|  |
| --- |
| 1) PROCEDURE Match(s)  2) 输入：中间状态，初始状态满足  3) 输出：两个图之间的所有匹配  4) IF M(s) 覆盖了的所有结点  5) 输出M(s)  6) ELSE  7) 计算可以被包含到M(s)中的候选点对集合P(s)  8) FOREACH P(s)中的点对p  9) IF 候选点对满足加入M(s)的所有约束规则  10) 将p加入M(s)，计算  11) CALL Match()  12) END IF  13) END FOREACH  14) 恢复数据状态  15) END IF  16) END PROCEDURE Match |

图xxx VF2子图同构算法流程

Feasibility Rules

符号说明：

N1和N2表示图1和图2中的点集。n和m分别表示图1和图2中的点。Pred(G,n)表示点n在图G中的前驱，Succ(G,n)表示点n在图G中的后继。Tin1(s)和Tin2(s)表示状态s在图1和图2中，指向当前已经匹配的点集的所有边的source点集合（边的起点）。Tout1(s)和Tout2(s)表示状态s在图1和图2中，从当前已经匹配的点集出发的所有边的target点结合（边的终点）。

T1(s)=Tin1(s)∪Tout1(s)，即当前状态s在图1中已经匹配的点集的所有一步邻居。N˜=N1−M1(s)−T1(s)，即图1中，除了s中已经匹配的点，和这些点的一步邻居以外的点。

前两条保证加入新的匹配对p后，两个子图仍然是同构的。设新加入的匹配对是(n,m)，则对于n在图1中的所有前驱（或后继），必须能在图2中m的前驱（或后继）里有相应的点与之对应。同样，对于m在图2中的所有前驱（或后继），也必须能在图1中n的前驱（或后继）里有相应的点与之对应。

之后的三条都是剪枝策略。其中Card表示求集合中元素的个数。

三四条表示，n在Tin1（或Tout1）中的前驱（或后继）的数目，必须大于等于m在Tin2（或Tout2）中的前驱（或后继）的数目。如果不满足，则说明对于query graph中新匹配的点m，其邻居个数是大于target graph中n的邻居个数的，所以说最终必然无法完全匹配query graph中所有的点。

第五条跟三四条思想类似，只不过考虑的两步邻居。具体来说，三四部中的考虑的邻居是Tin1和Tout1中的邻居。这些点即跟n相邻，又跟当前匹配中的其他点相邻。而第五条考虑的邻居，是只跟n相邻，跟当前匹配中其他店不相邻的邻居。这样细粒度的考虑的好处是，可以更细粒度的剪枝，从而提高剪枝效率。

约束1

约束2

约束3

约束4

约束5

放宽约束1、2

新的约束1

新的约束2

**候选集合中点对的属性约束**

算法图结点属性a(n)，架构图结点属性集合A(m)

属性约束规则1

映射最优筛选规则

对算法图中的节点加入拓扑信息（m, n），其中m表示算子出现的纵向上的顺序，n表示算子出现的横向上的顺序

对架构图中的节点加入拓扑信息（x, y），其中x表示PE纵向上的顺序，y表示PE横向上的顺序

规则1

从候选集中找出所有满足R=max(x)=min(max(x))的映射

这是映射最优选择的第一个条件，它保证选出的候选集的映射占有最少的行数

规则2

从候选集中找出所有满足C=max(y)=min(max(y))的映射

它保证选出的候选集的映射占有最少的列数

规则3

，其中表示第i行的y值。

在最少列数的基础上使映射尽可能靠右

规则4

对于映射点对集合M(a(m，n)，b(x，y))

。

最少交叉

规则3

对于映射点对集合M(a(m，n)，b(x，y))

。

尽量靠上