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.
Let $h(.)$ be consistent if $h(n) \leq c(n, n') + h(n') \land (n, n') \\
\therefore h(.)$ is monotone if $h(n) \leq c(n, n') + h(n') \land n, n' \in SCS(n)$.

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

Here is an example in which $A^*$, with non-monotone heuristics, re-expands a closed node:

$h(a) = 16 \neq c(a, d) + h(d) = 10 + 1 = 11$, so the heuristic in the table below is not monotone.
Here is an abbreviated run through A*:
s expanded, a expanded, d expanded
(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,
d re-expanded (g' = 8 < g = 12, f = 9),
e expanded, f closed.