高级算法设计与分析

计数问题

夏盟信
Xia, Mingji

中科院软件所
计算机科学国家重点实验室

2016.6

#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

- $F : \Sigma^* \to \mathbf{N}$ (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$F(x) = |\{y|R(x, y) = 1\}|$。

#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

- $F : \Sigma^* \to \mathbf{N}$ (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。
  - 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。
  - 一个二元函数R定义的计数问题是否是#P难的，和同一个R定义的判定问题是否是NP难的，这两种命题没有关系。

#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。
  - 一个二元函数R定义的计数问题是否是#P难的，和同一个R定义的判定问题是否是NP难的，这两种命题没有关系。
  - 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。
- $V = \{1, 2, \ldots, n\}$
  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。

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。

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。

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。

复杂性二分定理：一定范围内的问题要么难，要么容易

- 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问题类

每一个#CSP问题的实例（输入）和答案（输出）形式是一样的。

- 实例：
  作用于变量集合$X = \{x_1, x_2, \ldots, x_n\}$的一些约束：
  $R_1(x_{1,1}, \ldots, x_{1,r_1}), \ldots, R_1(x_{m,1}, \ldots, x_{m,r_m})$

#CSP问题类

每一个#CSP问题的实例（输入）和答案（输出）形式是一样的。

- 实例：
  作用于变量集合$X = \{x_1, x_2, \ldots, x_n\}$的一些约束：
  $R_1(x_{1,1}, \ldots, x_{1,r_1}), \ldots, R_1(x_{m,1}, \ldots, x_{m,r_m})$
- 答案：

$$\sum_{X \in D^n} \prod_{j=1}^{m} R_j$$

（D表示一个变量的定义域。）

#CSP问题类

每一个#CSP问题的实例（输入）和答案（输出）形式是一样的。

- 实例：
  作用于变量集合$X = \{x_1, x_2, \ldots, x_n\}$的一些约束：
  $R_1(x_{1,1}, \ldots, x_{1,r_1}), \ldots, R_1(x_{m,1}, \ldots, x_{m,r_m})$
- 答案：

$$\sum_{X \in D^n} \prod_{j=1}^{m} R_j$$

（D表示一个变量的定义域。）

给定一个函数集合$\mathcal{F}$，就定义了一个#CSP问题#CSP($\mathcal{F}$)，它的实例所用的约束R必须来自$\mathcal{F}$。

#CSP问题类

每一个#CSP问题的实例（输入）和答案（输出）形式是一样的。

- 实例：
  作用于变量集合$X=\{x_1,x_2,\ldots,x_n\}$的一些约束：
  $R_1(x_{1,1},\ldots,x_{1,r_1}),\ldots,R_1(x_{m,1},\ldots,x_{m,r_m})$
- 答案：

$$\sum_{X\in D^n}\prod_{j=1}^{m}R_j$$

（D表示一个变量的定义域。）

给定一个函数集合$\mathcal{F}$，就定义了一个#CSP问题#CSP$(\mathcal{F})$，它的实例所用的约束R必须来自$\mathcal{F}$。
#2SAT=#CSP$(\{F|F(y,z)=y\vee z$或者$\bar{y}\vee z$或者$y\vee\bar{z}$或者$\bar{y}\vee\bar{z}\})$

#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：$\mathcal{A}$

- $X = (x_1, x_2, \ldots, x_n)$。$x_j \in D$。$|D| = 2$。（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。）

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$\chi_{(AX=C)} \cdot i^{P(x_1, x_2, \ldots, x_n)}$。

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

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

- $$\sum_{x_j\in\{0,1\},j=1,\dots,n} \chi_{(AX=C)} i^{P(x_1,x_2,\dots,x_n)}$$
- 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$x_3=x_1+x_2+1 \mod 2$。
- $$\sum_{x_j\in\{0,1\},j=1,\dots,r} i^{P(x_1,x_2,\dots,x_r,L_{r+1}(X'),\dots,L_n(X'))}$$
- 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$x_{r+1}=L_{r+1}(X')$，$\ldots x_n=L_n(X')$。$X'=\{x_1,\ldots,x_r,1\}$。
- 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。

$\#CSP(\mathcal{A})$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

- $$\sum_{x_j\in\{0,1\},j=1,\dots,n} \chi_{(AX=C)} i^{P(x_1,x_2,\dots,x_n)}$$
- 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$x_3=x_1+x_2+1 \mod 2$。
- $$\sum_{x_j\in\{0,1\},j=1,\dots,r} i^{P(x_1,x_2,\dots,x_r,L_{r+1}(X'),\dots,L_n(X'))}$$
- 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。

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

- 无x_1项，或者x_1一次项系数是2。提取所有的$2x_1$公因子。

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

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

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

- 无x_1项，或者x_1一次项系数是2。提取所有的$2x_1$公因子。

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

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

- 有x_1项（系数是1或者3，以3为例）。提取$2x_1$，x_1的一次项不动。

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

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

$$\sum_{x_1}(-1)^{x_1L(X)}i^{3x_1} = (1-i)i^{L^2(X)}$$

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χ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。

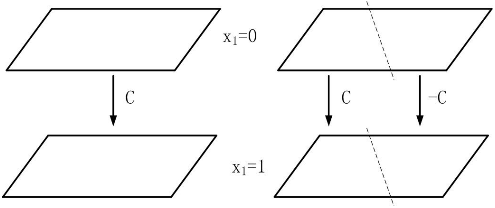


Figure: 虚线右侧区域表示仿射子空间$L(X) = 1$

第一种情况，$C = 1$。第二种情况，$C = \pm i$。

$\mathcal{A}$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，$V_G=\{x_1,\ldots,x_n\}$，$(j,k)\in E_G$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。

第二易解类：product type

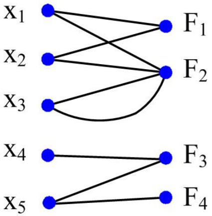
- 一个n元函数F在集合$\mathcal{E}$，当且仅当它在除某互补串对α和$\bar{\alpha}$之外的输入上值都是0。
  严格定义：$F \in \mathcal{E}$当且仅当存在$\alpha \in \{0,1\}^n$，
  $\forall X, X \notin \{\alpha, \bar{\alpha}\} \Rightarrow F(X) = 0$。

第二易解类：product type

- 一个n元函数F在集合$\mathcal{E}$，当且仅当它在除某互补串对α和$\bar{\alpha}$之外的输入上值都是0。
  严格定义：$F \in \mathcal{E}$当且仅当存在$\alpha \in \{0,1\}^n$，
  $\forall X, X \notin \{\alpha, \bar{\alpha}\} \Rightarrow F(X) = 0$。
- 一个变量赋值若要对和$\sum_{X \in D^n} \prod_{j=1}^{m} R_j$有非0贡献，它对$\mathcal{E}$中函数$F$的一个变量的赋值，决定了它对其他$F$的变量的唯一赋值。

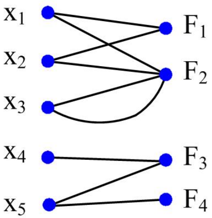
第二易解类：product type

- 一个n元函数F在集合$\mathcal{E}$，当且仅当它在除某互补串对α和$\bar{\alpha}$之外的输入上值都是0。
  严格定义：$F \in \mathcal{E}$当且仅当存在$\alpha \in \{0,1\}^n$，
  $\forall X, X \notin \{\alpha, \bar{\alpha}\} \Rightarrow F(X) = 0$。
- 一个变量赋值若要对和$\sum_{X \in D^n} \prod_{j=1}^{m} R_j$有非0贡献，它对$\mathcal{E}$中函数$F$的一个变量的赋值，决定了它对其他$F$的变量的唯一赋值。
- #CSP问题的实例可以画成偶图。



第二易解类：product type

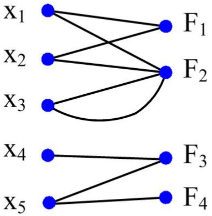
- 一个n元函数F在集合$\mathcal{E}$，当且仅当它在除某互补串对α和$\bar{\alpha}$之外的输入上值都是0。
  严格定义：$F \in \mathcal{E}$当且仅当存在$\alpha \in \{0,1\}^n$，
  $\forall X, X \notin \{\alpha, \bar{\alpha}\} \Rightarrow F(X) = 0$。
- 一个变量赋值若要对和$\sum_{X \in D^n} \prod_{j=1}^{m} R_j$有非0贡献，它对$\mathcal{E}$中函数$F$的一个变量的赋值，决定了它对其他$F$的变量的唯一赋值。
- #CSP问题的实例可以画成偶图。



- 连通的$\mathcal{E}$函数乘在一起，还是$\mathcal{E}$函数。

第二易解类：product type

- 一个n元函数F在集合$\mathcal{E}$，当且仅当它在除某互补串对α和$\bar{\alpha}$之外的输入上值都是0。
  严格定义：$F \in \mathcal{E}$当且仅当存在$\alpha \in \{0,1\}^n$，
  $\forall X, X \notin \{\alpha, \bar{\alpha}\} \Rightarrow F(X) = 0$。
- 一个变量赋值若要对和$\sum_{X \in D^n} \prod_{j=1}^m R_j$有非0贡献，它对$\mathcal{E}$中函数$F$的一个变量的赋值，决定了它对其他$F$的变量的唯一赋值。
- #CSP问题的实例可以画成偶图。



- 连通的$\mathcal{E}$函数乘在一起，还是$\mathcal{E}$函数。
- 一个函数是product type当且仅当能表示成$\mathcal{E}$中函数的乘积。

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