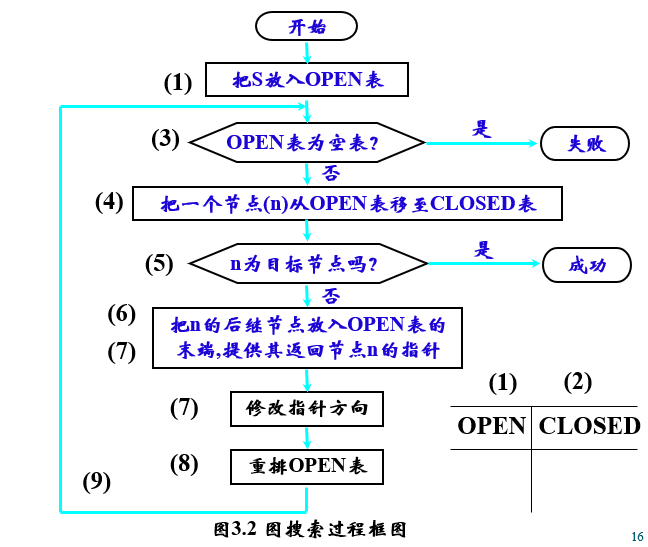
（掌握）图搜索过程

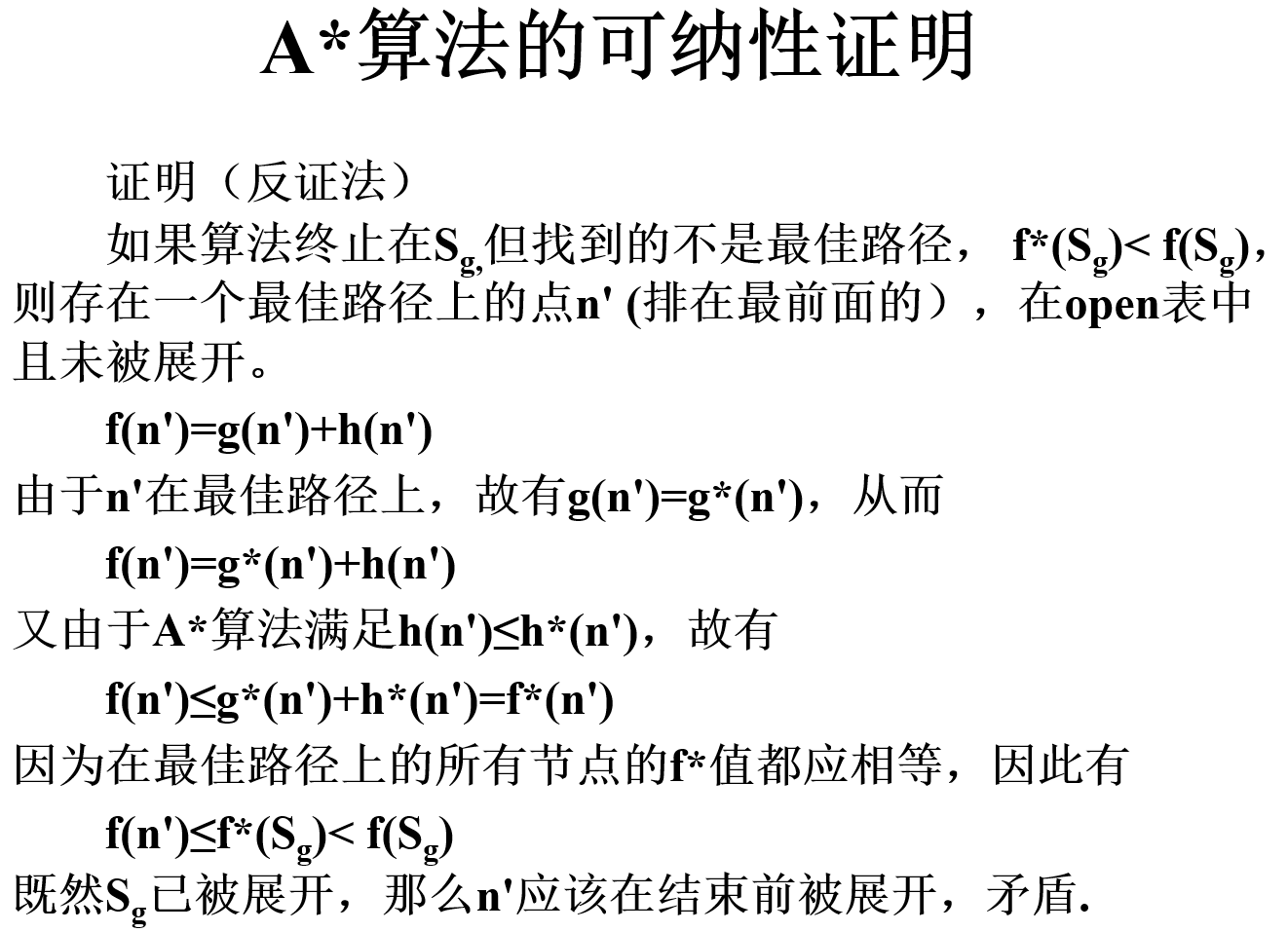
GRAPHSEARCH：

1. 建立一个只含有起始节点S的搜索图G，图中每个节点有一个指向其父节点的指针，S的这一指针为一特殊值(如0)，并把S放入未扩展节点表OPEN中。
2. 建立已扩展的节点表CLOSED，初始时该表为空。
3. LOOP：若OPEN表为空，则失败退出。
4. 把OPEN表中的第一个节点移出，放入CLOSED表中，称其为n节点。
5. 若n为目标节点，则成功退出。问题的解是沿指针追踪G中从n到S的路径而得到的。
6. 扩展节点n，生成不是n的祖先的那些后继节点的集合M。如果没有后继节点，则转LOOP。
7. 把那些不在G中的M的成员作为n的后继节点加入G，并设置一个通向n的指针，把它们加入OPEN表。对已在G中的M的成员，调整有关指针。
8. 按某种方式或按某个试探值，重排OPEN表。
9. GO LOOP。



（掌握）A\*算法

* 对节点n定义f\*(n)=g\*(n)+h\*(n)，表示从S开始通过节点n的一条最佳路径的代价。其中，g\*(n)表示从起始节点S到n的最佳路径的代价，h\*(n)表示从n到某目标节点的最佳路径的代价。
* **估价函数**：f(n)=g(n)+h(n)。其中g是g\*的估计，h是h\*的估计。g(n)是从初始节点S到节点n的实际代价(搜索树中目前从S到n的这段路径的最小代价)，显然有g(n)≥g\*(n)；h(n)是从节点n到目标节点的最优路径的估计代价,依赖于领域的启发信息,称为**启发式函数**。
* 定义1：在GRAPHSEARCH过程中, 如第8步的重排OPEN表是依据f(x)=g(x)+h(x)进行的,则称该过程为A算法。
* 定义2：在A算法中,如对所有的x存在h(x)≤h\*(x),则称h(x)为h\*(x)的下界,它表示某种偏于保守的估计。
* 定义3：采用h\*(x)的下界h(x)为启发函数的A算法，称为A\*算法。当h=0时，A\*算法就变为等代价搜索算法。
* **可纳性**：对任一状态空间图，当从初始节点到目标节点有路径存在时，如搜索算法总能在有限步骤内找到一条从初始节点到目标节点的最佳路径，并在此路径上结束，则称该搜索算法是可采纳的。
* A\* 算法的可纳性：A\* 算法的搜索效率很大程度上取决于估价函数h(n)。一般来说, 在满足h(n)≤h\*(n)的前提下，h(n)的值越大越好。h(n)的值越大，说明它携带的启发性信息越多，A\* 算法搜索时扩展的节点就越少，搜索效率就越高。



（掌握）规则演绎系统

规则演绎系统是运用If→Then规则来建立的问题求解系统，每个if可能与某断言集(事实)中的一个或多个断言匹配，有时把该断言集称为工作内存，then部分用于规定放入工作内存的新断言。在这种系统中，通常称每个if部分为前项，每个then部分为后项。





（理解）消解原理

* 消解原理是一种用于子句公式集的重要推理规则。**子句**是由文字的析取组成的公式，由子句、空子句所构成的集合称为**子句集**。消解原理, 力图从子句集中推导出一个表示矛盾的空子句。
* 定义：若P是原子谓词公式，则称P与﹁P为**互补文字**。
* 定义: 设C1和C2是子句集中的任意两个子句，如果C1中的文字L1与C2中的文字L2互补，那么可从C1和C2中分别消去L1和L2，并将C1和C2中余下的部分按析取关系构成一个新的子句C12，则称这一过程为**归结**，称C12为C1和C2的归结式，称C1和C2为C12的亲本(父辈)子句。

**求取子句集的步骤**：

1. 消去蕴涵符号
2. 小否定符号的辖域(狄·摩根定律)
3. 变量标准化(哑元唯一)
4. 消去存在量词(∀)
5. 化为前束
6. 化为合取范式(∧)
7. 消去全称量词(∀)
8. 消去连词符号(∧)
9. 更换变量名(同一变量名不出现在一个以上子句)

**消解反演推理方法**：把已知S证明L为真的问题，转化为证明S∧﹁ L为不可满足的问题。

1. 给出{S}， L；
2. 否定L，得～L；
3. 把～L添加到S中去，把新产生的集合｛～L, S｝化成子句集；
4. 应用消解原理，力图推出一个表示矛盾的空子句。