高级算法设计与分析

# 张量网络

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

# 张量网络

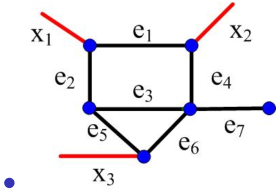


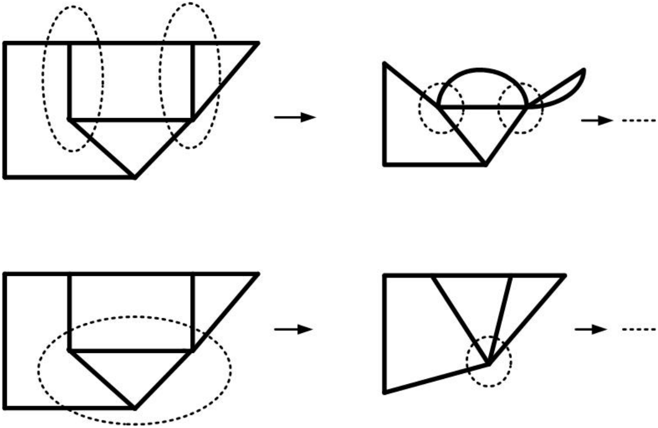

Figure: 图$G(V, E \cup X)$

- 边是变量，点是函数，点$v$被赋予函数$F_v$。
- $E$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$F_G$。

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

- $X$是空集时，定义了一个值。

## 嵌套使用，结果总相同




称之为张量网络的结合律。

# 计数问题及其Gadget

- 张量网络中，$X$外部边，$E$内部边。

# 计数问题及其Gadget

- 张量网络中，$X$外部边，$E$内部边。
- 局部约束定义的计数问题类#CSP, Holant等，都是问一个不含外部边的张量网络的值。

# 计数问题及其Gadget

- 张量网络中，$X$外部边，$E$内部边。
- 局部约束定义的计数问题类#CSP, Holant等，都是问一个不含外部边的张量网络的值。
- 不同计数问题类的要求不一样，例如，天生自带（免费使用）的函数不同，图的要求（平面图、偶图等）不同。

# 计数问题及其Gadget

- 张量网络中，$X$外部边，$E$内部边。
- 局部约束定义的计数问题类#CSP, Holant等，都是问一个不含外部边的张量网络的值。
- 不同计数问题类的要求不一样，例如，天生自带（免费使用）的函数不同，图的要求（平面图、偶图等）不同。
- 同一类中，材料不同，问题不同。
  #CSP($\mathcal{F}$)中，$\mathcal{F}$是可以使用的函数。

# 计数问题及其Gadget

- 张量网络中，$X$外部边，$E$内部边。
- 局部约束定义的计数问题类#CSP, Holant等，都是问一个不含外部边的张量网络的值。
- 不同计数问题类的要求不一样，例如，天生自带（免费使用）的函数不同，图的要求（平面图、偶图等）不同。
- 同一类中，材料不同，问题不同。
  #CSP($\mathcal{F}$)中，$\mathcal{F}$是可以使用的函数。
- #CSP($\mathcal{F}$)的实例，表示成张量网络，要求是偶图，左侧顶点的函数来自$\{=_j | j \in \mathbb{N}\}$，右侧顶点的函数都来自$\mathcal{F}$。

# 计数问题及其Gadget

- 张量网络中，$X$外部边，$E$内部边。
- 局部约束定义的计数问题类#CSP, Holant等，都是问一个不含外部边的张量网络的值。
- 不同计数问题类的要求不一样，例如，天生自带（免费使用）的函数不同，图的要求（平面图、偶图等）不同。
- 同一类中，材料不同，问题不同。
  #CSP($\mathcal{F}$)中，$\mathcal{F}$是可以使用的函数。
- #CSP($\mathcal{F}$)的实例，表示成张量网络，要求是偶图，左侧顶点的函数来自$\{=_j | j \in \mathbb{N}\}$，右侧顶点的函数都来自$\mathcal{F}$。
- Holant（Read-twice CSP）问题更一般，左侧都是$\{=_2\}$。

# 计数问题及其Gadget

- 张量网络中，$X$外部边，$E$内部边。
- 局部约束定义的计数问题类#CSP, Holant等，都是问一个不含外部边的张量网络的值。
- 不同计数问题类的要求不一样，例如，天生自带（免费使用）的函数不同，图的要求（平面图、偶图等）不同。
- 同一类中，材料不同，问题不同。
  #CSP($\mathcal{F}$)中，$\mathcal{F}$是可以使用的函数。
- #CSP($\mathcal{F}$)的实例，表示成张量网络，要求是偶图，左侧顶点的函数来自$\{=_j \mid j \in \mathbb{N}\}$，右侧顶点的函数都来自$\mathcal{F}$。
- Holant（Read-twice CSP）问题更一般，左侧都是$\{=_2\}$。
- 不可混淆问题集合的包含关系，和两个问题的实例集合的包含关系。

# 计数问题及其Gadget

- 张量网络中，$X$外部边，$E$内部边。
- 局部约束定义的计数问题类#CSP, Holant等，都是问一个不含外部边的张量网络的值。
- 不同计数问题类的要求不一样，例如，天生自带（免费使用）的函数不同，图的要求（平面图、偶图等）不同。
- 同一类中，材料不同，问题不同。
#CSP($\mathcal{F}$)中，$\mathcal{F}$是可以使用的函数。
- #CSP($\mathcal{F}$)的实例，表示成张量网络，要求是偶图，左侧顶点的函数来自$\{=_j \mid j \in \mathbb{N}\}$，右侧顶点的函数都来自$\mathcal{F}$。
- Holant（Read-twice CSP）问题更一般，左侧都是$\{=_2\}$。
- 不可混淆问题集合的包含关系，和两个问题的实例集合的包含关系。
- 最朴素的$A \le B$归约方法，就是构造$B$中的构件（Gadget,即张量网络）模拟$A$中的函数。

# 函数的矩阵形式

- 设$F$是一个$n+m$元函数，$x_1,\ldots,x_n,y_1,\ldots,y_m$是它的输入。对应$d^n\times d^m$的矩阵$M_F=(m_{x_1\cdots x_n,y_1\cdots y_m})$，其中

$$m_{x_1\cdots x_n,y_1\cdots y_m}=F(x_1,\ldots,x_n,y_1,\ldots,y_m)。$$

特别的，若取$m=0$，就表示成了列向量。

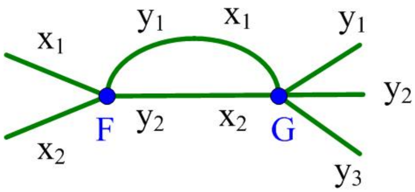
# 函数的矩阵形式

- 设$F$是一个$n+m$元函数，$x_1,\ldots,x_n,y_1,\ldots,y_m$是它的输入。对应$d^n\times d^m$的矩阵$M_F=(m_{x_1\cdots x_n,y_1\cdots y_m})$，其中
$$m_{x_1\cdots x_n,y_1\cdots y_m}=F(x_1,\ldots,x_n,y_1,\ldots,y_m)。$$
特别的，若取$m=0$，就表示成了列向量。
- 如下张量网络的函数可用矩阵乘法表示：

## 函数的矩阵形式

- 设$F$是一个$n+m$元函数，$x_1,\ldots,x_n,y_1,\ldots,y_m$是它的输入。对应$d^n\times d^m$的矩阵$M_F=(m_{x_1\cdots x_n,y_1\cdots y_m})$，其中

$$m_{x_1\cdots x_n,y_1\cdots y_m}=F(x_1,\ldots,x_n,y_1,\ldots,y_m)。$$

特别的，若取$m=0$，就表示成了列向量。
- 如下张量网络的函数可用矩阵乘法表示：
- 


$$M_F M_G$$

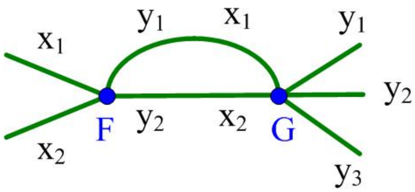
# 函数的矩阵形式

- 设$F$是一个$n+m$元函数，$x_1,\dots,x_n,y_1,\dots,y_m$是它的输入。对应$d^n\times d^m$的矩阵$M_F=(m_{x_1\cdots x_n,y_1\cdots y_m})$，其中

$$m_{x_1\cdots x_n,y_1\cdots y_m}=F(x_1,\dots,x_n,y_1,\dots,y_m)。$$

特别的，若取$m=0$，就表示成了列向量。

- 如下张量网络的函数可用矩阵乘法表示：

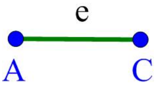
- 


$$M_F M_G$$

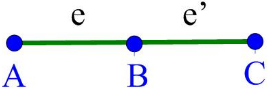
- 尽量遵循行（列）标的变量画在左（右）边。矩阵转置后怎么画？

# 张量网络特例:向量矩阵乘法

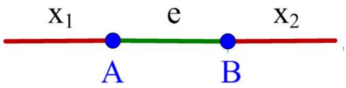
$$AC = \sum_{e\in[d]} A_e C_e$$

- 


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

- 


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

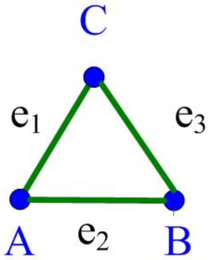
- 

# 特例：迹

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

# 特例：迹

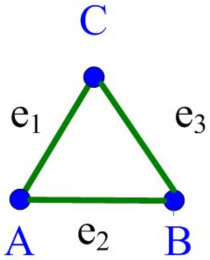
- $$\text{trace}(ABC) = \sum_{e_1,e_2,e_3\in[d]} A_{e_1e_2}B_{e_2e_3}C_{e_3e_1}$$




- 

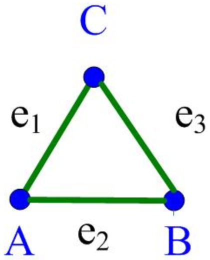
# 特例：迹

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

- 

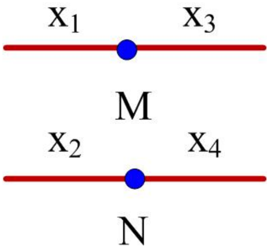

- 相同的张量网络图，
  不同的画法可表示$\mathrm{trace}(BCA)$和$\mathrm{trace}(C'B'A')$。

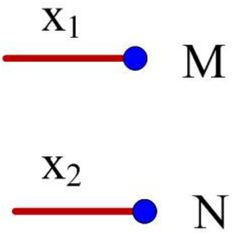
## 特例：迹

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

- 相同的张量网络图，
  不同的画法可表示$\text{trace}(BCA)$和$\text{trace}(C'B'A')$。
- 量子物理里用到partial trace。

# 特例：张量积




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

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

# 零元函数的张量积


$$(M \otimes N) = MN$$

**推论**

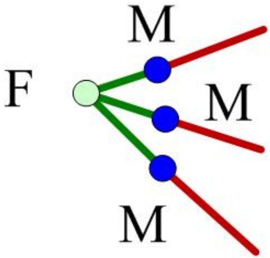
一个无外部边的张量网络的值，是它各个连通分支的值的乘积。

*Proof.*

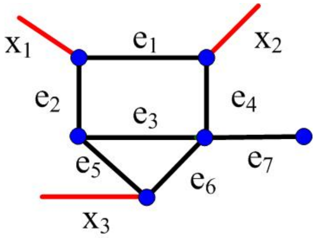
先用张量网络的结合律，把每个连通分支缩成点，然后零元函数的张量积。 $\square$

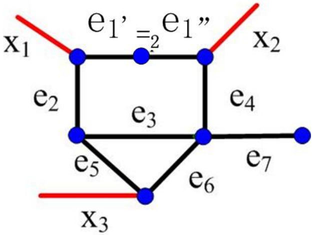
# 矩阵乘法和张量积的联合表示一个张量网络

$F(M^{\otimes 3})$



# 一条边实际上也是 "$=_2$"



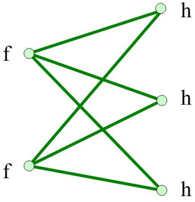



# 全息归约：Holant定理

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

# 全息归约：Holant定理

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

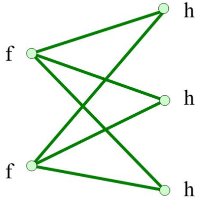


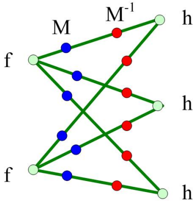
# 全息归约：Holant定理

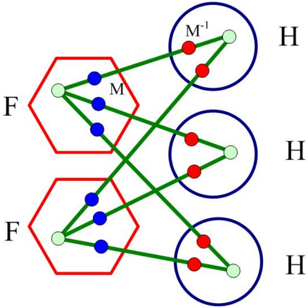
$AB = AEB = AMM^{-1}B = (AM)(M^{-1}B)$。($E$是单位阵)

# 全息归约：Holant定理

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

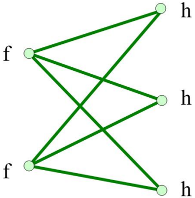


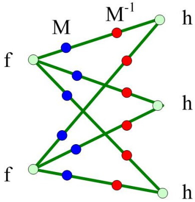



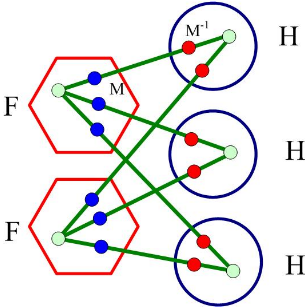



# 全息归约：Holant定理

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



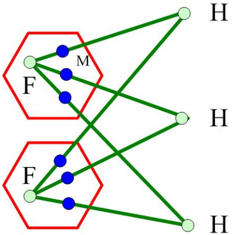

## 定理 (Valiant 2004)

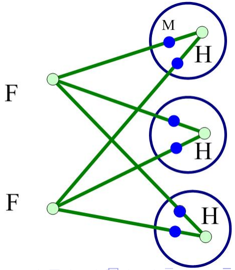
$\#\{F\}|\{G\}$和$\#\{f\}|\{g\}$在相同的图上的值相等。其中，

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

$$(M^{-1})^{\otimes 2} g = G。$$

类比$(AB)C = A(BC)$。



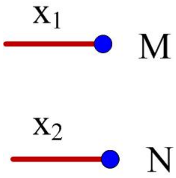

**定理**

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

# 图同态数目问题的一个易解类

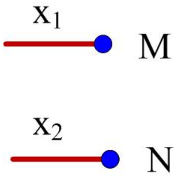
- 图H同态数目问题，问输入图$G$到$H$的同态映射数目。
- 就是一个二元函数$H$定义的#CSP问题。
- 如果二元函数$H$的矩阵形式的秩小于等于1，有多项式时间算法。
- 值域非负实数时，假设$H$联通，这是二分定理的一个易解类。

# $H$秩为1时的算法




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

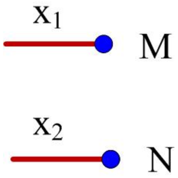
# $H$秩为1时的算法




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

- 因为$H$秩为1，设$H = ab'$。


# $H$秩为1时的算法




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

- 因为$H$秩为1，设$H = ab'$。

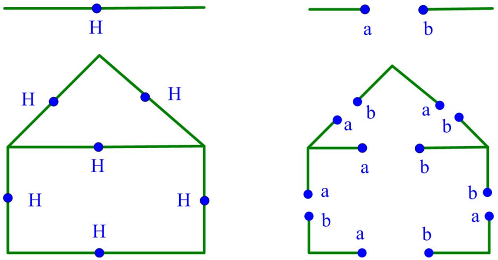



Figure: 作为输入的张量网络的两种等价形式

# 回顾——用图证明代数运算律

- 矩阵乘法和张量积的结合律，它们之间的分配律。
- 迹与矩阵乘法的律。
- 全息归约。（张量网络中的基变换）
- 二次张量积等同列向量乘行向量。（用于解释图同态易解类）

# 参考文献


- Matthew Cook, Networks of Relations, Ph.D Thesis 2005. (判定问题)
- https://simons.berkeley.edu/workshops/qhc2014-3 (workshop "Tensor Networks and Simulations", in simons institute for the theory of computing )
- http://arxiv.org/abs/1603.03039
- http://arxiv.org/abs/1306.2164