高级算法设计与分析-第三部分

# 计数问题的算法与复杂性

夏盟佶
Xia, Mingji

中科院软件所
计算机科学国家重点实验室

2021

# 接下来五次课

- 5月22日（本次）
- 5月24日
- 5月31日
- 6月7日
- 6月19日星期六（调6月14日星期一端午节所致）

- 理解#CSP等计数问题的定义和其复杂性二分定理，学习张量网络概念，学习若干计数问题（推广的方程组解数目问题、匹配数目问题等）的算法。
- 从张量网络的基变换角度，理解容斥原理、线性检测算法等。
- 了解多项式插值归约证明#P困难性的思路。

张量网络的概念和性质是核心，所以若点名就下次课点名。
习题中的一部分会被指定为作业。

# 向量、矩阵、张量（数组）

- 从形式上看，
  向量即一维数组，
  矩阵即二维数组，
  张量即数组。
- 用$[d]$表示大小为$d$的有限集合。
  例如，用以表示向量的指标变量$i$的取值范围$\{1, 2, \ldots, d\}$。
- $\mathbb{R}^d$里的向量$(F_i)$即一元(离散)函数$F(i) : [d] \to \mathbb{R}$。F(i)取值
  i的位置/取值
- 矩阵$(F_{i,j})$即二元函数$F(i, j)$。[d]×[d]→R
- k维张量$F_{i_1, i_2, \ldots, i_k}$即k元函数$F(i_1, i_2, \ldots, i_k)$。
  超高维矩阵

(后面自变量符号随习惯用$x, y, z, \ldots, x_i, \ldots$，也用图论中表示边的符号$e$。)

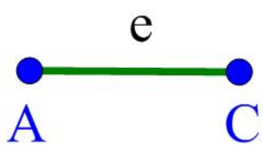
# “张量网络”中，“网络”即图

- 除了图之外，张量的网络还要给出顶点上的函数。
- 当顶点的函数不是对称函数时，要给出它的边的次序。

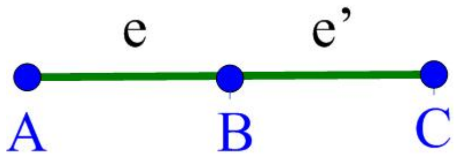
总结：
“张量”即高维数组，我们称为函数，
以避免混淆“张量积”和“张量网络”。

“网络”基本就是图。

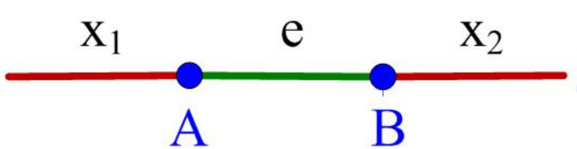
# 用点表示函数，其边表示该函数的自变量




$$AC = \sum_{e\in[d]} A_e C_e$$




$$ABC = \sum_{e,e'\in[d]} A_e B_{ee'} C_{e'}$$




$$AB_{x_1,x_2} = \sum_{e\in[d]} A_{x_1 e} B_{e x_2}$$

# 张量网络

- 图$G(V,E)$的每个点$v$被赋予一个$d_v$元函数$F_v$，$d_v$是$v$的度。
- 有限集$D$表示$F_v$的一个自变量的定义域。
- 边集合$E=\{e_1,\dots,e_m\}$。
- 张量网络的值定义为：

每个矩阵，每个位置的元素乘起来，加和

边/自变量　函数值

$$\sum_{e_1,\dots,e_m\in D}\boxed{\prod_{v\in V}F_v(e_{v,1},e_{v,2},\dots,e_{v,d_v})}$$

边取不同的值 算出的结果加和　张量表达式乘起来

其中$e_{v,1},e_{v,2},\dots,e_{v,d_v}$表示$v$的$d_v$条边。（图$G$记录了此次序。）
- 其他表示：

$$\sum_{\sigma:E\to D}\prod_{v\in V}F_v(\sigma(e_{v,1}),\sigma(e_{v,2}),\dots,\sigma(e_{v,d_v}))$$

$$\sum_{\sigma:E\to D}\prod_{v\in V}F_v(\sigma|_{Neighbor(v)})$$

# 张量网络

- 边是变量，点是函数，点$v$被赋予函数$F_v$。
- $X$是外部边集合，$E$是内部边集合。
- $E$中边有两个顶点。$X$中边有一个顶点。$E = \{e_j | j = 1, \ldots, m\}$。
- 定义一个以$X = \{x_1, x_2, \ldots, x_n\} \in D^n$为输入的函数$F_G$。

$$F_G(x_1, x_2, \ldots, x_n) = \sum_{e_j \in D} \prod_{v \in V} F_v(e_{v,1}, e_{v,2}, \ldots, e_{v,d_v})$$

? 有什么区别

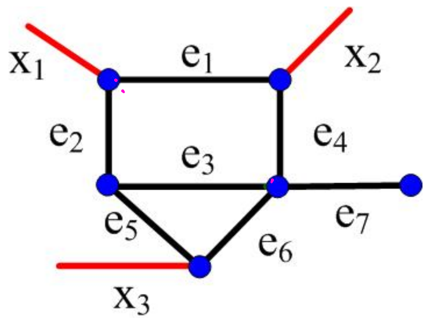



Figure: 图$G(V, E \cup X)$

# 张量网络

- 学习内容：
  - 特殊的张量网络与矩阵运算。
  - 张量网络基本性质—结合律，
    以及由此导出的基变换—全息归约。
  - 张量网络的计算。
    （正向用定义）
  - 示范一些问题如何建模成张量网络。
    （逆向，从原问题求和式出发，找出合适的张量网络）

- 张量网络和许多定义的本质相通。
  - 因子图、布尔线路，是其特殊情况。
  - 量子线路要加入测量等概念和约束。
  - 贝叶斯网络、Gibbs分布、爱因斯坦标记（求和约定）等。
  - 实际上就是构件（gadget），常见于计数问题之间的归约中。

# 迹

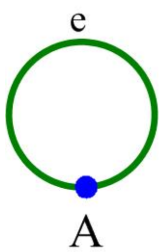
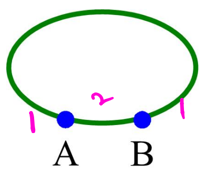



Figure: $\sum_e A_{e,e}$

$A$的迹等于$A'$的迹。

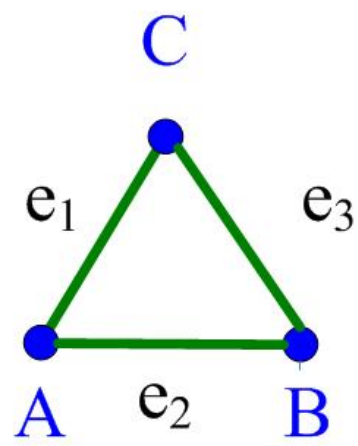



$AB$的迹等于$BA$的迹。

# 矩阵乘积的迹

- $$\mathrm{trace}(ABC) = \sum_{e_1,e_2,e_3\in[d]} A_{e_1e_2}B_{e_2e_3}C_{e_3e_1}$$

  首尾相同




- 相同的张量网络图，
  不同的画法可表示$\mathrm{trace}(ABC)$、$\mathrm{trace}(BCA)$和$\mathrm{trace}(C'B'A')$。
- 实对称矩阵的迹等于特征值之和。
  $B = M\Lambda M^{-1}$, $\mathrm{trace}(B) = \mathrm{trace}(\Lambda)$。
- 量子物理里有partial trace。

# 张量积定义

- 两个矩阵$\mathbf{M}_{dd}, \mathbf{N}_{dd}$，其中

$x_{11}y_{11}$ $\cdots$ $x_{11}y_{1d}$ $x_{12}$
$x_{11}y_{21}$
$\vdots$
$x_{11}y_{d1}$
$x_{21}y_{11}$
$\vdots$

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1d} \\ m_{21} & m_{22} & \dots & m_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ m_{d1} & m_{d2} & \dots & m_{dd} \end{pmatrix}$$

- 它们的张量积是一个$d^2 \times d^2$矩阵，有如下分块矩阵形式：

$$\mathbf{M} \otimes \mathbf{N} = \begin{pmatrix} m_{11}N & m_{12}N & \dots & m_{1d}N \\ m_{21}N & m_{22}N & \dots & m_{1d}N \\ \vdots & \vdots & \ddots & \vdots \\ m_{d1}N & m_{d2}N & \dots & m_{dd}N \end{pmatrix}$$

- 可用$\mathbf{M}$的行指标变量$x_1$和$\mathbf{N}$的行指标变量$x_2$联合起来的$x_1x_2 \in [d^2]$作为$\mathbf{M} \otimes \mathbf{N}$ 的行指标。

?

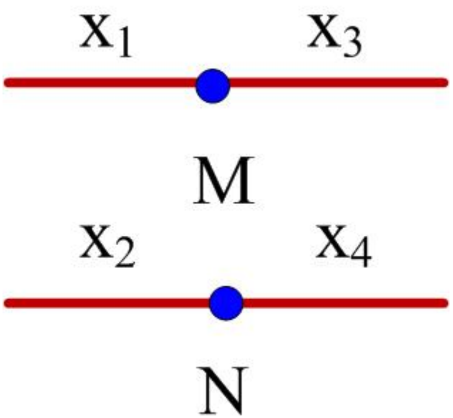
$$(\mathbf{M} \otimes \mathbf{N})_{x_1x_2,x_3x_4} = \mathbf{M}_{x_1,x_3}\mathbf{N}_{x_2,x_4}$$

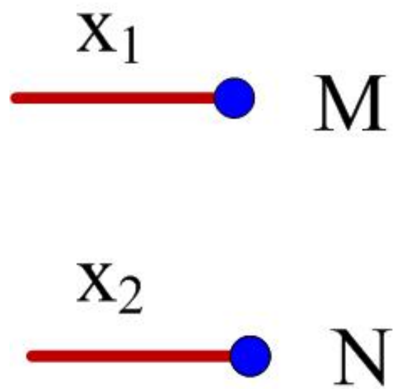
（$x_1x_2$是个符号二元组，不是数字乘法。行指标只是标记

# 张量积

跟张量net表示相同？

不相连的点



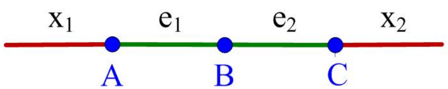



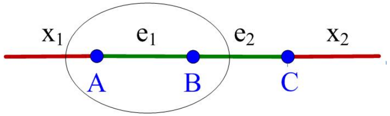

$$(M \otimes N)_{x_1 x_2, x_3 x_4} = M_{x_1, x_3} N_{x_2, x_4}$$

$$(M \otimes N)_{x_1, x_2} = M_{x_1} N_{x_2}$$

（记法$M^{\otimes 3} = M \otimes M \otimes M$）。

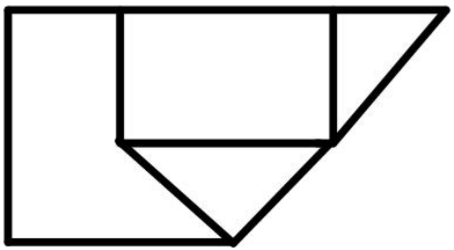
# 结合律

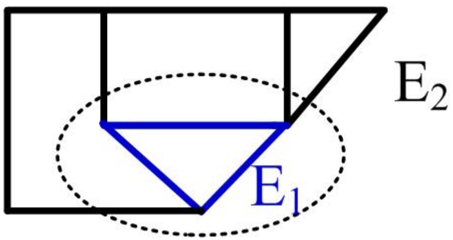



$$\sum_{e_1,e_2} A(x_1,e_1)B(e_1,e_2)C(e_2,x_2)$$

$$\sum_{e_2}(C(e_2,x_2)\sum_{e_1}A(x_1,e_1)B(e_1,e_2))$$

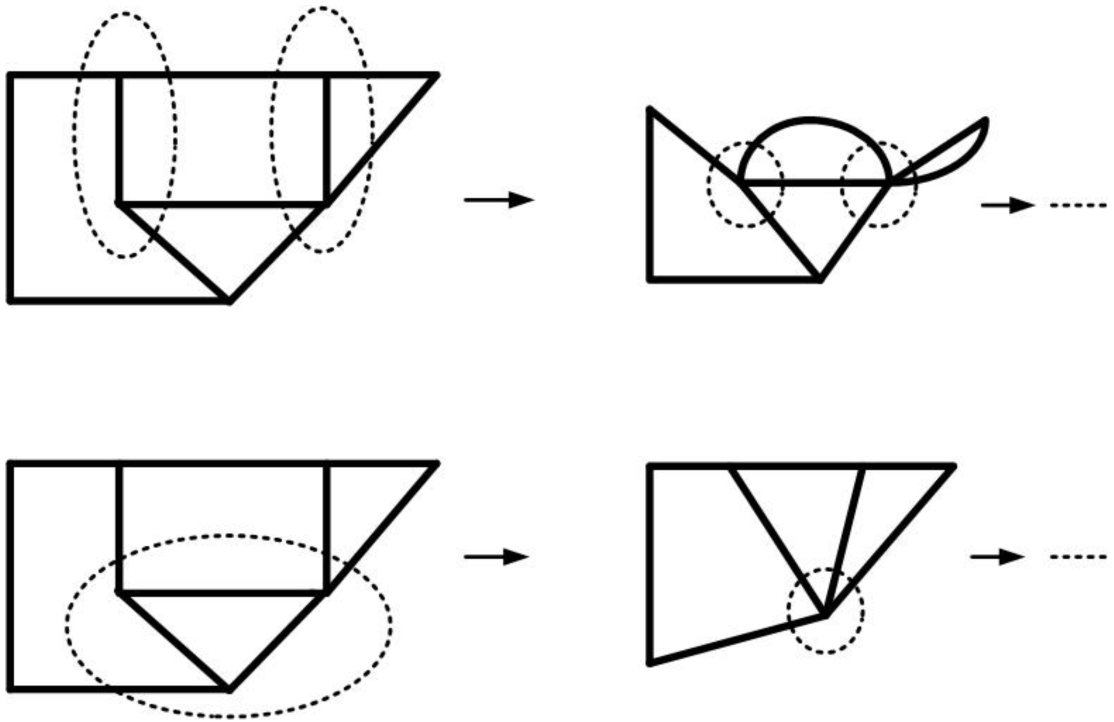





$$\sum_{e\in E_1\cup E_2}\prod_{v\in V_1\cup V_2} F_v$$

$$\sum_{e\in E_2}(\prod_{v\in V_2} F_v \sum_{e\in E_1}\prod_{v\in V_1} F_v)$$

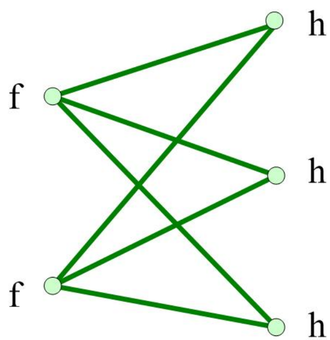
# 定义的不同嵌套次序，结果总相同

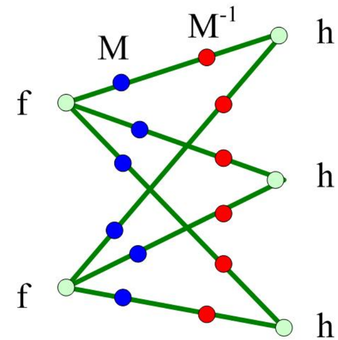


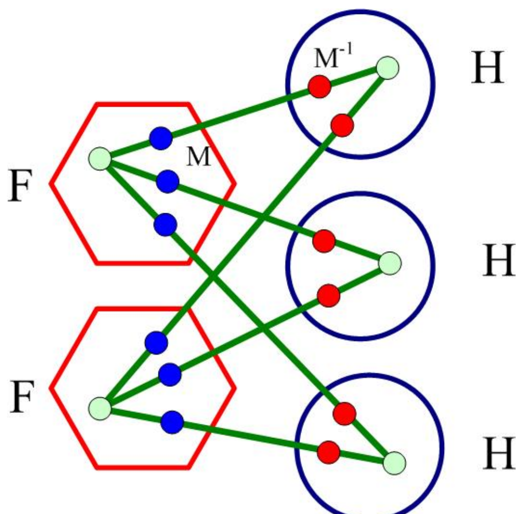
称之为张量网络的结合律。

# 全息归约：Holant定理

$AB = AEB = AMM^{-1}B = (AM)(M^{-1}B)$。($E$是单位阵)




## 定理 (Valiant 2004)

$\#\{F\}|\{H\}$和$\#\{f\}|\{h\}$在相同的图上的值相等。其中，

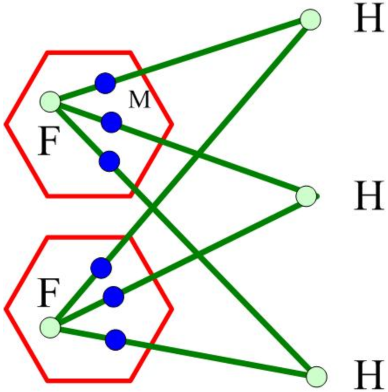
这表示？

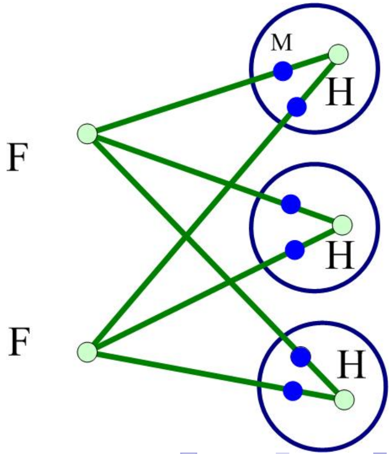
$$F = fM^{\otimes 3},$$

$$(M^{-1})^{\otimes 2}h = H。$$

# 全息归约另一个一般形式

类比$(AB)C = A(BC)$。




**定理**

$\#\{FM^{\otimes 3}\}|\{H\}$和$\#\{F\}|\{M^{\otimes 2}H\}$值相同。

# 迹

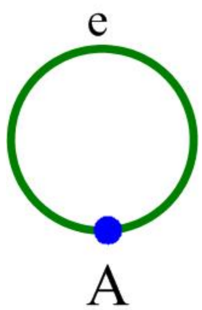



Figure: $\sum_e A_{e,e}$

$A$的迹等于$A'$的迹。

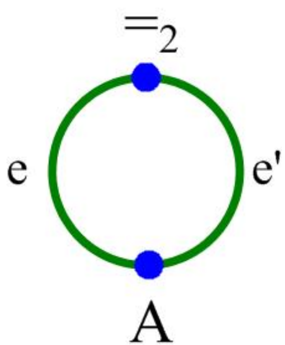



Figure: $\sum_{e,e'} A(e,e')$ "$=_2$ "$(e',e)$

# 布尔变量对称函数的表示

- $F$是对称函数，当且仅当对任意的置换$\pi$，任意$x_1,\ldots,x_n$，

$$F(x_1,\ldots,x_n)=F(x_{\pi(1)},\ldots,x_{\pi(n)})$$

- 布尔变量$x_j\in\{0,1\}$。
- $F$值取决于输入中有多少个0和1。
- 用$f_j$表示输入中有$j$个1时的$F$值，$j=0,1,\ldots,n$。
- 记

$$F=[f_0,f_1,\ldots,f_n]$$

- 记n元相等关系"$=_n$" $=[1,0,\ldots,0,1]$。

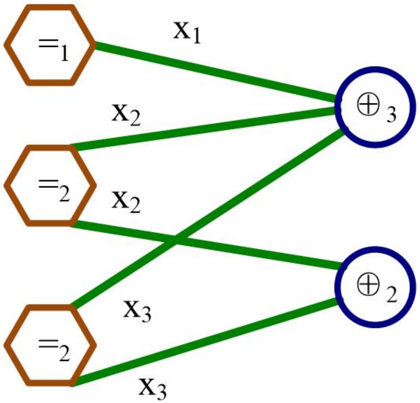
输入全0或全1时为1，否则为0

# 有限域上线性方程组的解集大小→ 张量网络值

把变量定义域[2]，看作大小为2的有限域。

例

$$\begin{cases} x_1 + x_2 + x_3 = 0 \mod 2 \\ x_2 + x_3 = 0 \mod 2 \end{cases}$$



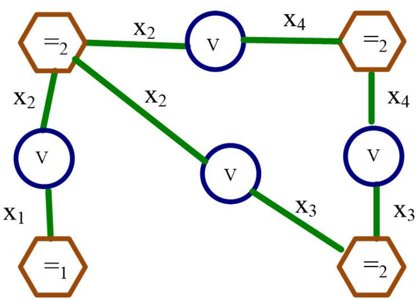

显然有多项式时间算法。by 方程

# 图的顶点覆盖数目问题

变量定义域[2]。

例

$$\begin{cases} x_1 \vee x_2 & = & 1 \\ x_2 \vee x_3 & = & 1 \\ x_2 \vee x_4 & = & 1 \\ x_3 \vee x_4 & = & 1 \end{cases}$$




这个问题是难解的——#P难。

# #P、归约、#P难

不带#：存在prob

# sharp 计数问题 有多少？ harder

- 一个函数$F$在#P中
  当且仅当存在多项式时间算法$R$,
  存在$k$,$R$的两个输入$x$和$y$总满足$|y| = |x|^k$,
  使得$F(x) = |\{y|R(x,y) = 1\}|$。
- 一个计算问题A可以归约到B，指存在一个调用函数B的计算A的多项式时间算法，其中B的运算时间不计在内。
- 如果#P里所有的问题都可以归约到B，B就是#P难的。
- 线性方程组解集大小和顶点覆盖数目，都在#P中。
- 但前者可以多项式时间计算，后者是#P难的。

# 研究对象

- 被研究的计数问题大多是计算具体的张量网络，甚至延申到数学上的函数，例如行列式（易算）、积和式（Permanent，难算）。
- $\mathcal{F}$是一个函数集合，定义Holant($\mathcal{F}$)问题：
- 输入：一个张量网络G，点上的函数来自$\mathcal{F}$。
  输出：G的值。

**例**

齐次线性方程组解的数目问题对应
$Holant(\{=_1, =_2, \ldots, \oplus_1, \oplus_2, \ldots\})$

**主要研究目标**

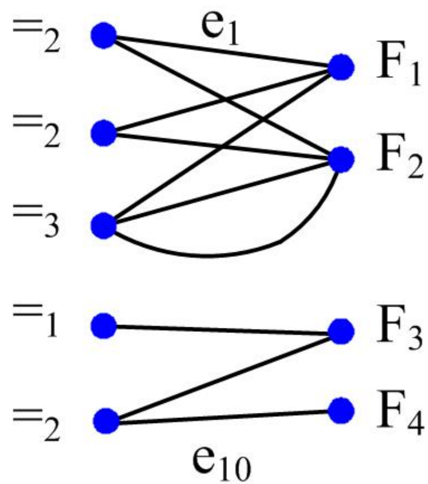
刻画所有Holant($\mathcal{F}$)问题的复杂性。

如果证明了一个问题集合中的问题要么是P的要么是#P难的，就叫做二分定理。

(计数复杂性——精确计数)

# #CSP 与 Holant

- #CSP($\mathcal{H}$)问题计算形如下张量网络。




#CSP与Holant的关系?

- 要求函数$F_i$来自集合$\mathcal{H}$。

是指F是H函数吧?

- #CSP($\mathcal{H}$)问题等价于Holant($\{=_1, =_2, \ldots, \} \cup \mathcal{H}$)。
-
$$\{\#\mathsf{CSP}(\mathcal{H}) | \mathcal{H}\} \subseteq \{\mathsf{Holant}(\mathcal{F}) | \mathcal{F}\}$$

# 布尔定义域复数值域#CSP二分定理中的第一易解类：$\mathcal{A}$

- $X = (\overset{0/1}{x_1}, x_2, \ldots, x_n)$。$x_j \in D = \{0, 1\}$。
- 定义仿射关系函数：$\chi_{(AX=C)}$，即$D$上$n$维空间的仿射子空间的指示函数。（$D$作为大小2的有限域。）
- $P(x_1, x_2, \ldots, x_n)$是一个整系数多项式，$x_j \in \{0, 1\}$，使用整数加法乘法运算。例如，$x_j^2 = x_j$。
- 要求$P$的最高次数是2；要求交错二次项的系数是偶数。例如：$3x_1 + 2x_2 + 2x_1x_3 + 4x_2x_3$。
- $F \in \mathcal{A}$，当且仅当有形式$\chi_{(AX=C)} \cdot i^{P(x_1, x_2, \ldots, x_n)}$。
  A中的函数形式

# #CSP(A)的算法

- $$\sum_{x_j\in\{0,1\},j=1,\ldots,n} \chi_{(AX=C)} i^{P(x_1,x_2,\ldots,x_n)}$$
- 假设$AX=C$的自由变量是$x_1,\ldots,x_r$，
  解是$x_{r+1}=L_{r+1}(X'),\ldots x_n=L_n(X') \mod 2$。$X'=(x_1,\ldots,x_r,1)$。
- 例如，方程组$x_3=x_1+x_2+1 \mod 2$
  的解表达式中$L_3(X')=x_1+x_2+1$。
- $$\sum_{x_j\in\{0,1\},j=1,\ldots,r} i^{P(x_1,x_2,\ldots,x_r,L_{r+1}(X'),\ldots,L_n(X'))} \quad ?$$
- 线性函数$L_j$在模2之后才是正确$x_j$值。
  而$P$采用整数（实际上可以是模4）运算。
- 下面把解表达式的模2运算融入模4运算。
  - 如果$P$里有一个一次项$x_j$，替换成$L_j(X')^2$。
    因为$L^2 \mod 4$等于$L \mod 2$。
  - 如果有交错项$2x_jx_{j'}$，替换成$2L_j(X')L_{j'}(X')$即可。

$$\sum_{x_j\in\{0,1\}} i^{P(x_1,x_2,\ldots,x_r)}$$

- $x_1$的一次项系数是偶数。提取含$2x_1$的项的公因子。

$$\sum_{x_2,\ldots,x_r}\sum_{x_1} i^{2x_1 L(X')+P'(x_2,\ldots,x_r)} = \sum_{x_2,\ldots,x_r}(i^{P'(x_2,\ldots,x_r)}\sum_{x_1}(-1)^{x_1 L(X')})$$

$$\sum_{x_1}(-1)^{x_1 L(X')} = 2\chi_{(L(X')=0)}\text{，其中 } X' = (x_2,\ldots,x_r,1)$$

- $x_1$的一次项系数是奇数。保留一个$x_1$，提取公因子。

$$= \sum_{x_2,\ldots,x_r}(i^{P'(x_2,\ldots,x_r)}\sum_{x_1}(-1)^{x_1 L(X')}{\color{red}i^{x_1}})$$

当$L(X') = 0 \mod 2$时，$F(1,x_2,\ldots,x_r) = iF(0,x_2,\ldots,x_r)$；
当$L(X') = 1 \mod 2$时，$F(1,x_2,\ldots,x_r) = -iF(0,x_2,\ldots,x_r)$。

$$\sum_{x_1}(-1)^{x_1 L(X')} i^{x_1} = (1+i)i^{3L^2(X')}$$

原本是关于$x_1, x_2, \ldots, x_n$的$2^n$个赋值对$\chi \cdot i^P$求和，已经看到了如何消除$\chi$中的非自由变量和$i^P$中的一个变量，代价是函数表达式$\chi \cdot i^P$的幅度受控的变化。
消除一个变量$x_i$，即从对两个大小$2^{n-1}$的超平面的点的函数值求和，转化成对其中一个超平面的点的函数值求和。

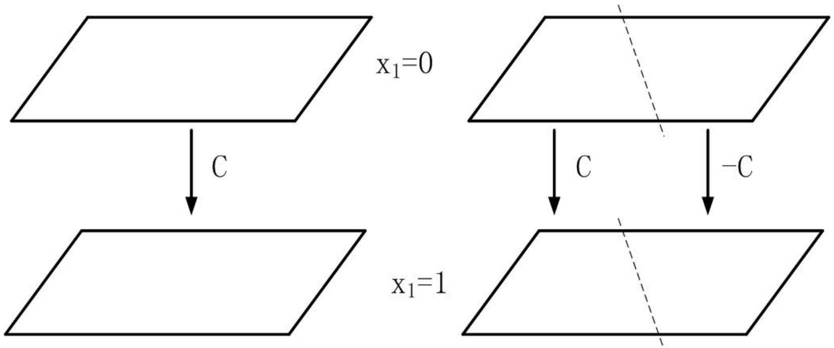



Figure: 虚线右侧区域表示仿射子空间$L(X)=1$

第一种情况，$C=1$。第二种情况，$C=\pm i$。
$(1+i)=i(1-i)$！。

# $\mathcal{A}$中的二元函数例子

- $$F = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$
- #CSP($\{F\}$)问题的一个实例，是一些$F$应用到变量$x_1, \ldots, x_n$。
- 这个实例对应图$G$，$V_G = \{x_1, \ldots, x_n\}$，$(x_j, x_k) \in E_G$当且仅当实例中有约束$F(x_j, x_k)$。
- 考虑被赋值1的顶点形成的子图$H$。如果$H$有奇（偶）数条边，所有约束的乘积是$-1$（resp. $1$）。
- #CSP($\{F\}$)($G$)=图$G$的偶数条边的这种子图数目$-$奇数条边的子图数目。
- 推论：图$G$的偶数条边的子图数目是多项式时间可以计算的。

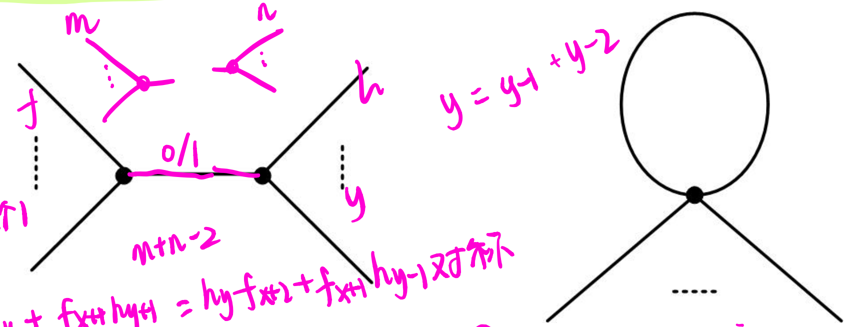
- 刚才的算法来自这篇论文。
- Jin-Yi Cai, Pinyan Lu, Mingji Xia:
  The complexity of complex weighted Boolean #CSP. J. Comput. Syst. Sci. 80(1): 217-236 (2014)
- $\#CSP(\mathcal{A})$与量子colliford线路关系紧密。
- Jin-Yi Cai, Heng Guo, Tyson Williams:
  Clifford gates in the Holant framework. Theor. Comput. Sci. 745: 163-171 (2018)
- BTW
  The Gödel Prize 2021
  Jin-Yi Cai and Xi Chen:
  Complexity of Counting CSP with Complex Weights J. ACM 64(3): 19:1 – 19:39 (2017).

# 脉络

- 我们先从矩阵运算出发，引出了张量网络的定义、结合律。
- 算法与计算复杂性研究关心
  张量网络带来的问题集合Holant={Holant($\mathcal{F}$)}。
- 先假设函数集合中有所有元的相等，被相等绑定的Holant，就成了#CSP。
- 我们看了线性方程组解数目问题，及其推广#CSP($\mathcal{A}$)。
- 接下来我们看一个松绑的Holant问题例子，斐波那契门。
  - 先从封闭性角度看斐波那契门的算法。
  - 再介绍张量网路的基变换——全息归约。
  - 松绑形态才容易被基变换转动，从全息归约角度再给斐波那契门一个算法。

# 算法一：利用封闭性质

- 斐波那契门关于如下两种运算封闭，因而任何斐波那契门构成的连通的张量网络也是斐波那契门。




$f_x h_y + f_{x+1} h_{y+1} = h_y f_{x+2} + f_{x+1} h_{y-1}$ 对称

斐 $S_i = S_{x,y} = S_{x,y-1} + S_{x,y-2}$

例

第二个运算，如果$F = [f_0, f_1, \ldots, f_k]$是斐波那契门，新得到的函数$[f_0 + f_2, f_1 + f_3, \ldots, f_{k-2} + f_k]$也是。

$F = \{f_0, \cdots, f_k\}$
$F' = \{f'_0, \cdots, f'_k\}$
$f'_0 = f_0 + f_2$

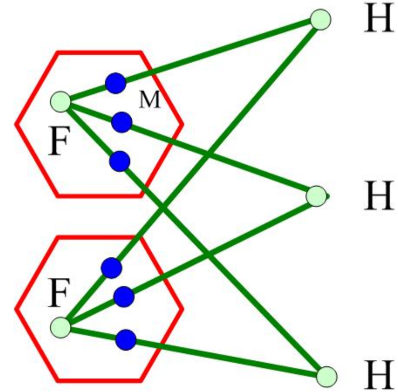
- 斐波那契门有多项式长度的表示，并且作为每种运算的输入和输出是多项式时间可以计算的。

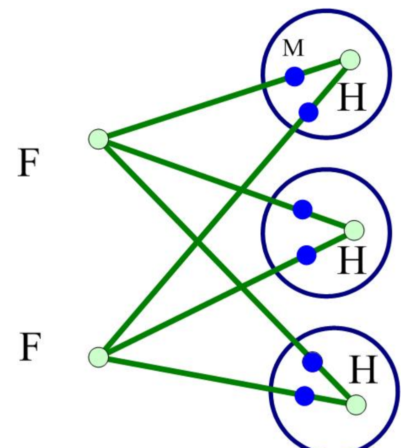
# 算法二：全息算法

- $$f_{i+2} = f_{i+1} + f_i$$
- 设方程$x^2 = x + 1$的两个根是$a, b$。显然$ab = -1$。
- ? $[1, a, a^2, \ldots, a^n]$满足递推关系，是斐波那契门。
- 此函数的指数长真值表向量形式：$(1, a)^{\otimes n}$。
- $$\begin{aligned} & c(1,a)^{\otimes n} + d(1,b)^{\otimes n} \\ = & \left(c(1,0)^{\otimes n} + d(0,1)^{\otimes n}\right) \begin{pmatrix} 1 & a \\ 1 & b \end{pmatrix}^{\otimes n} \\ = & \text{“}[c, 0, \ldots, 0, d]\text{”} \begin{pmatrix} 1 & a \\ 1 & b \end{pmatrix}^{\otimes n} \end{aligned}$$

# 回顾：全息归约

类比特例$(AB)C = A(BC)$。




**定理**

$\#\{FM^{\otimes 3}\}|\{H\}$和$\#\{F\}|\{M^{\otimes 2}H\}$值相同。

# 全息算法的基

- $M = \begin{pmatrix} 1 & a \\ 1 & b \end{pmatrix}$
- 右侧的二元相等变成了
- $$M^{\otimes 2}\text{“}[1,0,1]\text{”}$$
- 它等价于
$$MEM' = \begin{pmatrix} 1 & a \\ 1 & b \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ a & b \end{pmatrix} = \begin{pmatrix} 1+a^2 & 0 \\ 0 & 1+b^2 \end{pmatrix}$$
- 变到了第二易解类#CSP($\mathcal{P}$)——product type。
- 正交矩阵基保持二元相等关系。
- 在#CSP下，无法转动。（被反对角矩阵转动后还是自身。）在Holant下，可以被所有正交基转动。

# 脉络

————前半场脉络——————

- 算法与计算复杂性研究关心
  张量网络带来的问题集合Holant=$\{\mathrm{Holant}(\mathcal{F})\}$。
- 假设函数集合中有所有元的相等，被相等绑定的Holant，就成了#CSP。
- 我们看了线性方程组解数目问题，及其推广#CSP($\mathcal{A}$)。
- Holant（斐波那契门）。（松绑形态下，使用基变换描述易解类似乎必不可少。）

————后半场脉络——————

- 其他全息归约应用：计数算法、线性函数检测……
- Pfaffian和平面图完美匹配
- #P困难性证明中的多项式插值归约
- 拆开张量积的归约与#CSP 的pinning引理
- 介绍Ising模型、六点模型、#EO、Gibbs分布、近似计数、采样

# 前后呼应（跳跃无序）

- 算法与计算复杂性研究关心
  张量网络带来的问题集合Holant={Holant($\mathcal{F}$)}。
- 假设函数集合中有所有元的相等，就成了#CSP。
- 我们看了线性方程组解数目问题，及其推广#CSP($\mathcal{A}$)。函数集合的封闭性
- Holant （斐波那契门）。函数集合的封闭性

- 其他全息归约应用：计数算法、线性函数检测……
- Pfaffian和平面图完美匹配函数集合的封闭性
- #P困难性证明中的多项式插值归约
- 拆开张量积的归约与#CSP 的pinning引理
- 介绍Ising模型、六点模型、#EO、Gibbs分布、近似计数、采样

# 前后呼应（跳跃无序）

- 算法与计算复杂性研究关心
  张量网络带来的问题集合Holant={Holant($\mathcal{F}$)}。
- 假设函数集合中有所有元的相等，就成了#CSP。
- 我们看了线性方程组解数目问题，及其推广#CSP($\mathcal{A}$)。
- Holant （斐波那契门）。

- 其他全息归约应用：计数算法、线性函数检测建模成张量网络而简化计算……
- Pfaffian（建模成张量网络）和平面图完美匹配
- #P困难性证明中的多项式插值归约建模成张量网络而简化证明
- 拆开张量积的归约与#CSP 的pinning引理
- 介绍Ising模型、六点模型、#EO、Gibbs分布、近似计数、采样

# 前后呼应（跳跃无序）

- 算法与计算复杂性研究关心
  张量网络带来的问题集合Holant=$\{\text{Holant}(\mathcal{F})\}$。
- 假设函数集合中有所有元的相等，就成了#CSP（SAT与CSP问题（课程第二部分））。
- 我们看了线性方程组解数目问题，及其推广#CSP($\mathcal{A}$)。
- Holant（斐波那契门）。

- 其他全息归约应用：计数算法、线性函数检测……
- Pfaffian（推广了行列式）和平面图完美匹配（最大匹配的RNC算法（第一部分））
- #P困难性证明中的多项式插值归约
- 拆开张量积的归约与#CSP 的pinning引理
- 介绍Ising模型、六点模型、#EO（蕴含#CSP）、Gibbs分布、近似计数、采样