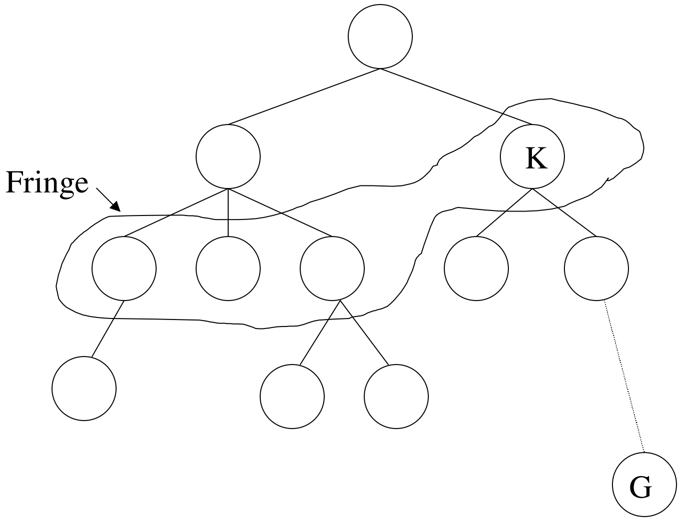
Important Results about A*

Several important results about the A* algorithm were shown in class; proofs were given for some, while others were just mentioned. Here we review these results and go through the proofs in a little more detail.

1) If a solution exists, A* terminates and returns a solution.

Take a "snapshot" of the fringe at some arbitrary time.



If a solution G exists, we know that there must be some node K in the fringe that is an ancestor of G. How do we know this? At the beginning, the fringe is initialized to the root, which is the ancestor of all nodes. The only way the fringe would no longer contain an ancestor of G before visiting G would be if we removed a node from the fringe without inserting all of its children. However, that's what SEARCH#2 does; when it removes a node from the fringe, it creates a new node for each successor and inserts them in the fringe. (Note that in this proof we're assuming we are not discarding nodes with re-visited states; it can be shown separately that under certain conditions certain nodes can be discarded.) Of course the algorithm could terminate before reaching G if it finds another goal first, but that's obviously okay, since we're just trying to show here that it returns **a** solution.

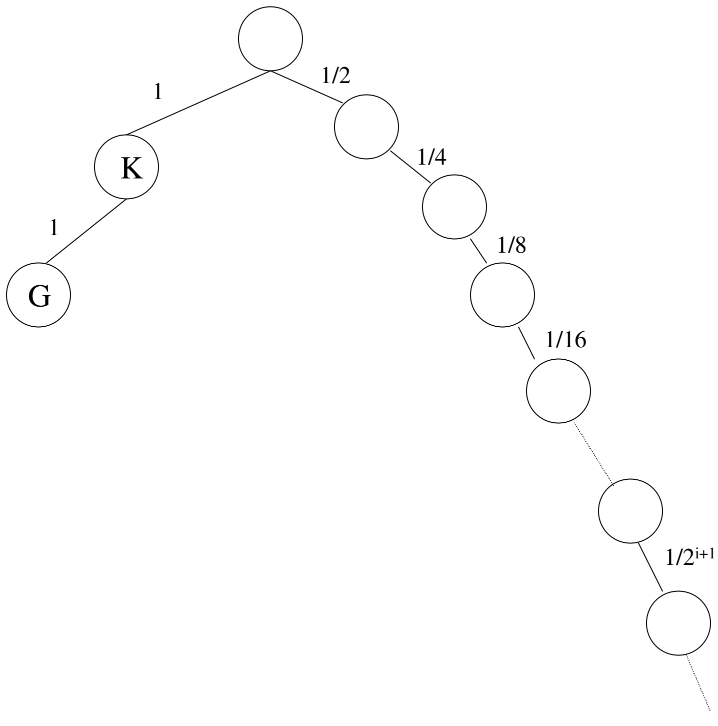
But how do we know that K will ever get removed from the fringe and expanded? If the search space is infinite, or if we start going into loops, might it not go off on some other path(s) forever? Note that, since we require that $c(N,N') \geq \varepsilon > 0$, for any node N, $g(N) \geq d(N)*\varepsilon$, where d(N) is the depth of node N in the search tree, since the node's depth is the number of steps we've taken to reach it, and each step must cost at least ε. Since we have an admissible heuristic, $h(N) \geq 0$, and $f(N) = g(N) + h(N)$, so obviously, $f(N) \geq d(N)*\varepsilon$.

Now what happens for nodes with d(N) > f(K)/ε ? This is equivalent to saying d(N)*ε > f(K). We just said f(N) ≥ d(N)*ε , so, by transitivity, f(N) > f(K). Since we always choose the smallest node in the fringe for expansion, no node at depth greater than f(K)/ε will ever be expanded before K.

Since ε is some positive number, f(K)/ε will be some finite depth. Obviously, if ε is very small, this could be a huge number, and there can be an exponential number of nodes above this depth, but, nevertheless, in some finite time, K will have to be expanded, and we will have made progress towards the goal. We can make the same guarantee that the next node along the path to the goal will also eventually be expanded, and so on, and so, after at most g(G)/ε such steps, we will have reached the goal.

As a side note, consider why we made the somewhat curious specification that c(N,N') ≥ ε > 0, rather than just say that each path cost must be positive. If we had the tree below, where the cost of the path between nodes at depth n and n+1 along the path on the right side is $1/2^{n+1}$ and, for simplicity of illustration, our heuristic function was h=0, then node K would never be expanded, because

$$\lim_{n\to\infty}\sum_{i=1}^{n}\frac{1}{2^i}=1$$

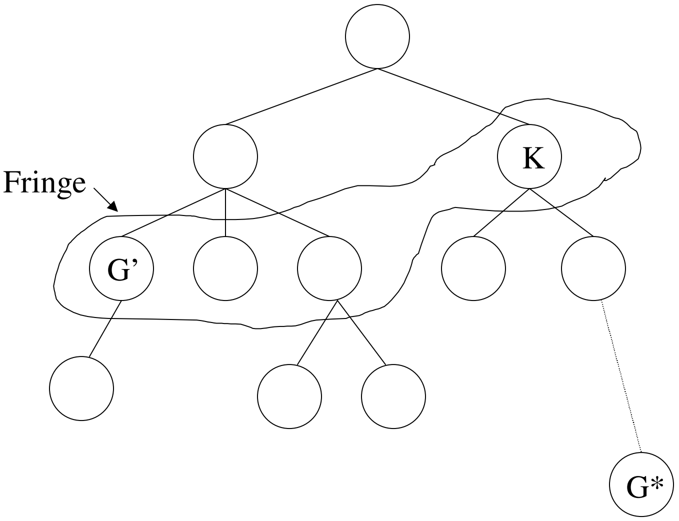


Thus, even though we add a positive path cost for each step along this path, f(N) never exceeds 1, and K is never chosen for expansion.

2) Whenever A* chooses to expand a goal node, the path to this node is optimal.

We now know that, if there is a solution, A* will return one. However, if there are multiple goals, will it always return the optimal one (or one of the optimal ones, if there are more than one with the same optimal cost), or might it find a sub-optimal one first and return it instead?

Let's assume, by way of contradiction, that there is an optimal goal G*, but A* instead returns some sub-optimal goal G'. If this assumption leads to a contradiction, we will know that A* could never do such a terrible thing. In SEARCH#2, it is clear that a node is tested to see if it is a goal (and returned if it is) immediately after extracting it from the fringe. Therefore, we can take a "snapshot" of the fringe immediately before the extraction of G'.



Just as we argued before that, if there is a solution, one of its ancestors K must be in the fringe at all times, by the same argument, for any specific (optimal) goal G*, it also must always have an ancestor K in the fringe. Also, as for any node, f(K) = g(K) + h(K). Since h is admissible, how far we've already actually gone to get to K (g(K)), plus an optimistic estimate of how much further we have left to go to G* (h(K)), must be less than or equal to the actual cost to reach G* (g(G*)), so $f(K) \leq g(G^*)$.

Since we just extracted G', it must also have been in the fringe immediately before the extraction. Since h is admissible, h(G') = 0, and so f(G') = g(G'). Since G' is a non-optimal goal, then obviously $g(G^*) < g(G') = f(G')$. We just showed that f(K) is less than or equal to something ($f(K) \leq g(G^*)$), and now we know that f(G') is greater than that same something ($f(G') > g(G^*)$), so, by transitivity, $f(K) < f(G')$. But A* always chooses the node in the fringe with the smallest f, so it couldn't have chosen G' instead of K. Thus, we've gotten our contradiction, and can rest assured that this travesty (A* choosing a non-optimal goal) cannot happen (given an admissible heuristic).

3) If A* uses a consistent heuristic, then f is non-decreasing along any path.

The definition of a consistent heuristic is that

1) For any node N and one of its children N’, h(N) ≤ c(N,N’) + h(N’).
2) For any goal node G, h(G) = 0.

As always, f(N) = g(N) + h(N). Substituting in the fact that h(N) ≤ c(N,N’) + h(N’), we get that f(N) ≤ g(N) + c(N,N’) + h(N’). However, the cost to get to N plus the cost to get from N to its child N’, g(N) + c(N,N’), is the same thing as the cost to get to N’, g(N’), so f(N) ≤ g(N’) + h(N’). By definition, g(N’) + h(N’) = f(N’), so we know f(N) ≤ f(N’). In other words, a node’s f is never less than it’s parent’s f, so f is non-decreasing along any path.

4) If A* uses a consistent heuristic, then whenever A* expands a node, it has already found an optimal path to this node’s state.

How might this statement not be true? Well, after expanding a node, we might later expand another node labeled by the same state but with a smaller g. Could this happen?

We expand a node K when we extract it from the fringe. Since we always choose the node in the fringe with the smallest f to extract, it must be true that all other nodes in the fringe at that time must have an f greater than or equal to K’s. Since the fringe is only changed by removing a node and inserting its children, only descendants of nodes currently in the fringe will ever be in the fringe. We just showed that f is non-decreasing along any path (Proof 3), so no node will ever again be in the fringe with an f smaller than those currently in the fringe, which are all already at least as large as f(K), so we will never again encounter a node with a smaller f than f(K).

Thus, for any node K’ expanded after K and labeled by the same state as K, f(K’) ≥ f(K). If two nodes are labeled by the same state, then they have the same value of h, since the heuristic is just a function of the state. Thus, g(K’) ≥ g(K), so we will never find a better path to this state.

5) If h is consistent, then h(N) ≤ c(N,N’) + h(N’) for any descendant N’ of N.

We know from the definition of a consistent heuristic that h(N) ≤ c(N,N’) + h(N’) for any **child** N’ of N. Does this apply only for direct children of N, or for all descendants of N?

Consider a grandchild N’’ of N, by way of N’. Assuming only that “child” refers to direct children, we know both that h(N) ≤ c(N,N’) + h(N’) and that h(N’) ≤ c(N’,N’’) + h(N’’). Substituting the second inequality for h(N’) in the first, we get h(N) ≤ c(N,N’) + c(N’,N’’) + h(N’’). However, the path cost from N to N’ (c(N,N’)) plus the path cost from N’ to N’’ (c(N’,N’’)) is just the path cost from N to N’’ (c(N,N’’)), so we have h(N) ≤ c(N,N’’) + h(N’’). Thus, the inequality holds for a grandchild N’’ of N as well as for a direct child N’. By a similar argument, we can extend this to any descendant of N.

6) If h is consistent, then h is admissible.

Again, the definition of a consistent heuristic is that

1) For any node N and one of its children N', $h(N) \leq c(N,N') + h(N')$.
2) For any goal node G, $h(G) = 0$.

For any node N, consider the best path from it to any goal node. This is the h*(N) that an admissible heuristic must not overestimate. (Note that if there is no path from N to a goal node, any value for h(N) is admissible.) We just showed that the N' in the definition of a consistent heuristic can be any descendant of N, so we can let N' be G, where G is this best reachable goal node. Then we have $h(N) \leq c(N,G) + h(G)$. By part two of the definition, $h(G) = 0$, so $h(N) \leq c(N,G)$. However, c(N,G) is the same thing as h*(N), so we have $h(N) \leq h^*(N)$, the definition of an admissible heuristic.

7) The set of nodes that are expanded before the optimal goal G* is extracted (and the search terminates) by A* using a consistent heuristic is exactly the set of nodes N such that $f(N) < g(G^*)$, plus possibly some nodes with $f(N) = g(G^*)$.

Again, think about the fringe right before we extract and return G*. Since G* is a goal, $f(G^*) = g(G^*)$. There are three places where nodes can be: in the fringe, below the fringe, or above the fringe. Those in the fringe (other than G*) and below the fringe obviously have not been expanded, and those above the fringe have been expanded (since nodes are only added to the fringe by having expanded their parents). Thus, we need to show that $f(N) \geq g(G^*)$ for nodes in and below the fringe, and $f(N) \leq g(G^*)$ for nodes above the fringe.

Since we extract the node from the fringe with the smallest f, it is clear that $f(N) \geq f(G^*) = g(G^*)$ for all nodes in the fringe. Since we are using a consistent heuristic, f is non-decreasing along any path (as shown in Proof 3), and all nodes below the fringe are descendants of nodes currently in the fringe (since nodes are only added to the fringe by expanding their parents), and we just showed that $f(N) \geq g(G^*)$ for all nodes in the fringe, so $f(N) \geq g(G^*)$ for all nodes below the fringe also.

Finally, how do we know $f(N) \leq g(G^*)$ for nodes above the fringe? Assume by way of contradiction that there were some node M above the fringe with $f(M) > g(G^*)$. Since M is above the fringe, M must have already been expanded (since nodes are only added to the fringe by expanding their parents). However, the fringe must have always included G* or one of its ancestors (as shown in Proof 1). Since $f(G^*) = g(G^*)$, and f is non-decreasing along any path, there must have always been an ancestor K of G* in the fringe with $f(K) \leq g(G^*)$. Thus, $f(M) > g(G^*) \geq f(K)$, so M could never have been chosen for expansion over K, and we have a contradiction, so our assumption that there could be a node M above the fringe with $f(M) > g(G^*)$ cannot be true.

8) If h_1 and h_2 are two consistent heuristics such that, for all nodes N, $h_1(N) \leq h_2(N)$, whenever a solution exists, all the nodes expanded by A* using h_2 as its heuristic are also expanded by A* using h_1 as its heuristic, except possibly some nodes with $f_1(N) = f_2(N) = g(G^*)$. (Note that $f_1(N)$ refers to f(N) using h_1 and $f_2(N)$ to f(N) using h_2.)

This is another way of saying that A* using any given consistent heuristic will never expand more nodes than it would have using a less accurate consistent heuristic (and could expand many fewer), aside for some "boundary case" exceptions (nodes with $f(N) = g(G^*)$. Just like when bidding on “The Price is Right”, it’s best for a heuristic to get as close as possible to the correct answer, without going over. (If it does sometimes go over, it could still be a very good and more efficient heuristic, but you lose the guarantee of getting an optimal solution.)

First, we will ignore the boundary cases, in which $f_2(N)$ or $f_1(N)$ equal $g(G^*)$. Then we can say (from Proof 7 above) that the set of nodes expanded by A* is exactly those such that $f(N) < g(G^*)$. Rewriting this as $g(N) + h(N) < g(G^*)$, we see that the nodes that are expanded are exactly those such that $h(N) < g(G^*) - g(N)$. Thus, if $h_2(N)$ is small enough such that N is expanded, then surely $h_1(N)$ is also small enough such that N will be expanded, since $h_1(N) \leq h_2(N)$, so a node expanded using h_2 will always also be expanded using h_1 .

Now to consider the boundary cases. If $f_2(N) = g(G^*)$ and $f_1(N) < g(G^*)$, N is expanded by A* using h_1, so it doesn't matter if it is expanded by A* using h_2, since the statement we're trying to prove gives a condition under which A* using h_1 must expand a node, but doesn't give any conditions under which it must not expand a node. If $f_1(N) = g(G^*)$ and $f_2(N) > g(G^*)$, N is not expanded by A* using h_2, so it doesn't matter if it is expanded by A* using h_1, since the statement we're trying to prove only makes a requirement of A* using h_1 in the case that A* using h_2 does expand the node. Therefore, the only undetermined case is when $f_1(N) = f_2(N) = g(G^*)$. Since ties may be broken arbitrarily, there is no guarantee that any node expanded by one of the heuristics will or will not be expanded by the other.