一种新的组织 P 系统变体的研究*)

徐 贤 董笑菊
(上海交通大学计算机科学与技术系 上海 200240)

摘 要 本文介绍了一种新的组织 P 系统的变体。定义的 P 系统改进了原始的设计,允许在通道(连接,生物上称为突触)上的规则应用中采用并行机制,以提高系统的运行效率。文中我们首先给出这种组织 P 系统的定义,然后描述它的运行机制,接着对它的计算能力做了一些简单的分析,并且用一个典型的例子说明了我们的 P 系统的运行过程的特点。

关键词 组织,P 系统,并行性

On a New Variant of Tissue P Systems

XU Xian DONG Xiao-Ju
(Department of Computer Science and Technology, Shanghai Jiao Tong University, Shanghai 200240)

Abstract In this paper, we introduce a new variant of tissue P systems, one class of P systems. The P systems we define improve the original design of the systems. They permit parallelism in rules application on channels (links, or synapses in biology), by which the running efficiency of the systems can be enhanced. We give the formal definition of our tissue P systems, describe their execution mechanism, and then give an initial analysis of their computational power. Also we illustrate the running of our P systems by a typical example.

Keywords Tissue, P system, Parallelism

1 概述

细胞型 P 系统(cell-like P systems)是一种分布式计算装置,其中最典型的一种是迁移 P 系统(transition P systems)[7~9]。其想法来源于细胞的组织结构和行为,主要的特征有:树形的嵌套膜结构,膜内具有反应规则,膜之间可以传递对象等。细胞型 P 系统(包括它的许多变体)在规则应用中使用最大并行性原则(maximal parallelism),这使它具有图灵机的计算能力。

组织 P 系统(tissue P systems)与细胞型 P 系统不同,它的思想来源于构成各类组织的细胞间的相互交互,更强调细胞间的合作,而不是细胞内部的信息处理。组织 P 系统由许多由通道连接的细胞构成,其结构构成一个图(而不是细胞型 P 系统中的树结构)。组织 P 系统的主要特征有:使用的主要规则类型是 symport 和 antiport;每个细胞(cell)含有状态,使得其内的对象具有更加丰富的行为;通道具有状态,以控制细胞间的通信;在通道上,规则以顺序方式运行,即每一步只能运用一条规则;在整个系统级别上,规则以并行方式作用;计算能力:两个细胞、足够的状态和小权值的 antiport 规则[4],就能够获得图灵机的计算能力;一个细胞,不管使用多少状态和怎样的规则,都只能得到由不带出现检测(appearance checking)的矩阵文法(matrix grammars)产生的语言的 Parikh 像。对组织 P 系统的更多相关解释可以参考文[5,6]。

我们将并行机制运用到细胞间通道上的规则,即由原来的串行作用变为并行作用,从而使得不仅在细胞级别上规则以并行方式运用。它与细胞型 P 系统中的最大并行性(maximal parallelism)并不完全相同,这将在后面的定义中看到。我们做这一扩展的原因可以从下面两方面说明。从生物学角度来看,各种不同类型的反应可以在符合条件时同时作用,而不是串行作用。在细胞间的通信中,只要通信的细胞间的特定条件(如温度)得到满足,就可以运用;这样,就可能有几种不同的规则同时在细胞间的通道上起作用,而决不是一条一条地作用。例如在许多信号传导的过程中,在信号触发时,往往不是只有一种信号分子,而是有若干种,它们同时作用到目标细胞上,形成不同的信号传导路径,产生促进或抑制的作用。

从计算的理论角度来看,并行性在各类科学计算领域的计算中,例如物理学、生物信息学等都是非常重要的要素。它在某种意义上比串行的计算方式更有优势,例如它可以提供更加简洁、直观的计算方式而不失有效性。目前,甚至已有学者提出所谓交互式计算,它可能拥有比图灵机更强的计算能力。同时,还有许多关于并行计算的技术目前已经应用到许多领域(如各类网络计算)。

2 通道上采用并行规则的组织 P 系统

在通道上使用并行规则的基本想法是将系统的每一步运行分成两步。以一个通道来说,第一步收集该通道上所有可能作用的规则,它们应具有互相独立且可以同时作用的特征(即某种并行性质),否则它们只能串行作用。第二步即简单地运行依据上述方法收集到的该通道上的规则。

在原先的带通道状态的组织 P 系统中,所有通道的状态都来自同一个状态集 K。但我们认为现实中可能并非如此,

*)国家杰出青年科学基金 NNSFC(60225012)、BDCC(03DZ14025)、中国国家自然科学基金(60473006)、MSRA、博士点基金(20010248033)资助。徐 贤 博士;董笑菊 博士。

因为在真正的生物系统中，不同的通道上的状态集很可能是互不相同的。所以，这里我们做一个简单的一般化，为每一个从细胞 i 到细胞 j 的通道设置一个与之相对应的状态集 K_{ij}。

2.1 定义

这里我们给出 P 系统的形式化定义。

定义 1 在通道上使用并行规则的组织 P 系统具有如下的结构：

$$\Pi=(O,\ T,\ Syn,\ (\omega_i)_{1\leqslant i\leqslant m},\ E,\ (K_{ij})_{(i,j)\in Syn},\ (s_{ij})_{(i,j)\in Syn},\ (R_{ij})_{(i,j)\in Syn},\ i_0)$$

其中，m 是细胞的数量，称为系统的度(degree)；

O 是有限的对象(object)集，即字母表；

$T\subseteq O$ 是终结(terminal)对象集；

Syn 是通道(channel，或者 link)，来源于突触(synapses)。$Syn\subseteq\{(i,j)\mid i,j=0,1,2,\ldots,m\}$(0 代表环境)，它是细胞集上一个非自反、非对称的关系；

$(\omega_i)\in O^*$ 是系统各细胞中的初始对象多重集(multi-set)，以下简称对象集；

$E\subseteq O$ 是环境中的对象多重集；

K_{ij} 是通道(i,j)上的有限状态集；

s_{ij} 是通道(i,j)的初始状态；

R_{ij} 是通道(i,j)上的有限规则集，规则具有下面的形式：

$(s_1,\ u/v,\ s_2),\ s_1,s_2\in K_{ij},\ u,v\in O^*$

这样的规则的意义是：在状态 s_1，通道(i,j)可以做一个 anti-port 动作。具体地，将对象集 u 从细胞 i 传输到细胞 j，作为交换将对象集 v 从细胞 j 传输到细胞 i，同时将状态转换至 s_2。

$i_0\in\{1,2,\cdots,m\}$是存放计算结果作为系统输出的细胞。

注解：

• 由上述定义的图是一个有向图，但是通道上使用的 antiport 规则使得对象可以双向移动。

• 从实现的角度来讲，我们可以设置某个相关于某个通道的某种池(pool)(如 $P_{ij}\subseteq R_{ij}$)，用以存放处理中的该通道上的规则集。

• 为方便讨论，我们这里给出与上述 P 系统相关的一些定义，包含以下三个方面：

①细胞的数量：m (同系统定义中)

②状态的数量：$k\triangleq\max_{(i,j)\in Syn}|K_{ij}|$

③规则的大小：$r\triangleq\max_{(i,j)\in Syn}\{|u|+|v|\mid(s_1,\ u/v,\ s_2)\in R_{ij}\}$

2.2 P 系统的运行机制

这里我们以算法的方式给出定义 1 中的 P 系统的一步运行机制。以通道(i,j)为例，其它的通道上的规则具有同样的运行机制。假设通道(i,j)当前在状态 s_1 中：

(1)首先，选择合适的可同时运行的规则集。

(a) 选择所有在通道上的规则$(s_1,\ u/v,\ s)$ $(s\in K_{ij})$以备处理；

(b)以状态 s 对规则进行分类，即以下面的等价，关系对规则进行划分：两条规则等价当且仅当它们的目标状态 s 相同。

这样我们得到若干个等价类；

(c) 选择最大的那个等价类，即包含最多规则的那个等价类。如果存在多个这样的等价类，则随机地选取其中的一个。设选中的等价类是 RC(rules class)，同时我们用$|RC|$表示其中的规则数。这里体现了在通道上使用并行性的具体意义，同时体现了不确定性。

(2) 然后的步骤：执行相应的规则集。

(a) 去除 RC 中那些不能执行的规则，例如那些所需要的对象不存在的规则；

(b) 如果 RC 中的规则没有冲突(我们将在下面解释)，我们就已经完成了执行前的工作。将 RC 重命名为 RC'，然后转到(d)；

(c) 如果有冲突存在，例如典型地(事实上这是唯一可能的冲突情形)，RC 中的若干条规则竞争一些 i 或者 j 中的对象，在这种情况下，执行其中的一条规则将使得另一条(或一些)规则无法运用。这里我们需要对 RC 做进一步的划分，使用下面的等价关系：

两个规则等价，当且仅当它们竞争 i 或者 j 中的相同的对象。

经过这样的划分之后我们得到的是一系列相对于$(i,j$ 中的)某些对象的等价类，并且在各个等价类之间没有之前的竞争冲突。[1)]

这样所要做的就是从各个等价类中选择一个代表元，作为冲突中胜出的规则。这样，就得到了一个新的规则集，它的大小与冲突等价类的数目相等。将它记为 RC'。

(d) 同时执行 RC' 中的各条规则，并且将通道(i,j)的状态设为其内的规则指定的(共同的)目标状态。

这样我们就完成了 P 系统的一步执行的过程的算法描述。

注解：在一步执行过程中，任何可以运用的规则都应被应用，即系统只要能运行就必须运行。反之，如果没有规则可用，则在该通道上不做任何事情，状态保持不变。

2.3 P 系统的计算过程

定义 1 中的 P 系统，计算过程可以描述如下：

(1)先是初始配置。系统有 m 个细胞，其中细胞 i 具有的对象多重集是 ω_i；

(2)然后开始离散式的计算过程(我们假设存在一个系统时钟)。在每一步中，运用上面描述的算法以更新系统细胞中的对象集；

(3)停机状态。当在任何通道上没有任何规则可用时，系统停止；

(4)结果。系统的运行结果将被置于细胞 i_0。系统的结果可以有两种类型：

(a) 细胞 i_0 中表示对象数量的向量，并且只取 T 中的对象。我们用 $\mathcal{P}_v(\Pi_{m,k,r})$(或 $\mathcal{P}_v(\Pi)$，当参数不重要时)表示这样的结果；

(b)更直接地，取细胞 i_0 中 T 或者 O 中的对象的多重集。我们用 $\mathcal{P}_s(\Pi_{m,k,r})$ (或 $\mathcal{P}_s(\Pi)$，当参数不重要时)表示这样的结果。

3 一个例子

这一节用一个例子来说明我们的 P 系统(定义 1)的运行。

[1)] 事实上，可以设计一个冲突检测算法来得到各个等价类。从单个对象开始，逐步合并相对于对象多重集的冲突规则集，得到各个实质性的冲突等价类，这是可行的，因为对象的字母表以及细胞内的对象集都是有限的。这里不展开具体细节。

例 1

$\Pi_1 = (O, T, Syn, \omega_1, \omega_2, \omega_3, E, (K_{ij})_{(i,j)\in Syn}, (s_{ij})_{(i,j)\in Syn}, (R_{ij})_{(i,j)\in Syn}, (P_{ij})_{(i,j)\in Syn}, i_0)$

- $m = 3$
- $O = \{a,b\}$
- $T = \{a,b\}$
- $Syn = \{s_1, s_2, s_3\}$，其中

s_1：(0,1)，s_2：(1,2)，s_3：(1,3)

- $\omega_1 = \omega_2 = \omega_3 = \lambda$
- 对于 E，我们作一个简化，假设将系统的输入预先置于 E 中
- $K_{01} = \{s, s'\}$，$K_{12} = \{t\}$，$K_{13} = \{p\}$
- $s_{01} = s$，$s_{12} = t$，$s_{13} = p$
- $R_{01} = \{r_{01}^1, r_{01}^2, r_{01}^3, r_{01}^4\}$，$R_{12} = \{r_{12}^1, r_{12}^2, r_{12}^3, r_{12}^4\}$，$R_{13} = \{r_{13}^1\}$，其中

r_{01}^1：$(s, a/\lambda, s)$	r_{12}^1：$(t, a/\lambda, t)$
r_{01}^2：$(s, b/\lambda, s)$	r_{12}^2：$(t, b/\lambda, t)$
r_{01}^3：$(s, a/\lambda, s')$	r_{12}^3：$(t, \lambda/a, t)$
r_{01}^4：$(s, b/\lambda, s')$	r_{12}^4：$(t, \lambda/b, t)$
r_{13}^1：$(p, ab/\lambda, p)$	

- $i_0 = 3$

图 1 给出了该例子的示意。在该图中，圆表示细胞；双重圆代表输出细胞；每个细胞内的对象（多重集）表示初始的情形；箭头表示通道，旁边标出了其上的规则。

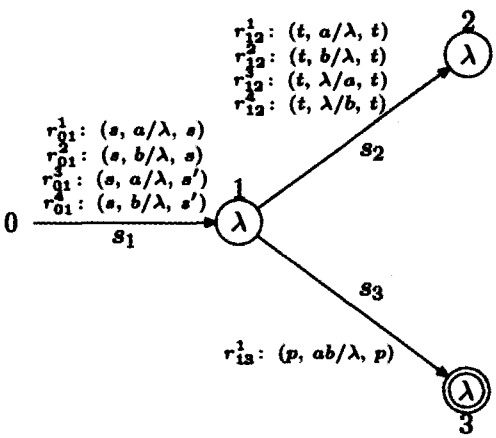

图 1　系统 Π_1（包括细胞、通道、规则和初始配置）

现在我们来看一下该系统的运行。首先，计算每个通道上的规则等价类。通道 $s_1((0,1))$ 的规则类是

$\{\{r_{01}^1, r_{01}^2\}, \{r_{01}^3, r_{01}^4\}\}$

通道 $s_2((1,2))$ 的规则类是

$\{\{r_{12}^1, r_{12}^2, r_{12}^3, r_{12}^4\}\}$

通道 $s_3((1,3))$ 的规则类是

$\{\{r_{13}^1\}\}$

然后我们计算 RC 和 RC'。

- 通道 $s_1((0,1))$：

—在状态 s，RC 可以是 $\{r_{01}^1, r_{01}^2\}$，RC' 则就是 RC；

—在状态 s'，RC 可以是 $\{r_{01}^3, r_{01}^4\}$，RC' 就是 RC。

- 通道 $s_2((1,2))$：

—在状态 t，RC 可以是 $\{r_{12}^1, r_{12}^2, r_{12}^3, r_{12}^4\}$，这样 RC' 就是 RC 的一个子集。例如当计算最终会中止时，RC' 可以是 $\{r_{12}^1, r_{12}^2\}$，$\{r_{12}^3, r_{12}^4\}$，RC 本身或者 $\emptyset$（空集）。而当计算最终不会中止时，RC' 可以是 $\{r_{12}^1, r_{12}^3\}$ 或 $\{r_{12}^2, r_{12}^4\}$。

- 通道 $s_3((1,3))$。这里只有一条规则，即只要当细胞 1 中有 a,b 时该规则应用。

整个系统的运行从各个通道的初始态、各个细胞的初始配置开始，不断地用上面的分析方法进行规则的选取和运用（基于前面的算法），最后得到计算结果。总的来说，我们能观察到：

(1)在状态 s，细胞 1 通过与环境的通道传入一定数量的 a 和 b。这一过程在状态由 s 变为 s' 时停止，此时细胞 1 中有 n 个 a 和 m 个 b。

(2)当细胞 1 中 a 和 b 的数量相等时，计算可以有一条成功的路径，将细胞 1 中的 a 和 b 以成对的 ab 的形式用通道 $s_3((1,3))$ 上的 r_{13}^1 规则传输到细胞 3 中。这将使得系统最终停止运行。

(3)当在细胞 1 中 a 和 b 的数量不同时，则当成对的 ab 被转运到细胞 3 中后，多余的 a 或 b 将使系统永不停止地运行下去（在细胞 1 和 2 之间），即循环地运用通道 $s_2((1,2))$ 上的规则：$\{r_{12}^1, r_{12}^3\}$（若有多余的 a）或 $\{r_{12}^2, r_{12}^4\}$（若有多余的 b）。值得注意的是，在上一种成功的计算中，与这里类似的通道 $s_2((1,2))$ 上的规则运用也是存在的，但这并不影响计算结果。

容易看出，上述的系统 Π_1 计算（接收）的是两个分量相等的自然数二维向量，即

$\mathcal{PS}_v(\Pi_1) = \{(n, n) \mid n \geq 1\}$

注意：至少有一个 a 被传入细胞 1。

该例子在一定程度上说明了在通道上使用并行方式的规则应用的有效性。当然，这里只是一个简单的例子，更多的例子可以进一步说明并行机制的用处，例如建模各种生物过程或者进行科学计算等。

4　计算能力

由于通道上引入了并行性，我们认为这样的 P 系统（定义 1）的计算能力显然应该不会比使用串行方式的系统[5]弱，因为我们认为后者所能计算的当然可以由 P 系统计算。这只需要通过某种技术手段将并行性弱化为串行性就可以实现。因为某种意义上，串行可视为并行的某种特殊情况。我们把它作为一个相对较小的问题加入到今后的工作计划中，这里暂不给出细节。

在我们的组织 P 系统中，除了在通道上使用 antiport 型的通信规则，还可以允许在细胞内部使用对象重写规则。当然这里并没有进一步考虑。这样的系统将能够描述更加复杂且有意义的过程（如生物中的基因调控）。但是设计中必须考虑到两种规则同时作用时可能存在的一些非简单的情况，即我们需要足够细致，以避免两类规则同时运作时产生不合要求的组合；或者两种规则应当能合作以完成特定的目标。

讨论　考察之后可以发现，我们在第 2 节中的算法可以做一些改进。在某些情况下，我们选择的 RC 可能最终使我们得到的 RC' 并非最佳的选择。也就是说，从尽量大地并行化的角度来讲，在选择 RC 时我们可以做得更好一些。至少在某些情况下，由于我们单以等价类的大小作为选择的标准，

（下转第 46 页）

结束语 无线传感器网络中，减少射频工作时间是降低节点能耗的有效方法。本文在分析现有同步机制缺点的基础上，为节点的休眠节能提供了一种高效简单的同步机制，通过集中发送、交错休眠和跨层路由，实现了同步机制与 MAC 和路由协议的紧密结合，减少了节点的同步及路由开销，增加了节点的休眠时间，有效降低了节点传送数据到基站的时延，提高了节点的吞吐量。

参 考 文 献

1 Akyildiz I F, Su W, Sankarasubramaniam Y, et al. Wireless Sensor Networks: a Aurvey. Computer Networks, 2002, 38:393～422
2 Raghunathan V, Schurgers C, Park S, et al. Energy-Aware Wireless Microsensor Networks. IEEE Signal Processing Magazine, Mar. 2002. 40～50
3 Ganesan D, Cerpa A, Ye W, et al. Networking Issues in Wireless Sensor Networks. Journal of Parallel and Distributed Computing, 2004, 64:799～814
4 Ye W, Heidemann J, Estrin D. Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks[R]. California: USC Information Sciences Institute, 2003
5 Li Y, Ye W, Heidemann J. Energy and Latency Control in Low Duty Cycle MAC Protocols[R]. California: USC Information Sciences Institute, 2004
6 孙利民，李建中，陈渝，等. 无线传感器网络. 北京：清华大学出版社，2005. 67～70
7 Lu G, Krishnamachari B, Raghavendra C. An Adaptive Energy-Efficient and Low-latency MAC for Data Gathering in Wireless Sensor Networks. In: Proceedings of the 18th International Parallel and Distributed Processing Symposium. San Francisco: IEEE Computer Society, 2004. 224～230
8 Woo A, Culler D. A Transmission Control Scheme for Media Access in Sensor Networks. In: Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking. San Francisco: IEEE Computer Society, 2001. 221～235

(上接第 18 页)

可能最终得到的 RC' 并不能体现这一要求。但是，这并非说明原算法有大的不妥，因为在一些情况下这样做仍然是合理的。例如，从时间复杂度上来讲，这可以减少每一步的系统运行时间。如果系统要求是实时的，那么这就比较重要了。

事实上，要做一些针对上面的讨论的改进并非十分困难。我们做一简单的描述如下：

在步骤 1 的第(b)步之后，

(c) 对在此时得到的每个规则等价类(用 RC_i 表示)应用步骤 2 第(a)、(b)、(c)步，得到对应的 RC'_i；

(d) 选择 RC'_i 中最大的那个，作为 RC'；

(e) 同时执行 RC' 中的规则。

我们能够看到这样的改进增加了系统运行的复杂性，主要是在计算 RC' 时。具体的复杂性细节需要进一步的分析，但我们相信不会有大的增长。

另一方面，有人或许会说，在通道上使用并行性是不必要的，因为通道上同时可以运用的规则(例如我们的算法中的 RC')实际上可以归为一条规则，这用简单的 antiport 规则的并就可以做到，例如下面的两条同一通道上的规则就可以做合并：

$(s_1, u_1/v_1, s_2), (s_1, u_2/v_2, s_2)$

可以由下一条规则代替

$(s_1, u_1u_2/v_1v_2, s_2)$

但是不幸的是，这一方法并非总是可行的，一般的并不正确。也就是说，这样处理得到的系统和原系统在功能上并不是等价的。而且，即使有时该方法可行，选择不合并要比合并在某些时候更加合理。并使系统的可读性增加(某种意义上更加模块化)。所以，我们认为上述的合并的观点并不十分合理(一般是做不到的)。例如在我们上面的例子中，系统 Π_1 就不能应用这样一种处理。

实际上，在合并可并行运用的规则这一方法可行的情况下，我们的算法隐含地提供了一种实现这一合并的方法。当然，这还不完整。但是，如果必要，这可以用于使系统最小化(相对于规则数目)，不过这可能会牺牲系统的可读性。

6 未来工作展望

我们这里的工作还是很初步的，更多的工作可以在此基础上展开，无论是在计算角度还是在应用角度。

·在细胞型的 P 系统中，有一种变体是引入动态可变的膜[1,2]。有鉴于此，是否可以考虑在我们的系统中引入动态可变的通道(连接)，即通道的拓朴结构在系统运行中可以动态变化(在文[3]中有一些类似讨论)。这或许可以使系统具有一些更加方便的计算特性，从而使它的应用更加广泛。

·更进一步，在上面的扩展基础上，再引入动态可变的膜结构(产生或消融细胞)(可以参考文[3]，以获取更多关于此扩展的描述)。我们这里的系统的相关要素(如通道上和/或细胞内的状态等)为这一扩展提供了一些初步的工作。当然，这一想法离目前的工作还比较远。但可以相信的一点是，这样的 P 系统某种意义上会更接近实际的组织中的细胞的行为。

·我们认为需要进一步给出一些更加精细和实际的例子来说明系统的有效性和表达能力。尽管我们之前的例子具有一定的典型意义，但在实际应用中它还是不够的。例如，在用我们的系统来描述信号通路时，需要描述一个更大、更复杂的交互网络(经过必要的分析和抽象)。这一方面需要系统机制上的支持，另一方面也可以反映出我们的系统在计算上与现实的生物过程的近似程度。

·在本文中，只给出了关于我们的 P 系统的计算能力的一些初步的讨论，但它们是间接的。我们需要给出一个对系统的计算能力的一个更加正式和直接的证明。我们认为，由于我们在系统的通道上使用了并行性，在一些计算能力的讨论上将可以得到更简洁的证明。当然，我们设计的实例也可以作为说明系统特点的一部分。

参 考 文 献

1 Alhazov A, Freund R, Păun Gh. P systems with active membranes and two polarizations. In:[10]. 20～36
2 Alhazov A, Ishdorj T. Membrane Operations in P Systems with Active Membranes. In: Second Brainstorming Week on Membrane Computing, Sevilla, Spain, February 2004. 37～44
3 Bernardini F, Gheorghe M. Population P systems. Journal of Universal Computer Science, 2004, 10(5):509～539
4 Freund R, Oswald M. A Short Note on Analysing P Systems with Antiport Rules. Bulletin of the EATCS, 2002, 78: 231～236
5 Freund R, Păun Gh, Pérez-Jiménez M J. Tissue-like P systems with channel states. In: Proceedings of the Second Brainstorming Week on Membrane Computing. Păun Gh, Riscos A, Romero A, et al. eds. Report RGNC 01/04, University of Seville, 2004. 206～223. and Theoretical Computer Science,, 2005, 330:101～116
6 Martín-vide C, Păun Gh, Pazos J, et al. Tissue P Systems: [TUCS Technical Report]. No. 421. Turku Centre for Computer Science, Sept 2001. Theoretical Computer Science, 2003, 296(2):295～326
7 Păun Gh. Computing with membranes: [TUCS Research Report]. No 208. Turku Centre for Computer Science, 1998. Journal of Computer and System Sciences, 2000, 61(1):108～143
8 Păun Gh. From Cells to Computers: Computing with Membranes (P Systems). BioSystems, 2001, 59(3):139～158
9 Păun Gh. Membrane Computing. An Introduction. Berlin: Springer, 2002
10 Păun Gh, Riscos-Núñez A, Romero-Jiménez, A, et al. Proceedings of the Second Brainstorming Week on Membrane Computing, Sevilla, February 2004. [Technical Report]. 01/04 of Research Group on Natural Computing, Sevilla University, Spain, 2004