高级算法设计与分析

全息归约

夏盟佶
Xia, Mingji

中科院软件所
计算机科学国家重点实验室

2016.6

斐波那契门

- 张量网络、函数，也可被叫做线路、门。
- 如果$f_{i+2} = f_{i+1} + f_i$，则称$[f_0, f_1, \ldots, f_k]$斐波那契门。

- 例

$$
\begin{aligned}
F &= [a_0, a_1] = [0, 1] \\
H &= [a_0, a_1, a_2] = [0, 1, 1] \\
K &= [a_0, a_1, a_2, a_3] = [0, 1, 1, 2] \\
\ldots
\end{aligned}
$$

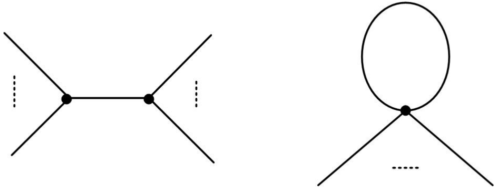
- 斐波那契门构成的张量网络求值，即$\#\{F, H, K, \ldots\}|\{=_2\}$问题，有多项式时间算法。

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。

算法一：利用封闭性质

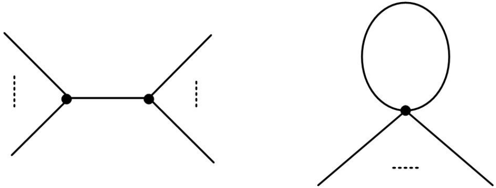
- 斐波那契门关于如下两种运算封闭，因而任何斐波那契门构成的连通的张量网络也是斐波那契门。



-

算法一：利用封闭性质

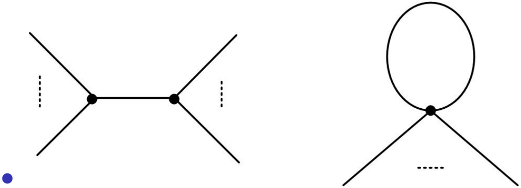
- 斐波那契门关于如下两种运算封闭，因而任何斐波那契门构成的连通的张量网络也是斐波那契门。



-

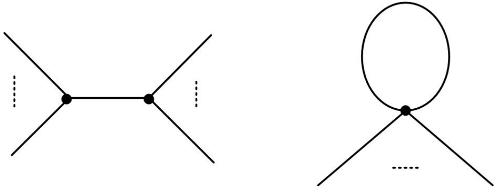
算法一：利用封闭性质

- 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。



- 例
  对第二个运算，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。

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。
- 

- 例

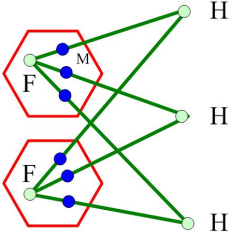
  对第二个运算，如果$F = [f_0, f_1, \ldots, f_k]$是斐波那契门，新得到的函数$[f_0 + f_2, f_1 + f_3, \ldots, f_{k-2} + f_k]$也是。
  - 斐波那契门有多项式长度的表示，并且作为每种运算的输入和输出是多项式时间可计算的。

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：$(1, a)^{\otimes n}$。
- $$\begin{aligned} & c(1,a)^{\otimes n} + d(1,b)^{\otimes n} \\ = & \left(c(1,0)^{\otimes n} + d(0,1)^{\otimes n}\right) \begin{pmatrix} 1 & a \\ 1 & b \end{pmatrix}^{\otimes n} \\ = & \text{“}[c, 0, \ldots, 0, d]\text{”} \begin{pmatrix} 1 & a \\ 1 & b \end{pmatrix}^{\otimes n} \end{aligned}$$

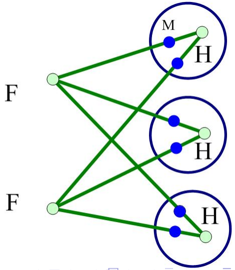
回顾：全息归约

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
$$MEM' = \begin{pmatrix} 1 & a \\ 1 & b \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ a & b \end{pmatrix} = \begin{pmatrix} 1+a^2 & 0 \\ 0 & 1+b^2 \end{pmatrix}$$
- 回顾product type。
- 正交矩阵基保持二元相等关系。

$[x, y, -x, -y, \ldots]$

- 仿照斐波那契门的算法，递推关系是$f_{i+2} + f_i = 0$。

$[x, y, -x, -y, \ldots]$

- 仿照斐波那契门的算法，递推关系是$f_{i+2} + f_i = 0$。
- 设方程$x^2 + 1 = 0$的两个根是$i, -i$。

$[x, y, -x, -y, \ldots]$

- 仿照斐波那契门的算法，递推关系是$f_{i+2} + f_i = 0$。
- 设方程$x^2 + 1 = 0$的两个根是$i, -i$。
- $$\begin{aligned} & c(1,i)^{\otimes n} + d(1,-i)^{\otimes n} \\ = & \left(c(1,0)^{\otimes n} + d(0,1)^{\otimes n}\right) \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}^{\otimes n} \end{aligned}$$

$[x, y, -x, -y, \ldots]$

- 仿照斐波那契门的算法，递推关系是$f_{i+2} + f_i = 0$。
- 设方程$x^2 + 1 = 0$的两个根是$i, -i$。
- $$\begin{aligned} & c(1, i)^{\otimes n} + d(1, -i)^{\otimes n} \\ = & \left(c(1,0)^{\otimes n} + d(0,1)^{\otimes n}\right) \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}^{\otimes n} \end{aligned}$$
- $M = \begin{pmatrix} 1 & a \\ 1 & b \end{pmatrix}$

$[x, y, -x, -y, \ldots]$

- 仿照斐波那契门的算法，递推关系是$f_{i+2} + f_i = 0$。
- 设方程$x^2 + 1 = 0$的两个根是$i, -i$。
- $$\begin{aligned} & c(1, i)^{\otimes n} + d(1, -i)^{\otimes n} \\ = & \left(c(1,0)^{\otimes n} + d(0,1)^{\otimes n}\right) \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}^{\otimes n} \end{aligned}$$
- $M = \begin{pmatrix} 1 & a \\ 1 & b \end{pmatrix}$
- 右侧的二元相等变成了

$[x, y, -x, -y, \dots]$

- 仿照斐波那契门的算法，递推关系是$f_{i+2} + f_i = 0$。
- 设方程$x^2 + 1 = 0$的两个根是$i, -i$。
-
$$\begin{aligned} & c(1, i)^{\otimes n} + d(1, -i)^{\otimes n} \\ = & \left(c(1,0)^{\otimes n} + d(0,1)^{\otimes n}\right) \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}^{\otimes n} \end{aligned}$$
- $M = \begin{pmatrix} 1 & a \\ 1 & b \end{pmatrix}$
- 右侧的二元相等变成了
-
$$M^{\otimes 2} \text{“}[1, 0, 1]\text{”}$$

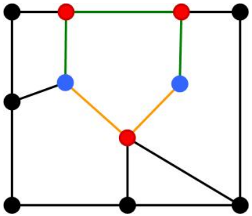
$[x, y, -x, -y, \ldots]$

- 仿照斐波那契门的算法，递推关系是$f_{i+2} + f_i = 0$。
- 设方程$x^2 + 1 = 0$的两个根是$i, -i$。
- $$\begin{aligned} & c(1, i)^{\otimes n} + d(1, -i)^{\otimes n} \\ = & \left(c(1,0)^{\otimes n} + d(0,1)^{\otimes n}\right) \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}^{\otimes n} \end{aligned}$$
- $M = \begin{pmatrix} 1 & a \\ 1 & b \end{pmatrix}$
- 右侧的二元相等变成了
- $$M^{\otimes 2} \text{“}[1, 0, 1]\text{”}$$
- 它等价于
$$MEM' = \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix} = \begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix} = 2\text{“}[0, 1, 0]\text{”}$$

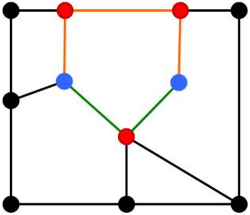
非偶图值为0

- 奇数长度的“ $\neq_2$ ”环不能被满足。
-

非偶图值为0

- 奇数长度的“$\neq_2$”环不能被满足。
- 直接从$[x, y, -x, -y, \ldots]$的张量网络看：
- 

非偶图值为0

- 奇数长度的“$\neq_2$”环不能被满足。
- 直接从$[x, y, -x, -y, \ldots]$的张量网络看：
- 

布尔函数的傅里叶形式

- $F:\{0,1\}^n \to \{0,1\}$
- 线性函数：$F(X+Y)=F(X)+F(Y)$
- $F(X) = a_1x_1 + a_2x_2 + \cdots + c_nx_n$

- $F:\{1,-1\}^n \to \{1,-1\}$
- 线性函数：$F(X\odot Y)=F(X)F(Y)$
- $F(X) = \chi_s(X) = \prod_{j\in s} x_j$

- F的真值表是一个2^n维向量，有标准基。

布尔函数的傅里叶形式

- $F:\{0,1\}^n \to \{0,1\}$
- 线性函数：
  $F(X+Y)=F(X)+F(Y)$
- $F(X)=a_1x_1+a_2x_2+\cdots+c_nx_n$

- $F:\{1,-1\}^n \to \{1,-1\}$
- 线性函数：
  $F(X\odot Y)=F(X)F(Y)$
- $F(X)=\chi_s(X)=\prod_{j\in s}x_j$

- F的真值表是一个2^n维向量，有标准基。
- 线性函数集合$\{\chi_s|s\subseteq[n]\}$是一组基。

布尔函数的傅里叶形式

- $F : \{0,1\}^n \to \{0,1\}$
- 线性函数：$F(X+Y) = F(X)+F(Y)$
- $F(X) = a_1x_1 + a_2x_2 + \cdots + c_nx_n$

- $F : \{1,-1\}^n \to \{1,-1\}$
- 线性函数：$F(X \odot Y) = F(X)F(Y)$
- $F(X) = \chi_s(X) = \prod_{j\in s} x_j$

- F的真值表是一个2^n维向量，有标准基。
- 线性函数集合$\{\chi_s | s \subseteq [n]\}$是一组基。

布尔函数的傅里叶形式

- $F:\{0,1\}^n \to \{0,1\}$
- 线性函数：$F(X+Y)=F(X)+F(Y)$
- $F(X) = a_1x_1 + a_2x_2 + \cdots + c_nx_n$

- $F:\{1,-1\}^n \to \{1,-1\}$
- 线性函数：$F(X\odot Y)=F(X)F(Y)$
- $F(X) = \chi_s(X) = \prod_{j\in s} x_j$

- F的真值表是一个2^n维向量，有标准基。
- 线性函数集合$\{\chi_s|s\subseteq[n]\}$是一组基。

- Proof.

Hadmard矩阵$H=\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$满秩。

$H^{\otimes n}$的行就是$\{\chi_s|s\subseteq[n]\}$。 $\square$

布尔函数的傅里叶形式

- $F:\{0,1\}^n \to \{0,1\}$
- 线性函数：$F(X+Y)=F(X)+F(Y)$
- $F(X)=a_1x_1+a_2x_2+\cdots+c_nx_n$

- $F:\{1,-1\}^n \to \{1,-1\}$
- 线性函数：$F(X\odot Y)=F(X)F(Y)$
- $F(X)=\chi_s(X)=\prod_{j\in s}x_j$

- F的真值表是一个2^n维向量，有标准基。
- 线性函数集合$\{\chi_s|s\subseteq[n]\}$是一组基。

- Proof.

Hadmard矩阵$H=\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$满秩。

$H^{\otimes n}$的行就是$\{\chi_s|s\subseteq[n]\}$。 □

- F的傅里叶形式，就是它在线性函数基下的坐标向量，

$$\hat{F}=H^{\otimes n}F,\quad \hat{F}(s)=\langle F,\chi_s\rangle$$

F与线性函数的距离

$\hat{F}$的长度:

- $\langle F, F\rangle = 2^n$
- $\hat{F} = H^{\otimes n} F$
- $\langle \hat{F}, \hat{F}\rangle = 2^{2n}$

F与线性函数的距离:

F与线性函数的距离

$\hat{F}$的长度:

- $\langle F, F\rangle = 2^n$
- $\hat{F} = H^{\otimes n}F$
- $\langle \hat{F}, \hat{F}\rangle = 2^{2n}$

F与线性函数的距离:

- $\hat{F}(s) = \langle F, \chi_s\rangle$
- $\hat{F}(s)$等于F和χ_s的相同点数目减去不同点数目。
- 如果F与所有χ_s的最小相对距离是ρ（不同点数除以总点数），
  $\forall s, \quad \hat{F}(s) \le (1-2\rho)2^n$。

线性检测

- 目的：检测F是否接近一个线性函数。
- 随机检测算法：随机抽取X, Y，如果$F(X \odot Y) = F(X)F(Y)$，就通过检测；否则，拒绝。
- 用p表示F被检测拒绝的概率。
- 如果F是线性函数，$p = 0$。

线性检测

- 目的：检测F是否接近一个线性函数。
- 随机检测算法：随机抽取X, Y，如果$F(X \odot Y) = F(X)F(Y)$，就通过检测；否则，拒绝。
- 用p表示F被检测拒绝的概率。
- 如果F是线性函数，$p = 0$。

- 定理

如果F与所有χ_s的最小相对距离是ρ，拒绝概率$p \geq \rho$。

线性检测

- 目的：检测F是否接近一个线性函数。
- 随机检测算法：随机抽取X,Y，如果$F(X \odot Y) = F(X)F(Y)$，就通过检测；否则，拒绝。
- 用p表示F被检测拒绝的概率。
- 如果F是线性函数，$p = 0$。

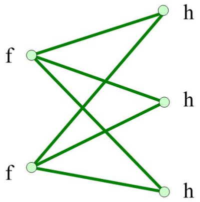
- **定理**
如果F与所有χ_s的最小相对距离是ρ，拒绝概率$p \geq \rho$。
  - 证明：

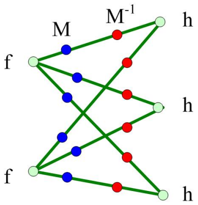
$$\sum_s \hat{F}^3(s) \leq \max_s \hat{F}(s) \langle \hat{F}, \hat{F} \rangle = (1-2\rho)2^{3n}$$

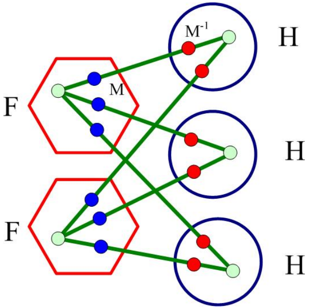
$(1-2p)2^{2n}$是2^{2n}组(X,Y)中，通过的数目减去被拒绝的数目，
只需证明这个数目是$\frac{1}{2^n}\sum_s \hat{F}^3(s)$。

回顾：全息归约

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







定理

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

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

$$(M^{-1})^{\otimes 2}g = G。$$

全息归约：线性检测的张量网络值是$\frac{1}{2^n}\sum_s \hat{F}^3(s)$

线性检测的张量网络：

- 其值应是线性检测中，通过的数目减去被拒绝的数目，即$\sum_{X,Y} F(X)F(Y)F(X \odot Y)$。

全息归约：

全息归约：线性检测的张量网络值是$\frac{1}{2^n}\sum_s \hat{F}^3(s)$

线性检测的张量网络：

- 其值应是线性检测中，通过的数目减去被拒绝的数目，即$\sum_{X,Y} F(X)F(Y)F(X\odot Y)$。
- 右侧：三个函数F。

全息归约：

全息归约：线性检测的张量网络值是$\frac{1}{2^n}\sum_s \hat{F}^3(s)$

线性检测的张量网络：

- 其值应是线性检测中，通过的数目减去被拒绝的数目，即$\sum_{X,Y} F(X)F(Y)F(X \odot Y)$。
- 右侧：三个函数F。
- 第一个函数F的n条边是$x_1, x_2, \ldots, x_n$，第二个函数F的n条边是$y_1, y_2, \ldots, y_n$，第三个函数F的n条边是$z_1, z_2, \ldots, z_n$。

全息归约：

全息归约：线性检测的张量网络值是$\frac{1}{2^n}\sum_s \hat{F}^3(s)$

线性检测的张量网络：

- 其值应是线性检测中，通过的数目减去被拒绝的数目，即$\sum_{X,Y} F(X)F(Y)F(X\odot Y)$。
- 右侧：三个函数F。
- 第一个函数F的n条边是$x_1, x_2, \ldots, x_n$，第二个函数F的n条边是$y_1, y_2, \ldots, y_n$，第二个函数F的n条边是$z_1, z_2, \ldots, z_n$。
- 希望对任何$j = 1, 2, \ldots, n$，$z_j = x_j \odot y_j$。

全息归约：

全息归约：线性检测的张量网络值是$\frac{1}{2^n}\sum_s \hat{F}^3(s)$

线性检测的张量网络：

- 其值应是线性检测中，通过的数目减去被拒绝的数目，即$\sum_{X,Y} F(X)F(Y)F(X \odot Y)$。
- 右侧：三个函数F。
- 第一个函数F的n条边是$x_1, x_2, \ldots, x_n$，第二个函数F的n条边是$y_1, y_2, \ldots, y_n$，第二个函数F的n条边是$z_1, z_2, \ldots, z_n$。
- 希望对任何$j = 1, 2, \ldots, n$，$z_j = x_j \odot y_j$。
- 左侧：第j个函数是$[1, 0, 1, 0]$（记为$\oplus_3$），作用于x_j, y_j, z_j。

全息归约：

全息归约：线性检测的张量网络值是$\frac{1}{2^n}\sum_s \hat{F}^3(s)$

线性检测的张量网络：

- 其值应是线性检测中，通过的数目减去被拒绝的数目，即$\sum_{X,Y} F(X)F(Y)F(X \odot Y)$。
- 右侧：三个函数F。
- 第一个函数F的n条边是$x_1, x_2, \ldots, x_n$，第二个函数F的n条边是$y_1, y_2, \ldots, y_n$，第二个函数F的n条边是$z_1, z_2, \ldots, z_n$。
- 希望对任何$j = 1, 2, \ldots, n$，$z_j = x_j \odot y_j$。
- 左侧：第j个函数是$[1, 0, 1, 0]$（记为$\oplus_3$），作用于x_j, y_j, z_j。

全息归约：

- 用H作用于右侧三个函数F。

全息归约：线性检测的张量网络值是$\frac{1}{2^n}\sum_s \hat{F}^3(s)$

线性检测的张量网络：

- 其值应是线性检测中，通过的数目减去被拒绝的数目，即$\sum_{X,Y} F(X)F(Y)F(X \odot Y)$。
- 右侧：三个函数F。
- 第一个函数F的n条边是$x_1, x_2, \ldots, x_n$，第二个函数F的n条边是$y_1, y_2, \ldots, y_n$，第二个函数F的n条边是$z_1, z_2, \ldots, z_n$。
- 希望对任何$j = 1, 2, \ldots, n$，$z_j = x_j \odot y_j$。
- 左侧：第j个函数是$[1, 0, 1, 0]$（记为$\oplus_3$），作用于x_j, y_j, z_j。

全息归约：

- 用H作用于右侧三个函数F。
- 用$\frac{1}{2}H$作用于左侧n个函数$\oplus_3$。

全息归约：线性检测的张量网络值是$\frac{1}{2^n}\sum_s \hat{F}^3(s)$

线性检测的张量网络：

- 其值应是线性检测中，通过的数目减去被拒绝的数目，即$\sum_{X,Y} F(X)F(Y)F(X\odot Y)$。
- 右侧：三个函数F。
- 第一个函数F的n条边是$x_1,x_2,\ldots,x_n$，第二个函数F的n条边是$y_1,y_2,\ldots,y_n$，第二个函数F的n条边是$z_1,z_2,\ldots,z_n$。
- 希望对任何$j=1,2,\ldots,n$，$z_j=x_j\odot y_j$。
- 左侧：第j个函数是$[1,0,1,0]$（记为$\oplus_3$），作用于x_j,y_j,z_j。

全息归约：

- 用H作用于右侧三个函数F。
- 用$\frac{1}{2}H$作用于左侧n个函数$\oplus_3$。
- 右侧变成$\hat{F}$。

全息归约：线性检测的张量网络值是$\frac{1}{2^n}\sum_s \hat{F}^3(s)$

线性检测的张量网络：

- 其值应是线性检测中，通过的数目减去被拒绝的数目，即$\sum_{X,Y} F(X)F(Y)F(X\odot Y)$。
- 右侧：三个函数F。
- 第一个函数F的n条边是$x_1,x_2,\dots,x_n$，第二个函数F的n条边是$y_1,y_2,\dots,y_n$，第二个函数F的n条边是$z_1,z_2,\dots,z_n$。
- 希望对任何$j=1,2,\dots,n$，$z_j=x_j\odot y_j$。
- 左侧：第j个函数是$[1,0,1,0]$（记为$\oplus_3$），作用于x_j,y_j,z_j。

全息归约：

- 用H作用于右侧三个函数F。
- 用$\frac{1}{2}H$作用于左侧n个函数$\oplus_3$。
- 右侧变成$\hat{F}$。
- 左侧变成$\frac{1}{8}$“$[1,0,1,0]$”$H^{\otimes 3}=\frac{1}{2}$“$[1,0,0,1]$”。

全息归约：线性检测的张量网络值是$\frac{1}{2^n}\sum_s \hat{F}^3(s)$

线性检测的张量网络：

- 其值应是线性检测中，通过的数目减去被拒绝的数目，即$\sum_{X,Y} F(X)F(Y)F(X\odot Y)$。
- 右侧：三个函数F。
- 第一个函数F的n条边是$x_1,x_2,\dots,x_n$，第二个函数F的n条边是$y_1,y_2,\dots,y_n$，第三个函数F的n条边是$z_1,z_2,\dots,z_n$。
- 希望对任何$j=1,2,\dots,n$，$z_j=x_j\odot y_j$。
- 左侧：第j个函数是$[1,0,1,0]$（记为$\oplus_3$），作用于x_j,y_j,z_j。

全息归约：

- 用H作用于右侧三个函数F。
- 用$\frac{1}{2}H$作用于左侧n个函数$\oplus_3$。
- 右侧变成$\hat{F}$。
- 左侧变成$\frac{1}{8}$“$[1,0,1,0]$”$H^{\otimes 3}=\frac{1}{2}$“$[1,0,0,1]$”。
- 新左侧要求新三组边X',Y',Z'相同。

线性子空间指示函数的傅里叶形式

- 全息归约的思想也可用于gadget，或者说开放的张量网络。
- 在Holant问题的二分定理中有很多这样不拘一格的用法。
- 用线性子空间的指示函数的傅里叶形式作为例子，
    - 在简单背景下展示一次较灵活的运用的方式，
    - 复习一下H，$=_k$，$\bigoplus_k$。

全息归约的一些更复杂的应用场景：

- 与多项式插值归约结合证明#P困难性
- 大定义域下的应用
    - 大小为3的定义域
    - 变定义域应用

- Hadmard矩阵$H = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
- k元相等函数$=_k$，$[1, 0, \ldots, 0, 1]$。
- k元奇偶性函数$\oplus_k$，$[1, 0, 1, 0 \ldots]$。
- H变$=_k$为$\oplus_k$，变$\oplus_k$为$=_k$。
- 布尔定义域的线性子空间，是一个其次线性方程组的解集合。

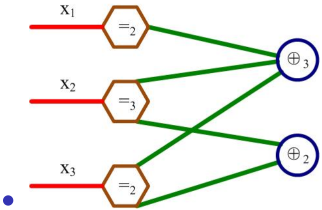
例

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

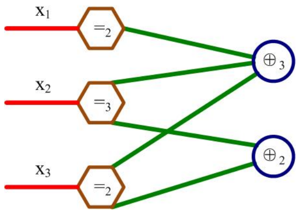
χ_L的张量网络

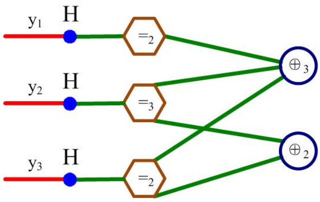
χ_L的张量网络

χ_L的张量网络

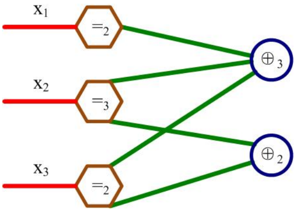


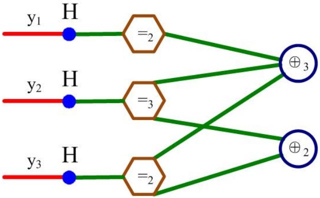
χ_L的张量网络

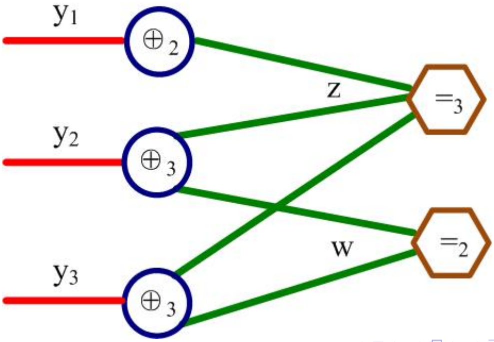
- 



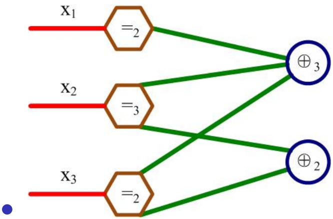
χ_L的张量网络

- 

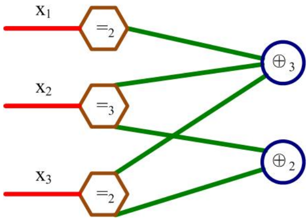


- 

“正交补空间”

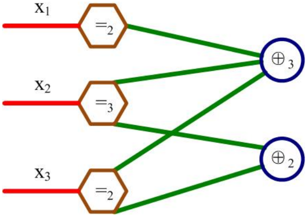


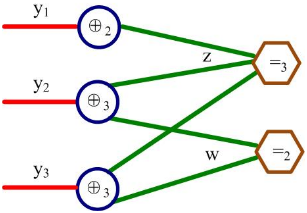
“正交补空间”



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

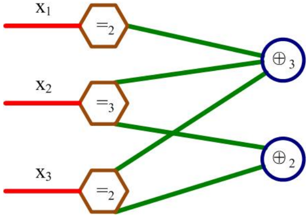
“正交补空间”

- 

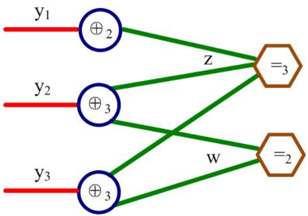
- 

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

“正交补空间”

- 

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

- 

$$\begin{cases} y_1 = z \\ y_2 = z + w \\ y_3 = z + w \end{cases}$$

回顾积和式——偶图的带权完美匹配数目

- A是$n \times n$矩阵。

回顾积和式——偶图的带权完美匹配数目

- A是$n \times n$矩阵。

定义

$$\mathit{Permanent}(A) = \sum_{\pi \in S_n} \prod_{j=1}^{n} A_{j,\pi(j)}$$

回顾积和式——偶图的带权完美匹配数目

- A是$n \times n$矩阵。

定义

$$\mathit{Permanent}(A) = \sum_{\pi \in S_n} \prod_{j=1}^{n} A_{j,\pi(j)}$$

- 按照此定义计算，需要$\mathcal{O}(n\ n!)$步。

Ryser公式

- $$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots \\ a_{21} & a_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$
- 考虑$\phi(A) = (a_{11}+a_{12}+\cdots+a_{1n})(a_{21}+a_{22}+\cdots+a_{2n})\cdots(a_{n1}+a_{n2}+\cdots+a_{nn})$
- 令
$$S_k = \sum_{B\text{是}A\text{的}n\times(n-k)\text{子式}} \phi(B)$$
- $$\text{Permanent}(A) = \sum_{k=0}^{n-1}(-1)^k S_0$$

全息归约解释

- A有两种图表示：n个点的有向图，和$2n$个点的偶图。

全息归约解释

- A有两种图表示：n个点的有向图，和$2n$个点的偶图。
- $V=\{1,2,\ldots,n\}$，$U=\{1',2',\ldots,n'\}$。
  无向偶图$H(V,U,E,W)$中，边(j,k')的权重$W(j,k)=A_{j,k}$。

全息归约解释

- A有两种图表示：n个点的有向图，和$2n$个点的偶图。
- $V = \{1, 2, \ldots, n\}$，$U = \{1', 2', \ldots, n'\}$。
  无向偶图$H(V, U, E, W)$中，边(j, k')的权重$W(j, k) = A_{j,k}$。
- Permanent(A)是H的所有完美匹配权重之和。

全息归约解释

- A有两种图表示：n个点的有向图，和$2n$个点的偶图。
- $V = \{1, 2, \ldots, n\}$，$U = \{1', 2', \ldots, n'\}$。
  无向偶图$H(V, U, E, W)$中，边(j, k')的权重$W(j, k) = A_{j,k}$。
- Permanent(A)是H的所有完美匹配权重之和。
- U, V的点都用函数$[0, 1, 0, \ldots, 0]$。

全息归约解释

- A有两种图表示：n个点的有向图，和$2n$个点的偶图。
- $V=\{1,2,\ldots,n\}$，$U=\{1',2',\ldots,n'\}$。
  无向偶图$H(V,U,E,W)$中，边(j,k')的权重$W(j,k)=A_{j,k}$。
- Permanent(A)是H的所有完美匹配权重之和。
- U,V的点都用函数$[0,1,0,\ldots,0]$。
- Permanent(A)是张量网络H的值。

全息归约解释

- A有两种图表示：n个点的有向图，和$2n$个点的偶图。
- $V=\{1,2,\ldots,n\}$，$U=\{1',2',\ldots,n'\}$。
  无向偶图$H(V,U,E,W)$中，边(j,k')的权重$W(j,k)=A_{j,k}$。
- Permanent(A)是H的所有完美匹配权重之和。
- U,V的点都用函数$[0,1,0,\ldots,0]$。
- Permanent(A)是张量网络H的值。
- 考虑张量网络H'，把V的点的函数换成$[0,1,\ldots,1]$，值不变。（想想原因）

全息归约解释

- A有两种图表示：n个点的有向图，和$2n$个点的偶图。
- $V = \{1, 2, \ldots, n\}$，$U = \{1', 2', \ldots, n'\}$。
  无向偶图$H(V, U, E, W)$中，边(j, k')的权重$W(j, k) = A_{j,k}$。
- Permanent(A)是H的所有完美匹配权重之和。
- U, V的点都用函数$[0, 1, 0, \ldots, 0]$。
- Permanent(A)是张量网络H的值。
- 考虑张量网络H'，把V的点的函数换成$[0, 1, \ldots, 1]$，值不变。（想想原因）
- $[0, 1, \ldots, 1] = (1, 1)^{\otimes n} - (1, 0)^{\otimes n}$

全息归约解释

- A有两种图表示：n个点的有向图，和$2n$个点的偶图。
- $V = \{1, 2, \ldots, n\}$，$U = \{1', 2', \ldots, n'\}$。
  无向偶图$H(V, U, E, W)$中，边(j, k')的权重$W(j, k) = A_{j,k}$。
- Permanent(A)是H的所有完美匹配权重之和。
- U, V的点都用函数$[0, 1, 0, \ldots, 0]$。
- Permanent(A)是张量网络H的值。
- 考虑张量网络H'，把V的点的函数换成$[0, 1, \ldots, 1]$，值不变。（想想原因）
- $[0, 1, \ldots, 1] = (1, 1)^{\otimes n} - (1, 0)^{\otimes n}$
- V的n点函数从$\{(1, 1)^{\otimes n}, -(1, 0)^{\otimes n}\}$中选一个，选完后是$n$个星，易算。

全息归约解释

- A有两种图表示：n个点的有向图，和$2n$个点的偶图。
- $V=\{1,2,\ldots,n\}$，$U=\{1',2',\ldots,n'\}$。
  无向偶图$H(V,U,E,W)$中，边(j,k')的权重$W(j,k)=A_{j,k}$。
- Permanent(A)是H的所有完美匹配权重之和。
- U,V的点都用函数$[0,1,0,\ldots,0]$。
- Permanent(A)是张量网络H的值。
- 考虑张量网络H'，把V的点的函数换成$[0,1,\ldots,1]$，值不变。（想想原因）
- $[0,1,\ldots,1]=(1,1)^{\otimes n}-(1,0)^{\otimes n}$
- V的n点函数从$\{(1,1)^{\otimes n},-(1,0)^{\otimes n}\}$中选一个，选完后是$n$个星，易算。
- 若都选$(1,1)^{\otimes n}$，值是$\phi(A)=(a_{11}+a_{12}+\cdots+a_{1n})(a_{21}+a_{22}+\cdots+a_{2n})\cdots(a_{n1}+a_{n2}+\cdots+a_{nn})$。

全息归约解释

- A有两种图表示：n个点的有向图，和$2n$个点的偶图。
- $V=\{1,2,\dots,n\}$，$U=\{1',2',\dots,n'\}$。
  无向偶图$H(V,U,E,W)$中，边(j,k')的权重$W(j,k)=A_{j,k}$。
- Permanent(A)是H的所有完美匹配权重之和。
- U,V的点都用函数$[0,1,0,\dots,0]$。
- Permanent(A)是张量网络H的值。
- 考虑张量网络H'，把V的点的函数换成$[0,1,\dots,1]$，值不变。（想想原因）
- $[0,1,\dots,1]=(1,1)^{\otimes n}-(1,0)^{\otimes n}$
- V的n点函数从$\{(1,1)^{\otimes n},-(1,0)^{\otimes n}\}$中选一个，选完后是$n$个星，易算。
- 若都选$(1,1)^{\otimes n}$，值是$\phi(A)=(a_{11}+a_{12}+\dots+a_{1n})(a_{21}+a_{22}+\dots+a_{2n})\cdots(a_{n1}+a_{n2}+\dots+a_{nn})$。
- 张量秩为2，经过全息变换之后，相当于有两个值的变量。

- 一般图的完美匹配数目也有2^npoly(n)算法，有没有全息归约解释？
- 张量网络H和H'的值相等，是全息归约定理逆命题不成立的情况之一。
- Jin-Yi Cai, Heng Guo, Tyson Williams:
  A complete dichotomy rises from the capture of vanishing signatures: extended abstract. STOC 2013: 635-644

References

- 斐波那契门算法和$\#([x,y,-x,-y])$的算法见：
  Jin-yi Cai, Pinyan Lu, Mingji Xia:
  A Computational Proof of Complexity of Some Restricted Counting Problems. TAMC 2009: 138-149
  Jin-yi Cai, Pinyan Lu, Mingji Xia:
  Computational Complexity of Holant Problems. SIAM J. Comput. 40(4): 1101-1132 (2011)
  （只有此篇文中有全息归约。其他全息归约解释参考文献中无。）
- 线性检测和线性子空间指标函数的傅里叶形式见：
  《Analysis of Boolean Functions》Ryan O'Donnell
  （张量网络和全息归约还可解释其他一些内容）
- 积和式的算法见：
  https://en.wikipedia.org/wiki/Computing_the_permanent