高级算法设计与分析

# 计数问题

夏盟佶
Xia, Mingji
（后三章）

中科院软件所
计算机科学国家重点实验室

2016.6

# 教学大纲

- 第十六章计数算法
  1）#P
  2）全息算法
- 第十七章密码学
  1）交互式证明系统
  2）零知识
- 第十八章流、提炼和抽样
  1）流数据的频率高阶矩
  2）抽样矩阵算法
  3）物理中的抽样

# 研究对象、目的、方法

- 计数问题
  问解的数目的计算问题，例如，线性方程组解的个数，#SAT；生成树数目，哈密尔顿回路数目。
  其他一些问题，例如，Permenant(积和式)，Pfaffian，行列式。
- 算法与复杂性
  （严格）计数问题：多项式时间算法，#P难。
  其他：参数计数问题的算法与复杂性，FPT与#W[1]难；近似计数问题；计数问题的指数时间算法。
- 归约方法
  全息归约，多项式插值归约；
  MCMC，correlation decay。

## 重点是深入理解一个定义

一条主线：
全息归约的张量网络解释，以及各种应用。
（张量网络就是计数问题及其gadget，理解透了就理解了全息归约）

# 重点是深入理解一个定义

一条主线：
全息归约的张量网络解释，以及各种应用。
（张量网络就是计数问题及其gadget，理解透了就理解了全息归约）

- 计数问题与复杂性二分定理
- 从张量网络到全息归约
- 全息归约的五个应用例子
- 完美匹配问题的平面图、参数、模情形

# #P

- $F : \Sigma^* \to \mathbf{N}$ (e.g. $\Sigma = \{0, 1\}$)

例

完美匹配数目问题（*#PerfectMatching*）
输入：图$G$（的编码）
输出：$G$的完美匹配数目

# #P

- $F : \Sigma^* \rightarrow \mathbf{N}$ (e.g. $\Sigma = \{0, 1\}$)

例

完美匹配数目问题（#PerfectMatching）
输入：图G（的编码）
输出：G的完美匹配数目

- 定义

$F \in \#P$
当且仅当存在多项式时间不确定型图灵机$M$，使得$M(x)$的接受计算路径数目等于$F(x)$，
当且仅当存在多项式时间算法$R$，$R$的两个输入$x$和$y$总满足$|y| = |x|^k$，使得$F(x) = |\{y|R(x, y) = 1\}|$。

# #P

- $F : \Sigma^* \to \mathbf{N}$ (e.g. $\Sigma = \{0, 1\}$)

例

完美匹配数目问题（*#PerfectMatching*）
输入：图*G*（的编码）
输出：*G*的完美匹配数目

- 定义

$F \in \#P$
当且仅当存在多项式时间不确定型图灵机$M$，使得$M(x)$的接受计算路径数目等于$F(x)$，
当且仅当存在多项式时间算法$R$，$R$的两个输入$x$和$y$总满足$|y| = |x|^k$，使得$F(x) = |\{y | R(x, y) = 1\}|$。

  - NP里的判定问题问有没有证据$y$，#P里的计数问题问有多少证据。

# #P

- $F : \Sigma^* \rightarrow \mathbf{N}$ (e.g. $\Sigma = \{0, 1\}$)

例

完美匹配数目问题（*#PerfectMatching*）
输入：图*G*（的编码）
输出：*G*的完美匹配数目

- 定义

$F \in \#P$
当且仅当存在多项式时间不确定型图灵机$M$，使得$M(x)$的接受计算路径数目等于$F(x)$，
当且仅当存在多项式时间算法$R$，$R$的两个输入$x$和$y$总满足$|y| = |x|^k$，使得$F(x) = |\{y|R(x, y) = 1\}|$。

  - NP里的判定问题问有没有证据$y$，#P里的计数问题问有多少证据。
  - 这个类由L. Valiant于1979年在文章“The complexity of computing the permanent”, Theoretical Computer Science，中首次提出。

# #P难

# #P难

- 定义

  $\#P$难问题：

  如果$\#P$里的所有问题都可以多项式时间图灵归约到$F$，那么$F$就是$\#P$难问题。

# #P难

- **定义**
  #P难问题：
  如果#P里的所有问题都可以多项式时间图灵归约到$F$，那么$F$就是#P难问题。
  - 如果一个问题是#P难的，那么也是NP难的。

# #P难

- 定义
  #P难问题：
  如果#P里的所有问题都可以多项式时间图灵归约到$F$，那么$F$就是#P难问题。
  - 如果一个问题是#P难的，那么也是NP难的。
  - 一个问题的计数版本是#P难的，和这个问题的判定版本是NP难的这两个命题没有关系。

# #P难

- 定义
  #P难问题：
  如果#P里的所有问题都可以多项式时间图灵归约到$F$，那么$F$就是#P难问题。
  - 如果一个问题是#P难的，那么也是NP难的。
  - 一个问题的计数版本是#P难的，和这个问题的判定版本是NP难的这两个命题没有关系。
  - Toda定理：
    $$PH \subseteq P^{\#P}。$$

# #P难问题

- #SAT，Permanent，哈密尔顿回路数目问题，……

# #P难问题

- #SAT，Permanent，哈密尔顿回路数目问题，……

**Theorem**

*#SAT是#P难的。*

# #P难问题

- #SAT，Permanent，哈密尔顿回路数目问题，……

**Theorem**

*#SAT是#P难的。*

- 证明类似cook定理。

# #P难问题

- #SAT，Permanent，哈密尔顿回路数目问题，……

**Theorem**

*#SAT是#P难的。*

- 证明类似cook定理。

**Theorem**

$0,1-$*Permanent是#P难的。*

# #P难问题

- #SAT，Permanent，哈密尔顿回路数目问题，……

**Theorem**

*#SAT是#P难的。*

- 证明类似cook定理。

**Theorem**

$0,1-$*Permanent是#P难的。*

- 因为#SAT可以归约到Permanent，并且归约有传递性。

# Permanent：偶图的带权完美匹配数目

- $A$是$n \times n$矩阵。

# Permanent：偶图的带权完美匹配数目

- $A$是$n \times n$矩阵。

**定义**

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

# Permanent：偶图的带权完美匹配数目

- $A$是$n \times n$矩阵。

**定义**

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

- $A$有两种图表示：$n$个点的有向图，和$2n$个点的偶图。

# Permanent：偶图的带权完美匹配数目

- $A$是$n \times n$矩阵。

**定义**

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

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

# Permanent：偶图的带权完美匹配数目

- $A$是$n \times n$矩阵。

定义

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

- $A$有两种图表示：$n$个点的有向图，和$2n$个点的偶图。
- $V = \{1, 2, \dots, n\}$
  有向图$G(V, E, W)$中，有向边$(j, k)$的权重$W(j, k) = A_{j,k}$。
- Permanent$(A)$是$G$的所有圈覆盖权重之和。

# Permanent：偶图的带权完美匹配数目

- $A$是$n \times n$矩阵。

**定义**

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

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

# Permanent：偶图的带权完美匹配数目

- $A$是$n \times n$矩阵。

**定义**

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

- $A$有两种图表示：$n$个点的有向图，和$2n$个点的偶图。
- $V = \{1, 2, \ldots, n\}$
  有向图$G(V, E, W)$中，有向边$(j, k)$的权重$W(j, k) = A_{j,k}$。
- Permanent$(A)$是$G$的所有圈覆盖权重之和。
- $V = \{1, 2, \ldots, n\}$，$U = \{1', 2', \ldots, n'\}$。
  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。

# 计数版本与判定版本

- #SAT是#P难的，其判定版本SAT是NP难的。
- 偶图的完美匹配数目问题是#P难的，其判定版本偶图是否存在完美匹配，是有多项式时间算法的。
  用图的最大匹配算法即可。
- #2SAT是#P难的，其判定版本2SAT有多项式时间算法。

## 介于难和容易之间的问题

- 如果P不等于NP，存在NP中的问题，它不在P中，也不是NP难的。（非此即彼不成立）
- 证明是采用延迟对角线方法的构造性证明。
- 即一个人造的问题例子。

# 介于难和容易之间的问题

- 如果P不等于NP，存在NP中的问题，它不在P中，也不是NP难的。（非此即彼不成立）
- 证明是采用延迟对角线方法的构造性证明。
- 即一个人造的问题例子。

与之对立的是复杂性二分定理。

- 绝大多数研究过的NP中的问题，或者是NP难的，或者在P里。
- 一个问题集合中的问题要么是容易的（P），要么是难的（NP难），这种结果称为复杂性二分定理。
- 一种常见的问题集合，CSP（约束满足）问题。

# Dichotomy theorem of CSP

$\mathcal{F}$ is a set of relations in Boolean variables.

**Theorem (Schaefer, STOC 1978)**

*Given a constraint set $\mathcal{F}$, the problem CSP($\mathcal{F}$) is in P, if $\mathcal{F}$ satisfies one of the conditions below, and CSP($\mathcal{F}$) is other wise NP-complete.*

- *$\mathcal{F}$ is 0-valid (1-valid).*
- *$\mathcal{F}$ is weakly positive (weakly negative). (Horn SAT)*
- *$\mathcal{F}$ is affine. (A system of linear equations)*
- *$\mathcal{F}$ is bijunctive. (2SAT)*

# #CSP的二分定理

布尔定义域的#CSP问题

- $\{0, 1\}$值域
  只有一类问题有多项式时间算法，其他都是#P难的。
  一类：仿射关系。

# #CSP的二分定理

布尔定义域的#CSP问题

- $\{0,1\}$值域
  只有一类问题有多项式时间算法，其他都是#P难的。
  一类：仿射关系。
- 非负实数值域
  两类：pure affine和product type

# #CSP的二分定理

布尔定义域的#CSP问题

- $\{0,1\}$值域
  只有一类问题有多项式时间算法，其他都是#P难的。
  一类：仿射关系。
- 非负实数值域
  两类：pure affine和product type
- 复数值域
  两类： $\mathcal{A}$ 和 $\mathcal{P}$(即product type)

## 第二易解类：product type

- 一个函数在集合$\mathcal{P}$，当且仅当它是一些一元函数、二元相等函数、二元不等函数的乘积。严格定义：

## 第二易解类：product type

- 一个函数在集合$\mathcal{P}$，当且仅当它是一些一元函数、二元相等函数、二元不等函数的乘积。严格定义：
- $$F : \{0,1\}^n \to \mathbb{C}$$

## 第二易解类：product type

- 一个函数在集合$\mathcal{P}$，当且仅当它是一些一元函数、二元相等函数、二元不等函数的乘积。严格定义：
- $$F : \{0,1\}^n \to \mathbb{C}$$
- 对称函数指输入变量的次序不影响函数值。

## 第二易解类：product type

- 一个函数在集合$\mathcal{P}$，当且仅当它是一些一元函数、二元相等函数、二元不等函数的乘积。严格定义：
- $$F:\{0,1\}^n \to \mathbb{C}$$
- 对称函数指输入变量的次序不影响函数值。
- 输入变量都是布尔变量，函数值只取决于有多少个1，即输入串的hamming weight。

## 第二易解类：product type

- 一个函数在集合$\mathcal{P}$，当且仅当它是一些一元函数、二元相等函数、二元不等函数的乘积。严格定义：
- $$F : \{0,1\}^n \to \mathbb{C}$$
- 对称函数指输入变量的次序不影响函数值。
- 输入变量都是布尔变量，函数值只取决于有多少个1，即输入串的hamming weight。
- $$F = [f_0, f_1, \ldots, f_n]$$

# 第二易解类：product type

- 一个函数在集合$\mathcal{P}$，当且仅当它是一些一元函数、二元相等函数、二元不等函数的乘积。严格定义：
- $$F : \{0,1\}^n \to \mathbb{C}$$
- 对称函数指输入变量的次序不影响函数值。
- 输入变量都是布尔变量，函数值只取决于有多少个1，即输入串的hamming weight。
- $$F = [f_0, f_1, \ldots, f_n]$$
- 一元函数具有形式$[a, b]$。
  二元相等函数“$=_2$”指$[1, 0, 1]$。
  二元不等函数“$\neq_2$”指$[0, 1, 0]$。

# 第二易解类：product type

- 一个函数在集合$\mathcal{P}$，当且仅当它是一些一元函数、二元相等函数、二元不等函数的乘积。严格定义：
- $$F : \{0,1\}^n \to \mathbb{C}$$
- 对称函数指输入变量的次序不影响函数值。
- 输入变量都是布尔变量，函数值只取决于有多少个1，即输入串的hamming weight。
- $$F = [f_0, f_1, \ldots, f_n]$$
- 一元函数具有形式$[a,b]$。
  二元相等函数“$=_2$”指$[1,0,1]$。
  二元不等函数“$\neq_2$”指$[0,1,0]$。
- 例子，$H_{a,b}(x_1,x_2,x_3) = “[a,b]”(x_1)“ =_2 ”(x_1,x_2)“ \neq_2 ”(x_1,x_3)$，$K(x_1,x_2,x_3,x_3,x_4,x_6) = H_{a,b}(x_1,x_2,x_3)H_{c,d}(x_4,x_5,x_6)$。

# 第一易解类：$\mathcal{A}$

- $X = (x_1, x_2, \ldots, x_n)$。$x_j \in D$。$|D| = 2$。（定义计数问题时，定义域$D$无需结构。）

# 第一易解类：$\mathcal{A}$

- $X = (x_1, x_2, \dots, x_n)$。$x_j \in D$。$|D| = 2$。（定义计数问题时，定义域$D$无需结构。）
- 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。）

# 第一易解类：$\mathcal{A}$

- $X = (x_1, x_2, \ldots, x_n)$。$x_j \in D$。$|D| = 2$。（定义计数问题时，定义域$D$无需结构。）
- 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。$x_j^2 = x_j$。

# 第一易解类：$\mathcal{A}$

- $X = (x_1, x_2, \ldots, x_n)$。$x_j \in D$。$|D| = 2$。（定义计数问题时，定义域$D$无需结构。）
- 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。$x_j^2 = x_j$。
- 要求$P$的最高次数是2；要求交错二次项的系数是偶数。

# 第一易解类：$\mathcal{A}$

- $X = (x_1, x_2, \ldots, x_n)$。$x_j \in D$。$|D| = 2$。（定义计数问题时，定义域$D$无需结构。）
- 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。$x_j^2 = x_j$。
- 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：$3x_1 + 2x_2 + 2x_1x_3 + 4x_2x_3$。

# 第一易解类：$\mathcal{A}$

- $X=(x_1,x_2,\ldots,x_n)$。$x_j\in D$。$|D|=2$。（定义计数问题时，定义域$D$无需结构。）
- 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。$x_j^2=x_j$。
- 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$\chi_{(AX=C)}i^{P(x_1,x_2,\ldots,x_n)}$。

# $\#CSP(\mathcal{A})$的算法

- $$\sum_{x_j \in \{0,1\}, j=1,\ldots,n} \chi_{(AX=C)} i^{P(x_1,x_2,\ldots,x_n)}$$

# $\#CSP(\mathcal{A})$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$x_{r+1}=L_{r+1}(X')$，…$x_n=L_n(X')$。$X'=\{x_1,\ldots,x_r,1\}$。

# $\#CSP(\mathcal{A})$的算法

- $$\sum_{x_j \in \{0,1\}, j=1,\dots,n} \chi_{(AX=C)} i^{P(x_1,x_2,\dots,x_n)}$$
- 假设$AX = C$的自由变量是$x_1, \dots, x_r$，
  解是$x_{r+1} = L_{r+1}(X')$，…$x_n = L_n(X')$。$X' = \{x_1, \dots, x_r, 1\}$。
- 例如，方程组是$x_3 = x_1 + x_2 + 1 \mod 2$。

# $\#CSP(\mathcal{A})$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$x_{r+1}=L_{r+1}(X')$，$\ldots x_n=L_n(X')$。$X'=\{x_1,\ldots,x_r,1\}$。
- 例如，方程组是$x_3=x_1+x_2+1 \mod 2$。
- $$\sum_{x_j\in\{0,1\},j=1,\ldots,r} i^{P(x_1,x_2,\ldots,x_r,L_{r+1}(X'),\ldots,L_n(X'))}$$

# $\#CSP(\mathcal{A})$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$x_{r+1}=L_{r+1}(X')$，$\ldots x_n=L_n(X')$。$X'=\{x_1,\ldots,x_r,1\}$。
- 例如，方程组是$x_3=x_1+x_2+1 \mod 2$。
- $$\sum_{x_j\in\{0,1\},j=1,\ldots,r} i^{P(x_1,x_2,\ldots,x_r,L_{r+1}(X'),\ldots,L_n(X'))}$$
- $P$采用整数运算，作为$i$的指数，可以模4运算。

# $\#CSP(\mathcal{A})$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$x_{r+1}=L_{r+1}(X')$，…$x_n=L_n(X')$。$X'=\{x_1,\ldots,x_r,1\}$。
- 例如，方程组是$x_3=x_1+x_2+1 \mod 2$。
- $$\sum_{x_j\in\{0,1\},j=1,\ldots,r} i^{P(x_1,x_2,\ldots,x_r,L_{r+1}(X'),\ldots,L_n(X'))}$$
- $P$采用整数运算，作为$i$的指数，可以模4运算。
- 下面把解表达式的模2运算融入模4运算。

## $\#CSP(\mathcal{A})$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$x_{r+1} = L_{r+1}(X')$，$\ldots x_n = L_n(X')$。$X' = \{x_1, \ldots, x_r, 1\}$。
- 例如，方程组是$x_3 = x_1 + x_2 + 1 \mod 2$。
- $$\sum_{x_j \in \{0,1\}, j=1,\ldots,r} i^{P(x_1,x_2,\ldots,x_r,L_{r+1}(X'),\ldots,L_n(X'))}$$
- $P$采用整数运算，作为$i$的指数，可以模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$x_3$，换成$L_3(X')^2$。（因为$L^2 \mod 4$等于$L \mod 2$。）

## $\#CSP(\mathcal{A})$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$x_{r+1}=L_{r+1}(X')$，…$x_n=L_n(X')$。$X'=\{x_1,\ldots,x_r,1\}$。
- 例如，方程组是$x_3=x_1+x_2+1 \mod 2$。
- $$\sum_{x_j\in\{0,1\},j=1,\ldots,r} i^{P(x_1,x_2,\ldots,x_r,L_{r+1}(X'),\ldots,L_n(X'))}$$
- $P$采用整数运算，作为$i$的指数，可以模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$x_3$，换成$L_3(X')^2$。（因为$L^2 \mod 4$等于$L \mod 2$。）
  - 如果有$2x_3x_4$，因为$2x_3x_4 \mod 4=2(x_3 \mod 2)(x_4 \mod 2)$，代入$L_3,L_4$即可。

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

- 无$x_1$项，或者$x_1$项系数是2

$$\sum_{x_2,\ldots,x_r}\sum_{x_1} i^{2x_1L(X)+P'(x_2,\ldots,x_r)} = \sum_{x_2,\ldots,x_r}(i^{P'(x_2,\ldots,x_r)}\sum_{x_1}(-1)^{x_1L(X)})$$

$$\sum_{x_1}(-1)^{x_1L(X)} = 2\chi_{(L(X)=0)}$$

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

- 无$x_1$项，或者$x_1$项系数是2

$$\sum_{x_2,\ldots,x_r}\sum_{x_1} i^{2x_1L(X)+P'(x_2,\ldots,x_r)} = \sum_{x_2,\ldots,x_r}(i^{P'(x_2,\ldots,x_r)}\sum_{x_1}(-1)^{x_1L(X)})$$

$$\sum_{x_1}(-1)^{x_1L(X)} = 2\chi_{(L(X)=0)}$$

- 有$x_1$项（系数是1或者3）

$$= \sum_{x_2,\ldots,x_r}(i^{P'(x_2,\ldots,x_r)}\sum_{x_1}(-1)^{x_1L(X)}i^{3x_1})$$

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，$F(1,x_2,\ldots,x_r)=iF(0,x_2,\ldots,x_r)$。

$$(1-i)i^{L^2(X)}$$

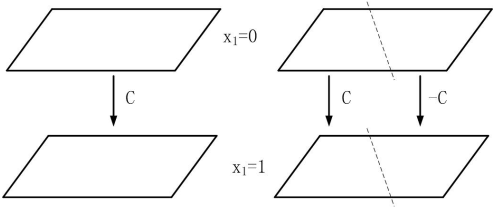


Figure: 虚线右侧区域表示仿射子空间$L(X) = 1$

第一种情况，$C = 1$。第二种情况，$C = \pm i$。

## $\mathcal{A}$中的二元函数例子

- $$\begin{pmatrix} 1 & 1 \\ 1 & i \end{pmatrix}$$
- $$F = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$
- #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$G$，$V_G = \{x_1, \ldots, x_n\}$，$(j, k) \in E_G$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$-1$（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。
- 图$G$的偶数条边的子图数目是多项式时间可以计算的。

# 参考文献


- Nadia Creignou, Miki Hermann:
  Complexity of Generalized Satisfiability Counting Problems. Inf. Comput. 125(1): 1-12 (1996)
- Martin E. Dyer, Leslie Ann Goldberg, Mark Jerrum:
  The Complexity of Weighted Boolean CSP. SIAM J. Comput. 38(5): 1970-1986 (2009)
- Jin-Yi Cai, Pinyan Lu, Mingji Xia:
  The complexity of complex weighted Boolean #CSP. J. Comput. Syst. Sci. 80(1): 217-236 (2014)
- 以上布尔定义域的#CSP问题复杂性，此外还有counting graph homomorphism，Holant等计数问题。
  关于这些问题的复杂性二分定理综合综述，蔡进一和陈汐的书草稿。
- Complexity Classifications of Boolean Constraint Satisfaction Problems, 2001
  Nadia Creignou, Sanjeev Khanna, Madhu Sudan
  包含了早期的判定问题、计数问题和优化问题的复杂性二分