Assignment 1 Search Algorithms

A screenshot of a computer

Description automatically generatedLeft tie breaking is used as the left most element, for example q4 is to the left of q14 because of the number being less.

1. Breadth-first search

* Order of expansion: S, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12, q13, q14, q15, q16 (goal)
* Found path: S, q4, q6, q7, q9, q13, q15, q16
* Path cost: 7

Breadth-first search (without checking for loops)

1. Depth-first search

* Order of expansion: S, q2, q3, q4, (q1, q2, q3, q4)... Infinite loop
* Found path: Ø
* Path cost: Ø

1. Uniform-cost search (without revisiting already visited nodes)

* Order of expansion: S, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12, q13, q14, q15, q16 (goal)
* Found path: S, q4, q6, q7, q9, q13, q15, q16
* Path cost: 7

## 1.2 Part two: informed search

A white background with black text

Description automatically generated



where w = wolf, g = goat and c = cabbage. Each of these three variables having the value of 0 (initial side) and 1 (other side).

So as an example, h(q2) =1+1+1=3 and h(q4) = 1+0+1 =2

1. An admissible search never overestimates the cost of reaching its goal (goal calculation using shortest path to goal from the node).

**h(n) <= h\*(n),** where h\*(n) is the true cost from our node to the goal.

To test if our heuristic function is admissible, we can check the extreme cases in our problem.

The maximum value for our true goal cost is 7 (from level 0 or start stage) and the minimum is 1 (from level 6 right above the last stage of our goal). Similarly let’s calculate it for the heuristic function as well. Minimum is 1 (excluding goal node) and maximum is 3.

If we check q15 (in stage 6) we see that its heuristic is 1, the cost to reach the goal from q15 is 1. Our equation 1 <= 1 holds true. If we try q13 in stage 5, we get 1 <= 2 for q13 and 2<=2 for q14. If stage 5 had nodes with a heuristic value of 3 or stage 6 had nodes with a heuristic value of 2 or 3 with paths that lead to the goal, our heuristic function would not have been admissible.

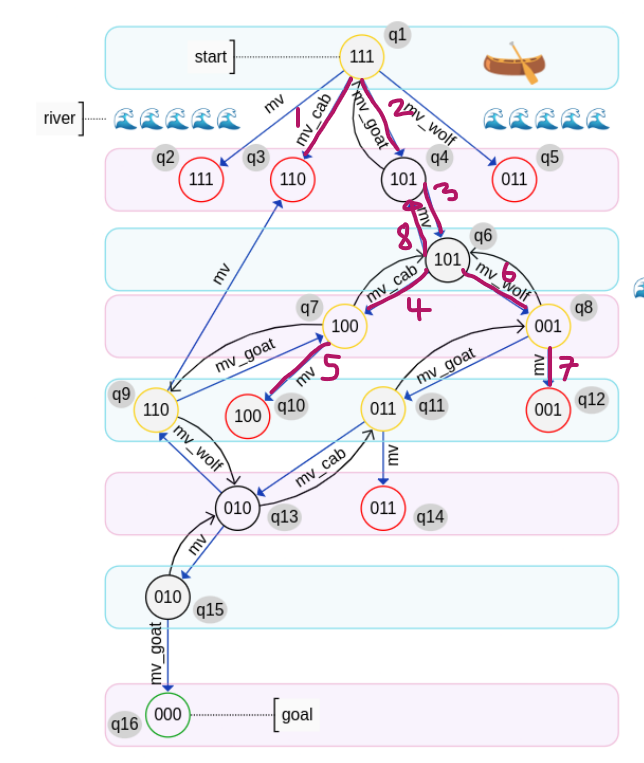
1. A consistent heuristic has the property + h(n’) where n are nodes and n’ are successor nodes of n. This must be true for every . Let’s evaluate this for our case. We know the cost from n to n’ is always 1 and we know that the maximum value for h(n’) is 3 (1+1+1) and the minimum for the node that isn’t the goal is 1. The maximum value for h(n) is also 3. Let’s check q1 and its neighbors to check if it holds for the first stage. For the first stage the equation becomes 3 <= 1 + (3,2), which becomes 3<= 1+2 (Satisfied for the first stage).

For the second stage let’s try q4 and its neighbors which becomes 2 <= 1 + . The same holds true for q3 and q5. Q4 becomes 2 <= 1+ , it still holds true. We continue to do this for each node and reach the conclusion that it is consistent which means it guarantees to find the shortest path. And proving this also proves that our heuristic function is admissible. All consistent heuristics are also admissible but not the other way around.

A close-up of a graph

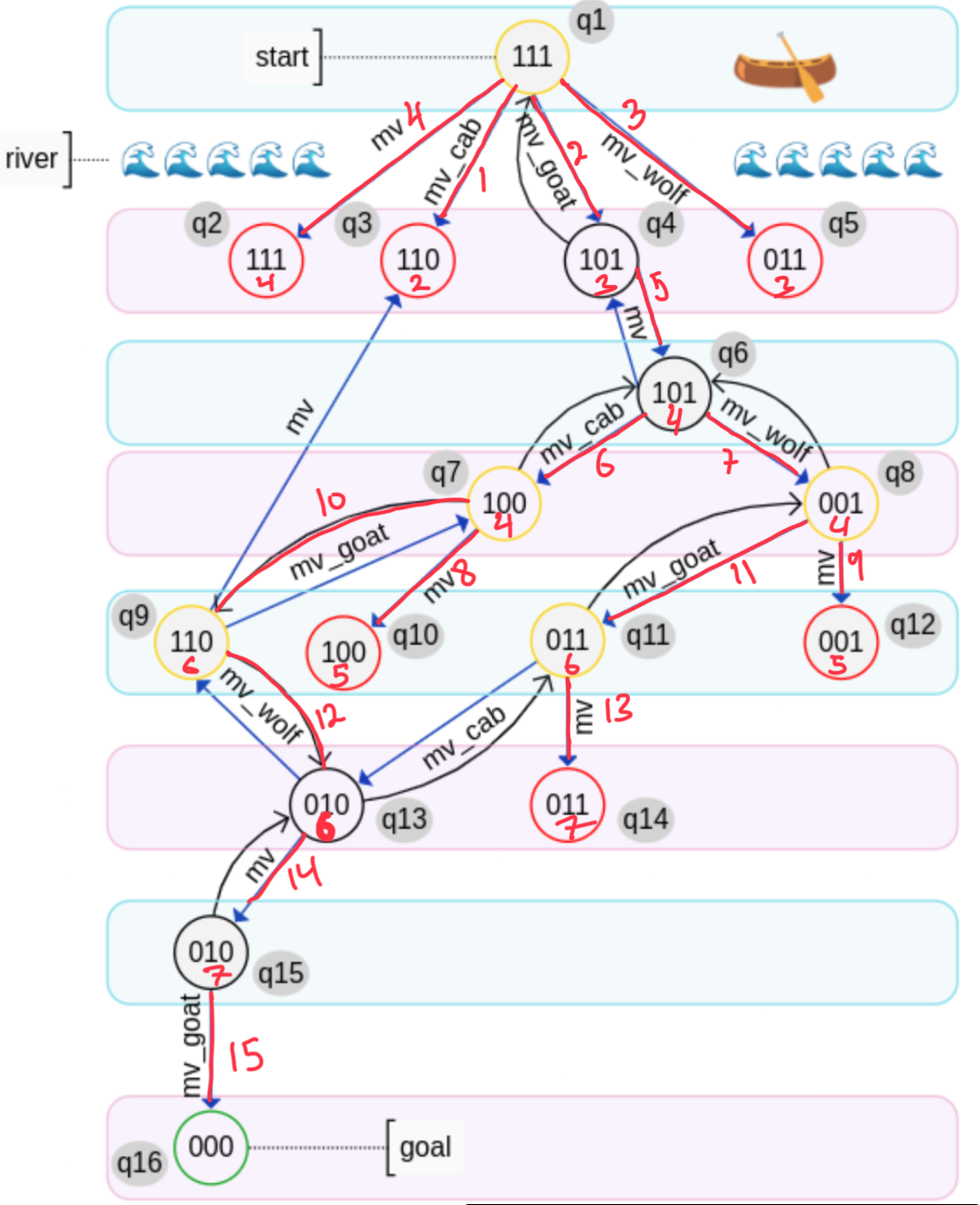
Description automatically generated

1. Greedy Best-First



* Order of expansion: S, q3, q4, q6, q7, q10, q8, q12, q4, (q6, q7, q10, q8, q12, q4) … infinite loop
* Found path: Ø
* Path cost: Ø

1. A\*



* Order of expansion: S, q3, q4, q5, q2, q6, q7, q8, q12, q9, q11, q13, q14, q15, q16
* Found path: S, q4, q6, q7, q9, q13, q15, q16
* Path cost: 7

# Sources

Wikipedia contributors. (2024, April 23). Consistent heuristic. In *Wikipedia, The Free Encyclopedia*. Retrieved 19:18, August 25, 2024, from <https://en.wikipedia.org/w/index.php?title=Consistent_heuristic&oldid=1220349845>