附录 C：A*算法的性质及证明

为了叙述方便，我们首先给出 A 算法的形式化描述，有关 A 算法的说明请见第一章内容。对于任意节点 n，当启发函数满足 $h(n) \leqslant h^*(n)$时，A 算法即为 A*算法。

A 算法：

1，初始化：OPEN=(S), CLOSED=( )，计算 f(S)；

2，循环做以下步骤直到 OPEN 为空结束：

3，循环开始

4，　　从 OPEN 中取出第一个节点，用 n 表示该节点；

5，　　如果 n 就是目标节点，算法结束，输出节点 n，算法成功结束；

6，　　否则将 n 从 OPEN 中删除，放到 CLOSED 中；

7，　　扩展节点 n，生成出 n 的所有子节点，用 m_i 表示这些子节点；

8，　　计算节点 m_i 的 f 值，由于可能存在多个路径到达 m_i，用 $f(n,m_i)$ 表示经过节点 n 到达 m_i 计算出的 f 值，不同的到达路径其 $g(m_i)$ 值可能不同，但是 $h(m_i)$ 是一样的，因为 $h(m_i)$ 是从 m_i 到目标节点路径代价的估计值，与如何从初始节点到达 m_i 无关；

9，　　如果 m_i 既不在 OPEN 中，也不在 CLOSED 中，说明这是一个新出现的节点，则将 m_i 加入到 OPEN 中，并标记 m_i 的父节点为 n；

10，　　如果 m_i 在 OPEN 中，并且 $f(n,m_i) < f(m_i)$，则 $f(m_i)=f(n,m_i)$，并标记 m_i 的父节点为 n；

11，　　如果 m_i 在 CLOSED 中，并且 $f(n,m_i) < f(m_i)$，则 $f(m_i)=f(n,m_i)$，并标记 m_i 的父节点为 n，将 m_i 从 CLOSED 中删除、重新加入到 OPEN 中；

12，　　对 OPEN 中的节点按照 f 值从小到大排序；

13，循环结束

14，没有找到解，算法以失败结束

定理 1：对有限图，如果从初始节点 s 到目标节点 t 有路径存在，则 A 算法一定成功结束，即一定能找到一条从初始节点 s 到目标节点 t 的路径。

证明：设 A 算法搜索失败，即没有找到解，则算法在第 2 步结束，OPEN 表变空，而 CLOSED 表中的节点是在结束之前被扩展过的节点。由于图有解，令（n_0=s，n_1，n_2，…，n_k=t）表示某一解路径，我们从 n_k 开始逆向逐个检查该序列的节点，找到出现在 CLOSED 表中的节点 n_i，即 $n_i \in$CLOSED，$n_{i+1} \notin$CLOSED（n_i 一定能找到，因为 $n_0 \in$CLOSED，$n_k \notin$CLOSED）。由于 n_i 在 CLOSES 中，必定在第 6 步被扩展，且 n_{i+1} 被加到 OPEN 中，因此在 OPEN 表空之前，n_{i+1} 一定会被从 OPEN 表中取出。若 n_{i+1} 是目标节点，则搜索成功，否则它被加入到 CLOSED 中，这两种情况都与搜索失败的假设矛盾，因此对有限图来说，当问题有解时，A 算法一定能找到解结束。[证毕]

因为 A*算法是是 A 算法的特例，因此它具有 A 算法的所有性质。这样对有限图来说，如果有解，则 A*算法一定能在找到到达目标的路径结束，下面要证明即使是无限图的情况下，A*算法不但一定能找到解，而且一定能找到最佳解结束。

在如下的所有定理证明中，均隐含了如下三个假设，这些假设在实际情况中是合理的。

（1）任何两个节点之间的耗散值都大于某个给定的大于 0 的常量，也即不能是负数，也不能是无穷小；

（2）h(n)对于任何 n 来说，都有 h(n)⩾0。如果定义的 h(n)存在小于 0 的情况，可以令其为 0，因为根据假设（1），这种处理是合理的。

（3）目标 t 的 h(t)函数等于 0，如果定义的 t 的 h 函数不为 0，则可以令其为 0。因为目标到目标的最佳路径耗散值一定为 0。

并用到了如下等式：

$$f^*(s) = f^*(t) = h^*(s) = g^*(t) = f^*(n)$$

其中 s 是初始节点，t 是目标节点，n 是 s 到 t 的最佳路径上的节点，$g^*(n)$是从从初始节点 s 到节点 n 的最佳路径的耗散值，$h^*(n)$是节点 n 到目标节点 t 的最佳路径的耗散值，$f^*(n)=g^*(n)+h^*(n)$，是从初始节点 s 出发经过节点 n 到达目标节点 t 的最佳路径的耗散值。理解了这几个符号的含义，就很容易理解上述等式为什么成立，因为 $f^*(s)$、$f^*(t)$、$h^*(s)$、$g^*(t)$、$f^*(n)$均表示的是从 s 到 t 的最佳路径的耗散值。

引理 2.1：对无限图，若有从初始节点 s 到目标点 t 的一条路径，则 A^*算法不结束时，在 OPEN 中即使最小的一个 f 值也将增到任意大，或有 $f(n)>f^*(s)$。

证明：设 n 是 A^*算法生成的搜索树中的叶节点，也即 n 在 OPEN 中，$d^*(n)$是从 s 到 n 最短路径经过的节点数（$d^*(s)=0$），最佳路径上任意两个相邻节点间的耗散值为 $C(n_i, n_{i+1})(i=0,1,\ldots, d^*(n))$。令$e = \min_i(C(n_i, n_{i+1})$，则 $g^*(n) \geqslant e \times d^*(n)$。而 $g(n) \geqslant g^*(n) \geqslant e \times d^*(n)$，故有：

$$f(n)=g(n)+h(n) \geqslant g(n) \geqslant e \times d^*(n)$$

如果 A^*算法不结束的话，将一直搜索下去，对于一个无限图，必有 $d^*(n)$趋于无穷大，所以有 f(n)值也将会趋于任意大。而从 s 到目标 t 的最佳路径耗散值一定是有界的，所以这种情况下，必有 $f(n)>f^*(s)$。[证毕]

引理 2.2：A^*算法结束前，OPEN 表中必存在一个节点 n，$f(n) \leqslant f^*(s)$，且 n 是在从 s 到目标节点 t 的最佳路径上的节点。

证明：设从初始节点 s 到目标节点 t 的一条最佳路径序列为：

$$(n_0=s，n_1，\cdots，n_k=t)$$

算法初始化时，s 在 OPEN 中，所以开始时 OPEN 中存在最佳路径上的节点 s。

在 A*算法结束前，最佳路径序列中前面的一些节点被放入到 CLOSED 表中，比如第一次扩展后，s 就被放入了 CLOSED 表中，一些节点还在 OPEN 表中。沿着最佳路径序列从 s 开始逆向查找，找到第一个在 OPEN 中的节点 n。由于算法还没有结束，这样的节点肯定存在。由于 n 是最佳路径上的节点，其在最佳路径序列上的祖先已经全部被扩展了（否则 n 不会是最佳路径序列中第一个在 OPEN 中的节点），所以这时就已经找到了从 s 到 n 的最佳路径，即 $g(n)=g^*(n)$。所以有：

$$f(n)=g(n)+h(n)=g^*(n)+h(n) \leqslant g^*(n)+h^*(n)=f^*(n)$$

由于 n 是最佳路径上的节点，所以有 $f^*(n)=f^*(s)$，所以 $f(n) \leqslant f^*(s)$。[证毕]

定理 2：对无限图，若从初始节点 s 到目标节点 t 有路径存在，则 A^*算法也一定成功结束，即一定能找到一条从初始节点 s 到目标节点 t 的路径。

证明：假定A^*不结束，由引理2.1有$f(n)>f^*(s)$，或OPEN表中最小的一个f值也变成任意大，这与引理2.2的结论矛盾，所以A^*只能成功结束。[证毕]

推论2.1：OPEN表上任一具有$f(n)<f^*(s)$的节点n，最终都将被A^*算法选作为扩展的节点。

证明：由定理2知A^*算法一定会成功结束，由A^*算法的结束条件，当OPEN表中$f(t)$最小时才结束。而$f(t)\geq f^*(t) = f^*(s)$，所以OPEN表中满足条件$f(n)<f^*(s)$的节点n，一定会被扩展。[证毕]

定理3：若存在初始节点s到目标节点t的路径，则A^*算法必能找到最佳解结束。

证明：

（1）由定理1、2知A^*算法一定会找到一个目标节点结束。

（2）设找到一个目标节点t结束，但找到的不是s到t的最佳路径，即：

$$f(t)=g(t)>f^*(s)$$

根据引理2.2知算法结束前OPEN表上有节点n，且处在最佳路径上，并有$f(n)\leqslant f^*(s)$，所以：

$$f(n)\leqslant f^*(s)<f(t)$$

这时A^*算法应该选n作为当前节点扩展，而不是选择目标节点t，这与假定A^*算法选t结束矛盾，所以A^*算法只能结束在最佳路径上。[证毕]

推论3.1：A^*算法选作扩展的任一节点n，有$f(n)\leqslant f^*(s)$。

证明：令n是由A^*算法选作扩展的任一节点，因此n不会是目标节点，且搜索没有结束，由引理2.2知在OPEN中有满足$f(n')\leqslant f^*(s)$的节点n'。若n=n'，则$f(n)\leqslant f^*(s)$，否则选n扩展，必有$f(n)\leqslant f(n')$，所以$f(n)\leqslant f^*(s)$成立。[证毕]

定理4：设有两个A^*算法A_1和A_2，若A_2比A_1有较多的启发信息，即对所有非目标节点均有$h_2(n)>h_1(n)$，则在具有一条从s到t的隐含图上，搜索结束时，由A_2所扩展的每一个节点，也必定由A_1所扩展，即A_1扩展的节点至少和A_2一样多。

这里所说的两个A^*算法A_1和A_2，指的是同一个问题定义了两个不同的启发函数$h_1(n)$和$h_2(n)$，分别用这两个启发函数得到的两个A^*算法。

证明：使用数学归纳法，对节点的深度进行归纳。

（1）对深度$d(n)=0$的节点（即初始节点s），定理结论成立，即若s为目标节点，则A_1和A_2都不扩展s，否则A_1和A_2都扩展了s（归纳法前提）。

（2）设深度$d(n)\leqslant=k$，对所有路径的端节点，定理结论都成立（归纳法假设）。

（3）要证明$d(n)=k+1$时，对所有路径的端节点，定理结论成立（归纳法推广）。

我们用反证法证明（3）。

设A_2搜索树上有一个节点n（$d(n)=k+1$）被A_2扩展了，而对应于A_1搜索树上的这个节点n，没有被A_1扩展。由于n被A_2扩展了，所以其父节点（d(n的父节点)=k）也肯定被A_2扩展了。由归纳法假设条件，A_1也扩展了n的父节点，所以n在A_1搜索树上，因此A_1结束时，n必定保留在其OPEN表上，因为n没有被A_1选择扩展。所以有：

$$f_1(n)\geqslant f^*(s)$$

即：

$$g_1(n)+h_1(n)\geqslant f^*(s)$$

有：

$h_1(n) \geqslant f^*(s) - g_1(n)$ （C.1）

另一方面 A_2 扩展了 n，有:

$f_2(n) \leqslant f^*(s)$

即：

$g_2(n)+h_2(n) \leqslant f^*(s)$

有：

$h_2(n) \leqslant f^*(s)-g_2(n)$ （C.2）

由于 d=k 时，A_2 扩展的节点，A_1 也一定扩展（归纳法假设），故有：

$g_1(n) \leqslant g_2(n)$ (C.3)

这是因为 A_1 扩展的节点包含了 A_2 扩展的节点，所以 A_1 扩搜索到的 $g_1(n)$不会比 $g_2(n)$大。

由式(C.1)、(C.3)有：

$h_1(n) \geqslant f^*(s) - g_1(n) \geqslant f^*(s) - g_2(n)$ （C.4）

比较式（C.2）、（C.4）可得：至少在节点 n 上有 $h_1(n) \geqslant h_2(n)$，这与定理的前提条件矛盾，因此存在节点 n 的假设不成立。[证毕]

定理 5：若 h(n)满足单调限制条件，则 A^*算法扩展了节点 n 之后，就已经找到了到达节点 n 的最佳路径。即若 A^*算法选 n 来扩展，在单调限制条件下有 $g(n)=g^*(n)$。

证明：设 n 是 A^*算法选作扩展的任一节点，若 n＝s，显然有 $g(s)=g^*(s)=0$，因此考虑 $n \neq s$ 的情况。

我们用序列 $P=(n_0=s, n_1, \cdots, n_k=n)$表示从 s 到达 n 的最佳路径。现在从 OPEN 中取出非初始节点 n 扩展时，假定没有找到 P，这时 CLOSED 中一定会有 P 中的节点（至少 s 是在 CLOSED 中，n 刚被选作扩展，不在 CLOSED 中），把 P 序列中（依顺序检查）最后一个出现在 CLOSED 中的节点称为 n_j，那么 n_{j+1} 是在 OPEN 中（$n_{j+1} \neq n$）。

由单调限制条件，对任意 i 有

$g^*(n_i) + h(n_i) \leqslant g^*(n_i) + C(n_i, n_{i+1}) + h(n_{i+1})$ （C.5）

因为 n_i 和 n_{i+1} 在最佳路径上，所以有

$g^*(n_{i+1})=g^*(n_i) + C(n_i, n_{i+1})$

代入式（C.5）有：

$g^*(n_i) + h(n_i) \leqslant g^*(n_{i+1}) + h(n_{i+1})$

这个不等式对 P 上所有相邻的节点都适用，若从 i＝j 到 i＝k–1 应用该不等式，并利用传递性有：

$g^*(n_{j+1})+h(n_{j+1}) \leqslant g^*(n_k)+h(n_k)$

由于序列中 n_k 就是 n，所以有：

$f(n_{j+1}) \leqslant g^*(n)+h(n)$ （C.6）

另一方面，A^*算法选择 n 扩展而没有选择 n_{j+1} 扩展，必有：

$f(n)=g(n)+h(n) \leqslant f(n_{j+1})$ （C.7）

比较式（C.6）、（C.7）有：

$g(n) \leqslant g^*(n)$

但已知 $g(n) \geqslant g^*(n)$，因此选 n 扩展时必有 $g(n) = g^*(n)$，即找到了从 s 到达 n 的最佳路径。[证毕]

定理 6：若 $h(n)$满足单调限制，则由 A^*算法所扩展的节点序列，其 f 值是非递减的，即 $f(n_i) \leqslant f(n_j)$。其中 n_j 在 n_i 的后面被扩展。

证明：如果 n_j 不是 n_i 的子节点，则既然 A^*算法先选择 n_i 扩展，则必有：

$$f(n_i) \leqslant f(n_j)$$

所以只需证明 n_j 是 n_i 的子节点的情况。

由单调限制条件：

$$h(n_i) - h(n_j) \leqslant C(n_i，n_j)$$

即：

$$(f(n_i) - g(n_i)) - (f(n_j) - g(n_j)) \leqslant C(n_i，n_j) \qquad (C.8)$$

由于 n_j 是 n_i 的子节点，所以有：

$$g(n_j) = g(n_i) + C(n_i，n_j)$$

带入式(C.8)有：

$$f(n_i) - g(n_i) - f(n_j) + g(n_i) + C(n_i，n_j) \leqslant C(n_i，n_j)$$

化简有：

$$f(n_i) - f(n_j) \leqslant 0$$

所以：

$$f(n_i) \leqslant f(n_j)$$

[证毕]