Lazy MT-Adaptive A* Proofs

Sven Koenig
University of Southern California
skoenig@usc.edu

In the following, we give a proof that Eager and Lazy MT-Adaptive A* use the same h-values if they break ties identically. Unfortunately, the proof is very terse. The line numbers refer to the search algorithms in Speeding up Moving-Target Search by Koenig, Likhachev and Sun.

Theorem 1 *During every search, Eager and Lazy MT-Adaptive A* use the same h-values if they break ties identically.*

Proof: The values at the end of the ith search are indicated via superscript i. The h-values at the end of the ith search are the same as those used during the ith search since Eager MT-Adaptive A* does not update any h-value during a search and Lazy MT-Adaptive A* calculates any h-value the first time it is needed during a search and then returns this h-value whenever it is needed again during the same search. The values of Eager MT-Adaptive A* are not overlined, while the values of Lazy MT-Adaptive A* are overlined. We do not make this distinction for s^i_{target} since $s^i_{target} = \bar{s}^i_{target}$ per construction.

We define $z^i(s) = h^i(s)$ if s was not expanded by Eager MT-Adaptive A* during the ith search and $z^i(s) = g^i(s^i_{target}) - g^i(s)$ otherwise. Similarly, we define $\bar{z}^i(s) = \bar{h}^i(s)$ if s was not expanded by Lazy MT-Adaptive A* during the ith search and $\bar{z}^i(s) = \bar{g}^i(s^i_{target}) - \bar{g}^i(s)$ otherwise.[1] $z^i(s^{i+1}_{target})$ is equal to $h(newtarget)$ computed by Eager MT-Adaptive A* on Line 39, and $\bar{z}^i(s^{i+1}_{target})$ is equal to $h(newtarget)$ computed by Lazy MT-Adaptive A* on Line 46. Eager and Lazy MT-Adaptive A* expand the same states during

[1] If Lazy MT-Adaptive A* expands a state s with $\bar{g}^i(s) + \bar{h}^i(s) = \bar{g}^i(s^i_{target})$, then it actually sets $\bar{z}^i(s) = h^i(s)$ but this does not cause a problem for our definition since $\bar{z}^i(s) = \bar{h}^i(s) = \bar{g}^i(s) + \bar{h}^i(s) - \bar{g}^i(s) = \bar{g}^i(s^i_{target}) - \bar{g}^i(s)$.

the same search when they use the same h-values and thus also calculate the same g- and z-values. For example, $\overline{deltah}(k) = \sum_{l=1}^{k-1} \bar{z}^l(s_{target}^{l+1})$ for all k with $k \geq 1$.

We prove the theorem by induction on the number of times Lazy MT-Adaptive A* calls InitializeState. Assume that Lazy MT-Adaptive A* calls InitializeState(s) during the jth search. Let x be equal to $search(s)$ at that point in time. These s, j and x are used in the remainder of the proof.

Lemma 1 *If $h^{k+1}(s) = H(s, s_{target}^{k+1})$ for at least one k with $0 \leq x \leq k < j$, then $h^j(s) = H(s, s_{target}^j)$.*

Proof: The lemma trivially holds if $k = j - 1$. Otherwise, we show that $h^{l+2}(s) = H(s, s_{target}^{l+2})$ if $h^{l+1}(s) = H(s, s_{target}^{l+1})$ for $k \leq l < j$, which implies the lemma. InitializeState(s) was called last during the xth search (or has not been called before iff $x = 0$). Thus, s was expanded last during or before the xth search (or has not been expanded yet iff $x = 0$) by Lazy MT-Adaptive A* and thus also by Eager MT-Adaptive A* according to the induction hypothesis since they expand the same states during the same search when they use the same h-values. $h^{l+2}(s) = \max(h^{l+1}(s) - z^{l+1}(s_{target}^{l+2}), H(s, s_{target}^{l+2})) = \max(H(s, s_{target}^{l+1}) - z^{l+1}(s_{target}^{l+2}), H(s, s_{target}^{l+2})) \leq \max(H(s, s_{target}^{l+1}) - H(s_{target}^{l+2}, s_{target}^{l+1}), H(s, s_{target}^{l+2})) \leq \max(H(s, s_{target}^{l+2}), H(s, s_{target}^{l+2})) = H(s, s_{target}^{l+2})$ since $H(s, s_{target}^{l+1}) \leq H(s, s_{target}^{l+2}) + H(s_{target}^{l+2}, s_{target}^{l+1})$ due to the triangle inequality of the H-values, $H(s_{target}^2, s_{target}^1) = h^1(s_{target}^2) \leq z^1(s_{target}^2)$ and $H(s_{target}^{l+2}, s_{target}^{l+1}) \leq \max(z^l(s_{target}^{l+2}) - z^l(s_{target}^{l+1}), H(s_{target}^{l+2}, s_{target}^{l+1})) = h^{l+1}(s_{target}^{l+2}) \leq z^{l+1}(s_{target}^{l+2})$ if $1 \leq l$, where the last inequality in the derivation holds since the h-value updates increase the h-values monotonically. Thus, $h^{l+2}(s) = H(s, s_{target}^{l+2})$ since also $h^{l+2}(s) = \max(h^{l+1}(s) - z^{l+1}(s_{target}^{l+2}), H(s, s_{target}^{l+2})) \geq H(s, s_{target}^{l+2})$. ■

If $x = j$, then InitializeState(s) does not change $\bar{h}(s)$. It was called last during the xth search, that is, the current search. It continues to hold that $h^j(s) = \bar{h}^j(s)$ according to the induction hypothesis. Otherwise, $0 \leq x < j$. We distinguish two cases:

- Case 1: Assume that $x = 0$ (induction basis). Then, $h^j(s) = H(s, s_{target}^j) = \bar{h}^j(s)$ since $h^1(s) = H(s, s_{target}^1)$ and thus $h^j(s) = H(s, s_{target}^j)$ according to the lemma.

- Case 2: Otherwise, $x > 0$. Assume that Eager and Lazy MT-Adaptive A* used the same h-values every time Lazy MT-Adaptive A* called InititalizeState so far. s was expanded last during or before the xth search by Lazy MT-Adaptive A* and thus also by Eager MT-Adaptive A* according to the induction hypothesis since they expand the same states during the same search when they use the same h-values. We distinguish two cases:

  - Case a: Assume that $h^{k+1}(s) = H(s, s^{k+1}_{target})$ for at least one k with $x \leq k < j$. Then, $h^j(s) = H(s, s^j_{target})$ according to the lemma. It holds that $z^x(s) - \sum_{l=x}^{j-1} z^l(s^{l+1}_{target}) \leq h^j(s)$ due to the monotonicity of the max operator used repeatedly in the calculation of $h^j(s)$. Thus, $\bar{h}^j(s) = \max(\bar{z}^x(s) - (\overline{deltah}(j) - \overline{deltah}(x)), H(s, s^j_{target})) = \max(\bar{z}^x(s) - \sum_{l=x}^{j-1} \bar{z}^l(s^{l+1}_{target}), H(s, s^j_{target})) = \max(z^x(s) - \sum_{l=x}^{j-1} z^l(s^{l+1}_{target}), H(s, s^j_{target})) \leq \max(h^j(s), H(s, s^j_{target})) = \max(H(s, s^j_{target}), H(s, s^j_{target})) = H(s, s^j_{target})$ since $z^x(s) = \bar{z}^x(s)$ and $z^l(s^{l+1}_{target}) = \bar{z}^l(s^{l+1}_{target})$ for all $x \leq l < j$ according to the induction hypothesis. Thus, $h^j(s) = H(s, s^j_{target}) = \bar{h}^j(s)$ since also $\bar{h}^j(s) = \max(\bar{z}^x(s) - (\overline{deltah}(j) - \overline{deltah}(x)), H(s, s^j_{target})) \geq H(s, s^j_{target})$.
  - Case b: Otherwise, $h^{x+1}(s) = z^x(s) - z^x(s^{x+1}_{target})$ and $h^{k+1}(s) = h^k(s) - z^k(s^{k+1}_{target})$ for all k with $x < k < j$ since $h^{x+1}(s) = \max(z^x(s) - z^x(s^{x+1}_{target}), H(s, s^{x+1}_{target})) \neq H(s, s^{x+1}_{target})$ and $h^{k+1}(s) = \max(h^k(s) - z^k(s^{k+1}_{target}), H(s, s^{k+1}_{target})) \neq H(s, s^{k+1}_{target})$ for all k with $x < k < j$. Then, $h^j(s) = \max(h^j(s), H(s, s^j_{target})) = \max(z^x(s) - \sum_{l=x}^{j-1} z^l(s^{l+1}_{target}), H(s, s^j_{target})) = \max(\bar{z}^x(s) - \sum_{l=x}^{j-1} \bar{z}^l(s^{l+1}_{target}), H(s, s^j_{target})) = \max(\bar{z}^x(s) - (\overline{deltah}(j) - \overline{deltah}(x)), H(s, s^j_{target})) = \bar{h}^j(s)$ since $z^x(s) = \bar{z}^x(s)$ and $z^l(s^{l+1}_{target}) = \bar{z}^l(s^{l+1}_{target})$ for all $x \leq l < j$ according to the induction hypothesis and $h^j(s) = \max(z^{j-1}(s) - z^{j-1}(s^j_{target}), H(s, s^j_{target})) \geq H(s, s^j_{target})$. ■