Hanwen Wang 260778557

No.

Date

the cat sat m
the cat sat on the
the cat on the
sc

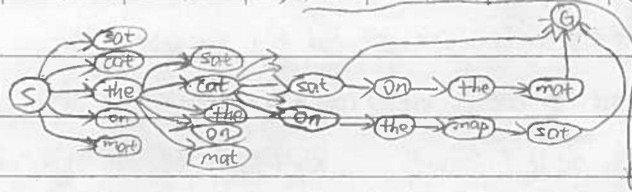
initial state: empty. goal state:

1. a) size: $(5C_1)^b = 15625$, state space: all possible configuration of words

Operator: add a word.
Path: add a word to an exist configuration

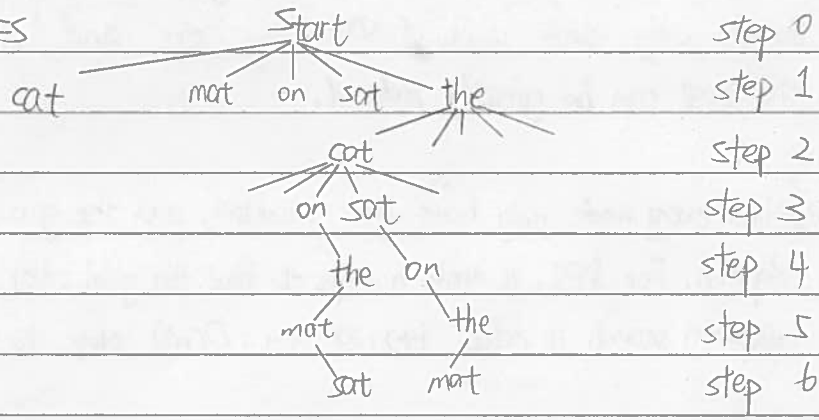
Path cost: cat 2, mat 2, on 1, sat 2, the 2

Optimal solution: the → cat → sa

b) 

In each node there are 5 options and only three states will lead to goal state. The search tree should be a tree-shaped graph with no loops, search graph can be random-shaped with loops, and in this graph, they all point to the same goal state at the end while they should all be separate in a tree graph.

c) i) BFS



ii) Uniform cost search: {(start, 0)} = {(on,1), (cat,2), (mat,2), (sat,2), (th

= {(cat,2), (mat,2), (sat,2), (the,2), (on on,2), (on cat,3), (on mat,3)

(on sat,3), (on the,3)} = ⋯ step 10: {(cat/mat on, 3)}, (on on on, 3),

(sat/the on, 3), (cat/mat cat/mat/sat/the, 4), (on on cat/mat/sat/the, 4), (sat/the cat/mat/sat/the, 4),

(on sat/mat/sat/the cat/mat/sat/the, 5)

No.

Date

iii) DFS start → cat → cat cat → cat cat cat → cat cat cat cat →

cat cat cat cat cat → cat cat cat cat cat cat →

cat ×5 mat → cat×5 on → cat ×5 sat → cat×5 the

iv) Iterative deeping depth 0 start depth 1 start cat mat on sat

so it is similar to BFS when depth 2 = step 2 in BFS ...

depth 6 = step 6 in BFS, it includes for example: {start, the, the cat, the cat on, the cat sat, ...}

d) start searching from "the cat", "on the mat" are combined, exclude this combination, never search any one of its elements. So the only remain ways of search are "sat" and "on the mat", the cost can be greatly reduced.

2. a) When every node only have one successor, and the goal is at depth n. For DFS, it needs n steps to find the goal; for iterative deepening search it needs $1+2+3+\dots+n = O(n^2)$ steps to get the goal.

b) If all the step costs are equal, the cost to a specific node is just the cost times depth, as a result, the old node (which is in the previous level) will always be visited prior to the new node (which is in the current level). Thus, uniform-cost behaves just like BFS in this way. So BFS is a special case of uniform-cost search.

No.

Date . .

c) In A* search, f=g+h while in uniform-cost search, f=g, thus when h=0, A* search will be the same as uniform-cost search, so the later one is a special case of the former one.

d) A* search always returns the optimal result. And it compares which path has the most negative value and returns it.

e) It is not because consistency should also be achieved. Otherwise the algorithm could enter a cycle. And also, the cost so far is not considered, it might not always be the optimal value.

f) No. If there are several ways with same number of states to reach the goal, it will still consider g(n), in this situation, it might not just go the optimal way.

Question 3

a)

For f1:

Step size	Mean (steps)	Standard deviation(steps)	Mean (final value)	Standard deviation (final value)
0.01	286.13	165.996	1.776	0.332
0.05	56.58	30.365	1.677	0.359
0.1	28.65	15.755	1.714	0.350
0.2	16.31	7.991	1.721	0.349

For f2:

Step size	Mean (steps)	Standard deviation(steps)	Mean (final value)	Standard deviation (final value)
0.01	577.84	247.644	9.1	2.142
0.05	117.16	48.115	8.8	2.400
0.1	60.41	23.647	8.98	2.254
0.2	30.73	10.779	8.86	2.354

From the data above, we can see that when step size decreases, mean and standard deviation value for steps increases. The trend of mean and standard deviation for final value is not that clear, to improve this situation, more data should be recorded for each step size to get the general trend solution.

b) step size = 0.01

For f1:

Beam width	Mean (steps)	Standard deviation(steps)	Mean (final value)	Standard deviation (final value)
2	194.01	117.685	1.708	0.351
4	131.34	73.914	1.651	0.358
8	81.68	47.726	1.733	0.343
16	59.25	33.446	1.613	0.354

For f2:

Beam width	Mean (steps)	Standard deviation(steps)	Mean (final value)	Standard deviation (final value)
2	454.47	211.017	8.56	2.562
4	353.56	187.290	8.08	2.799
8	233.24	125.683	7.66	2.926
16	165.77	98.389	7.6	2.939

From the data above, we can see that when beam width decreases, mean and standard deviation of steps increases. Mean and standard deviation values for final value is a little bit different. For f1, we need more data to conclude the general trend. For f2 however, when beam width increases, mean value decreases and standard deviation increases. Also, we can see that both f1 and f2 need fewer steps to achieve the goal state compared to the ones in the former question, which means that local beam search actually improve the hill climbing performance.