Artificial Intelligence

Assignment 5

Claudia Schon

schon@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Submission until: 17.06.2022, 7:00 a.m.
Tutorial on: 17.06.2022

GROUP HOLLERITH - SOLUTION

Group Members:

1. **Saborni Shernaj Binte Elahi (220202426) – (saborni@uni-koblenz.de)**
2. **M Rashedul Hasnat (220202415) – (rhasnat@uni-koblenz.de)**
3. **Kamrun Nahar (220202410) – (nahar@uni-koblenz.de)**
4. **Basitur Rahman Chowdhury (218100976) – (bchowdhury@uni-koblenz.de)**

1 Search problems (100 Points)

Encode the following problems as search problems analogous to the lecture. Define what a state is, the transition function as well as the initial state and set of goal state.

a) You have three containers that you can fill with 12, 8 and 3 liters of water respectively. Initially, all three containers are empty. Additionally, you have an infinite supply of water. You can fill up a container, put the water in one container into the other or throw it away. Your task is get exactly 1 liter of water into one of the containers.

b) Three missionaries and three cannibals come to a river. They are on the right side of the river and want to cross onto the left side. A rowboat that seats two is available. At least one person is necessary to row the boat to the other side. If the cannibals outnumber the missionaries on either bank of the river, the missionaries will be eaten. How shall they cross the river?

c) Four gnomes are (G_1, G_2, G_3, G_4) in the beginning (right side) of a wobbly bridge with four gnomes. Its dark and the group only has one torch. The task is to get everyone to the other side (left side). At most two gnomes can cross the bridge together. They need the torch in order to cross the bridge. The torch burns for 60 minutes. G_1 needs 5 minutes to get across, G_2 needs 10, G_3 needs 20 and G_4 needs 25 minutes. If two gnomes cross the bridge together, they need as much time as the slowest of them.

SOLUTION:

1.a:

The state space for this problem can be described as the set of ordered integers (x,y,z)

Where, x represents the quantity of water in the 12 liter container x=0,1,2,3,4,5,6,7,8,9,10,11,12

y represents the quantity of water in the 8 liter container y=0,1,2,3,4,5,6,7,8

z represents the quantity of water in the 3 liter container z=0,1,2,3

Start state: (0,0,0)

Goal state: (1,y,z) or (x,1,z) or (x,y,1)

We can perform 3 operations to reach our goal.

1. Fill water container
2. Empty water container
3. Transfer water container

Rule	State	Process
1	(x, y, z | x<12)	(12, y, z) Fill 12 liter container
2	(x, y, z | y<8)	(x, 8, z) Fill 8 liter container
3	(x, y, z | z<3)	(x, y, 3) Fill 3 liter container
4	(x, y, z | x>0)	(0, y, z) Empty 12 liter container
5	(x, y, z | y>0)	(x, 0, z) Empty 8 liter container
6	(x, y, z | z>0)	(x, y, 0) Empty 3 liter container
7	(x, y, z | x+y+z≥8 ∧ x>0)	(x-(8-y), 8, z) Pour water from 12 liter container into 8 liter container until 8 liter container is full
8	(x, y, z | x+y+z≥3 ∧ x>0)	(x-(3-z), y, 3) Pour water from 12 liter container into 3 liter container until 3 liter container is full
9	(x, y, z | x+y+z≥3 ∧ y>0)	(x, y-(3-z), 3) Pour water from 8 liter container into 3 liter container until 3 liter container is full
10	(x, y, z | x+y≤12 ∧ y>0)	(x+y, 0, z) Pour all water from 8 liter container to 12 liter container
11	(x, y, z | x+z≤12 ∧ z>0)	(x+z, y, 0) Pour all water from 3 liter container to 12 liter container
12	(x, y, z | y+z≤8 ∧ z>0)	(x, y+z, 0) Pour all water from 3 liter container to 8 liter container

Initialization:

Start state: (0,0,0)

Apply Rule 1:

Fill 12 liter container with water

Now the state is (12,y,z)

Iteration 1:

Current state: (12,y,z)

Apply Rule 7:

Pour water from 12 liter container to 8 liter container until 8 liter container is full

Now the state is (4,8,z)

Iteration 2:

Current state: (4,8,z)

Apply Rule 8:

Pour water from 12 liter container into 3 liter container until 3 liter container is full

Now the state is (1,8,3)

Goal Achieved

1.b:

States: triple (m, c, b) with 0 ≤ m, c, b ≤ 3, where m, c, and b represent the number of missionaries, cannibals and boats currently on the original bank.

Initial State: (3,3,1) & ({m, c, b}, r).

Successor function: From each state, either bring one missionary, one cannibal, two missionaries, two cannibals, or one of each type to the other bank.

Here, not all states are attainable (e.g. (0,0,1)), and some are illegal.

Goal State: (0,0,0)

States are tuples (R, ◦, D) with R ⊆ {m, c, b} and ◦ ∈ {r, l} and D ∈(|m ≥ c |, (l or r)).

Constraints: (|m ≥ c |, (l or r)).

Action: (b raid max ((c or m), (1 or 2))).

The set of goal states is as follows: {({}, l, D) | D ∈ (|m ≥ c |)}.

Stages:

0. Initial setup: _ & m, m, m, c, c, c, b.
1. Two cannibals cross over: b, c, c & m, m, m, c.
2. One comes back: c & m, m, m, c, c, b.
3. Two cannibals go over again: b, c, c, c & m, m, m.
4. One comes back: c, c & m, m, m, c, b.
5. Two missionaries cross: b, m, m, c, c & m, c.
6. A missionary and a cannibal return: m, c & m, m, c, c, b.
7. Two missionaries cross again: b, m, m, m, c & c, c.
8. A cannibal returns: m, m, m & c, c, c, b.
9. Two cannibals cross: b, m, m, m, c, c & c.
10. One cannibal return: m, m, m, c & c, c, b.
11. Bring over the third: m, m, m, c, c, c, b & _.

We can now encode the transition relation:

O = {((R, r, D), (R’, l, D’)) |X1, X2 ∈ R ∧ R’ = R \ {X_1, X_2} ∧ D’ = D ∧ (D’ > 0 in l)}

∪

{((R, l, D), (R’, r, D’)) |X1 ∈ {b, m, c} \ R ∧ R’ = {X_1} ∪ R∧ D’ = D ∧ (D’ > 0 in l)}.

1.c:

A state is a tuple S = (R, ∘, D) with R ⊆ {g1, g2, g3, g4}, ∘ ∈ {l, r} and D ∈ {0, ..., 60}.

Transitions which are valid, always send one or two gnomes across the bridge.

While sending gnomes from the right to the left side, it makes sense to always send two gnomes. Again, while returning from the left side to the right side, the most logical thing is to only send one gnome.

For this, it is required to have a function duration which representations the time needed for crossing the bridge:

$$\mathrm{duration}(W) = \begin{cases} 5 \text{ if } W = w1 \\ 10 \text{ if } W = w2 \\ 20 \text{ if } W = w3 \\ 25 \text{ if } W = w4 \end{cases}$$

We can now encode the transition relation:

O = {((R, r, D), (R’, l, D’)) |X1, X2 ∈ R ∧ R’ = R \ {X_1, X_2} ∧ D’ = D − max(duration(X_1), duration(X_2)) ∧ D’ ≥ 0}

∪

{((R, l, D), (R’, r, D’)) |X1 ∈ {w1, w2, w3, w4} \ R ∧ R’ = {X_1} ∪ R∧ D’ = D − duration(X_1)}.

The initial state is ({w1, w2, w3, w4}, r, 60).

The set of goal states is as follows: {({}, l, D) |D ≥ 0}.

2. RUN LENGTH ENCODING (OPTIONAL):

<u>Solution</u>:

Code:

```
encode (List, X):-
    findall([N, E], (bagof(true,member(N,List),Xs), length(Xs,E)), X).
```