Computing and Informatics, Vol. 27, 2008, 403–422

PEER-TO-PEER NETWORKS: A LANGUAGE THEORETIC APPROACH

Katalin Lázár

Department of Information Systems
Eötvös Loránd University
Pázmány Péter sétány 1/C
1117 Budapest, Hungary
e-mail: lazarka@inf.elte.hu

Erzsébet Csuhaj-Varjú

Computer and Automation Research Institute
Hungarian Academy of Sciences
Kende u. 13-17
1111 Budapest, Hungary
e-mail: csuhaj@sztaki.hu

András Lőrincz

Department of Information Systems
Eötvös Loránd University
Pázmány Péter sétány 1/C
1117 Budapest, Hungary
e-mail: lorincz@inf.elte.hu

Manuscript received 25 May 2006; revised 10 May 2007
Communicated by Ladislav Hluchý

Abstract. In this article a modification of a grammar systems theoretic construction, the so-called network of parallel language processors, is proposed to describe the behaviour of peer-to-peer (P2P) systems. In our model, the language processors form teams, send and receive information through collective and individual filters.

The paper deals with the dynamics of string collections. The connection between the growth function of a developmental system and the growth function of networks of parallel multiset string processors with teams of collective and individual filtering is also established.

Keywords: P2P networking, apprentice peers, networks of parallel multiset string processors with teams, collective and individual filtering, population dynamics

1 INTRODUCTION

Nowadays the Internet is witnessing a revolution hailed as peer-to-peer networking. For a detailed introduction to P2P computing, see [7, 15, 23].

In this paper we are going to study the peer-to-peer communication in a formal language theoretic framework, called networks of parallel multiset string processors with teams of collective and individual filtering.

The base of our model is the so-called network of parallel language processors (see e.g. [4, 5, 6, 10]), which consists of several language identifying devices or components associated with the nodes of a virtual graph. The language processors operate on strings, or more precisely, on sets or multisets of strings, by performing rewriting and communication steps alternatively. A system of network of parallel language processors functions by changing its states. At any step, the state of the network is described by the sets or multisets of strings present at the components. During the rewriting step, some strings present at some component are rewritten according to the rewriting rule set and rewriting mode of the component. During the communication step, some strings, or copies of some strings present at some component and satisfying some context condition are communicated to other components via input and output filters. Networks of language processors are intent on capturing some aspects of multi-agent systems: the language processors correspond to the agents and the string collections to the behaviour of the system.

In our model a peer is represented by a multiset string processor, situated at a given node of the network. Like peers in a P2P system, multiset string processors also possess identical functionalities. They form teams, the active components of which can correspond to peer groups. Both multiset string processors and teams have unique IDs. Each multiset string processor may belong to different teams simultaneously, as peers do in a P2P network. They operate on multisets of strings corresponding to advertisements or messages, by performing rewriting or communication steps alternatively. The rewriting step may either be the publication or the update of an advertisement, or the compilation or the modification of a message. The choice of parallel rewriting rules is motivated by the fact that the entire advertisement or message can be modified at a given time step. The lifetime of an advertisement or a message might be expressed by a disjoint alphabet and added to the string representing it, but it has no impact on the mathematical results established in this paper, and consequently, it is omitted. It may also be assumed that the various advertisement types are described by some letters in the strings representing them. The communication step corresponds to the application for or the provision of the advertisements or the dispatch or the receipt of a message in the P2P network. Some advertisements or messages can be provided or accessed only by peers within the given peer group, others by all of the peers. In the first case, the individual, in the second, the collective filtering mechanism is performed.

The formal language theoretic construction introduced in this paper is a simplified abstract model of P2P networks, but it contains the most significant properties of such systems. Clearly, more sophisticated features could be added, which might cast a new light on some additional aspects of P2P networking. The aim of this paper is twofold. First, it gives a formal language theoretic model to describe P2P systems. Secondly, it characterizes the dynamics of information in the network and discusses some related state-of-the-art issues.

2 P2P NETWORK: FORMAL DEFINITIONS

The reader is assumed to be familiar with the basics of formal language theory, for further details consult [10, 24, 25]. For the sake of legibility, only the most important notions used throughout this article are revised.

For an alphabet V, we denote by V^* the free monoid generated by V under the operation of concatenation. The elements of V^* are called strings. λ denotes the empty string, $|x|$ the length, $alph(x)$ the alphabet of $x \in V^*$ and $|x|_{V'}$ the number of occurrences of letters of a subset V' of V in $x \in V^*$. The cardinality of a finite set S is denoted by $\#(S)$, the set of natural numbers by $\mathbb{N}$ and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

Let U denote the set (the universe) of objects. A multiset is a pair $M = (V, f)$, where V is an arbitrary (not necessarily finite) set of objects of U and $f : U \to \mathbb{N}_0$ is a mapping assigning the multiplicity to each object such that if $a \notin V$ then $f(a) = 0$. The support of $M = (V, f)$ is defined by $supp(M) = \{a \in V | f(a) \geq 1\}$. M is a finite multiset in case $supp(M)$ is finite. The set of all finite multisets over the set V is denoted by V°. The number of objects in a finite multiset $M = (V, f)$, or in other words, the cardinality of M is defined by $card(M) = \sum_{a \in V} f(a)$. For instance, a multiset with elements a, a, a, b, b, c is denoted by $\{\{a, a, a, b, b, c\}\}$. It is claimed that $a \in M = (V, f)$, if $a \in supp(M)$, and $M_1 = (V_1, f_1) \subseteq M_2 = (V_2, f_2)$, if $supp(M_1) \subseteq supp(M_2)$ and for all $a \in V_1, f_1(a) \leq f_2(a)$. The union of two multisets is defined by $(M_1 \cup M_2) = (V_1 \cup V_2, f')$, where for all $a \in V_1 \cup V_2, f'(a) = f_1(a) + f_2(a)$, the difference by $(M_1 - M_2) = (V_1, f'')$, where $f''(a) = f_1(a) - f_2(a)$ for all $a \in V_1$, and the intersection by $(M_1 \cap M_2) = (V_1 \cap V_2, f''')$, where for all $a \in V_1 \cap V_2, f'''(a) = min(f_1(a), f_2(a))$ and $min(x, y)$ is the minimum of $x, y \in \mathbb{N}$. M is an empty multiset, denoted by ϵ, provided that $supp(M) = \emptyset$. A multiset M over the finite set of objects V can be represented as a string ω over the alphabet V with $|w|_a = f(a)$, $a \in V$ and λ representing the empty multiset ϵ. In the sequel, the finite multiset of objects with the word ω over V representing M is identified by $M = (V, f)$, hence $\omega \in V^\circ$ is written.

A $0L$ system (an *interactionless Lindenmayer system*) is a triplet $G = (V, \omega, P)$, where V is an alphabet, $\omega \in V^* \setminus \{\lambda\}$ is the axiom and P is a finite set of context-free rewriting rules over V such that for each $a \in V$ there is a rule $a \to x$ in P (we say that P is *complete*). For $z_1, z_2 \in V^*$ we write $z_1 \Longrightarrow z_2$ (with respect to G, if it is necessary, and we denote it by $\Longrightarrow_G$), if $z_1 = a_1 a_2 \dots a_r$, $z_2 = x_1 x_2 \dots x_r$, for $a_i \to x_i$ in P, $1 \leq i \leq r$.

If for each $a \in V$ there is exactly one production of the form $a \to x, x \in V^*$, then we speak of a deterministic $0L$, or a $D0L$ system. If the axiom is replaced by a finite language, then we have an $0L$ ($D0L$) system with a finite number of axioms, or in other words, a $F0L$ ($FD0L$) system.

The families of languages generated by $0L, D0L$, $F0L$ and $FD0L$ systems are denoted by *0L*, *D0L*, *F0L* and *FD0L*, respectively.

Since the production set P of a $D0L$ system $G = (V, \omega, P)$, defines a homomorphism $h : V \to V^*$, thus $G = (V, \omega, h)$ is often used instead of the first notation.

By a word sequence of a $D0L$ system $G = (V, \omega, h)$, the following sequence is meant: $h^0(\omega) = \omega, h(\omega), h^2(\omega), h^3(\omega), \dots$. The function $f : \mathbb{N}_0 \to \mathbb{N}_0$ defined by $f(t) = |h^t(\omega)|, t \geq 0$ is called the growth function of G, and the sequence $|h^t(\omega)|$ for $t = 0, 1, 2, \dots$ is said to be its growth sequence.

By a context condition ϱ over V^*, where V is an alphabet, we mean a computable mapping $\varrho : V^* \to \{true, false\}$. We say that ϱ is of type

reg, or it is a regular context condition over V^*, given by a regular language $L \subseteq V^*$, if $\varrho(\omega) = true$ for any $\omega \in V^*$, where $\omega \in L$, otherwise $\varrho(\omega) = false$.

rc, or it is of random context condition over V^*, given by a pair (Q, R), where $Q, R \subseteq V$, if $\varrho(\omega) = true$ for any $\omega \in V^*$ which contains each element of Q, but no element of R and $\varrho(\omega) = false$ otherwise. By definition, Q and R can be empty sets, in this case we omit the corresponding context check. Q is called the permitting and R the forbidding context condition.

In the sequel, we introduce the notion of network of parallel multiset string processors with teams of collective and individual filtering and define the way in which such a system works.

Definition 1. A network of parallel multiset string processors with teams of collective and individual filtering (a $T_{\mathrm{ci}}NMP_{\mathrm{F}0L}$ system) of degree n, $n \geq 1$, is a construct

$$\Gamma = (V, (t_1, \Theta_1, \Xi_1), \dots, (t_n, \Theta_n, \Xi_n)),$$

where

- V is an alphabet (the alphabet of the system),
- $t_i = \{c_{i,1}, \dots, c_{i,r_i}\}, 1 \leq i \leq n, r_i \geq 1$, is a team component, the i^{th} team,

- $\Theta_i = \{\theta_{i1}, \ldots, \theta_{ip_i}\}$, $\Xi_i = \{\xi_{i1}, \ldots, \xi_{iq_i}\}$, $1 \leq i \leq n, p_i, q_i \geq 1$, where θ_{ij}, ξ_{ik}, $1 \leq j \leq p_i, 1 \leq k \leq q_i$, are context conditions over V^*, called an exit filter and an entrance filter, respectively, recommended by the i^{th} team,
- $c_{i,j} = (P_{i,j}, F_{i,j}, \Psi_{i,j}, \Upsilon_{i,j}), 1 \leq i \leq n, 1 \leq j \leq r_i$, called the j^{th} component of the i^{th} team of the network, or in other words, the $(i,j)^{\text{th}}$ component of the network, where
  - $P_{i,j}$ is a finite set of $0L$ rules over V, the production set of the $(i,j)^{\text{th}}$ component, $1 \leq i \leq n, 1 \leq j \leq r_i$,
  - $F_{i,j} \in V^\circ$ is a non-empty finite multiset of strings, the multiset of axioms of the $(i,j)^{\text{th}}$ component, $1 \leq i \leq n, 1 \leq j \leq r_i$, and
  - $\Psi_{i,j} = \{\psi_{i,j_1}, \ldots, \psi_{i,j_{s_{i,j}}}\}$, $\Upsilon_{i,j} = \{\upsilon_{i,j_1}, \ldots, \upsilon_{i,j_{o_{i,j}}}\}$, $1 \leq i \leq n, 1 \leq j \leq r_i$, where $\psi_{i,j_k}, \upsilon_{i,j_l}$, $1 \leq k \leq s_{i,j}, 1 \leq l \leq o_{i,j}$, $s_{i,j}, o_{i,j} \geq 1$ are context conditions over V^*, called an exit filter and an entrance filter, respectively, recommended by the $(i,j)^{\text{th}}$ component.

A component or a multiset string processor corresponds to a peer in a P2P system, while a team contains the possible candidates that may join a peer group. Observe that the components residing in the nodes of the virtual graph are not assumed to be fully connected, as in the case of wireless networks. Consequently, the static neighbourhood relation applied in [5] is left out.

An element of $F_{i,j} \in V^\circ$, $1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1$, may either correspond to an advertisement or a message. Whether the underlying string is an advertisement or a message might be expressed by a disjoint alphabet, but it has no impact on the mathematical results established in the paper; thus it is omitted. Using the terminology of networks of language processors, the component, in fact, is a multiset string processor. The choice of a multiset string processor is motivated by the fact that in P2P networks multiple instances of an advertisement or a message may exist on the members of a peer group, and each receiver of an advertisement or a message takes away its own copy.

In case of an advertisement, the filters θ_{ij}, ξ_{ik}, $1 \leq i \leq n$, $1 \leq j \leq p_i$, $1 \leq k \leq q_i$, $p_i, q_i \geq 1$, limit access to the advertisement which may be available to every multiset string processor (collective filtering of information), whilst the filters $\psi_{i,j_k}, \upsilon_{i,j_l}$, $1 \leq i \leq n$, $1 \leq j \leq r_i$, $1 \leq k \leq s_{i,j}$, $1 \leq l \leq o_{i,j}$, $r_i, s_{i,j}, o_{i,j} \geq 1$, to those that are available only to the components of the given team (individual filtering of information).

In case of a message, the filters θ_{ij}, ξ_{ik}, $1 \leq i \leq n$, $1 \leq j \leq p_i$, $1 \leq k \leq q_i$, $p_i, q_i \geq 1$, are the pipe endpoints referred to as the output pipe (the sending end) and as the input pipe (the receiving end) at the collective information filtering level, whereas the filters $\psi_{i,j_k}, \upsilon_{i,j_l}$, $1 \leq i \leq n$, $1 \leq j \leq r_i$, $1 \leq k \leq s_{i,j}$, $1 \leq l \leq o_{i,j}$, $r_i, s_{i,j}, o_{i,j} \geq 1$, are the pipe endpoints referred to as the output pipe (the sending end) and as the input pipe (the receiving end) at the individual information filtering level, respectively.

According to the type of the filters and the type of the productions sets we distinguish different classes of $T_{\mathrm{ci}}NMP$ systems. We denote by $T_{\mathrm{c_X i_Y}}NMP_{\mathrm{Z}}$ the class of $T_{\mathrm{ci}}NMP$ systems with (X)-type collective and (Y)-type individual filter, where $X, Y \in \{reg, rc\}$ and $Z \in \{0L, D0L, F0L, \dots\}$.

The $T_{\mathrm{ci}}NMP_{\mathrm{F0L}}$ system functions by changing its states.

Definition 2. By a state (or a configuration) of a $T_{ci}NMP_{F0L}$ system $\Gamma = (V, (t_1, \Theta_1, \Xi_1), \dots, (t_n, \Theta_n, \Xi_n))$, $n \geq 1$, we mean a tuple $s = (M_{1,1}, \dots, M_{1,r_1}, \dots, M_{n,1}, \dots, M_{n,r_n})$, where $M_{i,j} \in V^{\circ}$, $1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1$, is called the state of the $(i,j)^{\mathrm{th}}$ component and represents the multiset of strings which are present at component (i,j) at that moment. $s_0 = (F_{1,1}, \dots, F_{1,r_1}, \dots, F_{n,1}, \dots, F_{n,r_n})$ is said to be the initial state of the system.

Definition 3 (Configuration transmission)**.** Let $\Gamma = (V, (t_1, \Theta_1, \Xi_1), \dots, (t_n, \Theta_n, \Xi_n))$, $n \geq 1$ be a $T_{ci}NMP_{F0L}$ system. Let $s_1 = (M_{1,1}, \dots, M_{1,r_1}, \dots, M_{n,1}, \dots, M_{n,r_n})$, $s_2 = (M'_{1,1}, \dots, M'_{1,r_1}, \dots, M'_{n,1}, \dots, M'_{n,r_n})$ be two states of Γ. We say that

1. s_2 is derived from s_1 by a rewriting step in Γ, written as

$$(M_{1,1}, \dots, M_{1,r_1}, \dots, M_{n,1}, \dots, M_{n,r_n}) \Rightarrow (M'_{1,1}, \dots, M'_{1,r_1}, \dots, M'_{n,1}, \dots, M'_{n,r_n}),$$

   if $M_{i,j} = \{\{\alpha_{i,j_1}, \dots, \alpha_{i,j_{g_{i,j}}}\}\}, M'_{i,j} = \{\{\beta_{i,j_1}, \dots, \beta_{i,j_{g_{i,j}}}\}\}$, where $\alpha_{i,j_k}, \beta_{i,j_k} \in V^*$, $\alpha_{i,j_k} \Rightarrow \beta_{i,j_k}$ in $P_{i,j}$, $1 \leq k \leq g_{i,j}, g_{i,j} \geq 1$, $1 \leq i \leq n, 1 \leq j \leq r_i$.

2. s_2 is derived from s_1 by a communication step in Γ, written as

$$(M_{1,1}, \dots, M_{1,r_1}, \dots, M_{n,1}, \dots, M_{n,r_n}) \vdash (M'_{1,1}, \dots, M'_{1,r_1}, \dots, M'_{n,1}, \dots, M'_{n,r_n}),$$

   if for every $1 \leq i \leq n$, $1 \leq j \leq r_i, r_i \geq 1$, one of the following conditions holds:

   (a) $M'_{i,j} = M_{i,j} \cup \{\{\gamma | \gamma \in M_{k,l}, \theta_{kx}(\gamma) = true, \xi_{iy}(\gamma) = true, 1 \leq k \leq n, 1 \leq l \leq r_k, 1 \leq x \leq p_k, 1 \leq y \leq q_i, r_k, p_k, q_i \geq 1, (k,l) \neq (i,j)\}\}$, or
   (b) $M'_{i,j} = M_{i,j} \cup \{\{\gamma | \gamma \in M_{i,k}, \psi_{i,k_u}(\gamma) = true, \upsilon_{i,j_v}(\gamma) = true, 1 \leq k \leq r_i, 1 \leq u \leq s_{i,k}, 1 \leq v \leq o_{i,j}, r_i, s_{i,k}, o_{i,j} \geq 1, j \neq k\}\}$, or
   (c) $M'_{i,j} = M_{i,j} \cup \{\{\gamma | \gamma \in M_{k,l}, \theta_{kx}(\gamma) = true, \xi_{iy}(\gamma) = true, 1 \leq k \leq n, 1 \leq l \leq r_k, 1 \leq x \leq p_k, 1 \leq y \leq q_i, r_k, p_k, q_i \geq 1, (k,l) \neq (i,j)\}\} \cup \{\{\gamma | \gamma \in M_{i,k}, \psi_{i,k_u}(\gamma) = true, \upsilon_{i,j_v}(\gamma) = true, 1 \leq k \leq r_i, 1 \leq u \leq s_{i,k}, 1 \leq v \leq o_{i,j}, r_i, s_{i,k}, o_{i,j} \geq 1, j \neq k\}\}$.

If the underlying string is an advertisement, then Condition 1 of Definition 3 corresponds to the publication or the update of the advertisement. Should the string be a message, then Condition 1 refers to the compilation or the modification of the message. Parallel rewriting rules have been selected to be applied, since the entire advertisement or message can be modified at a given time step. The choice of a multiset is motivated by the fact that initially multiple instances of an advertisement or a message may exist on a peer, and every rewriting step may create some identical strings, which correspond either to advertisements or to messages, the contents of which are the same.

Conditions 2a, 2b, 2c of Definition 3 are called the collective, the individual and the simultaneous collective and individual filtering mechanism, respectively.

In Condition 2, if the string to be communicated is an advertisement, then a component can apply either for an advertisement that may be available to arbitrary member of an arbitrary team (Condition 2a), for an advertisement that may be available only to the members of the team the given component belongs to (Condition 2b), or for both of the previous two types of advertisements (Condition 2c), in case some context conditions are met. Should the string to be communicated be a message, it might be transferred either via the pipe that connects two components belonging to arbitrary teams (Condition 2a) via the pipe that connects two members of the team the given component belongs to (Condition 2b), or via both of the previous two types of pipes (Condition 2c), provided that some context conditions are fulfilled. In case of a message, the satisfiability of the given context condition means that the component which is intent on sending/receiving the message is able to send/receive it.

A sequence of subsequent states determines computation in Γ.

Let $\Gamma = (V, (t_1, \Theta_1, \Xi_1), \dots, (t_n, \Theta_n, \Xi_n))$, $n \geq 1$, be a $T_{\mathrm{ci}}NMP_{\mathrm{F0L}}$ system. By computation C in Γ we mean a sequence of states $s_0, s_1, \dots$, where $s_k \Rightarrow s_{k+1}$, if $k = 2j+1, j \geq 0$, and $s_k \vdash s_{k+1}$, if $k = 2j, j \geq 1$.

2.1 Information Dynamics

In the following we show that by using the previous formalism, the dynamics of information in P2P networks can be characterized in some particular cases.

Definition 4. Let $\Gamma = (V, (t_1, \Theta_1, \Xi_1), \dots, (t_n, \Theta_n, \Xi_n))$, $n \geq 1$, be a$T_{\mathrm{crc}\,\mathrm{irc}}NMP_{\mathrm{FD0L}}$ system and let $(M_{1,1}^{(t)}, \dots, M_{1,r_1}^{(t)}, \dots, M_{n,1}^{(t)}, \dots, M_{n,r_n}^{(t)})$ be the state of Γ at step t during the computation in Γ, where $t \geq 0, r_i \geq 1, 1 \leq i \leq n$. Assume that for all $t = 2k'+1, k' \geq 0$, $M_{i,j}^{(t+1)} \in V^\circ$ is obtained from $M_{i,j}^{(t)} \in V^\circ$, $1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1$, by the application of the same condition $y, y \in \{2a, 2b, 2c\}$ of Definition 3.

1. The function $m(t) : \mathbb{N}_0 \to \mathbb{N}_0$ defined by $m(t) = \sum_{i=1}^{n} \sum_{j=1}^{r_i} |M_{i,j}^{(t)}|$, for $t \geq 0$ is called the growth function of Γ.
2. The function $m_{i,j}(t) : \mathbb{N}_0 \to \mathbb{N}_0$ defined by $m_{i,j}(t) = |M_{i,j}^{(t)}|$, for $t \geq 0$ is called the growth function of Γ at node (i,j), $1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1$.
3. (Communication functions.)

   (a) The function $f^c_{(i,j)(k,l)}(t) : \mathbb{N}_0 \to \mathbb{N}_0$ defined by $f^c_{(i,j)(k,l)}(t) = |\{\{\gamma \in M_{i,j}^{(t)} | \theta_{ix}(\gamma) = true, \xi_{ky}(\gamma) = true, 1 \leq k \leq n, 1 \leq l \leq r_k, 1 \leq x \leq p_i, 1 \leq y \leq q_k, p_i, q_k, r_k \geq 1, (k,l) \neq (i,j)\}\}|$ for $t = 2k', k' \geq 0$, is called the communication function of Γ from node (i,j) to node (k,l) using collective filtering (including the simultaneous collective and individual filtering, as well).

(b) The function $f^i_{(i,j)(i,k)}(t) : \mathbb{N}_0 \to \mathbb{N}_0$ defined by $f^i_{(i,j)(i,k)}(t) = |\{\{\gamma \in M^{(t)}_{i,j} | \psi_{i,j_u}(\gamma) = true, \upsilon_{i,k_v}(\gamma) = true, 1 \leq k \leq r_i, 1 \leq u \leq s_{i,j}, 1 \leq v \leq o_{i,k}, s_{i,j}, o_{i,k}, r_i \geq 1, j \neq k\}\}|$ for $t = 2k', k' \geq 0$, is called the communication function of Γ from node (i,j) to node (i,k) using individual filtering (including the simultaneous collective and individual filtering as well).

The growth function of Γ describes the increase in the number of pieces of information in the network, the growth function of Γ at node (i,j) the increase in the number of pieces of information at node (i,j), and the communication function of Γ from node (i,j) to node (k,l) $((i,k))$ using collective (individual) filtering the increase in the number of pieces of information during the communication between node (i,j) and node (k,l) $((i,k))$ using collective (individual) filtering at a given time step, respectively.

We demonstrate that the change of the rewritten and the communicated string collections using random context filters can be described by development systems.

Theorem 1. Let $\Gamma = (V, (t_1, \Theta_1, \Xi_1), \ldots, (t_n, \Theta_n, \Xi_n))$, $n \geq 1$, be a $T_{\mathrm{crc}\,\mathrm{irc}}NMP_{\mathrm{FD0L}}$ system and let $(M^{(t)}_{1,1}, \ldots, M^{(t)}_{1,r_1}, \ldots, M^{(t)}_{n,1}, \ldots, M^{(t)}_{n,r_n})$ be the state of Γ at step t during the computation in Γ, where $t \geq 0, r_i \geq 1, 1 \leq i \leq n$. Assume that for all $t = 2k'+1, k' \geq 0$, $M^{(t+1)}_{i,j} \in V^\circ$ is obtained from $M^{(t)}_{i,j} \in V^\circ$, $1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1$, by the application of the same condition $y, y \in \{2a, 2b, 2c\}$ of Definition 3. Then a $D0L$ system $H = (\Sigma, \omega, h)$ can be constructed such that

1. $m(t) = f(t)$, where $m(t)$ is the growth function of Γ and $f(t)$ is the growth function of H,
2. $m_{i,j}(t) = |\bar{h}_{i,j}(h^t(\omega))|$ for some erasing homomorphism $\bar{h}_{i,j} : \Sigma \to \Sigma$, where $m_{i,j}(t)$ is the growth function of Γ at node (i,j).
3. (Communication functions.)

   (a) $f^c_{(i,j)(k,l)}(t) = |\bar{h}_{(i,j)(k,l)}(h^t(\omega))|$ for some erasing homomorphism $\bar{h}_{(i,j)(k,l)} : \Sigma \to \Sigma$, where $f^c_{(i,j)(k,l)}$ is the communication function of Γ from node (i,j) to node (k,l), $t = 2k', k' \geq 0, 1 \leq i,k \leq n, 1 \leq j \leq r_i, 1 \leq l \leq r_k, r_i, r_k \geq 1, (k,l) \neq (i,j)$, using collective filtering.

   (b) $f^i_{(i,j)(i,k)}(t) = |\bar{h}_{(i,j)(i,k)}(h^t(\omega))|$ for some erasing homomorphism $\bar{h}_{(i,j)(i,k)} : \Sigma \to \Sigma$, where $f^i_{(i,j)(i,k)}$ is the communication function of Γ from node (i,j) to node (i,k), $t = 2k', k' \geq 0, 1 \leq i \leq n, 1 \leq j,k \leq r_i, r_i \geq 1, j \neq k$, using individual filtering.

Proof.

1. Due to the fact that $D0L$ systems define homomorphism and the number of strings with a fixed minimal alphabet at a node is known, the number of strings

with the same minimal alphabet at this node can be calculated after performing a rewriting step. The application of context conditions check the presence and/or absence of some symbols in the string. Since the (minimal) alphabet of the string is known, it is decidable whether the given string satisfies the underlying context condition, and consequently any multiset of string present at some stage of computation in Γ can be represented by a multiset of symbols identifying the different alphabets in a unique manner.

It can be shown that at any computation step (both the rewriting and the communication step) in Γ, the multiset of these symbols equals the multiset of letters of a word of a $D0L$ system H, which generates only words that represent the states (string collections) of Γ in the above-described manner.

To prove the statement, we construct a $D0L$ system $H = (\Sigma, \omega, h)$.

Let the homomorphisms $h_{i,1}, \ldots, h_{i,r_i}$ be defined by the production set $P_{i,1}, \ldots, P_{i,r_i}$, $1 \leq i \leq n, r_i \geq 1$, of Γ and let the context conditions θ_{ix}, ξ_{iy}, and $\psi_{i,j_u}, \upsilon_{i,j_v}$, $1 \leq i \leq n, 1 \leq j \leq r_i, 1 \leq x \leq p_i, 1 \leq y \leq q_i, 1 \leq u \leq s_{i,j}, 1 \leq v \leq o_{i,j}, r_i, p_i, q_i, s_{i,j}, o_{i,j} \geq 1$ be expressed explicitly by the notations (Q_{ix}, R_{ix}), (Q_{iy}, R_{iy}) and (Q_{i,j_u}, R_{i,j_u}), (Q_{i,j_v}, R_{i,j_v}), respectively, where $Q_{ix}, Q_{iy}, Q_{i,j_u}, Q_{i,j_v}$ are the corresponding sets of permitting and $R_{ix}, R_{iy}, R_{i,j_u}, R_{i,j_v}$, $1 \leq i \leq n, 1 \leq j \leq r_i, 1 \leq x \leq p_i, 1 \leq y \leq q_i, 1 \leq u \leq s_{i,j}, 1 \leq v \leq o_{i,j}, r_i, p_i, q_i, s_{i,j}, o_{i,j} \geq 1$ are the corresponding sets of forbidding symbols, respectively.

Let $\{V_1, \ldots V_{2^m}\}$ be the set of subsets of V, where $m = card(V)$ and let $\Sigma = \{a_{i,j_w}, b_{i,j_w} | 1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1, 1 \leq w \leq 2^m\}$.

For the sake of legibility, instead of defining the homomorphism h of H, the corresponding production set P is presented.

For every $i, j, w, 1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1, 1 \leq w \leq 2^m$, the construction of the rules of P is as follows:

(a) $a_{i,j_w} \to b_{i,j_z}$, if $alph(h_{i,j}(V_w)) = V_z$, where $1 \leq z \leq 2^m$.

(b) $b_{i,j_w} \to a_{i,j_w} a_{d_1,k_{1_w}} \ldots a_{d_{s'},k_{s'_w}} a_{i,e_{1_w}} \ldots a_{i,e_{s_w}}$, $1 \leq d_l \leq n, 1 \leq k_l \leq r_{d_l}, 1 \leq l \leq s'$, $1 \leq e_{l''} \leq r_i, 1 \leq l'' \leq s, r_{d_l}, s', r_i, s \geq 1$, where

  (i) $(d_l, k_l) \neq (d_{l'}, k_{l'})$, if $l \neq l'$ (pairwise different), and $\{(d_1, k_1), \ldots, (d_{s'}, k_{s'})\}$ is the maximal subset of $\{(1,1) \ldots, (1, r_1), \ldots, (n, 1) \ldots, (n, r_n)\} - \{(i,j)\}$ such that for every (d_l, k_l) there are x and y, $1 \leq x \leq p_i, 1 \leq y \leq q_{d_l}, 1 \leq d_l \leq n, 1 \leq k_l \leq r_{d_l}, 1 \leq l \leq s', p_i, q_{d_l}, r_{d_l}, s' \geq 1$, for which it holds that $Q_{ix} \subseteq V_w, R_{ix} \cap V_w = \emptyset$ and $Q_{d_l y} \subseteq V_w, R_{d_l y} \cap V_w = \emptyset$, and components $c_{i,j}, c_{d_l,k_l}$ use collective filtering (including the simultaneous collective and individual filtering, as well), and

  (ii) $e_1, \ldots, e_s, s \geq 1$ are pairwise different numbers, and $\{e_1, \ldots, e_s\}$ is the maximal subset of $\{1, \ldots, r_i\} - \{j\}$ such that for every $e_{l''}$ there are u and v, $1 \leq u \leq s_{i,j}, 1 \leq v \leq o_{i,e_{l''}}, 1 \leq e_{l''} \leq r_i, 1 \leq l'' \leq s, s_{i,j}, o_{i,e_{l''}}, r_i, s \geq 1$, for which it holds that $Q_{i,j_u} \subseteq V_w, R_{i,j_u} \cap V_w = \emptyset$ and $Q_{i,e_{l''_v}} \subseteq V_w, R_{i,e_{l''_v}} \cap$

$V_w = \emptyset$, and components $c_{i,j}, c_{i,e_{l''}}$ use individual filtering (including the simultaneous collective and individual filtering as well).

Should there be no (d_l, k_l), no $e_1, \dots, e_s$, or no (d_l, k_l) and $e_1, \dots, e_s$, $1 \leq d_l \leq n, 1 \leq k_l \leq r_{d_l}, 1 \leq l \leq s', s, s' \geq 1$, with the above properties, then P has production of the form $b_{i,j_w} \to a_{i,j_w} a_{i,e_{1_w}} \dots a_{i,e_{s_w}}$, $b_{i,j_w} \to a_{i,j_w} a_{d_1,k_{1_w}} \dots a_{d_{s'},k_{s'_w}}$, or $b_{i,j_w} \to a_{i,j_w}$, respectively, where $1 \leq d_l \leq n, 1 \leq k_l \leq r_{d_l}, 1 \leq l \leq s', 1 \leq e_{l''} \leq r_i, 1 \leq l'' \leq s, , r_{d_l}, s', r_i, s \geq 1$.

Let $F_{i,j} = \{\{v_{i,j_1}, \dots, v_{i,j_{m_{i,j}}}\}\}$, where $m_{i,j} = card(F_{i,j}), 1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1$. Let

$$g(F_{i,j}) = \begin{cases} g(v_{i,j_1}) \dots g(v_{i,j_{m_{i,j}}}), & \text{if } F_{i,j} \neq \emptyset, \\ \lambda, & \text{otherwise,} \end{cases}$$

where $g(v_{i,j_w}) = a_{i,j_z}$, $alph(v_{i,j_w}) = V_z, 1 \leq z \leq 2^m$.

Let $\omega = g(F_{1,1}) \dots g(F_{1,r_1}) \dots g(F_{n,1}) \dots g(F_{n,r_n})$.

We show that the growth function of H equals the population growth function of Γ. Obviously, symbol $a_{i,j_k}, 1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1$, in ω corresponds to a word in $F_{i,j}$ with alphabet V_k, and vice versa. Consequently, the length of ω equals the number of axioms of Γ.

In fact, productions given by condition 1a describe a rewriting step in Γ: a word of alphabet V_z is derived by means of $P_{i,j}$ from a word of alphabet V_w. The productions are applied in parallel manner, thus all strings are represented after a rewriting step at node (i, j), $1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1$.

Productions given by condition 1b, on the other hand, describe the communication step. If a word of alphabet V_w at node (i, j), $1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1$, can be communicated both to nodes $(d_1, k_1), \dots, (d_{s'}, k_{s'})$, and to nodes $(i, e_1), \dots, (i, e_s)$, $1 \leq e_{l''} \leq r_i, 1 \leq l'' \leq s$, $1 \leq d_l \leq n, 1 \leq k_l \leq r_{d_l}, 1 \leq l \leq s', r_i, s, r_{d_l}, s' \geq 1$, components $c_{i,j}, c_{d_l,k_l}$ use the collective (including the simultaneous collective and individual) and $c_{i,j}, c_{i,e_{l''}}$ the individual (including the simultaneous collective and individual) filtering method, then a new word, or in other words, a copy of the string over V_w will appear at those nodes, otherwise the underlying word will remain at the given node.

Provided that ω_t is the t^{th} member of the $D0L$ sequence of H, the length of ω_t equals the total number of strings present at the nodes at step t during a computation in Γ. Thus $m(t) = f(t)$ holds.

2. By choosing $\bar{h}_{i,j} : \Sigma \to \Sigma$ as $\bar{h}_{i,j}(a_{i,j_w}) = a_{i,j_w}, \bar{h}_{i,j}(b_{i,j_w}) = b_{i,j_w}$, $1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1, 1 \leq w \leq 2^m$ and $\bar{h}_{i,j}(a_{k,l_w}) = \lambda$, $\bar{h}_{i,j}(b_{k,l_w}) = \lambda$, $1 \leq k \leq n, 1 \leq l \leq r_k, r_k \geq 1, (k,l) \neq (i,j), 1 \leq w \leq 2^m$, the result is obtained immediately.

3. By choosing

   (a) $\bar{h}_{(i,j)(k,l)} : \Sigma \to \Sigma$ as $\bar{h}_{(i,j)(k,l)}(a_{k',l'_w}) = \lambda, \bar{h}_{(i,j)(k,l)}(b_{i,j_z}) = a_{k,l_z}$, if $b_{i,j_z} \to a_{i,j_z}\alpha a_{k,l_z}\beta, \alpha, \beta \in \Sigma^*$, and λ otherwise, $1 \leq i, k, k' \leq n, 1 \leq j \leq r_i, 1 \leq l \leq r_k, 1 \leq l' \leq r_{k'}, r_i, r_k, r_{k'} \geq 1, (i,j) \neq (k,l), 1 \leq w, z \leq 2^m$, and

   (b) $\bar{h}_{(i,j)(i,k)} : \Sigma \to \Sigma$ as $\bar{h}_{(i,j)(i,k)}(a_{k',l'_w}) = \lambda, \bar{h}_{(i,j)(i,k)}(b_{i,j_z}) = a_{i,k_z}$, if $b_{i,j_z} \to a_{i,j_z}\alpha a_{i,k_z}\beta, \alpha, \beta \in \Sigma^*$, and λ otherwise, $1 \leq i, k' \leq n, 1 \leq j, k \leq r_i, 1 \leq l' \leq r_{k'}, r_i, r_{k'} \geq 1, k \neq j, 1 \leq w, z \leq 2^m$,

   the statements follow. □

Theorem 1 describes how to construct a communication graph by means of communication functions, since the sequence of communication functions with respect to a given time step defines a sequence of communication graphs.

By the theory of $D0L$ systems (see [24] for details), we obtain the following corollaries:

Corollary 1. Let $\Gamma = (V, (t_1, \Theta_1, \Xi_1), \ldots, (t_n, \Theta_n, \Xi_n))$, $n \geq 1$, be a $T_{\mathrm{crcirc}}NMP_{\mathrm{FD0L}}$ system and let $(M_{1,1}^{(t)}, \ldots, M_{1,r_1}^{(t)}, \ldots, M_{n,1}^{(t)}, \ldots, M_{n,r_n}^{(t)})$ be the state of Γ at step t during the computation in Γ, where $t \geq 0, r_i \geq 1, 1 \leq i \leq n$. Assume that for all $t = 2k' + 1, k' \geq 0$, $M_{i,j}^{(t+1)} \in V^\circ$ is obtained from $M_{i,j}^{(t)} \in V^\circ$, $1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1$, by the application of the same condition $y, y \in \{2a, 2b, 2c\}$ of Definition 3. Then the growth function of a $T_{\mathrm{crcirc}}NMP_{\mathrm{FD0L}}$ is either exponential or polynomially bounded, which is decidable.

Corollary 2. Let $\Gamma = (V, (t_1, \Theta_1, \Xi_1), \ldots, (t_n, \Theta_n, \Xi_n))$, $n \geq 1$, be a $T_{\mathrm{crcirc}}NMP_{\mathrm{FD0L}}$ system and let $(M_{1,1}^{(t)}, \ldots, M_{1,r_1}^{(t)}, \ldots, M_{n,1}^{(t)}, \ldots, M_{n,r_n}^{(t)})$ be the state of Γ at step t during the computation in Γ, where $t \geq 0, r_i \geq 1, 1 \leq i \leq n$. Assume that for all $t = 2k' + 1, k' \geq 0$, $M_{i,j}^{(t+1)} \in V^\circ$ is obtained from $M_{i,j}^{(t)} \in V^\circ$, $1 \leq i \leq n, 1 \leq j \leq r_i, r_i \geq 1$, by the application of the same condition $y, y \in \{2a, 2b, 2c\}$ of Definition 3. Suppose that $H = (\Sigma, \omega, h)$ is the $D0L$ system for which the Conditions 1, 2 and 3 of Theorem 1 hold. Let $\omega = \omega_0, \omega_1, \omega_2, \ldots,$ be the word sequence generated by the $D0L$ system H. Then the sets $\Sigma_i = alph(\omega_i), i \geq 0$ form an almost periodic sequence, i.e., there are numbers $p > 0$ and $q \geq 0$, such that $\Sigma_i = \Sigma_{i+p}$ holds for every $i \geq q$. If a letter $a \in \Sigma$ occurs in some Σ_i, then it occurs also in some Σ_j, with $j \leq \#(\Sigma) - 1$.

According to Corollary 2, it can be claimed that after some time the function of these P2P networks results in the saturation of information.

Corollary 3. Let $\Gamma_z = (V_z, (t_{1_z}, \Theta_{1_z}, \Xi_{1_z}), \ldots, (t_{n_z}, \Theta_{n_z}, \Xi_{n_z}))$, $n \geq 1$, $z = 1, 2$, be a $T_{\mathrm{crcirc}}NMP_{\mathrm{FD0L}}$ system and let $(M_{1,1_z}^{(t)}, \ldots, M_{1,r_{1_z}}^{(t)}, \ldots, M_{n,1_z}^{(t)}, \ldots, M_{n,r_{n_z}}^{(t)})$ be the state of Γ_z at step t during the computation in Γ_z, where $t \geq 0, r_{i_z} \geq 1, 1 \leq i \leq n$. Assume that for all $t = 2k' + 1, k' \geq 0$, $M_{i,j}^{(t+1)} \in V^\circ$ is obtained from $M_{i,j}^{(t)} \in V^\circ$, $1 \leq$

$i \le n, 1 \le j \le r_i, r_i \ge 1$, by the application of the same condition $y, y \in \{2a, 2b, 2c\}$ of Definition 3. Then the sequence and language equivalence problems are decidable for the $D0L$ systems $H_z = (\Sigma_z, \omega_z, h_z), z = 1,2$, constructed for $\Gamma_z, z = 1,2$, and satisfy the conditions 1, 2 and 3 of Theorem 1.

Corollary 3 implies that in practice it is decidable for two P2P networks whether they function in the same manner concerning the dynamics of information.

Using analogous considerations to those of Theorem 1 we obtain the following statement:

Theorem 2. Let $\Gamma = (V, (t_1, \Theta_1, \Xi_1), \ldots, (t_n, \Theta_n, \Xi_n))$, $n \ge 1$ be a $T_{\mathrm{crc}\,\mathrm{irc}}NMP_{\mathrm{FD0L}}$ system and let $(M_{1,1}^{(t)}, \ldots, M_{1,r_1}^{(t)}, \ldots, M_{n,1}^{(t)}, \ldots, M_{n,r_n}^{(t)})$ be the state of Γ at step t during the computation in Γ, where $t \ge 0, r_i \ge 1, 1 \le i \le n$. Then an $0L$ system $H = (\Sigma, \omega, h)$ can be constructed such that for any $t \ge 0$ and ω generated by H at step t, $|\omega|$ is equal to the number of words in Γ in a configuration obtained at $2t^{\mathrm{th}}$ steps of computation, and vice versa.

Proof. The reader can easily verify by the proof of Theorem 1 that the population growth of the network can be described by an $0L$ system, since the proof of condition 1b of Theorem 1 can be modified so that there are more than one applicable rules for the same letter. □

3 DISCUSSION

Peer-to-peer (P2P) networking is a rapidly growing domain of computer science but with only a few theoretical considerations. Thus theoretical foundations are justifiable by all means. To the best of our knowledge, this paper is the first work aiming at extending the language theoretic approach to P2P networking.

The aim of this paper is to adopt a different formal approach on the issues of information dynamics in P2P networks. Demonstration via simulations and experimental evaluation (see, for instance [8, 12, 17, 27]) of the theoretical aspects are outside the scope of this work, though our mathematical results may open up new directions in the development of applications that we review in the next subsections.

3.1 Related Works

In the sequel, we are going to give an overview of the state-of-the-art research in the field of P2P computing, focusing mainly on potential further generalizations of the formal model used for the description of P2P networks.

A relevant concept in P2P networks is self-organization. De Meer and Koppen claim that self-organization includes complexity, feedback, emergence, criticality, heterarchy, stigmergy or perturbation in the context of P2P networks (see [23], Chapter 15), whereas Aberer et al. (see [23], Chapter 10) assert that it resides in the distribution of control, the locality of processing and the emergence of global structures from local interactions.

In our formal language theoretic model self-organization can be viewed as the emergence of the intensive interaction of a multiset string processor with its dynamic environment, i.e. a collection of separated sub-environments, each belonging to an other multiset string processor.

Web service technology provides functionality over the Internet that can be accessed through well-defined interfaces (see [23], Chapter 14). Hillenbrand and Müller propose some design goals that have been adopted to either technology and might prove beneficial in the other. Nonetheless the combination of P2P and web service technology poses some serious issues such as security to be overcome.

Through a well-defined input filter system we can enforce access control and restrict availability to resources to guarantee security and defend the network against undesirable effects in our mathematical model.

It is demonstrated that P2P networks bear a close resemblance to the ubiquitous computing world (see [23], Chapter 27) with regard to ad hoc communication, feasibility of wireless network structure, rapid information flow, collaboration and resource sharing. Thus it can be deduced that the mathematical model delineated in this article surpasses the formalization of P2P networks.

Distributed information retrieval is of prime importance in P2P networks [19]. To this end, we introduce some kind of hierarchy into the network. We maintain the list of faulty and faultless peers. We apply the constraints of the grammar to detect faulty messages. Local satisfaction of these constraints guarantees the global testing of the system. Testing is restricted to testing of a peer and peers engage in communication with the given peer, thus combinatorial tests can be avoided. Unlike [19], we do not assume strong cooperation among all the peers herein.

Kant et al. [16] propose a taxonomy for the classification of P2P technologies. The taxonomy reveals several research issues to be explored, such as friendliness, security and access control. The formal language theoretic construction introduced in our paper is a simplified abstract model of a P2P system, but it can be extended so as to incorporate some security and access control related topics. The concept of friendliness in a hostile environment can be realized through the introduction of apprentice peers aiding the peers to determine with whom it is worth communicating.

In [17] Kant and Iyer study the evolution of P2P communities in terms of the path of the response, reachability, abandonment or retry of a query and the hybrid nature of the P2P architecture. In our model, the path of the reply can be determined through the communication functions. The various communication functions describe the reachablility of the peers and their contribution to the community, i.e. the satisfaction of the demands of other peers, concerning the requested information, either at the collective, individual or both levels. The communication functions guarantee that it is possible to realize load balancing in the model, i.e. the support of intelligent redistribution based on the access frequency and the location of the content [17]. As a consequence, a minority of nodes providing the queriers with information may be prevented from becoming hot-spots [17]. The abandonment or the retry of a query may exhaust the resources of the information provider, which can pose some security requirements in P2P networks to be dealt with. These security considerations can be fulfilled in some ways. The utilization of the apprentice peers is one method not only to overcome the security concerns, but also to avoid combinatorial explosion. The hybrid architecture may be described in our mathematical model provided that we impose some restrictions on the direction of the communication.

P2P systems are vulnerable to attrition attacks including denial-of-service attacks. Giuli et al. [8] elaborate a defensive framework against malicious adversaries in the form of filters. In our work, it is also the filter system that protects the network against perpetrators. The peers are equipped with one-step buffers, called apprentice peers. The apprentice peer maintains the list of the friends, which it dynamically updates at each time step. Owing to the constraints imposed on the grammars, it is not needed to have the votes of a quorum as in [8], through local satisfaction of the constraints of the grammar malicious peers can be detected, and as a result, combinatorial tests can be avoided.

3.2 Main Achievements and Further Considerations

In this paper we have equipped the networks of parallel multiset string processors with teams of collective filtering introduced in [5] with the individual filtering mechanism. The individual filtering mechanism makes it possible for peers to use a hybrid filtering method, since the collective and individual filters can be of different types (for instance, the access to the resources of a P2P system depends on whether the collective or the individual filtering mechanism is employed).

Another generalization of the model described in [5] is that there is no need to apply a static neighbourhood relation, a multiset string processor is allowed to communicate with another in case certain conditions are satisfied, and as a consequence the relationship between two multiset string processors varies dynamically, as in the case of wireless networks. Context conditions are given in the form of filters. Should the communicated string be a message, then the output filter (either at the level of the multiset string processor or at that of the team the given multiset string processor belongs to) can be regarded as the output pipe, and the input filter as the input pipe. In this way, pipes can be realized in the networks of parallel multiset string processors with teams of collective and individual filtering.

Both the output and the input pipe can be only in two states, namely they can be off or on. Should an output pipe of a given multiset string processor be off (which corresponds to the fact that the underlying context condition is not satisfied), then the multiset string processor is not able to send a message. If an input pipe of the multiset string processor is on (which corresponds to the fact that the underlying context condition is satisfied), then it is able to send the previously compiled message with a time limit. Any message may be accepted on behalf of the receiver within that time limit. The satisfiability of the time limit can be guaranteed by the application of context conditions encoding it (the use of a disjoint alphabet to describe the time limit has no influence on the mathematical results demonstrated in this paper, therefore it is omitted).

In theory, a well-defined input filter system can protect the network against overloading and faults stemming from the function of the system. In P2P systems different types of faults can occur. Information dispatched to other peers may be erroneous, lacking, or present in an undesirable amount. An additional problem may arise in case the required information does not arrive in due course. Faults emerging during the function of the network may substantially influence how such a system works. The faults may superpose each other regardless of how high percentage of the nodes of the network function in a correct manner. It is essential to maintain the filters in the appropriate state in order to protect the system against undesirable effects. It can be assumed that each peer has some tool to determine whether the message received from a given peer is defective or not, which should enable the peer to decrease the frequency of the communication with the malicious peer or terminate the process immediately. The question may arise how it can be realized.

One possibility is that a peer sends message to itself, which contains the list of faulty peers or its complement, the list of friends.

Another potential is that the peers are equipped with one-step buffers. For the sake of future generalizations, the peers will be called master peers, whereas the one-step buffers apprentice peers. The apprentice peer maintains the list of the faulty or faultless peers, which it sends to its master peer at each time step. Afterwards, the master peer can control its inputs and outputs subject to the list, rewritten according to the actual information received from other members of the P2P network. The introduction of a buffer does not have an influence on the results of Theorem 1, because it can be regarded as a special multiset string processor, whose rewritings are identical ones, hence the solution is viable. Indeed a general multiset string processor is capable of doing more and this is why the master and the apprentice peer are distinguished. In effect, more than one apprentice peer may belong to a master peer, and as a consequence, a hierarchy of processors might be constructed. These concepts are elaborated below.

The maintenance of the list of peers is motivated by Internet crawler experiments (see e.g. [21]), where the list of good URLs corresponds to the list of friends of the underlying Internet crawler, with whom it is worth 'communicating'. If the crawler happens to receive faulty (insignificant or obsolete) information from an URL then its list of friends is updated. The maintenance of such lists has proven to be very efficient [22] in scale-free small world networks (see e.g. [2] and references therein). Evolving structures seem to generate scale-free small world structure.

There are several approaches in the literature [1, 3, 9, 18, 26, 29] that help peers decide which members of a P2P system send them a faulty message. Collaboration may be facilitated by the selection of trustworthy partners. Obviously, the identification and the exclusion of malicious and egoistic peers, and the rehabilitation of reliable ones, contribute enormously to the efficiency of the network.

Another application of the apprentice peer is that it introduces memory into the system and enables feedback. In neural network parlance [11] let the experienced input at time t be denoted by $x_e(t)$. For input $x_e(t)$ the system produces the output $y(t)$. The output may serve some goals and in general the system may have *expectations* of future inputs. If the expectations are not fulfilled then the system may generate correcting outputs. This type of functioning is enabled by the apprentice peer. The master peer may derive the output as well as its desired next input $x_d(t+1)$ in reply to the input $x_e(t)$ at time t. The output can be sent to other peers, whilst the desired next input $x_d(t+1)$ to the apprentice, which sends it back at the next time instant. Thus, at the next time step, the master will have access both to the next experienced input $x_e(t+1)$ and the desired input $x_d(t+1)$, and it can derive correcting outputs provided that they do not coincide. This type of computation is called first order feedback scheme. Analogously, second order feedback schemes can be constructed if the one-step buffer is extended to a multistep buffer. Apprentices assist peers with their feedbacks so that error correction may be realizable in P2P networks.

The concept of apprentice peers surpasses multistep buffers. It can be easily seen that pipeline systems and pipelined operations fit in with our considerations [20]: the master peer sends the input to the pipeline and the last apprentice of the hierarchy provides the other peers awaiting the result of the computation with the output of the pipeline.

Furthermore, it can be anticipated that the adaptive grammatical model of P2P networks can take a crucial part in the automation of a testing process. It is widely known that nowadays distributed hardware systems possess all characteristics of complex systems and are difficult, sometimes impossible, to be tested. A compiler that is able to check at a higher level the logical consistency of a software as a rule-based system, for instance, as a language theoretic construction, does not exist.

In telecommunications [28] most software units can be viewed as reactive systems that receive stimuli from their environment and respond to them by emitting observable output signals after their internal state has been altered. As a result, it is a natural way to model such systems as finite state machines [10, 13, 14, 25]. Nonetheless rarely do the software units function in a way as they are expected. Unexpected inputs might emerge in lieu, which means that in this respect they behave rather like stochastic machines. An essential future direction of the language theoretic approach described in this paper is that the features of the P2P network may be warranted by the grammar. At present, if the constraints of the grammar are satisfied in the case of each peer then the system does not need to be tested globally. Instead, testing can be restricted to the individual testing of the peers, and as a result combinatorial tests can be avoided.

In conclusion, owing to the language theoretic formulation of P2P systems several advantages can be gained. In particular, the concept of apprentice peer makes the P2P system flexible in certain aspects. In the first place, it can render the P2P network adaptive through the maintenance of lists and make task-based dynamic configuration possible. In the second place, apprentice peers can be used for computing correcting outputs, the tool of short term adaptation. Lastly, the introduction of the apprentice peers enables pipelined operations in P2P networks.

REFERENCES

[1] ADAMS, W. J.—DAVIS IV, N. J.: Decentralized Trust-Based Access Control for Dynamic Coalitions. USMA IA Workshop, June 15–17, 2005.

[2] BARABÁSI, A.-L.: Linked: The New Science of Networks. Perseus Publishing Cambridge, Ma. 2002.

[3] BONATTI, P.—DUMA, C.—OLMEDILLA, D.—SHAHMEHRI, N.: An Integration of Reputation-Based and Policy-Based Trust Management.

[4] CSUHAJ-VARJÚ, E.: Networks of Language Processors. EATCS Bulletin. Vol. 63, 1997, pp. 120–134.

[5] CSUHAJ-VARJÚ, E.: Networks of Language Processors: A Language Theoretic Approach to Filtering and Cooperation. In: L. Kovács (Ed.): ERCIM Workshop Proceedings, Fifth DELOS Workshop on Filtering and Collaborative Filtering, 1997, pp. 91–97.

[6] CSUHAJ–VARJÚ, E.—SALOMAA, A.: Networks of Parallel Language Processors. In: Gh. Păun and A. Salomaa (Eds.): New Trends in Formal Languages, Control, Cooperation and Combinatorics. Lecture Notes in Computer Science, Vol. 1218, 1997, pp. 299–318.

[7] FLENNER, R., ABBOTT, M., BOUBEZ, T., COHEN, F., KRISHNAN, N.—MOFFET, A.—RAMAMURTI, R.—SIDDIQUI, B.—SOMMERS, F.: Java P2P Unleashed. Sams Publishing, 2001.

[8] GIULI, TH. J.—MANIATIS, P.—BAKER, M.—ROSENTHAL, D. S. H.—ROUSSOPOULOS, M.: Attrition Defenses for a Peer-to-Peer Digital Preservation System. USENIX Annual Technical Conference, General Track 2005, pp. 163–178.

[9] Handbook of Applied Cryptography. A. Menezes, P. van Oorschot, S. Vanstone (Eds.), CRC Press, 1996.

[10] Handbook of Formal Languages. J. Dassow, Gh. Păun, G. Rozenberg (Eds.). Springer-Verlag, Berlin-Heidelberg, 1997.

[11] HAYKIN, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, New Jersey, USA, 1999.

[12] HE, Q.—AMMAR, M. H.—RILEY, G. F.—RAJ, H.—FUJIMOTO, R.: Mapping Peer Behavior to Packet-Level Details: A Framework for Packet-Level Simulation of Peer-to-Peer Systems. 11th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS '03), 2003, pp. 71–78.

[13] HOPCROFT, J. E.—ULLMAN, J. D.: Formal Languages and Their Relations to Automata. Addison-Wesley, 1969.

[14] HOPCROFT, J. E., ULLMAN, J. D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley, 1979.

[15] JXTA V2.3.X: JAVA^{TM} PROGRAMMER'S GUIDE, SUN MICROSYSTEMS, 2005. AVAILABLE ON: `http://www.jxta.org/`.

[16] KANT, K.—IYER, R.—TEWARI, V.: A Framework for Classifying Peer-to-Peer Technologies. Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID 2002), pp. 368–375.

[17] Kant, K.—Iyer, R.: A Performance Model for Peer-to-Peer File-Sharing Services. Available on: http://citeseer.ist.psu.edu/kant01performance.html.

[18] Malkhi, D.: Secret Sharing. An Advanced Course in Computer and Network Security. The Hebrew University of Jerusalem, 2002.

[19] Nejdl, W.—Siberski, W.—Thaden, U.—Balke, W.-T.: Top-k Query Evaluation for Schema-Based Peer-to-Peer Networks. In: S. A. McIlraith et al. (Eds.): ISWC 2004, LNCS Vol. 3298, pp. 137-151. Springer-Verlag, Berlin-Heidelberg, 2004.

[20] Palotai, Zs.—Kandár, T.—Mohr, Z.—Visegrády, T.—Ziegler, G.—Arató, P.—Lőrincz, A.: Value Prediction in HLS Allocation Problems Using Intellectual Properties. Applied Artificial Intelligence, Vol. 16, 2002, pp. 151–192.

[21] Palotai, Zs.—Mandusitz, S.—Lőrincz, A.: Distributed Novel News Mining from the Internet with an Evolutionary News Forager Community. Int. Joint Conf. on Neural Networks 2004, July 26–29, 2004, Budapest, Paper No. 1095. IJCNN2004 CD-ROM Conference Proceedings, IEEE Catalog Number: 04CH37541C, ISBN: 0-7803-8360-5, IEEE Operations Center, Piscataway, NJ, 08855-1331.

[22] Palotai, Zs.—Farkas, Cs.—Lőrincz, A.: Is Selection Optimal for Scale-Free Small Worlds? European Conference on Complex Systems 2005, November 14–18, 2005, Paris, ComPlexUs Vol. 3, 2006, pp. 158–168.

[23] P2P Systems and Applications. R. Steinmetz and K. Wehrle (Eds.), LNCS Vol. 3485, 2005, pp. 1–5. Springer-Verlag, Berlin-Heidelberg, 2005.

[24] Rozenberg, G.—Salomaa, A.: Mathematical Theory of L Systems. Academic Press, New York, 1980.

[25] Salomaa, A.: Formal Languages. Academic Press, New York, 1973.

[26] Tanenbaum, A. S.—van Steen, M.: Distributed Systems: Principles and Paradigms. Prentice-Hall, New Jersey, 2002.

[27] Ting, Ny. S.—Deters, R.: 3LS – A Peer-to-Peer Network Simulator. Proceedings of the Third International Conference on Peer-to-Peer Computing (P2P '03) 2003, pp. 212–213.

[28] What is SDL? Available on: http://www.sdl-forum.org/SDL.

[29] Ziegler, G.—Farkas, Cs.—Lőrincz, A.: A Framework for Anonymous but Accountable Self-Organizing Communities. Information and Software Technology, Vol. 48, 2006, No. 8, pp. 726–744.

Katalin Lázár received an M. Sc. degree in computer science from the Eötvös Loránd University, Budapest in 2002. She is a Ph. D. candidate in computer science at the Eötvös Loránd University. She was a visiting researcher at the University of South Carolina, Columbia, USA in 2006. Her current research interests include artificial intelligence, grammar systems theory, information theory and information security. She is the (co-)author of several scientific papers.

Erzsébet Csuhaj-Varjú works with the Computer and Automaton Research Institute of the Hungarian Academy of Sciences as the head of the Theoretical Computer Science Research Group. Her main research interests are formal languages, distributed systems, and nature-motivated computing. In these areas she has more than 130 papers, a monograph, and seven edited volumes published in international publication forums. She gave numerous talks at international conferences and research seminars, she was visiting researcher at several research sites in Europe, in Canada, and Japan. Her results in these fields have received considerable attention, she has more than 600 independent citations. She was supervisor (principal investigator) and participant of several Hungarian and bilateral granted research projects, she was the Hungarian site manager of the Thematic Network "MolCoNet", IST-2001-32008 in molecular computing, workpackage leader of the EC project Centre of Excellence in Information Theory, Computation, and Control, ICA1-CT-2000. She is the Hungarian team leader of the European Molecular Computing Consortium. She is a member of the College of Science, Hungarian Scientific Research Fund, and a founding member and a member of the Doctoral Committee of the Ph. D. School in Informatics at the Loránd Eötvös University in Budapest, and an external member of the Ph. D. School in Mathematics and Computer Science at the Szeged University. In the last three years, she was a member of the Computer Science and Informatics Committee of the Hungarian Academy of Sciences, and a member of the General Assembly of the Hungarian Academy of Sciences as a representative of the D.Sc holders. She is member of the editorial board of International Journal of Foundations of Computer Science and ERCIM NEWS, the quarterly of the European Research Consortium for Informatics and Mathematics. She has been programme committee chair, programme committee member and organizer of several international workshops and conferences.

András Lőrincz has been teaching at the Faculty of Informatics of the Eötvös Loránd University, Budapest since 1998. His research focuses on distributed intelligent systems and their applications in neurobiological and cognitive modelling, as well as medicine. He has founded the Neural Information Processing Group of Eötvös University and he directs a multidisciplinary team of mathematicians, programmers, computer scientists and physicists. He has acted as the PI of several successful international projects in collaboration with Panasonic, Honda Future Technology Research and the Information Directorate of the US Air Force in the fields of hardware-software co-synthesis, image processing and human-computer collaboration. He is a member of two EU FP6 research projects, the NEW TIES: New and Emergent World Models Through Individual, Evolutionary, and Social Learning project and the PERCEPT: Perceptual Consciousness – Explication and Testing project. He graduated in physics at the Eötvös Loránd University in 1975 where he received his Ph. D. in 1978 and his C. Sc. in 1986 in experimental and theoretical solid-state physics and chemical physics, respectively. He habilitated and became a university professor in 1988 at the Szeged University. He conducted research and taught quantum control, photoacoustics and artificial intelligence at the Hungarian Academy of Sciences, University of Chicago, Brown University, Princeton University, the Illinois Institute of Technology and ETH Zurich. He authored about 200 peer reviewed scientific publications on different disciplines with about 600 independent citations. He also has 10 patents. In 1997–1998 he was the scientific director of the Hungarian subsidiary of US-based Associative Computing Ltd. He has received the Széchenyi Professor Award, Master Professor Award and the Széchenyi István Award in 2000, 2001, and 2004, respectively. Four of his students won the prestigious Pro Scientia Gold Medal in the field of information science over the last 6 years. In 2004, he was awarded the Kalmár Prize of the John von Neumann Computer Society of Hungary. He has become an elected Fellow of the European Coordinating Committee for Artificial Intelligence in 2006.