Methods of ai - Assignment 2

Sebastian Bitzer Robert Freund

December 3, 2002

1 Exercise 1

1.1 (i)

There are only two possibilities a search algorithm could not terminate. It could either

- get stuck in an infinite graph
- get trapped in a cycle

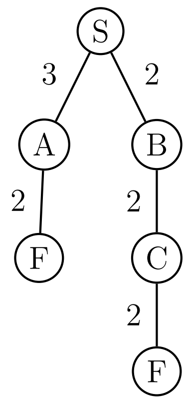
Since a finite graph is assumed in the task, the first can not be the case. It is also impossible for A* to become caught up in a cycle since A* maintains a closed list which enables it to do cycle checks.

1.2 (ii)

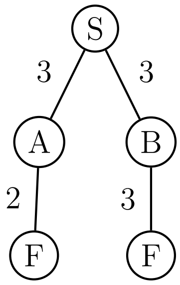
Space complexity: A^* on level d keeps for each preceding level b nodes in memory. $O(b \cdot d)$
Time complexity: A^* on level d has considered b nodes in each preceding level. $O(b \cdot d)$

1.3 (iii)

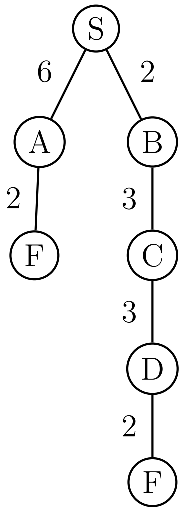
Local maximum problem occuring in state S:



Plateau problem occuring in state S:



Ridge problem:



2 Exercise 2 (Properties of heuristic functions)

2.1 (i)

premise (heuristic function obeys triangle inequality):

$$h(A \rightarrow C) \leq h(A \rightarrow B) + h(B \rightarrow C)$$

assertion (f-costs along any path is nondecreasing):

$$f(A) \leq f(B), A \leq B$$

proof:

$$f(A) = g(X \to A) + h(A \to C)$$

where $g(X \to A)$ are the costs from the starting state X till A and $h(A \to C)$ are the estimated costs from A to final state C

according to premise it is valid that:

$$g(X \to A) + h(A \to C) \leq g(X \to A) + h(A \to B) + h(B \to C)$$

where B is a node between A and the final state C

$h(X)$ has to be admissible, i.e.:

$$h(A \to B) \leq g(A \to B)$$

therefore:

$$g(X \to A) + h(A \to B) + h(B \to C) \leq g(X \to A) + g(A \to B) + h(B \to C)$$

and these are the costs for B:

$$g(X \to A) + g(A \to B) + h(B \to C) = f(B)$$

so:

$$f(A) \leq f(B), A \leq B \quad \square$$

2.2 (ii)

h_1 and h_2 are admissible

(a) $h(s) = h_1(s) + h_2(s)$
is not admissible in general, but if $h_1(s) + h_2(s) \leq g(s)$ then $h(s)$ is admissible, too

(b) $h(s) = |h_1(s) - h_2(s)|$
is admissible (let h_1 be the bigger one: then $h_1 - h_2 \leq h_1 \leq g$)

(c) $h(s) = max(h_1(s), h_2(s))$

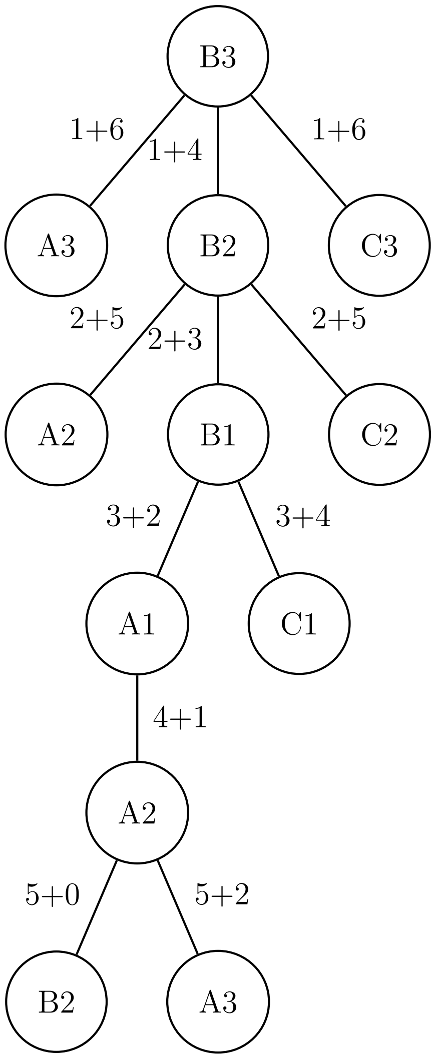
is admissible (if both functions are admissible, then, of course, so is the maximum of them)

(d) $h(s) = min(h_1(s), h_2(s))$
is admissible (if both functions are admissible, then, of course, so is the minimum of them)

(e) $h(s) = \frac{h_1(s)+h_2(s)}{2}$
is admissible (let h_1 be the bigger one: then $\frac{h_1+h_2}{2} \leq \frac{h_1+h_1}{2} = \frac{2h_1}{2} = h_1$ and h_1 is admissible)

3 Exercise 3

The names of the nodes denote the position of the empty part of the puzzle:
A1: First line, first column of puzzle;
A2: First line, second column, etc.
The edges are labeled with the costs: $f^*(n) = g(n) + h^*(n)$



$$\begin{array}{|c|c|c|}\hline 2 & 8 & 3 \\ \hline 1 & 6 & 4 \\ \hline 7 & & 5 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline 2 & 8 & 3 \\ \hline 1 & & 4 \\ \hline 7 & 6 & 5 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline 2 & & 3 \\ \hline 1 & 8 & 4 \\ \hline 7 & 6 & 5 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline & 2 & 3 \\ \hline 1 & 8 & 4 \\ \hline 7 & 6 & 5 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline & 8 & 4 \\ \hline 7 & 6 & 5 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 8 & & 4 \\ \hline 7 & 6 & 5 \\ \hline \end{array}$$

4 Exercise 4 (Human Problem Solving)

X, Y are variables for (possibly empty) strings of letters
m is an integer
f is a function

$$Iter(X, m, f) \rightarrow X f^1(X) \dots f^{m-1}(X)$$

$$R\_Alt(X, f, \langle Y_1, \dots, Y_n \rangle) \rightarrow X Y_1 f^1(X) Y_2 \dots f^{n-1}(X) Y_n$$

$$L\_Alt(X, f, \langle Y_1, \dots, Y_n \rangle) \rightarrow Y_1 X Y_2 f^1(X) \dots Y_n f^{n-1}(X)$$

$$Con_n(X_1, \dots, X_n) \rightarrow X_1 \dots X_n$$

$$Unit_n(X_1, \dots, X_n) \rightarrow (X_1 \dots X_n)$$

4.1 (i)

(a) aabbcc

$$R\_Alt(a, succ, \langle a, b, c \rangle)$$

$$L\_Alt(a, succ, \langle a, b, c \rangle)$$

(b) abccba

$$Con_2(Iter(a, 3, succ), Iter(c, 3, pre))$$

$$Con_3(Iter(a, 2, succ), Iter(c, 2, id), Iter(b, 2, pre))$$

(c) afbcfdf

$$L\_Alt(f, id, \langle a, bc, d \rangle)$$

$R\_Alt(a, succ, \langle f, \emptyset, f, f \rangle)$ where $\emptyset$ is the empty string

(d) abcdef

$$Iter(a, 6, succ)$$

$$Con(Iter(a, 3, succ), Iter(d, 3, succ))$$

(e) acbcccdc

$$L\_Alt(c, id, \langle a, b, c, d \rangle)$$

$$R\_Alt(a, succ, \langle c, c, c, c \rangle)$$

4.2 (ii)

$aabb : bbbccc :: ccdd : dddeee$

$aabb : bbbccc$ can be represented as:

$$Iter(Iter(X, 2, id), 2, succ) : Iter(Iter(Y, 3, id), 2, succ)$$

$$succ(X) = Y \wedge X = "a"$$

where function $succ(Z)$ is defined as increasing every single character in the string for its own and thus obtaining bb from $succ(aa)$

and $ccdd : dddeee$ can be represented as:

$$Iter(Iter(X, 2, id), 2, succ) : Iter(Iter(Y, 3, id), 2, succ)$$

$$succ(X) = Y \wedge X = "c"$$

It is obvious that this is an analogy, because the structures are equal.

W hat could be a measure for different descriptions to figure out which description is the preferred one for humans?

We could imagine several possibilities:

- fewest operator applications
- value function for representations, where the most intuitive operator gets the smallest value and the function is an expression over all occurring operators in the representation
- fewest nesting of operators