Note Title 2014-03-04

A* with non monotone heuristics (example)

A* (with non monotone heuristics)

1. Put the start node s in OPEN.

2. If OPEN is empty, exit with failure.

3. Remove from OPEN and place in CLOSED a node n for which f(n) is minimum.

4. If n is a goal node, exit with the solution obtained by tracing back pointers from n to s.

5. Expand n, generating all of its successors. For each successor n' of n:

   a. Compute g'(n'); compute f'(n')=g'(n')+h(n')

   b. if n' is already on OPEN or CLOSED and g'(n')<g(n'), let g(n')=g'(n'), let f(n')=f'(n'), redirect the pointer from n' to n and, if n' is on CLOSED, move it to OPEN.

   c. if n' is neither on OPEN nor on CLOSED, let f(n')=f'(n'), attach a pointer from n' to n, and place n' on OPEN.

6. Go to 2.

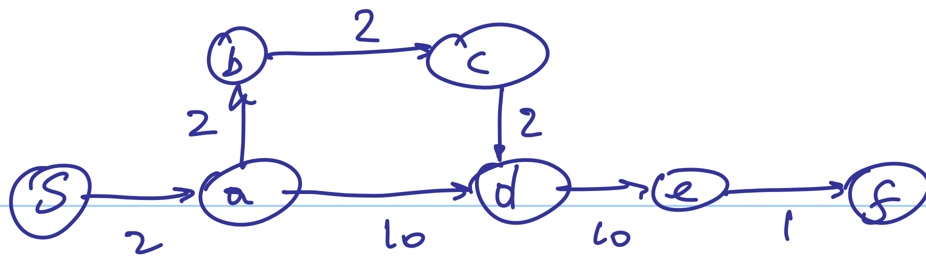
$h(\cdot)$ is <u>consistent</u> if $h(n) \leq c(n,n') + h(n') \;\forall (n, n')$.

$h(\cdot)$ is <u>monotone</u> if $h(n) \leq c(n,n') + h(n') \;\forall n, n'$ s.t. $n' \in SCS(n)$.

One can show that, if $h(\cdot)$ is admissible (i.e., it is a lower bound on $h^*(\cdot)$), then consistency and monotonicity are equivalent.

Here is an example in which A*, with non-monotone heuristics, re-expand a closed node.

$h(a) = 16 \not\leq c(a,d) + h(d) = 10 + 1 = 11$, so the heuristic in the table below is not monotone.



node	h	h^*
s (start)	–	19
a	16	17
b	14	15
c	12	13
d	1	11
e	1	1
f (goal)	0	0

Here is an abbreviate run through A*:
s expanded, a expanded, d expanded
(and closed, $g(d) = 12$, $f(d) = 13$,
backpointer to a; OPEN contains
b with $f(b) = 14$, e with $f(e) = 23$) →
b expanded, c expanded,
<u>d re-expanded</u> ($g' = 8 < g = 12$, $f = 9$),
e expanded, f closed.