ISSN 1000 9825 Journal of Software 软件学报 2000,11(10):1286~1293

并行密码体制的构造*

卿斯汉

(中国科学院软件研究所信息安全国家重点实验室 北京 100080)
(中国科学院信息安全技术工程研究中心 北京 100080)
E-mail: qsihan@yahoo.com

摘要 如何构造并行密码体制是值得关心的重要问题,也为构造与传统密码不同的密码体制开辟了一个新的思路.分析了串行环境和并行环境的内在不同点,根据并行环境的特征,基于并行复杂性理论提出一系列构造并行密码体制的基本理论和要求.

关键词 并行计算,串行计算,密码体制,计算复杂性理论.
中图法分类号 TP393

近年来,与传统密码不同的各类密码体制的研究逐渐深入.例如,Naor 和 Shamir 提出的视觉密码学(visual cryptography)[1]、Desmedt 等人提出的理智密码学(cerebral cryptography)[2]、Desmedt 等人提出的声学密码学(audio cryptography)和光学密码学(optical cryptography)[3]等等.对于量子密码学的研究也已经日渐成熟.

随着并行理论和各种并行机的迅速发展,如何利用并行计算构造并行密码体制是值得我们关心的重要问题,也为我们构造与传统密码不同的密码体制开辟了一个新的思路[4].首先,我们关心下述两个问题:

(1) 当转换一个串行算法为一个并行算法时,可否期望"显著"的加速?

(2) 有无本质上是串行的问题?亦即用并行的手段不可能"显著"地使其加速.

显然,如果能较好地解答上述问题,会有很大的实践意义.起码,对于一个应用问题,我们或者致力于探索好的并行算法,或者集中精力改善它的串行算法,努力增加串行机的处理速度等.

在密码背景下,我们显然希望加解密算法能在并行环境中提高速度,各种用于密码分析的算法亦然(便于攻击别人).反之,对自己设计的密码体制进行攻击时,希望密码分析是一个本质上串行的问题(能够保护自己).

但是,对并行环境的研究比对串行环境的研究困难得多.相对于串行环境,在并行环境中我们缺乏:

(1) 良好的数据结构;

(2) 有效的理论工具,如深度优先搜索算法等;

(3) 对计算问题进行合理的分类;

(4) 类似于 NP-困难性的概念;

(5) 类似于图灵机这样的计算模型.

特别地,在并行环境中,我们需要数学模型和计算复杂性理论.能否研制出一些类似于NP-完全性的概念呢?能否区分并行问题(即在并行环境下可望显著地加速)和串行问题(即本质上是串行的问题)这两类问题呢?如果并行理论是成功的,它应能做到:

(1) 将问题区分成上述两大类,在理论上是清晰而无二义性的;

(2) 这种区分方法是"现实"的,亦即它将实际问题分为容易和困难两类,其中容易类问题的并行算法明显比它的串行算法有效;

* 本文研究得到国家重点基础研究发展规划项目(No. G1999035810)资助.作者卿斯汉,1939 年生,研究员,博士生导师,主要研究领域为信息安全理论和技术.

本文通讯联系人:卿斯汉,北京 100080,中国科学院软件研究所

本文 2000-05-31 收到原稿,2000-06-30 收到修改稿

(3) 这种区分方法是与模型无关的.例如,在串行环境中,NP 完全性的概念是用图灵机定义的,但对任何合理的计算模型都成立.

本文探讨构造并行密码体制的一些基本理论和要求.为了下文的需要,这里先作一些说明和准备.

首先要说明的是,我们不但重视一个计算问题对时间的需求量,还重视它对硬件的需求量.在串行环境中,由于在 n 个时间单元内,最多可以访问 n 个存储单元,因此,硬件需求量受时间需求量的制约.而在并行环境下,由于任意多个处理机可以同时工作,因而上述结论不成立.其次,如何计算硬件资源呢?我们将把硬件与处理机视为同义语.类似于在串行环境中,我们视存储量为问题规模的函数.同样地,我们也视处理机个数为问题规模的函数.我们希望这样得到的并行时间界仅与问题的本质有关,而不依赖于某一个具体算法.

显然,一个计算问题在并行机上对时间与硬件需求量的乘积至少等于同一问题在串行机上对时间的需求量.因此,任何一个在并行机上用多项式时间与硬件可解的问题,在串行机上都属于 P 类问题.这样,作为较易下手的第 1 步,我们可以先讨论 P 类问题的子结构.例如,将本质上属于串行问题的类定义为在 P 中是 $\leqslant_m^{\log}$ 完全的问题,它们在某种意义上是 P 中最困难的问题.

下面给出一些下文中要用到的定义.

定义 1. 一个图灵机 TM 是一个 8 元组 $M=(Q,F,\Sigma,\Delta,\Gamma,q_0,\mathcal{B},\delta)$.其中 Q 为有限集(状态集),$F\subseteq Q$(最终状态),Γ 为有限集(工作带字符集),$\Sigma\subseteq\Gamma$(输入字符集),$\Delta\subseteq\Gamma$(输出字符集),$\mathcal{B}\in\Gamma-(\Sigma\cup\Delta)$(空白符),$q_0\in Q$(初始状态),$\delta:(Q-F)\times\Gamma\to Q\times\Gamma\times\{L,R\}$为部分函数,且 $Q\cap\Gamma=\varnothing$.

定义 2. $g:\Delta^4\to\Delta$ 函数定义如下:

$$g(t_{i-1},t_i,t_{i+1},t_{i+2})=\begin{cases}t_i, & \text{若 } t_{i-1},t_i,t_{i+1}\in Q;\\ q, & \text{若 } t_{i-1}\in Q,\text{且 } \delta(t_{i-1},t_i)=(q,a,R);\\ a, & \text{若 } t_{i-1}\in Q,\text{且 } \delta(t_{i-1},t_i)=(q,a,L);\\ a, & \text{若 } t_i\in Q,\text{且 } \delta(t_{i-1},t_{i+1})=(q,a,R);\\ t_{i-1}, & \text{若 } t_i\in Q,\text{且 } \delta(t_i,t_{i-1})=(q,a,L)\text{且 } t_{i-1}\neq\#;\\ t_i, & \text{若 } t_{i+1}\in Q,\text{且 } \delta(t_{i+1},t_{i+2})=(q,a,R);\\ q, & \text{若 } t_{i+1}\in Q,\text{且 } \delta(t_{i+1},t_{i+2})=(q,a,L);\\ \#, & \text{若 } t_i=\#;\\ \text{其余无定义.}\end{cases}$$

定义 3. 称 A 是 log 空间归约于 B 的,记为 $A\leqslant_m^{\log}B$,若存在一个 log 空间可计算函数 f(即计算 f 所需的空间为 $O((\log n))$),使得 $X\in A$ 当且仅当 $f(x)\in B$.

定义 4. 称语言 L 是 P 中$\leqslant_m^{\log}$-完全的,若下述二式成立:

(1) $L\in P$;

(2) $\forall L'\in P, L'\leqslant_m^{\log}L$.

1 并行环境下的数学模型

适用于并行环境的数学模型有线路,ATM,PRAM,反向有界的 TM 等等.

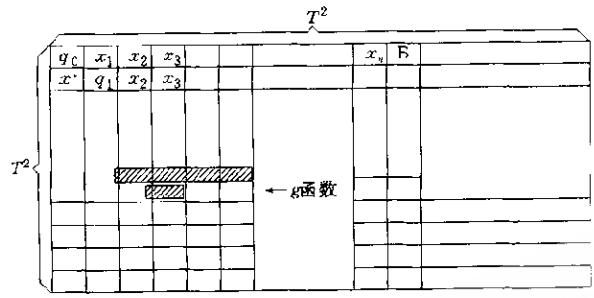
本文选中布尔组合线路,我们认为这是一种便于处理的模型,因为实际的计算机是用这种线路构成的.

定义 1.1. 布尔组合线路是 DAG,其结点(门)的入度$\leqslant\alpha$(通常 $\alpha=2$),其中 α 值固定.输入结点的入度为 0,输出结点的出度为 0.除输入结点外,所有入度为 k 的结点都标有一个布尔函数 $f_k:B^k\to B, B=\{0,1\}$.输入结点则标记为"x".线路大小是结点的个数,线路的深度是由输入到输出的最长路径的长度.

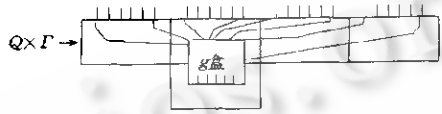
定义 1.2. PSIZE 是线路大小可用多项式表示的布尔组合线路.

定理 1.3. $P\subseteq$PSIZE.

证明:设 m 为运行时间是 $T(n)$的多带 TM,其中 $T(n)$为多项式.设 m' 为运行时间 $T^2(n)$且识别同一语言的单带 TM.我们构造一个模拟 m'的线路.考虑在每一时刻的带的内容:



其中 g 函数的定义见定义 2,它由 4 个符号计算 1 个符号.将其放大后,可以得到:



g 盒的大小和深度为 $O(1)$.

在上述图表中每一小方块中都放置一个 g 盒(边缘方块需特殊处理),就构造出一个线路,其大小与深度分别为 $O(T^4)$ 与 $O(T^2)$. □

定义 1.4. 线路族 C 是序列 $\{C_0,C_1,\ldots,C_n,\ldots\}$,其中 C_i 有 i 个输入.称 C 的大小和深度复杂性为 $Z(n)$ 和 $D(n)$,若 $\forall n$,C_n 的大小 $\leqslant Z(n)$ 且 C_n 的深度 $\leqslant D(n)$.称族 $C=\{C_0,C_1,\ldots,C_n\}$,识别语言 $L\subseteq\{0,1\}^*$,若 C_n 识别 $L\cap\{0,1\}^n$.亦即,若 $x=x_1,\ldots,x_n$,则 $X\in L$ 当且仅当 C_n 对于输入 $x_1,\ldots,x_n$ 输出 1.

定理 1.5. 存在大小与深度均为 1 且可计算非递归函数的线路族*.

证明:令 L 为任意非递归集.令

$$\hat{L}=\{y\mid \text{length}(y)\in L\}.$$

因此

$$\hat{L}\cap\{0,1\}^n=\begin{cases}\varnothing, & \text{若 } n\in L\\ \{0,1\}^n, & \text{若 } n\in L\end{cases}.$$

我们定义

$$C_n=\begin{cases}0, & \text{若 } n\in L\\ 1, & \text{若 } n\in L\end{cases}.$$

于是 C_n 识别 $\hat{L}\cap\{0,1\}^n$. □

我们知道,线路的大小和深度不一定表示一个问题的复杂性.由定理 1.5 可知,C_n 是不容易计算的.因此,在某种意义上,我们希望研究"均匀"的线路.例如,定义均匀性的概念为给定 n,存在计算线路 C_n 的算法.

定义 1.6. 设 C 为线路,则 C 的标准编码(记为 $\bar{c}$)是形为 $[g,t,g_L,g_R]$ 的四元组的串,其中 $g\in\{0,1\}^*$ 为门编号;$t\in\{x,\wedge,\vee,\neg,\oplus,\ldots\}$ 是门类型,其中"x"=输入,其余为布尔函数算符;$g_L,g_R\in\{0,1\}^*$ 是门 g 的左、右门的编号(若类型为 t 的门 g 有 1 个或 0 个输入,则省去 g_R,或省去 g_L 与 g_R).

定义 1.7. 深度和大小分别为 $T(n)$ 和 $Z(n)$ 的线路族 $C=\{C_0,C_1,\ldots\}$ 是

(1) U_P-均匀的,若通过 DTM(确定型图灵机),函数 $n\rightarrow\bar{c}_n$ 在 $Z(n)$ 的多项式时间内是可计算的;

(2) U_{BC}-均匀的,若通过 DTM,函数 $n\rightarrow\bar{c}_n$ 在 $\log Z(n)$ 空间内是可计算的;

(3) U_R-均匀的,若通过 DTM,函数 $n\rightarrow\bar{c}_n$ 是在 $T(n)$ 空间内可计算的.

对于上述定义以及下面将给出的其他定义,$X\text{-}SIZE(Z(n))$ 表示由大小复杂性为 $O(Z(n))$ 的 X-均匀线路可以识别的语言类.$X\text{-}DEPTH(T(n))$ 的含义类似.$X\text{-}SIZE,DEPTH(Z(n)T(n))$ 表示由同时有大小为 $O(Z(n))$ 且深度为 $O(T(n))$ 的 X-均匀线路可以识别的语言类.

---

* 在定理的构造中,不计输入结点的大小.

下述定理说明，线路大小是一种合理的复杂性度量标准.

定理 1.8. $P=U_{BC}\text{-}SIZE(n^{O(1)})=U_P\text{-}SIZE(n^{O(1)})$.

证明：我们下面将分以下 3 步加以证明：

(1) $P\subseteq U_{BC}\text{-}SIZE(n^{O(1)})$；

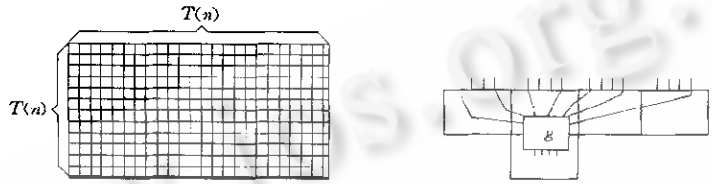
(2) $U_{BC}\text{-}SIZE(n^{O(1)})\subseteq U_P\text{-}SIZE(n^{O(1)})$；

(3) $U_P\ SIZE(n^{O(1)})\subseteq P$.

对(1)的证明. 我们证明下述较强的论断：

$$ONETAPE\text{-}DTIME(T(n))\subseteq U_{BC}\text{-}SIZE(T^2(n)).$$

设给定运行时间为 $T(n)$的单带 DTM，对应的线路为 $T(n)\times T(n)$的"黑盒"表，该黑盒计算函数 g.



为了计算线路的标准编码，令三元组(i,j,k)为门编号，i 和 j 说明 g 盒的行与列号；$1\leqslant i,j\leqslant T$；$K$ 指明黑盒内所考虑的门，$1\leqslant k\leqslant K$，K 与 n 无关. 为方便计，我们将按(i,j,k)的字典顺序输出门. 显然，对于每一个三元组(i,j,k)：

$$g=(i,j,k),$$

t 由 i,j,k 确定.

g_L,g_R：若门 g 不是黑盒的输入，则 g 的输入有相同的 i 和 j 值，且由黑盒的内部结构可得出 k 值. 若门 g 是黑盒的输入，例如 $g=(i,j,k)$，则 $g_L=(i',j',k')$，其中 $i'=i-1,j-1\leqslant j'\leqslant j+2$，且通过黑盒的结构可以确定 j' 和 k'值.

因此，我们可以输出$[g,t,g_L,g_R]$. 显然，输出的大小为 $O(T^2(n))$. 那么，计算$[g,t,g_L,g_R]$需用多少空间呢？计算 k 并存储黑盒的内部结构需常数空间. 对于每一对 i 和 j，计算 $T(n)$需用空间 $\log T(n)$. 对(1)证毕.

对(2)的证明. 对于 U_{BC}-均匀线路，计算 $n\to\bar{c}_n$ 所用的空间仅为 $\log Z(n)$. 因此，它仅需运行时间 $2^{O\log Z(n)}=Z(n)^{O(1)}$. 于是，

$$U_{BC}\text{-}SIZE(Z(n))\subseteq U_P\text{-}SIZE(Z(n)^{O(1)}).$$

但我们只对 $Z(n)$为 n 的多项式这种线路感兴趣，因此对(2)证毕.

对(3)的证明. 我们将证明：对 $Z(n)=\Omega(n)$，

$$U_P\text{-}SIZE(Z(n))\subseteq DTIME(Z(n)).$$

给定一个大小为 $Z(n)$的 U_P-均匀线路族，构造一个如下操作的DTM：

(a) 求 n(输入长度)；

(b) 对于 n 运行"均匀机器"，生成 $\bar{c}_n$；

(c) 对于给定的输入模拟该线路.

求 n 所需的时间为 n，计算 $\bar{c}_n$ 所需的时间为 $Z(n)$的多项式，模拟线路所需时间$\leqslant Z(n)$的多项式. 因此，总运行时间为 $Z(n)$的多项式. 对(3)证毕. □

已知 $DSPACE(\log n)\subseteq DTIME(n^{O(1)})$，且人们相信(无法证明)$\subseteq$应为$\subset$. 因此，$U_{BC}$-均匀性比 U_P-均匀性限制更强. 定理 1.8 表明，$U\text{-}SIZE(n^{O(1)})$是一种与模型无关的度量，因为它对 U_{BC}-最严格的均匀性类和对 U_P-最松的均匀性类是一样的(就此 3 种定义而言).

定义 1.9. 线路族 $C=\{C_0,C_1,\ldots\}$的直接连接语言 L_{DC}是形如$\langle n,g,p,y\rangle$的字符串集合. 其中 $n\in\{0,1\}^*$为二进制输入，$g\in\{0,1\}^*$为 C_n 中的门号，$p\in\{\in,L,R\}$，$y\in\{x,\wedge,\vee,\neg,\ldots\}\cup\{0,1\}^*$或为门类型，或为门号；

且 P 中的 3 种情形分别表示：

$\in$：$Y\in\{X,\wedge,\vee,\neg,\ldots\}$且 C_n 中的门 g 类型为 y.

L：$Y\in\{0,1\}^*$且 C_n 中的门 Y 是门 g 的左输入.

R：$Y\in\{0,1\}^*$且 C_n 中的门 Y 是门 g 的右输入.

定义 1.10. C 的扩展连接语言 L_{EC} 的定义同 L_{DC}. 其不同之处为 $P\in\{L,R\}^*$，$|P|\leqslant\log$(C_n 的大小)，且若 $|P|\geqslant 1$，条件变为 $y\in\{0,1\}^*$，并且在 C_n 中，门 y 是由门 g 开始，跟随输入路径 P 所达到的门.

显然，L_{DC} 完全勾画出线路族. 四元组$\langle n,-,-,-\rangle$表示 C_n. 四元组$\langle n,g,-,-\rangle$指出 C_n 中出现哪些门. 四元组$\langle n,g,\in,y\rangle$指出门 g 的类型，而四元组$\langle n,g,L,Y\rangle$和$\langle n,g,R,Y\rangle$指出门 g 的输入. L_{EC} 除含有上述信息外，还包含若干沿路径的信息.

定义 1.11. 深度和大小分别为 $T(n)$ 和 $Z(n)$ 的线路族 C 是

(1) U_D-均匀的，若存在一个对于形为$\langle n,-,-,-\rangle$的输入，能在时间 $O(\log Z(n))$ 内识别 L_{DC} 的 DTM；

(2) U_E-均匀的，若存在一个对于形为$\langle n,-,-,-\rangle$的输入，能在时间 $O(\log Z(n))$ 内识别 L_{EC} 的 DTM；

(3) U_{E^*}-均匀的，若存在一个对于形为$\langle n,-,-,-\rangle$的输入，能在时间 $O(T(n))$ 和空间 $O(\log Z(n))$ 内识别 L_{EC} 的 ATM(Alternating 图灵机).

定理 1.12. 对于 $S(n)=\Omega(\log Z(n))$，下述 3 个条件等价：

(1) $n\to C_n$ 是 $DSPACE(S(n))$ 内可计算的；

(2) L_{EC} 中形为$\langle n,-,-,-\rangle$的串是 $DSPACE(S(n))$ 内可识别的；

(3) L_{DC} 中形为$\langle n,-,-,-\rangle$的串是 $DSPACE(S(n))$ 内可识别的；

证明：(1)⇒(2)：给定输入$\langle n,g,p,y\rangle$，(a) 若 $P=\in$，则运行计算 $n\to C_n$ 的机器，直到生成$\langle g,t,-,-\rangle$，且接受当且仅当 $t=y$. (b) 若 $p=L_{p'}$，其中 $p'\in\{L,R\}^*$，则运行计算 $n\to C_n$ 的机器，直到生成$[g,t,g_L,-]$，然后以$\langle n,g_L,p',y\rangle$为输入，重复运行. (c) 若 $P=RP'$，类似于(b)的处理.

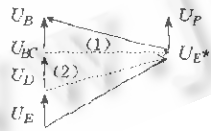
所需空间为，运行 $n\to C_n$ 的机器用空间 $S(n)$. 当 $|P|\geqslant 1$ 时，我们还需空间求 P' 并存储 g_L，为此，空间 $S(n)$ 是足够的. □

(2)⇒(3)：由 $L_{DC}\subseteq L_{EC}$ 立即可以得到.

(3)⇒(1)：给定 n 和 L_{DC} 识别器，我们可以由 $g=1$ 开始，按数字顺序输出各门. 首先，测试 C_n 中有无门号为 g 的门，即在 L_{DC} 识别器对各种可能的 t 值测试$\langle n,g,\in,t\rangle$，若存在编号为 g 的门，则输出“$[g,t,$”.

其次，通过在 L_{DC} 识别器中对 $\forall i\geqslant 0$ 测试$\langle n,g,L,i\rangle$求出输入(如果有的话). 将 i 值增 1，重复上述过程，直到找出输入，届时输出“i”，类似地试$\langle n,g,R,j\rangle$，直至找出输入，并输出“j]”为止. 定理证毕.

下面的定理给出了均匀线路的 6 种不同定义之间的关系.



(1) if $T(n)\geqslant\log^2 Z(n)$①
(2) if $T(n)\geqslant\log Z(n)\cdot\log\log Z(n)$②

①当$T(n)\geqslant\log^2 Z(n)$时成立，②当$T(n)\geqslant\log Z(n)\cdot\log\log Z(n)$时成立.

Fig. 1
图1

定理 1.13. 令 $\zeta=\{C_0,C_1,\ldots\}$为大小和深度分别是 $Z(n)$ 和 $T(n)$ 的线路族，图 1 给出了 6 种不同的均匀条件之间的关系，图中 $A\cdot\to\cdot B$ 表示根据定义 A 的均匀性⇒根据定义 B 的均匀性.

证明：其中 7 个关系的证明如下.

(1) $U_E\to U_{E^*}$：因为 $T(n)\geqslant\log Z(n)$，且 $DTIME(T(n))\subseteq ATIME(T(n))$. 于是，

$$DTIME(\log Z(n))\subseteq ATIME,SPACE(T(n),\log Z(n)).$$

(2) $U_E\to U_D$：因为 $L_{DC}=L_{EC}\cap R$，而 R 是正则的.

(3) $U_D \to U_{BC}$：由定理 1.12 和下述事实立即可以得到

$$DTIME(\log Z(n)) \subseteq DSPACE(\log Z(n)).$$

(4) $U_{BC} \to U_B$：因为入度有界，故有 $T(n) \geqslant \log Z(n)$，因此得到

$$DSPACE(\log Z(n)) \subseteq DSPACE(T(n)).$$

(5) $U_{BC} \to U_P$：因为下述论断成立：

$$DSPACE(\log Z(n)) \subseteq DTIME(Z(n)^{O(1)}).$$

(6) $U_{E^*} \to U_B$：由定理 1.12 及下述论断立即可以得到

$$ATIME(T(n)) \subseteq DSPACE(T(n)).$$

(7) $U_{BC} \to U_{E^*}$：若 $n \to \bar{c}$，在 $DSPACE(\log Z(n))$ 中可计算的，则由定理 1.12，L_{EC} 是在 $DSPACE(\log Z(n))$ 中可识别的，因而在 $ASPACE, TIME(\log Z(n), \log Z(n))$ 中可识别(Savitch 定理)．于是，只要 $T(n) \geqslant \log^2 Z(n)$，$L_{EC}$ 即在 $ASPACE, TIME(\log Z(n), T(n))$ 中可识别，即线路 U_{E^*} 是均匀的． □

由此可见，在上述定义中，U_E 是最严格的．因为，任何 U_E-均匀的线路也是其他定义下均匀的线路，但是，我们不知道 U_P 和 U_B 之间的关系，因为我们无法确定 $T(n)$ 与 $Z(n)$ 之间的一般关系．

2 NC 类与 P-完全性问题类

由前面的结果，可以定义重要的 NC 类．

定义 2.1.

$$NC^K = U_E\text{-}SIZE, DEPTH(N^{O(1)}, \log^k n),$$

$$NC = \bigcup_K NC^K (\text{语言族}).$$

可以证明，NC 在定义 U_E, U_{E^*}, U_D 和 U_{BC} 下是相同的．此外，当 $K \geqslant 2$ 时，NC^K 在上述 4 种定义下亦相同．当 $K \geqslant 1$ 时，NC^K 在定义 U_E 与 U_{E^*} 下相同．还可以证明：

$$NC^K = ASPACE, TIME(\log n, \log^K n).$$

考虑这一类问题的理由如下：(1) 时间的加速(相对于串行机)是近指数的，同时，硬件量的增加是合理的；(2) 许多实际问题属于这一类；(3) 除了某些模型(如网状与树型连接的并行机)以外，较好地实现了与模型无关的分类要求；(4) 在大多数模型中，对于输入为 n 的非平凡函数，时间下界为 $\log n$．因此，时间 $(\log n)^{O(1)}$ 是近于最优的．例如，在布尔组合线路中，若门的扇入为 2，则为了由 n 个输入获得 1 个输出，我们需要一棵高度为 $\log n$ 的树．此外，这样分类还有一些技术上的方便之处．例如，在空间的归约下封闭，等等．

定理 2.2. $NSPACE(\log n) \subseteq NC^2 \subseteq DSPACE(\log^2 n)$.

证明：由 Savitch 定理，

$$NSPACE(\log n) \subseteq ASPACE, TIME(\log n, \log^2 n) = NC^2. \text{ 并且，}$$

$$NC^2 = ASPACE, TIME(\log n, \log^2 n) \subseteq ATIME(\log^2 n) \subseteq DSPACE(\log^2 n).$$ □

根据上述定理以及其他知识，我们将若干重要的复杂性类之间的关系作以下分析，如图 2 所示．对图 2，我们作以下说明：

(1) 对图中所示的 $\subseteq$ 关系，我们猜测(无法严格证明)是 $\subset$ 关系；

(2) 由空间层次定理可得 $DSPACE(\log n) \subsetneq DSPACE(\log^2 n)$；

(3) $P \neq SPACE(\log^{O(1)} n)$．$P$ 中有 $\leqslant_m^{\log}$-完全集(例如，CVP，即线路值问题)，但由空间层次定理可知，$SPACE(\log^{O(1)} n)$ 中没有这种集合．但是我们不清楚它们之间的关系，例如，是 $\subseteq$ 还是 $\supseteq$ 关系呢？多半，它们是不可比较的．

(4) 更重要的问题是：NC 与 P 之间的关系如何？对 NC 而言不存在层次定理．问题在于，我们必须同时考虑大小与深度，因此无法使用“对角线法”这一强有力的工具．因而，不知道是否 $NC^1 = NC$，也不知道是否 $NC^1 = P$，甚至不知道是否 $NC^1 = NP$.

(5) 我们知道 $NC \subseteq P$ 且 NC 与空间有关，但是，空间本身并不确定 NC 的特征，因为它允许大小特别大的线路存在．

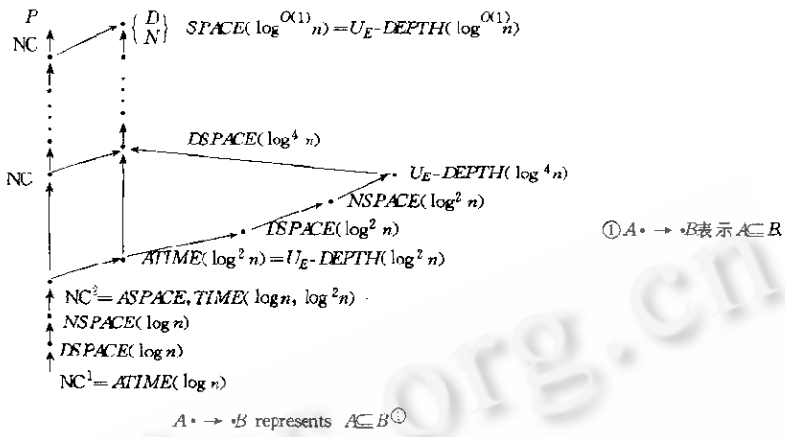


Fig. 2
图2

(6) 深度为 $\log^2 n$ 的线路的大小可以达到 $2^{\log^2 n}$,因而我们怀疑 U_E-$DEPTH(\log^2 n)\subseteq P$ 的正确性.

由以上结论可知,许多重要问题有快速的并行解.例如,$DSPACE(\log n)$中有下述问题:

(i) 整数加法与减法;

(ii) n 个 n 比特整数的求和;

(iii) 整数乘法;

(iv) 排序问题:给定 n 个数为输入,输出排好序的 n 个数;

(v) K-团(clique)问题:若 K 作为输入的一部分,则为 NP-完全问题.此处,我们讨论 K 固定时的情形.

又如,$DSPACE(\log n)$中含有下述问题:

(i) 图连通性问题:给定图 G 和两个顶点 s 和 t,确定由 s 到 t 有无直接路径.

(ii) 最短路径(当边权为 $O(\log n)$比特时)问题:由 s 到 t 的最短路径长度是否小于 K?

然而,仍有许多重要的实际问题,不可能有明显的并行加速,它们属于 P-完全性问题,即在 P 中$\leqslant_m^{\log}$-完全性问题(定义 4).目前已知此类问题有几十种,我们下面仅讨论其中一种,即 CVP(线路值)问题.

给定一个布尔组合线路及其各输入值,问其输出值是否为 1,此即 CVP 问题.不难证明,CVP$\in P$.实际上,在 TM 中由线路的输入值计算其输出值所需运行时间为 $O(n\log n)$.其次,令 $L\in P$,则由定理 1.8 可知,

$$P=U_{BC}\text{-}SIZE(n^{O(1)}).$$

因此,$L\leqslant_m^{\log}$CVP.从而证明了 CVP 是 P-完全性问题.对于 CVP 这一类问题,不可能存在好的并行加速算法.

我们有理由相信,NC 类与 P-完全性类是一种区分两类不同问题的方法,前者是并行问题,后者是串行问题.可以证明:若 L 是 P 中$\leqslant_m^{\log}$-完全的且 $L\in NC^K(K\geqslant 2)$,则 $P=NC^K$.但迄今为止,尚未发现任何一个同属于 NC^K 及 P 完全性类的问题.下述观点也强有力地支持 $P\neq NC$ 这一猜测:虽然我们只能用限制性很强的计算模型(例如,掷卵石博奕模型),而不能用通用模型(例如 TM)来证明这一猜测,但还没有人在反证这一猜测方面获得任何进展.

最后,我们指出,这一分类法还远远不够完善.有许多问题既不能证明属于 NC 类,也不能证明属于 P-完全性类.而在串行环境,大多数问题均属于 NP-困难类或 P 类,只有少数问题(例如,分解因式、图的同构)无法分类.此外,我们曾指出过,这一分类法还不是完全与模型无关的.

就密码学的应用而言,目前并行理论的研究更是远远不能满足需要.特别是,NP-困难类问题有无并行加速的可能?无论是肯定或者是否定的一般性答复,显然均将对密码学产生重大影响.

如何构造并行密码体制是值得我们关心的重要问题,也为我们构造与传统密码不同的密码体制开辟了一个

新的思路.本文分析了串行环境和并行环境的内在不同点,根据并行环境的特征,基于并行复杂性理论提出一系列构造并行密码体制的基本理论和要求.

参考文献

1 Naor M, Shamir A. Visual cryptography. In: Santis A D ed. Advances in Cryptology—— Eurocrypt'94 Proceedings. LNCS 950, Berlin: Springer-Verlag, 1995. 1~12

2 Demedt Y, Hou S, Quisquater J J. Cerebral cryptograpny. In: Workshop on Information Hiding Proceedings. Portland, Oregon, USA, 1998. 21~32

3 Demedt Y, Hou S, Quisquater J J. Audio and optical cryptography. In: Ohta K, Pei D eds. Advances in Cryptology—— Asiacrypt'98 Proceedings. LNCS 1514, Berlin: Springer-Verlag, 1998. 392~404

4 Qing Si-han. Cryptographic systems in parallel environment. Technical Report, Engineering Research Center for Information Security Technology, The Chinese Academy of Sciences, 1999
(卿斯汉.并行环境下的密码体制.科技报告,中国科学院信息安全技术工程研究中心,1999)

Construction of Parallel Cryptographic Systems

QING Si-han

(*State Key Laboratory of Information Security Institute of Software The Chinese Academy of Sciences Beijing* 100080)
(*Engineering Research Center for Information Security Technology The Chinese Academy of Sciences Beijing* 100080)

Abstract How to construct a parallel cryptographic system is an important issue which needs to be considered. In this direction people may have a new way to build a cryptosystem different from the traditional cryptography. In this paper, the author analyzed the differences between serial and parallel environments, and the features of parallel environment. Based upon the computational complexity theory, the basic theory and requirements for building parallel cryptographic systems are presented.

Key words Parallel computation, serial computation, cryptographic system, computational complexity theory.