**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)≤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mi表示这些子节点；

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

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

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

11， 如果mi在CLOSED中，并且f(n,mi)＜f(mi)，则f(mi)=f(n,mi)，并标记mi的父节点为n，将mi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（n0=s，n1，n2，…，nk=t）表示某一解路径，我们从nk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ni，即ni∈CLOSED，ni+1∉CLOSED（ni一定能找到，因为n0∈CLOSED，nk∉CLOSED）。由于ni在CLOSED中，必定在第6步被扩展，且ni+1被加到OPEN中，因此在OPEN表空之前，ni+1一定会被从OPEN表中取出。若ni+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]