BÁO CÁO THỰC HÀNH NHẬP MÔN TRÍ TUỆ NHÂN TẠO

Tuần 8

Bài toán. Cho 2 bình rỗng X, Y có thể tích lần lượt là V_x, V_y. Dùng 2 bình này để đong ra z lít nước.

(a) Xây dựng cơ sở tri thức để giải bài toán.

(b) Cài đặt tri thức trên máy tính để giải quyết bài toán.

(c) Viết chương trình minh hoạ để giải bài toán.

Lời giải

(a) Phát biểu bài toán theo dạng hình thức:

- Giả sử $V_x < V_y$. Gọi lượng nước chứa trong bình X là x ($0 \leq x \leq V_x$). Lượng nước chứa trong bình Y là y ($0 \leq y \leq V_y$).
- Điều kiện kết thúc bài toán là $x = z$ hoặc $y = z$.
- Điều kiện đầu tiên của bài toán là $x = 0$ và $y = 0$.
- Quá trình giải được thực hiện bằng cách xét lần lượt các luật, luật nào thoả mãn thì được áp dụng. Lúc này các luật chính là các "kinh nghiệm" hay tri thức mà ta đã chuyển giao cho máy tính.

(b) Cài đặt tri thức trên máy tính để giải quyết bài toán: x = 0; y = 0;

```
WHILE ((x = Vx) AND (y <> z)) DO
BEGIN
IF (x = Vx) THEN x = 0;
IF (y = 0) THEN y = Vy;
IF (y > 0) THEN
    BEGIN
    K = MIN(Vx-x, y);
    x = x + k;
    y = y - k;
    END
END
```

(c) Chương trình minh hoạ

```
:- dynamic visited_state/5.  /* Create a database */

min(X, Y, K) :- X > Y, K is Y.
min(X, Y, K) :- X =< Y, K is X.

input(Vx, Vy, Z) :-
    X = 0, Y = 0,
        /* Remove all clauses in database before we begin */
        retractall(visited_state(_,_,_,_,_)),
        state(X, Y, Vx, Vy, Z).

/* if Z > max(Vx, Vy) then the problem has no solution */
state(_, _, Vx, Vy, Z) :-
        Z > Vx, Z > Vy,
        write('No solution!').

state(Z, _, _, _, Z) :- write('Goal state').
state(_, Z, _, _, Z) :- write('Goal state').

state(X, Y, Vx, Vy, Z) :-
        Y = 0,
        New_Y is Vy,
        not(visited_state(X, New_Y, Vx, Vy, Z)),
        assertz(visited_state(X, Y, Vx, Vy, Z)),
        format('Pour ~d lit water into jug Y: (~d, ~d).', [Vy, X, New_Y]),
    nl,
        state(X, New_Y, Vx, Vy, Z).

state(X, Y, Vx, Vy, Z) :-
        X = Vx,
        New_X is 0,
        not(visited_state(New_X, Y, Vx, Vy, Z)),
        assertz(visited_state(X, Y, Vx, Vy, Z)),
        format('Pour ~d lit water of jug X out: (~d, ~d).', [Vx, New_X, Y]),
        nl,
```

```
        state(New_X, Y, Vx, Vy, Z).

state(X, Y, Vx, Vy, Z) :-
        not(Y = 0), X < Vx,
        min(Y, Vx - X, K), New_X is X + K, New_Y is Y - K,
        not(visited_state(New_X, New_Y, Vx, Vy, Z)),
        assertz(visited_state(X, Y, Vx, Vy, Z)),
        format('Pour ~d lit water from jug Y to jug X: (~d, ~d)', [K, New_X, New_Y]),
        nl,
        state(New_X, New_Y, Vx, Vy, Z).

/*
For example, we start a program by using the query:
input(7, 4, 2).
We will receive the result.
*** Note that: z <= max(Vx, Vy).
*/
```