, Cost = 3499 D = 2

X=cityA, Y = cityC, Cost = 5499, D = 2

cheapest\_flight2(X, Y, Cost, D)

Cost = 3499

Cost = 5499

Cost1 = 1000

Cost2 = 1500

Cost1 = 3000

Cost2 = 1500

Cost = Cost1 + Cost2 + 999

Cost = 1500

Cost = 3000

flight(flight2, cityB, cityC, \_, \_, \_, 1500)

flight(flight5, cityB, cityC, \_, \_, \_, 3000)

Yes

No

X=5499

**X=3499**

not(cheaper\_connection2(cityA, cityC, X, \_, \_))

not(cheaper\_connection2(cityA, cityC, X, \_, \_))

OtherCost = 3499

Cost = 5499

Yes

No

OtherCost = 5499

Cost = 3499

OtherCost<Cost

connection2(cityA, cityC, 3499, \_,\_, D),

connection2(cityA, cityC, 5499, \_,\_, D),

X=cityA, Y = cityC, OtherCost = 3499

X=cityA, Y = cityC, OtherCost = 5499

connection2(X, Y, OtherCost, \_,\_, D),