Chapter 2

Пространство на състоянията и търсене

2.1 Лекции

Решаването на всяка задача е свързано с действия, които последователно променят предметната област (света) до достигане на целевото и състояние (решението на зазачата). За това е необходимо формално представяне на състоянията на света. Начини за реализация:

- Таблица от двойки "състояние-действие", пълно описващи всички възможности и съответното правилно действие.

- Използуване на знания за моделиране на състоянията на света и за начина, по който действията биха ги променили (преходите от едно в друго състойние) – *симулиране на изпълнението на действия с цел намиране на последователност, която би довела до целевото състояние.* Това обикновено се прави чрез *търсене в пространството на състоянията.*

2.1.1 Фармализация на задачата

- Състояния (допустими, начално)

- Оператори

- Целево състояние или тест за проверка на целевото състояние

2.1.2 Задача

- Дадено: начално и целево състояние

- Търси се: последователност от оператори (път), които водят от началното до целевото състояние.

2.1.3 Решаване на задачата чрез търсене

- Разширение на състояние (възел).

- Функция за избор на наследник.

- Дърво на търсене.

- Алгоритми: слепи (пълно изчерпване, uninfromed, blind, exhaustive) и евристични (informed).

- Оценка на алгоритмите: цена на пътя и цена (сложност) на търсенето.

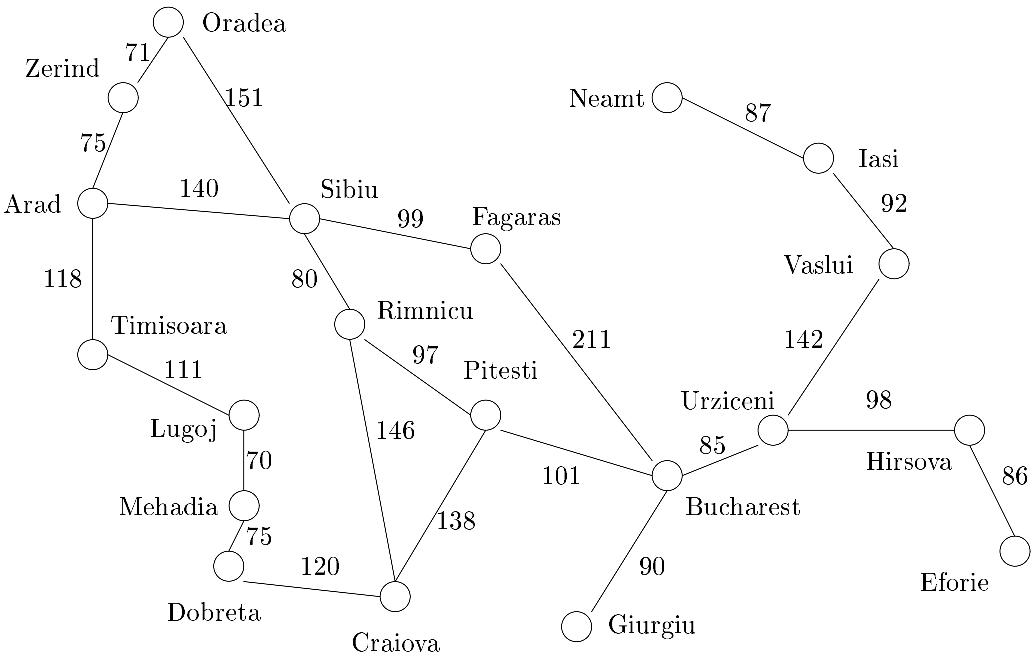

Figure 2.1: Пътна карта на Румъния

2.1.4 Пример 1: Търсене на път между два града по карта

```
path(X,Y,W) :-    % От X до Y има път
    arc(X,Y,W).   % ако от X до Y има директен път (дъга в графа).
path(X,Y,W) :-    % От X до Y има път
    arc(X,Z,W1),  % ако от X до Z има дъга и
    path(Z,Y,W2), % от Z до Y има път.
    W is W1+W2.
```

Проблеми:

- Повторение на състояния (зацикляне) – маркиране на посетените състояния. Предикати за работа с базата данни на Пролог.

```
path(X,Y,W) :-
    arc(X,Z,W1),             % Намери ново състояние Z.
    \+ clause(passed(Z),_), % Ако Z не е маркирано,
    assertz(passed(Z)),      % то маркирай Z и
    path(Z,Y,W2),            % продължи.
    W is W1+W2.
```

- Показване на пътя – използуване на базата данни.

```
?- listing(passed).
```

- Намиране на най-краткия път – избор на подходяща стратегия за търсене.

2.1.5 Пример 2: Задача за фермера, вълка, козата и зелето

```
Състояние = {B,W,G,C}, където (s - южен бряг, n - северен бряг):
    B - положение на лодката (s/n);
    W - положение на вълка (s/n);
    G - положение на козата (s/n);
    C - положение на зелето (s/n);
```

```
change_state(s,W,G,C,n,W,G,C).
change_state(n,W,G,C,s,W,G,C).
change_state(s,s,G,C,n,n,G,C).
change_state(n,n,G,C,s,s,G,C).
change_state(s,W,s,C,n,W,n,C).
change_state(n,W,n,C,s,W,s,C).
change_state(s,W,G,s,n,W,G,n).
change_state(n,W,G,n,s,W,G,s).

illegal(s,n,n,_).
illegal(n,s,s,_).
illegal(s,_,n,n).
illegal(n,_,s,s).

solve(s,s,s,s).
solve(B,W,G,C) :-
    change_state(B,W,G,C,B1,W1,G1,C1),
    \+ illegal(B1,W1,G1,C1),
    \+ clause(passed(B1,W1,G1,C1),_),
    assertz(passed(B1,W1,G1,C1)),
    solve(B1,W1,G1,C1).

?- solve(n,n,n,n).    % Решаване на задчата
?- listing(passed).   % Показване на решението
```

2.2 Семинарни упражнения

- Списъци в Пролог – представяне чрез вложени термове и достъп чрез унификация:
  - $[X|Y] = [1,2,3] \rightarrow \{X/1, Y/[2,3]\}$
  - $[X,Y|Z] = [1,2,3] \rightarrow \{X/1, Y/2, Z,[3]\}$
  - $[X|Y] = [1] \rightarrow \{X/1, Y/[]\}$
- Рекурсивна обработка на списъци – предикатите `member` и `append`.
- Два начина на използуване на списъците при рекурсивна обработка:
  - "сглобяване" на списъка при връщане от рекурсията:
  - списък с натрупване (акумулатор) и предаване на стойността от дъното на рекурсията с допълнителна променлива (R) с начална стойност []:

```
no_double1([],[]).
no_double1([X|T],V) :-
    member(X,T), !,
    no_double1(T,V).
no_double1([X|T],[X|V]) :-
   no_double1(T,V).
```

```
no_double2([],R,R).
no_double2([X|T],L,R) :-
    member(X,L), !,
    no_double2(T,L,R).
no_double2([X|T],L,R) :-
   no_double2(T,[X|L],R).
```

2.3 Лабораторни упражнения

- Експеримерти с предиката `path` – работа с базата данни на Пролог (`assert` и `retract`) и намиране на алтернативни решения.
- Решаване на задачата за фермера, вълка, козата и зелето (предиката `solve`).
- Модификация на предикатите `path` и `solve` за работа със списъци - избягване на зациляне (използуване на списък с натрупване) и показване на решението.