### 第二十四次2020年6月1日-6月7日

#### 1 exact cover

Exact cover：

P = ⟨V, S⟩。V是有限的点的集合，S是V的子集的集合。P的一个解S⋆是S的子集，使得V中的每个点在S⋆中恰好出现一次（只出现在S⋆的一个集合中）精确覆盖问题包含三种类型：决策，计数，枚举。决策回答是否存在解，计数给出解的个数，枚举显示给出每个解。

An exact cover instance is a pair P = ⟨V, S⟩, where V is a finite set of points and S is a set of subsets of V . A solution to P is a subset S⋆ of S such that each point v ∈ V appears in exactly one set in S⋆. The exact cover problem has three natural variants: decision, counting, and enumeration. The first variant asks us to determine whether a solution exists, the second variant asks us to determine the number of solutions, and the third variant asks us to explicitly construct every solution.

#### 2 DLX(舞蹈链)算法

介绍DLX算法之前我们先了解X算法。X算法是一种常规的回溯算法，我们可以用一个递归过程求精确覆盖问题。递归过程为：

1.如果矩阵A为空（没有任何列），则当前局部解即为问题的一个解，返回成功；否则继续。

2.根据一定方法选择第c列。如果某一列中没有1，则返回失败，并去除当前局部解中最新加入的行。

3.选择第r行，使得Ar, c = 1（该步是不确定的）。

4.将第r行加入当前局部解中。

5.对于满足Ar, j = 1的每一列j，从矩阵A中删除所有满足Ai, j = 1的行，最后再删除第j列。

6.对所得比A小的新矩阵递归地执行此算法。

例子：全集U = {1, 2, 3, 4, 5, 6, 7} ，现有U的六个子集S= {A, B, C, D, E, F}

A = {1, 4, 7}；B = {1, 4}；C = {4, 5, 7}；D = {3, 5, 6}；E = {2, 3, 6, 7}；F = {2, 7}。

矩阵表示：

用X算法得出本问题只有一个解：S⋆ = {B, D, F}

高德纳主要想通过X算法体现舞蹈链的实用性。他发现了使用舞蹈链的X算法效率极高，并把这一过程称为DLX。DLX用矩阵来表示精确覆盖问题，在内部的存储结构为舞蹈链。舞蹈链是一个双向环形链表，每个矩阵中的1都有一个指针指向其左、右、上、下的1。因为精确覆盖问题中的矩阵一般都是稀疏的，所以舞蹈链中的元素很少，既很省时间，又很省空间。可见使用舞蹈链的DLX算法无论在选择行时还是回溯错误的选择时效率都很高。

DLX算法一个实例化libexact。