第3卷第4期
1992年11月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 3, No. 4
Nov., 1992

关于概率无限寄存器机器PURM及其程序可模拟的随机函数

党 哲　　周维芳

（南开数学研究所，天津 300071）　（南开大学计算机与系统科学系，天津 300071）

THE PROBABILISTIC UNLIMITED REGISTER MACHINE (PURM) AND THE RANDOM FUNCTIONS THAT CAN BE SIMULATED BY PURM PROGRAMS

Dang Zhe

(*Nankai Institute of Mathematics, Tianjin* 300071)

Zhou Weifang

(*Department of Computer Science, Nankai University, Tianjin* 300071)

Abstract In this paper, a new model for randomized computation——the Probabilistic Unlimited Register Machine (PURM) which is simpler than the Probabilistic Turing Machine (PTM)[4,5] is given. We prove the equivalence between PURM program and computable PTM. Moreover, by constructing the PURM programs, we get the sufficiant conditions for a random function that can be simulated by PURM program or PTM. At last, some simple properties of PTM and PURM are discussed.

摘要 本文给出了一种新的随机计算的机器模型：概率无限寄存器机器PURM，它比概率Turing机[4,5](PTM)更为简单.我们证明了PURM程序与可计算的PTM之间的等价性.基于对PURM程序的构造，我们给出了随机函数可被PURM程序或可计算的PTM模拟的充分条件.最后，讨论了PTM和PURM的一些简单性质.

本文1990年1月7日收到，1990年11月15日定稿.作者**党哲**，助教，1990年硕士毕业于南开大学，现工作单位为河北工学院计算机系，主要从事理论计算机科学，并行处理，人工智能方面的研究工作.**周维芳**，1989年本科毕业于南开大学，现为在读硕士生，目前主要从事人工智能、理论计算机科学方面的研究工作.

§1. 引 言

作为随机算法(randomized algorithm)[7]的机器模型,概率 Turing 机(PTM)[4,5]和概率自动机(PA)[1,3]扮演着十分重要的角色.基于无限寄存器机器(URM)我们构造比 PTM 更为直观和简单的概率无限寄存器机器(PURM),并证明了 PURM 与可计算的 PTM 间的等价性.利用这一结果,本文给出了随机函数可被 PURM 或可计算的 PTM 模拟的充分条件.

§2. 概率无限寄存器机器 PURM

定义 2.1:$PURM(A)$包括有穷字母表 $A=\{a_0,a_1,\cdots,a_t\}$,$t\geqslant 1$,和一条可枚举的无限寄存器序列 $R_0,R_1,\cdots$.以 r_i 标记 R_i 的内容,$i\geqslant 0$,其中 $r_0\in\{a_0,a_1\}$,$r_j\in A^*$,$j\geqslant 1$,A^* 为 A 上的字集.$PURM(A)$的指令为:(1)$Z(n)$:标记 R_n 为空(即 $r_n=0$),$n>0$;(2)$S^{(i)}(n)$:r_n 末端加上字符 a_i,$i\leqslant t$,$n>0$;(3)$T(m,n)$:以 r_m 替换 r_n,$m,n>0$;(4)$J(m,n,q)$:若 $r_m=r_n$ 则转到第 q 条指令执行,否则执行下一条指令,$m,n\geqslant 0$;(5)$TOSS$:以 1/2 的概率替换 r_0 为 a_0 或 a_1.

定义 2.2:$PURM(A)$程序 P 是 $PURM(A)$的指令的有穷序列 $I_1,\cdots,I_s$.起始带形式是寄存器内容序列 $a_0,a_1,\cdots,a_n,0,0,\cdots$(简记为 $a_0,a_1,\cdots,a_n,0^\infty$),其中 $a_i\in A^*$,$1\leqslant i\leqslant n$,称 $a_1,\cdots,a_n$ 为输入.程序运行结束(即无下一条指令可执行)时,以 R_1 的内容为输出.记 $Prob(P(\alpha^{(n)})=\beta)$为程序 P 以 $\alpha^{(n)}=a_1,\cdots,a_n$ 为输入,$\beta\in A^*$ 为输出的概率.

注意到任何 $PURM(A)$程序 P 只涉及到有穷多个寄存器,显然有:

定理 2.1:给定任意自然数 N_0 和 $PURM(A)$程序 P,存在程序 P'满足:在相同输入下 P 和 P'运行结束时在 $R_0,\cdots,R_{N_0}$各寄存器上有相同内容的概率相等且 P'由以下指令构成($M=1,2,\cdots$):(1)$S_M^{(i)}(n)$:r_n 末端加上字符 $a_i\in A$,对 $m\neq n$,$0\leqslant m\leqslant M$,r_m 不变,但 $m>M$,r_m 可能改变;(2)$D_M(n)$:删除 r_n 末端字符,对 $m\neq n$,$0\leqslant m\leqslant M$,r_m 不变,但 $m>M$,r_m 可能改变;(3)$\overline{J}_M(n)[E_1]$:若 $r_n\neq 0$,则转到 E_1 执行,否则执行下一条指令,对 $0\leqslant m\leqslant M$,r_m 不变,但 $m>M$,r_m 可能改变;(4)$TOSS_M$:以 $\frac{1}{2}$ 概率替换 r_0 为 a_0 或 a_1,对 $1\leqslant m\leqslant M$,r_m 不变,但 $m>M$,r_m 可能改变.

§3. 概率 Turing 机(PTM)

本节中部分定义选自文[4,5]中略作修改的形式.

定义 3.1:概率 Turing 机(PTM)包括带符号集 A,内部状态集 S,转移函数 $p:S\times A\times V\times S\to[0,1]$和初始函数 $h:S\to[0,1]$.其中 $A\cap S=\varnothing$,$\cdot\in S$,$V=A\cup\{R,L\}$,$R,L\notin A$ 且 p,h 满足 (1)$\sum_{v\in V}\sum_{s'\in S}p(s,a,v,s')=1$,对每个 $a\in A$,$s\in S$ 成立;(2)$\sum_{s\in S}h(s)=1$.与 *Turing* 机类似,*PTM* 作用在单向无穷长存贮带上,$R,L,\cdot$ 分别代表读头右移一格,左移一格和停机.转移函数 p 给出了 *PTM* 在当前状态为 s,扫视带符号 a 时"下一个动作"的条件概率,其中若 $p(s,a,v,s')\neq 0$,则"下一个动作"可以是:(1)若 $v\in A$,则用 v 替换 a 并进入状态 s';(2)若 $v=R$(或 L),则右移(或左移)一格并进入状态 s';(3)若 $s'=\cdot$,则停机.

称 $s_0\in S$ 为 *PTM* 的初始状态若 $h(s_0)=1$,$s_0\neq\cdot$.如果 *PTM* 的起始带形式为 $\alpha 0^\infty$(0 表示空),停机时的最终带形式为 $\beta 0^\infty$,$\alpha,\beta\in A^*$,则称 α,β 分别为输入和输出.对于 k 维输入 $\alpha^{(k)}=$

$(\alpha_1,\cdots,\alpha_k)\in\underbrace{A^*\times\cdots\times A^*}_{k}$，扩充带符号集为 $A\cup\{\pi\}$，$\pi\notin A$，$\alpha^{(k)}$为输入当且仅当起始带形式为 $\pi\alpha_1\pi\cdots\pi\alpha_k\pi 0^\infty$；$k$ 维输出可类似地处理.

定义 3.2：从 X^n 映到 Y 的随机函数 f 被特征函数 $\mu_f: X^n\times Y\to[0,1]$ 刻划，其中对任 $x^{(n)}\in X^n$，$y\in Y$，$\mu_f(x^{(n)},y)$是 $f(x^{(n)})$取值 y 的概率且 $\sum_{y\in Y}\mu_f(x^{(n)},y)\leqslant 1$.

定义 3.3：给定任意正整数 k 和 PTM Z，定义随机函数 $\Phi_Z^{(k)}$ 从 $\underbrace{A^*\times\cdots\times A^*}_{k}$ 映到 A^* 满足：

$\mu_{\Phi_Z^{(k)}}(\omega^{(k)},\omega)=\sum_{\substack{\beta\in\psi(Z)\\ \langle\beta\rangle=\omega}}\sum_{s\in S}h(s)t_z(\overbrace{s\omega^{(k)}}^{k},\beta)$，式中 $\omega^{(k)}\in\underbrace{A^*\times\cdots\times A^*}_{k}$，$\omega\in A^*$，$\psi(Z)$为 Z 的瞬时描述集，$\langle\beta\rangle=\omega$ 表示瞬时描述为 β 时其带形式是 $\omega 0^\infty$，$t_Z(\overbrace{s\omega^{(k)}}^{k},\beta)$表示 Z 在状态为 $s\in S$，带形式为 $\omega^{(k)}0^\infty$时经过有穷步运算到达瞬时描述 β 的概率. 符号的详细说明见[5].

直观上，$\mu_{\Phi_Z^{(k)}}(\omega^{(k)},\omega)$表示 Z 在输入为 $\omega^{(k)}$时停机且输出 ω 的概率. 当 $k=1$ 时，记 $\Phi_Z^{(1)}$ 为 Φ_Z. 为直观起见又写 $\mu_{\Phi_Z^{(k)}}(\omega^{(k)},\omega)$为 $Prob(\Phi_Z^{(k)}(\omega^{(k)})=\omega)$.

定义 3.4：称实数 $\alpha\in[0,1]$是可计算的实数当且仅当 α 的二进展式 $\sum_{i=0}^{\infty}\alpha_i 2^{-i}$满足 α_n 对 n 是递归的；称 PTM Z 是可计算的 PTM 当且仅当其转移函数 p 和初始函数 h 的值域中的所有值均是可计算的实数.

正如文[2]中所指出的那样，可计算的 PTM 是与这样的 PTM 等价的：其转移函数的值域为 $\{0,\frac{1}{2},1\}$，具有初始状态 s_0.

§4. 等价性结果

定理 4.1：任给 $PURM(A)$程序 P，存在一个以 $A'\supset A$ 为带符号集的可计算的 PTM Z 满足对任 $r_1,\cdots,r_n,\xi\in A^*$，

$Prob(\Phi_Z^{(n)}(r_1,\cdots,r_n)=\xi)=Prob(P(r_1,\cdots,r_n)=\xi)$.

证明：设 $P=I_1,\cdots,I_k$，M_P为 P 中出现的最大寄存器编号. 设 $A'=A\cup\{\$,\pi,\varepsilon\}$，$\$,\pi,\varepsilon\notin A$. Z 的起始带形式为 $\$a_0\pi r_1\pi\cdots\pi r_n\pi\cdots\pi 0^\infty$，最终带形式为 $\$r_0\pi\xi\pi\cdots\pi 0^\infty$，两式中均有 M_P+1 个 π，S 为 Z 的状态集，$s_1\in S$ 为 Z 的初始状态. 由定理 2.1(N_0 不妨取作 M_P)，对每个 I_m，$1\leqslant m\leqslant k$ 只要进行如下的模拟(M 为自然数)：

若 I_m 是 $TOSS_M$，则有：$p(s_m,a,L,s_m)=1$，任 $a\in A'$，$a\neq\$$；$p(s_m,\$,R,s'_m)=1$；$p(s'_m,a,a_0,s''_m)=\frac{1}{2}$，任 $a\in A'$；$p(s'_m,a,a_1,s''_m)=\frac{1}{2}$，任 $a\in A'$；$p(s''_m,a,R,s_{m+1})=1$，但 $a\in\{a_0,a_1\}$，若 $m=k$，则 $s_{m+1}=\cdot$.

以上模拟中 s_m,s'_m,s''_m 均为 S 中元素.

若 I_m 是 $S_M^{(i)}(n)$，则有：$p(s_m,a,L,s_m)=1$，任 $a\in A'\backslash\{\$\}$；$p(s_m,\$,R,s'_m)=1$；$p(s'_m,a,R,s'_m)=1$，任 $a\in A$；$p(s'_m,\pi,R,s_{m1})=1$；$p(s_{mj},a,R,s_{mj})=1$，任 $a\in A'\backslash\{\pi\}$，$1\leqslant j\leqslant M_P$；$p(s_{mj},\pi,c_j,s_{m(j+1)})=1$，$1\leqslant j\leqslant M_P$，式中当 $j=n$ 时 $c_j=\varepsilon\in A'\backslash A$，当 $j=M_P$ 时 $c_j=\pi$，$s_{m(j+1)}=s''_m$，其余情

况 $c_j=R;p(s''_m,a,R,<a,s''_m>)=1$,任 $a\in A'\backslash\{\varepsilon\};p(<*,s''_m>,a,*,s'_m)=1$,任 $*,a\in A';p(s^1_m,a,L,s^2_m)=1$,任 $a\in A';p(s^2_m,a,L,s''_m)=1$,任 $a\in A';p(s''_m,\varepsilon,R,s^3_m)=1;p(s^3_m,a,\pi,s^4_m)=1$,任$a\in A';p(s^4_m,a,L,s_m{}^5)=1$,任$a\in A';p(s^5_m,a,a_i,s_{m+1})=1,a\in A'$,若$m=k$,则$s_{m+1}=\cdot$.

以上模拟中 $s_m,s'_m,s''_m,s_{mj},1\leqslant j\leqslant M_P,s^l_m,1\leqslant l\leqslant 5$ 为 S 中元素.

若 I_m 是 $D_M(n)$,则可仿 $S^{(i)}_M(n)$处理(略).

若 I_m 是 $\overline{J}_M(n)[E_1]$,设 E_1 标识第 h 条指令,不失一般性设 $h\leqslant k$,则有:$p(s_m,a,L,s_m)=1$,任 $a\in A'\backslash\{\$\};p(s_m,\$,R,s'_m)=1;p(s'_m,a,R,s'_m)=1$,任 $a\in A;p(s'_m,\pi,R,s_{m1})=1;p(s_{mi},a,R,s_{mi})=1$,任 $a\in A'\backslash\{\pi\},1\leqslant i\leqslant n-1;p(s_{mi},\pi,R,s_{m(i+1)})=1,1\leqslant i\leqslant n-1;p(s_{mn},a,a,s_h)=1$,任 $a\in A'\backslash\{\pi\};p(s_{mn},\pi,\pi,s_{m+1})=1$,若 $m=k$,则 $s_{m+1}=\cdot$.

以上各 $s_m,s'_m,s_{mj},1\leqslant j\leqslant n$ 均为 S 中元素.

注意到在以上模拟中,若 $PURM(A)$在执行 P 的过程中从 I_m 转移到 I_n,则 Z 就有从状态 s_m 通过一系列动作到状态 s_n 的相应模拟,且其转移概率相等,故对 $r_1,\cdots,r_n,\xi\in A^*$,

$Prob(\Phi^{(n)}_Z(r_1,\cdots,r_n)=\xi)=Prob(P(r_1,\cdots,r_n)=\xi)$. 证毕.

定理 4.2:任给以 A 为带符号集的可计算的 PTM Z,存在 $PURM(A')$程序 P',$A'\supset A$ 有穷,满足对任 $r_1,\cdots,r_n,\xi\in A^*$,$Prob(\Phi^{(n)}_Z(r_1,\cdots,r_n)=\xi)=Prob(P(r_1,\cdots,r_n)=\xi)$.

证明:不失一般性,我们仅对 $n=1$ 的情形证明.

设 Z 的转移函数的值域为$\{0,1/2,1\}$,具有初始状态 $s_0\in S$,S 为状态集. 设 $g:N\times G(A^*)\times N\times S\to N$ 是 $URM-$可计算的 $1-1$ 函数,其中 N 是非负整数集,$G(A^*)$是 $URM-$可计算的 $1-1$ 函数 $G:A^*\to N$ 的值域. 以 $g(m,G(a_1,\cdots,a_k),i,s),m\in N,a_1,a_k\neq 0,a_i\in A,1\leqslant i\leqslant k$ 表示 Z 的带形式为 $0^m_{a_1},\cdots a_k0^\infty$,状态为 s,当前扫视在第 i 个带格上时的格局编码. 记 g^{-1}的第 j 分量为 g^{-1}_j. L 为可计算函数满足:

$$L(\alpha,s)=\begin{cases}g'(1,\alpha_1,s_1,\Lambda,\Lambda,\alpha,s) & \text{若 } p(s,\alpha,\alpha_1,s_1)=1,\\ g'(2,\alpha_1,s_1,\alpha_2,s_2,\alpha,s) & \text{若 } p(s,\alpha,\alpha_1,s_1)=\frac{1}{2},p(s,\alpha,\alpha_2,s_2)=\frac{1}{2}.\end{cases}$$

式中 $\Lambda\notin A\cup S$,g'是可计算的 $1-1$ 函数取值于 N 中.

当 Z 从第 n 步动作进行到第 $n+1$ 步动作时,P'计算相应的带形式及状态等的变化并模拟该步动作. 设 C_n 为第 n 步动作结束时 Z 的格局编码,则 $C_1=g(0,G(r_1),1,s_0)$表示 r_1 为输入,初始状态为 s_0,读头在第一个带格上. P'根据 C_n 和 Z 的动作计算 C_{n+1}如下:

step1:如果 $g^{-1}_4(C_n)=\cdot$,则打印 $G^{-1}(g^{-1}_2(C_n))$于 $PURM(A')$的 R_1 中作为输出并停机,否则做 step2.

step2:计算 $g'^{-1}_1(L(\alpha,s))$,其中 $\alpha=G_{g^{-1}_3(C_n)}(g^{-1}_2(C_n)),s=g^{-1}_4(C_n)$. 如果结果为 1 则转向 step3,否则转向 step4;

step3:计算 $C_{n+1}=Replace(C_n,\alpha_1,s_1)$,其中 $\alpha_1=g'^{-1}_2(L(\alpha,s)),s_1=g'^{-1}_3(L(\alpha,s)),\alpha,s$ 如 step2 中所述. 将 n 加 1 转向 step1;

step4:$TOSS$. 如果 $r_0=a_0$,则 $C_{n+1}=Replace(C_n,\alpha_1,s_1),\alpha_1,s_1$ 如step3中所述;如果 $r_0=a_1$,则 $C_{n+1}=Replace(C_n,\alpha_2,s_2),\alpha_2=g'^{-1}_4(L(\alpha,s)),s_2=g'^{-1}_5(L(\alpha,s)),\alpha,s$ 如 step2 中所述;$n+1$ 转向 step1.

以上是算法的描述. 其中可计算函数 *Replace* 如下给出：

$$Replace(C_n,\alpha_1,s_1)=\begin{cases}g(Q_1,G(G_1^{-1}(Q_2)\cdots G_{\bar{Q}_3^1-1}(Q_2)\alpha_1 G_{\bar{Q}_3^1+1}(Q_2)\cdots G_{PL(C_n)}(Q_2)),Q_3,s_1) & \text{如果 } \alpha_1\in A,\\ g(Q_1,Q_2,Q_3+1,s_1) & \text{如果 } \alpha_1=R,\\ g(Q_1,Q_2,Q_3-1,s_1) & \text{如果 } \alpha_1=L \text{ 且 } Q_3-1\geqslant 0,\\ \text{无定义} & \text{其余情况.}\end{cases}$$

式中$s_1\in S$，$\alpha_1\in V$，$PL(C_n)$表示$G^{-1}(Q_2)$的长度，$G_k^{-1}(Q_2)$表示$G^{-1}(Q_2)$的第k字符，$Q_j=g_j^{-1}(C_n)$，$1\leqslant j\leqslant 3$.

注意到算法可在某 PURM(A′)上实现(因为 URM 是 PURM 的特殊情形)，且对 n 施归纳不难看出该算法是对 Z 的每个动作及其转移概率的完全模拟，故有：$Prob(\Phi_Z(r_1)=\xi)=Prob(P'(r_1)=\xi)$，$r_1,\xi\in A^*$，$P'$是该算法所确定的某 PURM(A′)程序.

§5. 关于随机函数的若干结果

本节中 PTM 指可计算的 PTM，A 指有穷字母表.

定义 5.1：称随机函数 $f:X\to Y$ 是 *PURM*(或 *PTM*)可模拟的，其中 $X,Y\subseteq A^*$，如果存在 *PURM*(*A*)程序 P(或 *PTM* Z)使得对任 $x\in X,y\in Y$，$Prob(P(x)=y)=\mu_f(x,y)$(或 $Prob(\Phi_Z(x)=y)=\mu_f(x,y)$).

定义 5.2：称函数 $f:A^*\to[0,1]$是 *URM*(或 *TM*)－可接受的如果存在 *URM* 程序(或 *Turing* 机)计算 $\delta(\alpha,n)$，$n\geqslant 0,\alpha\in A^*$，其中 $f(\alpha)=\sum_{n=0}^{\infty}\delta(\alpha,n)2^{-n}$. f 是多元函数时可类似定义.

定义 5.3：称函数 $f:A^*\times A^*\to[0,1]$是有限 *URM*(或 *TM*)－可接受的如果 f 是 *URM*(或 *TM*)－可接受的且 $G_x=\{y:f(x,y)>0\}$，$x\in A^*$，有穷，同时存在 *URM* 程序(或 *Turing* 机)使得输入 x 时可在正则顺序下产生 G_x 且 $\sum_{y\in G_x}f(x,y)\leqslant 1$.

定理 5.1：随机函数 $f:A^*\to A^*$ 是 *PURM* 可模拟的充分条件是 $\mu_f:A^*\times A^*\to[0,1]$是有限 *URM*－可接受的.

证明：设 μ_f 是有限 *URM*－可接受的，由定义 5.3，设 *URM* 程序 P' 产生 $G_x=\{y:\mu_f(x,y)>0\}$. 记 $N_x=|G_x|$(集 G_x 中元素的个数). 对任 $x\in A^*$，令$\{\mu_f(x,y):y\in G_x\}=\{\alpha_{x_1},\cdots,\alpha_{x_{N_x}}\}$其中 $\alpha_{x_1}\geqslant\cdots\geqslant\alpha_{x_{N_x}}$，$y_{x_i}$满足 $\alpha_{x_i}=\mu_f(x,y_{x_i})$，$1\leqslant i\leqslant N_x$.

若 $\mu_f(x,y)=1$，某 $x,y\in A^*$，则定理所要求的程序 P 当输入为 x 时直接将 y 输出以模拟 f. 因此不失一般性，设对任 $x,y\in A^*$，$\mu_f(x,y)<1$；对任 $y\in G_x$，令 $\alpha_{x_i}=\sum_{j=1}^{\infty}\varepsilon_{x_{ij}}2^{-j}$，$1\leqslant i\leqslant N_x$. 首先对图 1 所示二叉树给出 *Marking* 算法. 设树的顶点集 $ND=\{N_{01},N_{11},N_{12},N_{22},N_{23},N_{24},\cdots\}$，对任 $x\in A^*$，算法如下：

(1)初始化：令 $Mark(N)=\Lambda$(空)，$N\in ND$；

(2)标记第 1 层顶点 N_{11},N_{12}：若某个 $\alpha_{x_{j_1}}>\frac{1}{2}$(或 $\alpha_{x_{j_1}}=\alpha_{x_{j_2}}=\frac{1}{2}$)，$j_1\neq j_2$，$1\leqslant j_1,j_2\leqslant N_x$，则 $Mark(N_{11})=j_1$，$NMark(1)=\{N_{12}\}$(或 $Mark(N_{11})=j_1$，$Mark(N_{12})=j_2$，$NMark(1)=\varnothing$)；

(3)设第 j 层结点已标记完毕，对 $j+1$ 层结点有以下标记方法：

令 $PM(j+1)=\bigcup_{N_{ji_k}\in NMark(j)}\{N_{(j+1)(2i_k-1)},N_{(j+1)(2i_k)}\}$,又记 $PM(j+1)=\{N_{(j+1)i'_k}:1\leqslant k\leqslant 2\cdot|NMark(j)|\}$,且 $i'_k<i'_{k+1}$,$1\leqslant k\leqslant 2|NMark(j)|$.

做如下循环:

For $i:=1$ to N_x do

begin $q:=\sum_{p=1}^{i}\varepsilon_{x_{p(j+1)}}$;

if $\varepsilon_{x_{i(j+1)}}=1$ then $Mark(N_{(j+1)i'_q})=i$;

end;

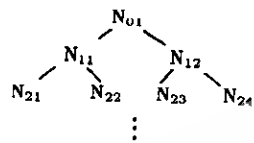

图1

同时,$NMark(j+1)=\{N\in PM(j+1):Mark(N)=\Lambda\}$.

我们断言以上算法不会产生矛盾,即不会出现这样的情形:某层没有足够的NMark中的结点可以被标记.反设若某j层结点满足:进行以上Marking算法时均被标成非空,且存在某 $\varepsilon_{x_{ij}}=1$,使 $i\neq Mark(N_{jk})$,N_{jk}为任意 j 层结点,则有:$\sum_{y\in G_x}\mu_f(x,y)\geqslant\varepsilon_{x_{ij}}2^{-j}+1$ (因第 j 层已标记满)$=1+2^{-j}>1$,矛盾.

容易看出Marking算法可在URM上实现.

下面构造模拟随机函数 f 的算法,由该算法所确定的PURM(A'),$A'\supset A$,程序记为 P.算法如下:

$x\in A^*$ 作为输入由 P' 产生 G_x;$M:=1$;$L:=0$;

$*$:$TOSS$;$M:=2M$;$L:=L+1$;

若 r_0($TOSS$ 的结果)$=1$ 则若 $Mark(N_{LM})\neq\Lambda$ 则打印 $y_{x_{Mak(N_{LM})}}$ 作为输出并停机;

若 $r_0=0$ 且 $Mark(N_{L(M-1)})\neq\Lambda$,则打印 $y_{x_{Mark(N_{L(M-1)})}}$ 作为输出并停机;

$goto$ $*$.

根据以上算法,有:任 $y\in A^*$,若 $y\notin G_x$ 则 $\mu_f(x,y)=0$,由 P 的构造知 $Prob(P(x)=y)=0$;若 $y\in G_x$,不妨设 $y=y_{x_i}$,$1\leqslant i\leqslant N_x$,由 P 的构造知:

$Prob(P(x)=y)=\sum_j Prob(P$ 的执行过程中存在 p,使结点 N_{pj} 满足 $Mark(N_{pj})=i$ 且 P 输出 $y_{x_i})=\sum_{j:\varepsilon_{x_{ij}}=1}2^{-j}=\sum_{j=1}^{\infty}\varepsilon_{x_{ij}}2^{-j}=\mu_f(x,y_{x_i})=\mu_f(x,y)$.

故而对任 $x,y\in A^*$,$Prob(P(x)=y)=\mu_f(x,y)$,定理得证.证毕.

由定理4.1,定理4.2易知,

推论5.1:随机函数 $f:A^*\to A^*$ 是 PTM 可模拟的充分条件是 $\mu_f:A^*\times A^*\to[0,1]$ 是有限 TM—可接受的.

作为定理5.1的特殊情形,以下推论成立:

推论5.1':若 $M:A^*\to[0,1]$ 是 URM—可接受的,则存在 $PURM(A')$ 程序 P,$A'\supset A$,满足 $Prob(P(x)=a_0)=M(x)$,$Prob(P(x)=a_1)=1-M(x)$,$x\in A^*$,$a_0,a_1\in A$.

对 PTM 也有类似于推论5.1'的结论.

定义5.4:称 $PURM(A)$ 程序 P(或 PTM Z)是可接受的如果 $Prob(P(x)=y)$(或 $Prob(\Phi_z(x)=y)$)是 URM(或 TM)—可接受的,$x,y\in A^*$.

定义5.5:称 $PURM(A)$ 程序 P(或 PTM Z)是有限可停的(FS 的)如果存在 $M>0$,P(或

Z)对任何输入 $x\in A^*$ 在 M 步计算内停机.

定理 5.2:若 $PURM(A)$程序 P 是 FS 的则 P 是可接受的.

定理 5.2':若 PTM Z 是 FS 的则 Z 是可接受的.

定义 5.6:称 PTM Z 是稳定的如果对任 $\epsilon>0$ 存在 $\delta>0$ 满足:任 PTM Z',Z' 与 Z 具有相同带符号集 A,如果 $\sum_{\substack{s,s'\in S\\ u\in U,v\in V}}(p(s,u,v,s')-p'(s,u,v,s'))^2<\delta$,式中 p,p' 分别是 Z,Z' 的转移函数,且 s_0 为 Z,Z' 的初始状态,则有 $|Prob(\Phi_Z(x)=y)-Prob(\Phi_{Z'}(x)=y)|<\epsilon, x,y\in A^*$.

定理 5.3:FS 的 PTM 是稳定的.

§6. 结束语

本文中我们定义了 PURM 机器,并证明了其程序与可计算的 PTM 之间的等价性. 我们还部分地回答了这样的问题:一个随机函数可被 PTM 或 PURM 程序模拟的充要条件是什么? 我们把这一问题作为公开问题提出,它的完全解决依赖于对 PTM 或 PURM 的进一步的认识.

作者感谢 E. —E. Doberkat 教授,徐书润副教授和王永革同志.

参考文献

[1]E.-E. Doberkat, Stochastic Automata: Stability, Nondeterminism and Prediction, Lecture Notes in Computer Science 113, Springer-Verlag, Berlin, 1981.

[2]J. T. Gill, Computational Complexity of Probabilistic Turing Machines, SIAM. J. Comput. 6, 1977, 675-695.

[3]M. O. Rabin Probabilistic Automata, Information. and Control 6, 1963, 230-245.

[4]E. S. Santos, Probabilistic Turing Machines and Computability Proc. Amer. Math. Soc. 22, 1969, 704-710.

[5]E. S. Santos, Computability by Probabilistic Turing Machines, Trans. Amer. Math. Soc. 4, 1971, 165-184.

[6]J. C. Shepherdson and H. E. Sturgis, Computability of Recursive Functions, J. Assoc. Comput. Mach. 10, 1963, 217-255.

[7]D. J. A. Welsh, Randomized Algorithms, Discrete Appl. Math. 5, 1983, 133-146.