HIT CS&E

第7章　概率方法与去随机化

骆吉洲
计算机科学与技术学院

HIT CS&E

导引

目的：结构或性质的存在性和构造性

存在性
- 概率论证法
  - 构造概率空间
  - 证明结构或性质的概率>0

构造性
- 随机算法
  - 结构或性质的概率p>0
  - 找到它需要平均抽样$1/p$次
- 确定型算法
  - 去随机化
  - 直接设计

HIT CS&E

提纲

HIT CS&E

参考文献

- 《Randomized Algorithms》
  第5章
- 《概率与计算》
  第6章

HIT CS&E

7.1 概率论证法

- 7.1.1 基本计数论证
- 7.1.2 期望论证
- 7.1.3 二阶矩方法
- 7.1.4 Lovasz局部引理

HIT CS&E

7.1.1 基本计数论证

HIT CS&E

计数论证示例

问题: 能否用两种颜色对K_n的边着色使得同色K_k子图不出现?

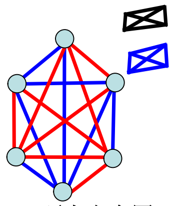
构造一个概率空间来解决问题

对$N=\binom{n}{2}$条边中的每条边掷一枚硬币

- 头面向上，该边用红色
- 背面向上，该边用蓝色

得到2^N种着色方案上的均匀分布

每种着色方案上的概率均为2^{-N}



K_n: n-顶点完全图

K_k子图共有$M=\binom{n}{k}$个,依次编号为1,2,…,M

A_i:随机着色方案中,第i个K_k子图是单色的

HIT CS&E

$\Pr[A_i]=2\cdot 2^{-\binom{k}{2}}$ K_k中的每条边只能同取两色之一

$\Pr[\bigcup_{i=1}^{\binom{n}{k}} A_i] \le \sum_{i=1}^{\binom{n}{k}} 2^{-\binom{k}{2}+1} = \binom{n}{k} 2^{-\binom{k}{2}+1}$ Union Bound

$\Pr[\bigcap_{i=1}^{\binom{n}{k}} \overline{A_i}] = 1-\Pr[\bigcup_{i=1}^{\binom{n}{k}} A_i] = 1-\binom{n}{k} 2^{-\binom{k}{2}+1} > 0$

定理: 如果$\binom{n}{k} 2^{-\binom{k}{2}+1} < 1$,则能用两种颜色对$K_n$的边着色使得结果中不出现同色$K_k$子图.

问题：怎么用高效算法找出这样一个着色方案?

HIT CS&E

随机算法框架

K_n的无单色K_k-子图着色算法

输入：K_n, k

输出：K_n的无单色K_k-子图2-着色

```
1.  For i=1 To N  Do                          //N应该取多大
2.      随机抽取Kn的一种着色方案A               //抽样过程是否高效
3.      If  A中无单色Kk子图  Then               //检验过程是否高效
4.          输出方案A，结束
5.  输出“未找到”
```

For循环每遍成功的概率是$p>0$, 成功需要的期望遍数$1/p$

循环遍数: $N=2/p$, 则算法成功的概率大于1/2

关键步骤1: 设计高效的抽样方法

关键步骤2: 设计高效的性质检验方法

HIT CS&E

7.1.2 期望论证

HIT CS&E

直观理解期望论证

《高级算法》平均成绩是90分

结论1：肯定有人成绩≥90分

结论2：肯定有人成绩≤90分

HIT CS&E

期望论证

引理: 设S是一个概率空间，X是S上的一个随机变量。如果$E[X]=\mu$, 则$\Pr[X\ge\mu]>0$ 且 $\Pr[X\le\mu]>0$.

证明: $\mu = E[X] = \sum_x x\Pr[X=x]$

若 $\Pr[X\ge\mu]=0$, 则 $X<\mu$恒成立。于是

$$\begin{aligned}\mu &= E[X]\\ &= \sum_x x\Pr[X=x]\\ &< \sum_x \mu\Pr[X=x]\\ &= \mu\sum_x \Pr[X=x]\\ &= \mu\end{aligned}$$

矛盾。因此， $\Pr[X\ge\mu]>0$

类似地， $\Pr[X\le\mu]>0$

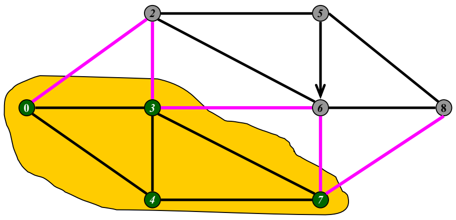
应用:最大割问题

最大割问题

输入：连通图 $G=(V,E)$, 记$|E|=m$

输出：V的划分$S,V\text{-}S$使得介于S和$V\text{-}S$之间的边数$c(S)$最大

- 最大割问题是NP难问题
- 最小割问题存在多项式时间算法



最大割的期望论证

概率空间

- 创建标记A,B
- $\forall v\in V$, 将v均匀随机地标记为A或B
- $S=\{v\in V|\ v$的标记为$A\}$　　$V\text{-}S=\{v\in V|\ v$的标记为$B\}$

期望论证

- $\forall e\in E$, 端点标记相同的概率为1/2，不同的概率为1/2

$$X_e=\begin{cases}1 & e\text{的端点标记不同}\\ 0 & e\text{的端点标记相同}\end{cases}\qquad \Pr[X_e]=1/2,\quad E[X_e]=1/2$$

- $c(S)=\sum_{e\in E}X_e$
- $E[c(S)]=E[\sum_{e\in E}X_e]=\sum_{e\in E}E[X_e]=m/2$
- 存在大小至少为$m/2$的割

估计概率界

- $p=\Pr[\ c(S)\geq |E|/2\ ]$
- $c(S)\leq m$恒成立

$$\begin{aligned} m/2 &= E[\ c(S)] \\ &= \sum_i i\cdot\Pr[c(S)=i] \\ &= \sum_{i<m/2} i\cdot\Pr[c(S)=i]\ +\sum_{i\geq m/2} i\cdot\Pr[c(S)=i] \\ &\leq (m/2-1)\cdot(1-p)\ +m\cdot p \end{aligned}$$

- $p\geq 1/(m/2+1)$

最大割的Las Vegas算法

最大割问题的Las Vegas算法

输入：连通图 $G=(V,E)$, 记$|E|=m$

输出：V的划分$S,V\text{-}S$使得介于S和$V\text{-}S$之间的边数$c(S)$最大

1. $c\leftarrow 0$, $S\leftarrow\varnothing$
2. For i=1 To m Do
3. 　$\forall v\in V$, 以1/2的概率将v放入S_i
4. 　$c_i\leftarrow$介于S_i和$V\text{-}S_i$之间的边条数
5. 　If $c_i>c$ Then $c\leftarrow c_i$, $S\leftarrow S_i$
6. 输出S,c

性能分析

- 每一遍执行For循环，$c_i>m/2$的概率至少为$p\geq 1/(m/2+1)$
- $c>m/2$, 执行For循环的期望遍数为$1/p\leq m/2+1$
- 由Markov不等式可知
  For循环执行$m/2$遍，$c<m/2$的概率至多为1/2

作业

本章结束后，将该算法改造成确定型算法并进行分析

抽样修改+期望论证

两阶段概率论证

第一阶段

从概率空间抽样(样本不一定具有要求的性质)

第二阶段

修改样本使其具有要求的性质

在两阶段中结合期望论证得出结论

HIT CS&E

示例1：独立集的大小

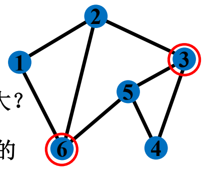
独立集: 图$G=(V,E)$顶点子集I: $\forall u,v\in I$均有$uv\notin E$

例. {3,6}是独立集　{1,3},{1,4}, {4,6},{2,4}均是独立集

顶点的平均度: $d= 2|E|/|V|= 2m/n$

问题: 图G的最大独立集至少为多大?

注意: 求图G的最大独立集是NP难的



HIT CS&E

第一步：对顶点抽样

$\forall v\in V$,独立地以$1\text{-}1/d$的概率删除v及其邻边

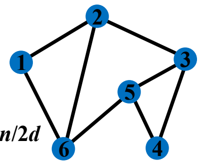
- 用X表示留下来的顶点个数
  X是随机变量

  $E[X] = n/d$　每个顶点以$1/d$概率留下

- 用Y表示留下来的边的数量
  Y也是随机变量

  边e留下 ⇔ e的端点均留下

  $E[Y] = m\cdot(1/d)^2 = (nd/2)\cdot(1/d)^2 = n/2d$



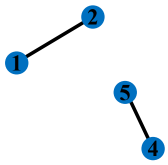
HIT CS&E

第二步：修改样本

对剩下的每条边，删除边及它的一个邻接顶点

- 最终剩下的顶点组成一个独立集
  相互之间没有边相连
- 最终剩下的顶点有$X\text{-}Y$个

  $E[X\text{-}Y]= E[X]\text{-}E[Y]$
  $= n/d - n/2d$
  $= n/2d$
  $= n^2/m$



HIT CS&E

结论

引理: 图$G=(V,E)$存在大小为$|V|^2/|E|$的独立集

作业

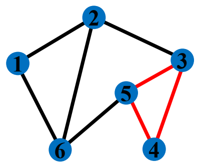
设计一个Las Vegas算法使得它输出图$G=(V,E)$的大小至少为$|V|^2/|E|$的独立集的概率大于0

HIT CS&E

示例2：随机图的围长

围长: 图$G=(V,E)$最短环的长度

例. 下图的围长是3



随机图: 随机生成一个图G(n,p)
　n个顶点
　每对顶点之间以概率p添加边

直观上: 图越稠密，围长越小

问题: 是否存在围长较大的稠密图?

HIT CS&E

第一步：随机图抽样

$p = n^{1/k}/n$,生成一个随机图$G\in G_{n,p}$

- X表示G中边的条数
  X是随机变量

  $$E[X] = p\binom{n}{2} = \frac{1}{2}(1-\frac{1}{n})n^{1/k+1}$$

- 用Y表示G中长度$<k$的环的个数
  Y也是随机变量

  任意i个顶点有$(i\text{-}1)!/2$种顺序连接成环

  任意i个顶点按一种顺序连成长度为i的环的概率为p^i

  n个顶点中取i个顶点的方案数为$\binom{n}{i}$

  $$E[Y] = \sum_{i=3}^{k-1} p^i\binom{n}{i}\frac{(i-1)!}{2} \le \sum_{i=3}^{k-1} n^i p^i = \sum_{i=3}^{k-1} n^{i/k} < kn^{(k-1)/k}$$

HIT CS&E

第二步：样本修改

随机图G中，将长度$<k$的每个环删去一条边

- 得到的图仅含长度$\geq k$的环
  围长$\geq k$
- 得到的图的边数的数学期望
  $E[X-Y] = E[X] - E[Y]$
  $\geq \frac{1}{2}(1-\frac{1}{n})n^{1/k+1} - kn^{(k-1)/k}$
  $\geq \frac{1}{4}n^{1/k+1}$

HIT CS&E

结论

引理：$k\geq 3$，则存在边数至少为$\frac{1}{4}n^{1/k+1}$且围长至少为k的图

思考

能否设计一个高效的Las Vegas算法来产生这样的图？
算法的复杂性如何呢？

HIT CS&E

7.1.3 二阶矩方法

- Review——《计算建模》GMM方法
- GMM——General Moment Method
- 了解方法原理和应用

HIT CS&E

补充

定理：如果X是非负随机变量，则 $\Pr[X=0] \leq \frac{E[X^2]}{(E[X])^2}$

证明：$\Pr[X=0] \leq \Pr[\,|X-E[X]| \geq E[X]] \leq \frac{E[X^2]}{(E[X])^2}$

定理：如果$X_i\,(i\geq 1)$ 是0-1随机变量且$X=\sum_{i=1}^{n} X_i$，则

$$\Pr[X>0] \geq \sum_{i=1}^{n} \frac{\Pr[X_i=1]}{E[X\,|\,X_i=1]}$$

证明思路：令

$$Y = \begin{cases} 1/X & X>0 \\ 0 & X=0 \end{cases}$$

则 $\Pr[X>0] = E[XY]$

然后根据$E[XY]$的定义式即可得出定理

HIT CS&E

7.1.4 Lovasz局部引理

- 直观地理解Lovasz局部引理
- 对称形式
- 一般形式（或非对称形式）
- 简单应用
- 算法式LLL

HIT CS&E

Lovasz引理的直观含义

- $A_1, A_2, \ldots, A_n$是一系列“坏”事件，它们不具备期望的性质
- 期望的性质 ＝ 避开所有的“坏”事件
  亦即，人们对下式是否成立感兴趣
  $$\Pr[\bigcap_{i=1}^{n} \bar{A}_i] > 0$$
- 如果$\sum_i \Pr[A_i]<1$, 则避开所有“坏” 事件的概率大于0
- 然而，通常情况下，$\sum_i \Pr[A_i] >> 1 \geq \Pr[\cup_i A_i]$
- 如果$A_1, \ldots, A_n$相互独立，且$\Pr[A_i] \leq p<1$均成立，则
  $$\Pr[\bigcap_{i=1}^{n} \bar{A}_i] = \prod_{i=1}^{n} \Pr[\bar{A}_i] \geq (1-p)^n > 0$$
- Lovasz引理将上述情况推广到“有限度独立”的情况

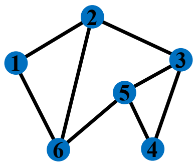
HIT CS&E

依赖图

依赖图

$A_1,A_2,\ldots,A_n$是概率空间中的事件，有向图G=({1,2,…,n},E)称为$A_1,A_2,\ldots,A_n$的依赖图，如果A_i独立于A_j($ij\notin E$)对i=1,2,…,n均成立

A_1与A_3,A_4,A_5相互独立
A_2与A_4,A_5相互独立
A_3与A_1,A_6相互独立
…



HIT CS&E

对称形式

Lovasz Local Lemma

设$A_1,A_2,\ldots,A_n$是任意概率空间中的n个事件，这些事件的依赖图的度≤d，且$\Pr[A_i]\le p<1$对i=1,2,…,n均成立。如果下列条件之一成立，则 $\Pr[\bigcap_{i=1}^{n}\overline{A_i}]>0$

引理1. (Lovasz and Erdos 1973, 正式发表于1975)

$4pd<1$

引理2. (Lovasz 1977)

$ep(d+1)<1$

引理3. (Shearer 1985)

$p=1/2 \quad d=1$
或
$p<\frac{(d-1)^{d-1}}{d^d} \quad d>1$

HIT CS&E

一般形式

Lovasz Local Lemma

设$A_1,A_2,\ldots,A_n$是任意概率空间中的n个事件，这些事件的依赖图是有向图G=({1,2,…,n},E)。如果存在实数$x_1,x_2,\ldots x_n\in[0,1)$使得下式对$i$=1,2,…,$n$均成立

$$\Pr\left[\bigwedge_{i=1}^{n}\bar{A}_i\right]\le x_i\prod_{(i,j)\in E}(1-x_j)$$

则

$$\Pr\left[\bigwedge_{i=1}^{n}\bar{A}_i\right]\ge\prod_{i=1}^{n}(1-x_i)$$

HIT CS&E

Lovasz引理的证明

请大家自觉阅读

《概率与计算》第6.7节 (对称形式的证明)

《概率与计算》第6.9节 (一般形式的证明)

HIT CS&E

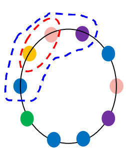
应用1：环着色

环着色

- 将$11n$个点依次连接,形成一个环
- 用n种颜色对点着色，每种颜色恰好用于11个点

随机抽取每种颜色的一个顶点，共n个顶点

- 各个顶点被抽中的概率是1/11
- 相邻顶点被一起抽中的概率是p=1/121
- E_i—抽中的第i色顶点v有相邻顶点u被抽中，p<1/121
  - E_i之间不独立，有多少E_j与E_i相关
  - v周围至多4种颜色顶点与E_i相关
  - 第i色有11个位置，共44个位置与E_i关联
  - 除去u,v的两色，还有42个E_j与E_i关联



HIT CS&E

环着色

环着色

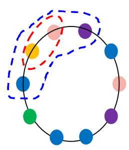
- 将$11n$个点依次连接,形成一个环
- 用n种颜色对点着色，每种颜色恰好对11个点着色

随机抽取每种颜色的一个顶点，共n个顶点

- 各个顶点被抽中的概率是1/11
- 相邻顶点被一起抽中的概率是p=1/121
- E_i—抽中的第i色顶点v有相邻顶点u被抽中，p<1/121
- $E_1,\ldots,E_n$依赖图的度$d\le42$
- $ep(d+1)\approx0.966<1$
- $\Pr[\cap_i\overline{E_i}]>0$

任意着色方案存在不相邻的每色一点选取方案

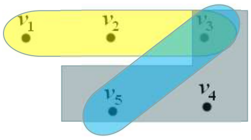
能否设计算法来完成这种方案的选取呢？



HIT CS&E

应用2：超图着色

超图——图的推广形式，每条边可以包含任意个数的顶点
- 图的边包含两个顶点
- 超图的边是任意的顶点子集
- 顶点的度是包含该顶点的超边的条数



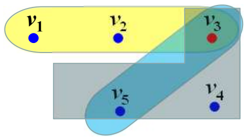
$H(X,E)$
$X=\{v_1,v_2,v_3,v_4,v_5\}$
$E=\{e_1,e_2,e_3\}$
各个顶点的度是多少？

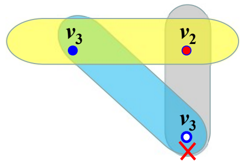
- 超图能对复杂信息结构建模
- 应用领域越来越广泛

HIT CS&E

超图着色问题

超图2-着色问题
输入：超图H(X,E)
输出：用两种颜色对所有顶点着色使得没有同色超边



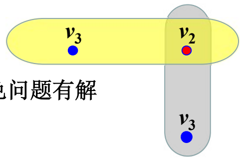


- 如果相交边太多，则问题无解
- 相交边条数上界是多少，才能确保问题有解呢？

HIT CS&E

一致超图的结果

k-一致超图：每条边均包含k个顶点



k-一致超图的2-着色
边数少于2^{k-1}的k-一致超图的2-着色问题有解

推广(用Lovasz局部引理)
定理：设超图H的每条超边至少有k个顶点且每条超边至多与d条超边相交。如果$e(d+1) < 2^{k-1}$, 则H的2-着色问题有解。

HIT CS&E

定理证明

证明：概率空间是随机2-着色
定义“边e_i是同色的”为“坏事件”A_i,共$|E|$个坏事件
$\Pr[A_i] \le 2^{-(k-1)} = p$
A_i至多依赖于d个坏事件

$ep(d+1) \le 2^{-(k-1)} e(d+1) < 1$

由Lovasz局部引理可知，

$$\Pr[\bigcap_{i=1}^{|E|} \overline{A_i}] > 0$$

亦即，存在2-着色方案使得所有边不同色

问题：怎么用高效算法为这种图找到2-着色方案呢？

HIT CS&E

超图2-着色参考资料

超图2-着色的多项式时间算法性方法

1. J. Beck, An algorithmic approach to the Lovasz local lemma, Random Structures and Algorithms, 2(4)(1991) , pp. 343–365

超图2-着色的构造性方法

1. Robin A. Moser: Derandomizing the Lovasz Local Lemma more effectively. CoRR abs/0807.2120 (2008)
2. Robin A. Moser: A constructive proof of the Lovasz Local Lemma. CoRR abs/0810.4812 (2008)

HIT CS&E

算法式Lovasz引理

算法式Lovasz引理
输入：一组随机变量$X_1,\ldots,X_n$
要避开的“坏”事件$A_1,\ldots,A_m$
输出：避开所有坏事件的随机变量取值

1. $x_1,x_2,\ldots,x_n \leftarrow X_1,\ldots,X_n$的一组随机赋值
2. **while** ($\exists i$: A_i发生) **Do**
3. 对A_i相关的随机变量重新随机赋值
4. 返回最终赋值

HIT CS&E

例：构造可满足k-SAT的解

构造可满足性k-SAT的解

输入：变量$x_1,\ldots,x_n$上的CNF公式F=$C_1\wedge C_2\wedge\ldots\wedge C_m$

每个C_j是至多k个(否定)变量的或,即$|C_j|\le k$,

$x_i,\neg x_i$未同时出现在同一C_j中

每个布尔变量至多出现在$d\le 2^{k-2}$个子句中

输出：满足F的一组布尔变量赋值

1. $x_1,x_2,\ldots,x_n\leftarrow X_1,\ldots,X_n$的一组均匀随机赋值
2. while ($\exists j$: C_j未被满足) Do
3. 对C_j中的随机变量均匀随机地重新赋值
4. 返回最终赋值

作业：用LLL证明算法的输入是可满足的

结论：算法找到满足性赋值的期望运行遍数为$O(n+km\log m)$

文献：Robin A. Moser. The Lovász Local Lemma and Satisfiability. Efficient Algorithms 2009: 30-54

HIT CS&E

7.2 去随机化

- 7.2.1 MAX-SAT随机算法的去随机化
- 7.2.2 集合平衡配置随机算法去随机化
- 7.2.3 随机电路去随机化

HIT CS&E

7.2.1 MAX-SAT随机算法去随机化

- 随机抽样算法的去随机化
- 随机舍入算法的去随机化
- 随机混合算法的去随机化

HIT CS&E

随机抽样算法的去随机化

HIT CS&E

随机抽样算法

MAX-SAT问题的随机抽样算法RandSample

输入：n个文字及其上的CNF公式$F=C_1\wedge\ldots\wedge C_m$

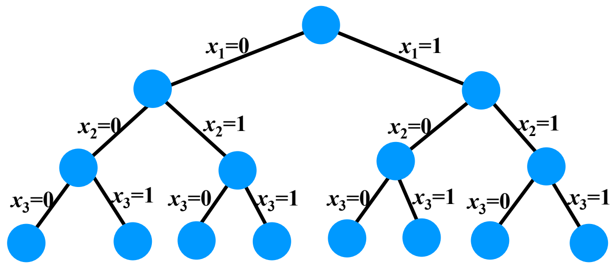
输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

1. For i=1 To n Do
2. 第i个文字以概率1/2取真,以概率1/2取假
3. 返回1-2步得到的随机赋值

性能：$O(n)$时间E[2]-近似随机算法

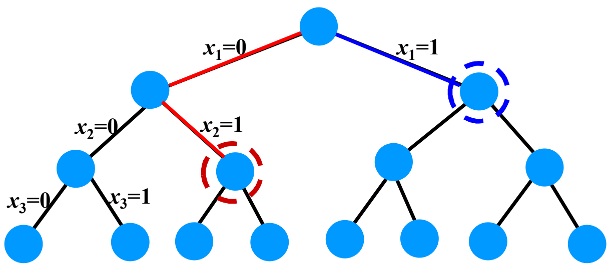
HIT CS&E

赋值树示意图



HIT CS&E

每个节点v都定义了一个部分赋值$f(v)$

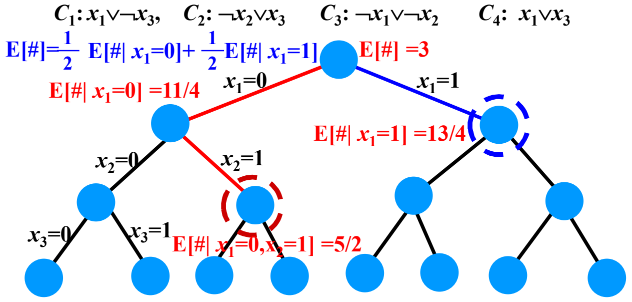


HIT CS&E

条件期望

条件期望E[#| $f(v)$]: 给定节点v的部分赋值$f(v)$后被满足的子句的个数的期望值

$C_1: x_1\vee\neg x_3$, $C_2: \neg x_2\vee x_3$ $C_3: \neg x_1\vee\neg x_2$ $C_4: x_1\vee x_3$



HIT CS&E

条件期望的多项式时间计算

$C_1: x_1\vee\neg x_3$, $C_2: \neg x_2\vee x_3$ $C_3: \neg x_1\vee\neg x_2$ $C_4: x_1\vee x_3$

E[#| x_1=0,x_2=1] =?

$C_1: x_1\vee\neg x_3$在x_1=0, x_2=1的条件下被满足的概率为1/2

$C_2: \neg x_2\vee x_3$在x_1=0, x_2=1的条件下被满足的概率为1/2

$C_3: \neg x_1\vee\neg x_2$肯定被$x_1$=0, x_2=1满足

$C_4: x_1\vee x_3$在x_1=0, x_2=1的条件下被满足的概率为1/2

E[#| x_1=0,x_2=1] =1/2+1/2+1+1/2 = 5/2

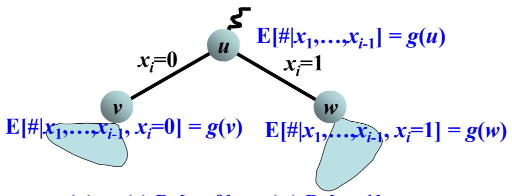
结论：任意节点的条件期望可以在多项式时间内被计算

HIT CS&E

重要事实

E[#]= $\frac{1}{2}$ E[#| x_1=0]+ $\frac{1}{2}$ E[#| x_1=1]

E[#] = E[#| x_1=0]·Pr[x_1=0] + E[#| x_1=1]·Pr[x_1=1]



$g(u)$= g(v)·Pr[x_i=0] + $g(w)$·Pr[x_i=1]
= 0.5[g(v)+g(w)]

g(v)≥g(u) 或 g(w)≥g(u)必然成立

HIT CS&E

MAX-SAT确定型赋值算法

MAX-SAT问题的确定型赋值算法DetAssign

输入：n个文字及其上的CNF公式$F=C_1\wedge\ldots\wedge C_m$

输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

1. u←赋值树的树根
2. For i=1 To n
3. v ←u的左孩子 //x_i=0
4. w ←u的右孩子 //x_i=1
5. 分别计算$g(v)$和$g(w)$ //条件数学期望
6. If $g(v)\geq g(w)$ Then u←v,取定x_i=0
7. Else u←w, 取定x_i=1
8. 返回得到的赋值$x_1,\ldots,x_n$

HIT CS&E

算法分析

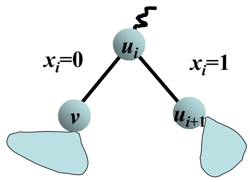
将算法执行过程中选定的节点u依次记为$u_0,u_1,\ldots,u_n$

E[#] = g(u_0)

$g(u_{i-1})\leq g(u_i)$ i=1,2,…,n

E[#] = g(u_0) ≤ $g(u_1)$≤… ≤ $g(u_n)$=x_1,..,x_n满足的子句数sat

$$\frac{opt}{E[\#]}\leq 2 \Rightarrow \frac{opt}{sat}\leq 2$$



结论：DetAssin的近似比为2
其运行时间是多项式的

$g(u_i)$= 0.5[g(v)+g(u_{i+1})]
g(v)≤g(u_{i+1})

随机舍入算法的去随机化

问题转化

MAX-SAT问题

输入：n个文字及其上的CNF公式$F=C_1\wedge\ldots\wedge C_m$

输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

$$x_i=\begin{cases}1 & \text{第}i\text{个文字取真}\\ 0 & \text{第}i\text{个文字取假}\end{cases}\qquad y_j=\begin{cases}1 & \text{子句}C_j\text{被满足}\\ 0 & \text{子句}C_j\text{未被满足}\end{cases}$$

MAX-SAT表示为0-1规划

$$\max\ y_1+y_2+\ldots+y_m$$

$$\begin{aligned} s.t.\quad & \Sigma_{i\in C_j^+}x_i+\Sigma_{i\in C_j^-}(1\text{-}x_i)\ge y_j && 1\le j\le m\\ & x_i\in\{0,1\} && 1\le i\le n\\ & y_j\in\{0,1\} && 1\le j\le m\end{aligned}$$

问题转化

MAX-SAT问题

输入：n个文字及其上的CNF公式$F=C_1\wedge\ldots\wedge C_m$

输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

$$x_i=\begin{cases}1 & \text{第}i\text{个文字取真}\\ 0 & \text{第}i\text{个文字取假}\end{cases}\qquad y_j=\begin{cases}1 & \text{子句}C_j\text{被满足}\\ 0 & \text{子句}C_j\text{未被满足}\end{cases}$$

松弛0-1规划中的约束条件得线性规划问题

$$\max\ y_1+y_2+\ldots+y_m$$

$$\begin{aligned} s.t.\quad & \Sigma_{i\in C_j^+}x_i+\Sigma_{i\in C_j^-}(1\text{-}x_i)\ge y_j && 1\le j\le m\\ & x_i\in[0,1] && 1\le i\le n\\ & y_j\in[0,1] && 1\le j\le m\end{aligned}$$

随机舍入算法

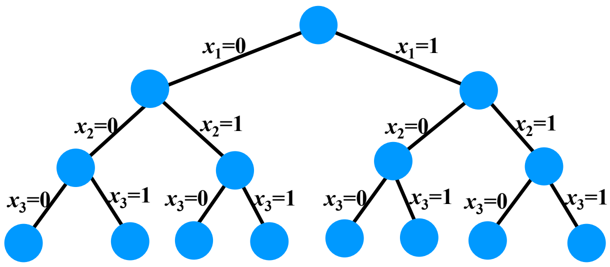
MAX-SAT问题的随机舍入算法RandRound

输入：n个文字及其上的CNF公式$F=C_1\wedge\ldots\wedge C_m$

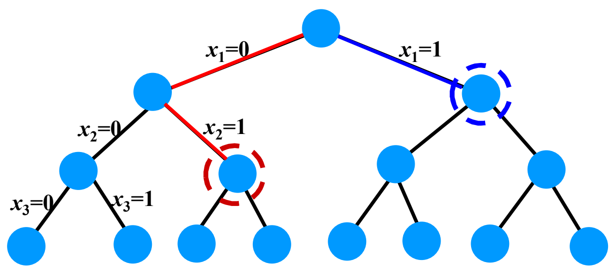
输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

1. 将MAX-SAT表示为0-1规划问题并松弛为线性规划问题
2. 调用多项式时间算法求得线性规划问题的解(x^*,y^*)
   // $x^*=(x_1^*,\ldots,x_n^*)$，每个分量对应一个文字
   // $y^*=(y_1^*,\ldots,y_m^*)$，每个分量对应一个子句
   //能同时被满足的子句的个数不超过$y_1^*+\ldots+y_m^*$
3. For i=1 To n Do
4. 　第i个文字以概率x_i^*取真,以概率$1\text{-}x_i^*$取假
5. 返回3-4步得到的赋值

赋值树示意图



每个节点v都定义了一个部分赋值$f(v)$



HIT CS&E

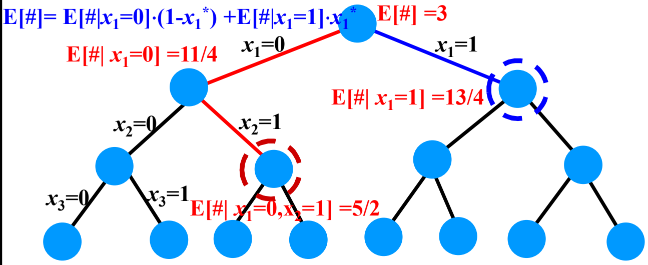
条件期望

条件期望E[#| $f(v)$]: 给定节点v的部分赋值$f(v)$后被满足的子句的个数的期望值

$C_1: x_1 \vee \neg x_3$, $C_2: \neg x_2 \vee x_3$ $C_3: \neg x_1 \vee \neg x_2$ $C_4: x_1 \vee x_3$

$x_1^* = x_2^* = x_3^* = 0.5$ $y_1^* = y_2^* = y_3^* = y_4^* = 1$

E[#]= E[#|x_1=0]·(1-x_1^*) +E[#|x_1=1]·x_1^*



HIT CS&E

条件期望的多项式时间计算

$C_1: x_1 \vee \neg x_3$, $C_2: \neg x_2 \vee x_3$ $C_3: \neg x_1 \vee \neg x_2$ $C_4: x_1 \vee x_3$

E[#| x_1=0,x_2=1] =?

$C_1: x_1 \vee \neg x_3$在x_1=0, x_2=1的条件下被满足的概率为(1-x_3^*)

$C_2: \neg x_2 \vee x_3$在x_1=0, x_2=1的条件下被满足的概率为x_3^*

$C_3: \neg x_1 \vee \neg x_2$肯定被$x_1$=0, x_2=1满足

$C_4: x_1 \vee x_3$在x_1=0, x_2=1的条件下被满足的概率为x_3^*

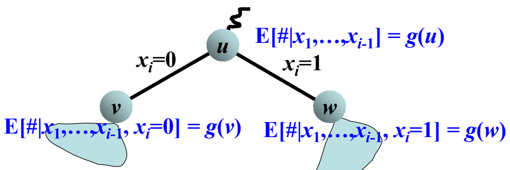
E[#| x_1=0,x_2=1] =(1-x_3^*)+x_3^*+1+x_3^* = 5/2

结论：任意节点的条件期望可以在多项式时间内被计算

HIT CS&E

重要事实

E[#] = E[#| x_1=0]·Pr[x_1=0] + E[#| x_1=1]·Pr[x_1=1]
= E[#| x_1=0]·(1-x_1^*)+E[#| x_1=1]·x_1^*



g(u)= g(v)·Pr[x_i=0] + g(w)·Pr[x_i=1]
= g(v)·(1-x_i^*)+g(w)·x_i^*

g(v)≥g(u) 或 g(w)≥g(u)必然成立

HIT CS&E

MAX-SAT确定型赋值算法

MAX-SAT问题的确定型赋值算法DetAssign

输入：n个文字及其上的CNF公式$F=C_1 \wedge \ldots \wedge C_m$

输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

1. 将问题表示为0-1规划，松弛，求得优化解(x^*,y^*)
2. u←赋值树的树根
3. For i=1 To n
3. v ←u的左孩子 (//x_i=0) w ←u的右孩子(//x_i=1)
5. 分别计算g(v)和g(w) //根据x^*计算条件数学期望
6. If g(v)≥g(w) Then u←v,取定x_i=0
7. Else u←w, 取定x_i=1
8. 返回得到的赋值$x_1,\ldots,x_n$

HIT CS&E

算法分析

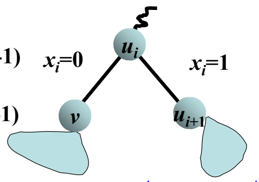
将算法执行过程中选定的节点u依次记为$u_0,u_1,\ldots,u_n$

E[#] = g(u_0)

g(u_{i-1})≤ g(u_i) i=1,2,…,n

E[#] = g(u_0) ≤ g(u_1)≤… ≤ g(u_n)=x_1,...,x_n满足的子句数sat

$\frac{opt}{E[\#]} \le e/(e-1) \Rightarrow \frac{opt}{sat} \le e/(e-1)$



结论：DetRound的近似比为$e/(e-1)$
其运行时间是多项式时间

g(u_i)= g(v)·(1-x_i^*)+g(u_{i+1})·x_i^*
g(v)≤g(u_{i+1})

HIT CS&E

随机混合算法的去随机化

HIT CS&E

混合算法

MAX-SAT问题的混合随机算法RandMix

输入：n个文字及其上的CNF公式$F=C_1\wedge\ldots\wedge C_m$

输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

1. 调用RandSample得赋值A,A满足的子句个数记为X
2. 调用RandRound的赋值B,B满足的子句个数记为Y
3. If $X>Y$ Then $C=A$
4. Else $C=B$
5. 返回赋值C

$$\frac{opt}{E[\#]}\le\frac{4}{3}$$

HIT CS&E

混合算法去随机化

MAX-SAT问题的混合随机算法DetMix

输入：n个文字及其上的CNF公式$F=C_1\wedge\ldots\wedge C_m$

输出：文字赋值$x_1,\ldots,x_n$使得$C_1,\ldots,C_m$被同时满足的子句最多

1. 调用DetAssin得赋值A',A'满足的子句个数记为X'
2. 调用DetRound的赋值B',B'满足的子句个数记为Y'
3. If $X'>Y'$ Then $C'=A'$
4. Else $C'=B'$
5. 返回赋值C'

$X'\ge E[X]$ 且 $Y'\ge E[Y]$ $\Rightarrow sat\ge E[\#]$

$$\frac{opt}{E[\#]}\le\frac{4}{3}\Rightarrow\frac{opt}{sat}\le\frac{4}{3}$$

结论：DetMix是一个多项式时间4/3-近似算法

HIT CS&E

7.2.2 集合平衡配置随机算法的去随机化

HIT CS&E

集合平衡配置问题

问题定义

输入：$n\times n$的0-1矩阵A

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}=\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix}$$

输出：向量$x=(x_1,\ldots,x_n)^T\in\{1,-1\}^m$使得 $|Ax|_\infty$最小

问题背景请大家复习5.2节

HIT CS&E

深入理解计算问题

$$A=\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}\qquad x=\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}\qquad x'=\begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$$

$$Ax=\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}=\begin{pmatrix} 0 \\ -2 \\ 1 \\ -1 \end{pmatrix}$$

$\|Ax\|_\infty=2$

$$Ax'=\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}\begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix}=\begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$\|Ax'\|_\infty=1$

HIT CS&E

平衡配置的随机算法

输入：0-1矩阵$A_{n\times n}$

输出：$x\in\{-1,+1\}^n$使 $\min\|Ax\|_\infty$

$(\forall 1\le j\le m)$随机独立取 $x_j\in\{-1,+1\}^n$

$$x_j=\begin{cases} +1 & \Pr[x_j=+1]=1/2 \\ -1 & \Pr[x_j=-1]=1/2 \end{cases}$$

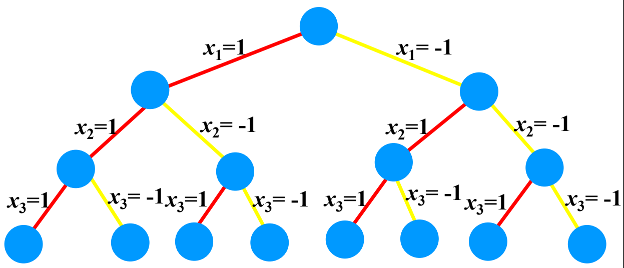
参见5.2节

性能：$\Pr[\ \|Ax\|_\infty>4\sqrt{n\ln n}\ ]<\frac{2}{n}$ 模仿5.2节给出证明

问题：是否存在确定型算法使得其输出x满足$\|Ax\|_\infty<4\sqrt{n\ln n}$

HIT CS&E

平衡配置树示意图



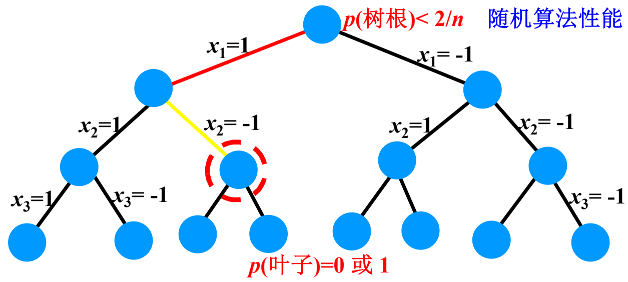
HIT CS&E

平衡配置树示意图

每个节点v都定义了一个部分配置$c(v)$，叶子节点对应完整配置

$p(v)$= Pr[$\|Ax\|_\infty > 4\sqrt{n\ln n}$ $|c(v)$]

- 给定部分配置$c(v)$的条件下，完整配置是“坏配置” 的概率

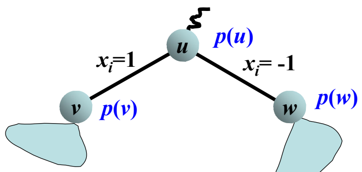


HIT CS&E

重要事实

$p(u)= p(v)\cdot\Pr[x_i=1] + p(w)\cdot\Pr[x_i= -1]$ 全概率公式

$= 0.5[p(v)+p(w)]$



$p(v)\geq p(u)$ 或 $p(w)\geq p(u)$必然成立

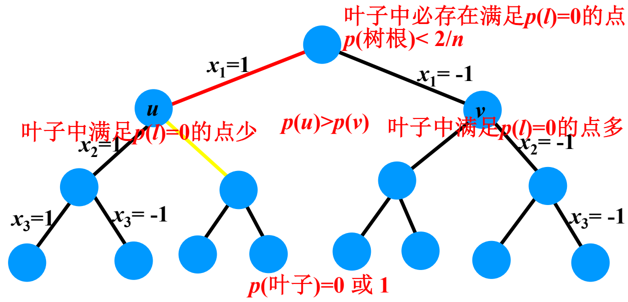
选择小概率的方向进行集合配置

HIT CS&E

效果

每个节点v都定义了一个部分配置$c(v)$，叶子节点对应完整配置

$p(v)$= Pr[$\|Ax\|_\infty > 4\sqrt{n\ln n}$ $|c(v)$]



HIT CS&E

集合平衡配置确定型算法框架

集合平衡配置问题的确定型算法框架

输入：$n\times n$的0-1矩阵A

输出：n维向量$(x_1,\ldots,x_n)\in\{-1,1\}^n$使得$\|Ax\|_\infty$最小

1. $u\leftarrow$集合平衡配置树的树根
2. For i=1 To n
3. $v\leftarrow u$的左孩子 (//x_i=1)；$w\leftarrow u$的右孩子 (//x_i=-1)
4. 分别计算$p(v)$和$p(w)$ //概率计算 ,怎么算?
5. If $p(v)\geq p(w)$ Then $u\leftarrow w$,取定x_i= -1
6. Else $u\leftarrow v$, 取定x_i=1
7. 返回配置结果$(x_1,\ldots,x_n)$ $\|Ax\|_\infty < 4\sqrt{n\ln n}$

问题：*nice*？有什么问题呢？

HIT CS&E

技巧

找到$p(u)$的一个能在多项式时间内计算的上界$q(u)$来代替$p(u)$

$q(u) = \min(1, \sum_j \Pr[\ |A_jx| > 4\sqrt{n\ln n}\ |c(u)])$ A_j是A的第j行

$\geq \Pr[\ \vee_j |A_jx| > 4\sqrt{n\ln n}\ |c(u)]$

$= \Pr[\ \|Ax\|_\infty > 4\sqrt{n\ln n}\ |c(u)]$

$=p(u)$

性质1： $q(u)=0 \Rightarrow p(u)=0$

性质2： q(叶子)=0 或 1

性质3： q(树根)≤ 2/n

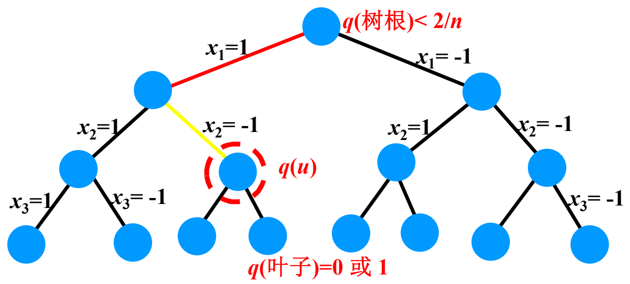
$\Pr[\ A_jx > 4\sqrt{n\ln n}\ ] \leq 1/n^2$ 套用5.2节的证明

$\Pr[\ A_jx < -4\sqrt{n\ln n}\ ] \leq 1/n^2$ 套用5.2节的证明

HIT CS&E

每个节点v都定义了一个部分配置$c(v)$，叶子节点对应完整配置

$q(u) = \min(1, \sum_j \Pr[\ |A_j x| > 4\sqrt{n\ln n}\ \ |c(u)])$



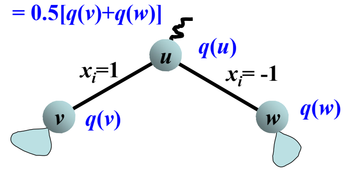
HIT CS&E

重要事实

$q(u) = \min(1, \sum_j \Pr[\ |A_j x| > 4\sqrt{n\ln n}\ \ |c(u)])$

$\Pr[\ |A_j x| > 4\sqrt{n\ln n}\ \ |c(u)] = \Pr[\ |A_j x| > 4\sqrt{n\ln n}\ \ |c(v)]\ \Pr[x_i=1] + \Pr[\ |A_j x| > 4\sqrt{n\ln n}\ \ |c(w)]\ \Pr[x_i=-1]$

$q(u)= q(v)\cdot\Pr[x_i=1] + q(w)\cdot\Pr[x_i= -1]$
$= 0.5[q(v)+q(w)]$



$q(v)\geq q(u)$ 或 $q(w)\geq q(u)$必然成立

选择$q(\cdot)$的方向进行集合配置

HIT CS&E

集合平衡配置确定型算法

集合平衡配置问题的确定型算法DetColoring

输入：$n\times n$的0-1矩阵A

输出：n维向量$(x_1,\ldots,x_n)\in\{-1,1\}^n$使得$\|Ax\|_\infty$最小

1. $u\leftarrow$集合平衡配置树的树根
2. For i=1 To n
3. $v\leftarrow u$的左孩子 (//x_i=1)；$w\leftarrow u$的右孩子 (//x_i=-1)
4. 在多项式时间内计算$q(v)$和$q(w)$
5. If $q(v)\geq q(w)$ Then $u\leftarrow w$,取定x_i= -1
6. Else $u\leftarrow v$, 取定x_i=1
7. 返回配置结果$(x_1,\ldots,x_n)$　　$\|Ax\|_\infty < 4\sqrt{n\ln n}$

问题：如何在n的多项式时间内计算$q(u)$呢？🤔

HIT CS&E

作业

1. 为什么$p(u)$无法在多项式时间计算？
2. 为什么$q(u)$能够在多项式时间计算？设计算法

结论

定理：确定型算法DetColoring能在多项式时间内求得x满足
$\|Ax\|_\infty < 4\sqrt{n\ln n}$

开放问题

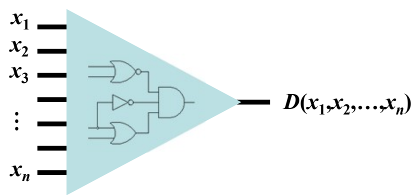
问题：是否存在确定型算法在多项式时间内求得x满足
$\|Ax\|_\infty = o(\sqrt{n\ln n}\ )$

HIT CS&E

7.2.3 随机电路去随机化

HIT CS&E

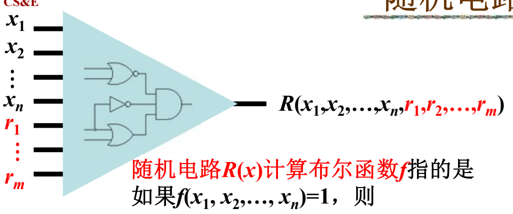
确定型电路D



确定型电路$D(x)$计算布尔函数$f(x)$指的是
$D(x) = f(x)$
在$x=(x_1,x_2,\ldots,x_n)$的2^n个不同取值上恒成立

HIT CS&E

随机电路R



随机电路$R(x)$计算布尔函数f指的是

如果$f(x_1, x_2,\ldots, x_n)=1$，则

$R(x_1, x_2,\ldots, x_n, r_1, r_2,\ldots, r_m) = 1$

至少对$r=(r_1,r_2,\ldots,r_n)$的2^{m-1}个不同取值上成立

也就是说

$\Pr[R(x,r)=0] = 1$　　若$f(x)=0$

$\Pr[R(x,r)=1] \geq 1/2$　　若$f(x)=1$

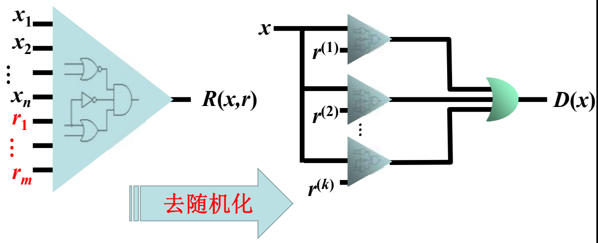
*注：1/2可以是[1/2,1)中的任意数

HIT CS&E

Adleman定理

定理[Focs1978]：如果是布尔函数$f(x)$可以用多项式规模的随机电路计算，则它也可以用多项式规模的确定型电路来计算

证明思路



HIT CS&E

去随机化过程

第一步：穷举所有x,r,列出$R(x,r)$的值

每列用$r=(r_1,\ldots,r_m)$的一组取值标定

2^m列

每行用$x=(x_1,\ldots,x_n)$的一组取值标定

2^n行

0	1	1	1	0	1	1	…	0
1	1	0	1	1	0	1	…	1
0	0	0	0	0	0	0	…	0
1	1	1	0	1	0	1	…	1
0	0	0	0	0	0	0	…	0
0	1	0	0	0	1	1	…	1
…	…	…	…	…	…	…	…	…
1	1	1	1	1	1	0	0	1

行x：恒等于0　$\Rightarrow f(x)=0$

行x：不恒等于0　$\Rightarrow f(x)=1$

$\Rightarrow \Pr[D(x,r)=1] \geq 1/2$

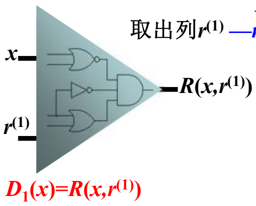
$\Rightarrow$该行至少一半以上的元素等于1

HIT CS&E

第二步：删除全0行，剩下的矩阵一半以上元素为1

至少有一列$r^{(1)}$一半以上元素为1

取出列$r^{(1)}$ —$r^{(1)}$对应$r=(r_1,\ldots,r_m)$的一组具体取值



$D_1(x)=R(x,r^{(1)})$

2^m列

2^n行

0	1	1	1	0	1	1	…	0
1	1	0	1	1	0	1	…	1
0	0	0	0	0	0	0	…	0
1	1	1	0	1	0	1	…	1
0	0	0	0	0	0	0	…	0
0	1	0	0	0	1	1	…	1
…	…	…	…	…	…	…	…	…
1	1	1	1	1	1	0	0	1

性质

在列$r^{(1)}$取1的所有行x上

$D_1(x)$均能正确计算$f(x)$

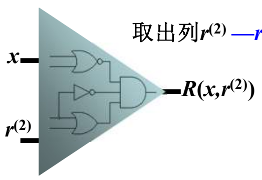
一半以上的行不用再考虑

HIT CS&E

第三步：删除$r^{(1)}$中元素等于1的行，至多剩下一半的行

至少有一列$r^{(2)}$一半以上元素为1

取出列$r^{(2)}$ —$r^{(2)}$对应$r=(r_1,\ldots,r_m)$的一组具体取值



$D_2(x)=R(x,r^{(2)})$

2^m列

2^n行

0	1	1	1	0	1	1	…	0
1	1	0	1	1	0	1	…	1
0	0	0	0	0	0	0	…	0
1	1	1	0	1	0	1	…	1
0	0	0	0	0	0	0	…	0
0	1	0	0	0	1	1	…	1
…	…	…	…	…	…	…	…	…
1	1	1	1	1	1	0	0	1

性质

在列$r^{(2)}$取1的所有行x上

$D_2(x)$均能正确计算$f(x)$

又有一半以上的行不用再考虑

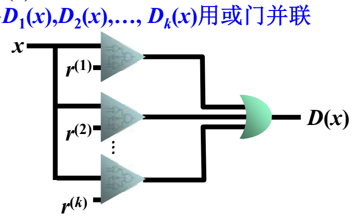
HIT CS&E

重复第三步

- 每次至少删掉一半以上的行
- 经过$k(k\leq n, k\leq m)$轮，所有行都将被删掉
- 得到$D_1(x),D_2(x),\ldots, D_k(x)$

重组得$D(x)$

- 将$D_1(x),D_2(x),\ldots, D_k(x)$用或门并联



HIT CS&E

Adleman定理

定理[Focs1978]：如果是布尔函数$f(x)$可以用多项式规模的随机电路计算，则它也可以用多项式规模的确定型电路来计算

- 是否意味着：每个蒙特卡罗算法都可以被去随机化吗？

  Yes, 但必须是“一致谏言”的

  $r^{(1)},\ldots,r^{(k)}$不是问题输入-谏言

  $r^{(1)},\ldots,r^{(k)}$需要确定型算法存储、处理

  带谏言的图灵机 =? 普通的图灵机
- 是否意味着：RP$\subseteq$P

  No. 仅仅是RP $\subseteq$ P/poly