Journal of Universal Computer Science, vol. 1, no. 5 (1995), 252-268
submitted: 8/2/95, accepted: 11/5/95, appeared: 28/5/95, ©Springer Pub. Co.

Conditional Tabled Eco-Grammar Systems versus (E)T0L Systems[1]

Erzsebet CSUHAJ-VARJU
Computer and Automation Institute of the Hungarian Academy of Sciences
Kende u. 13 – 19, 1111 Budapest, Hungary

Gheorghe PĂUN
Institute of Mathematics of the Romanian Academy of Sciences
PO Box 1–764, 70700 Bucureşti, Romania

Arto SALOMAA
Academy of Finland and University of Turku, Department of Mathematics
20500 Turku, Finland

Abstract. We investigate the generative capacity of the so-called conditional tabled eco-grammar systems (CTEG). They are a variant of eco-grammar systems, generative mechanisms recently introduced as models of the interplay between *environment* and *agents* in eco-systems. In particular, we compare the power of CTEG systems with that of programmed and of random context T0L systems and with that of ET0L systems. CTEG systems with one agent only (and without extended symbols) are found to be surprisingly powerful (they can generate non-ET0L languages). Representation theorems for ET0L and for recursively enumerable languages in terms of CTEG languages are also presented.

C.R. CATEGORIES: F4.2 [Mathematical Logic and Formal Languages]: Grammars and other Rewriting Systems: *Grammar types*, F4.3 [Mathematical Logic and Formal Languages]: Formal Languages

KEY WORDS: Grammar systems, L systems, Artificial Life

1. Introduction

According to [8], one of the most important classes of models already developed in Theoretical Computer Science and useful for Artificial Life are the L systems. The same author emphasizes the fact that of particular interest for Artificial Life is the *pattern of life*, the structure of living organisms and systems, the cooperation between parts of such systems. As an attempt to model such a cooperation at the level of an eco-system, starting with the basic relationship between *environment* and *agents* living/acting in/on that environment,

[1] Research supported by the Academy of Finland, Project 11281

the notion of eco-grammar system has been introduced in [3] (presented first in [4]). Basically, it is a variant of grammar systems, integrating features both of cooperating distributed grammar systems and of parallel cooperating grammar systems – see [2] for details about grammar systems theory. In short, several *agents*, described by 0L systems, and an *environment*, also described by a 0L system, interact as follows: The evolution of agents (the rules used at a given step) depends on the string describing the environment. The agents have also associated some pure rewriting rules, by which they act, locally (one rule only is used in every time unit), on the environment. The action depends on the state of the agent in the current time unit. Schematically, one obtains the picture in figure 1.

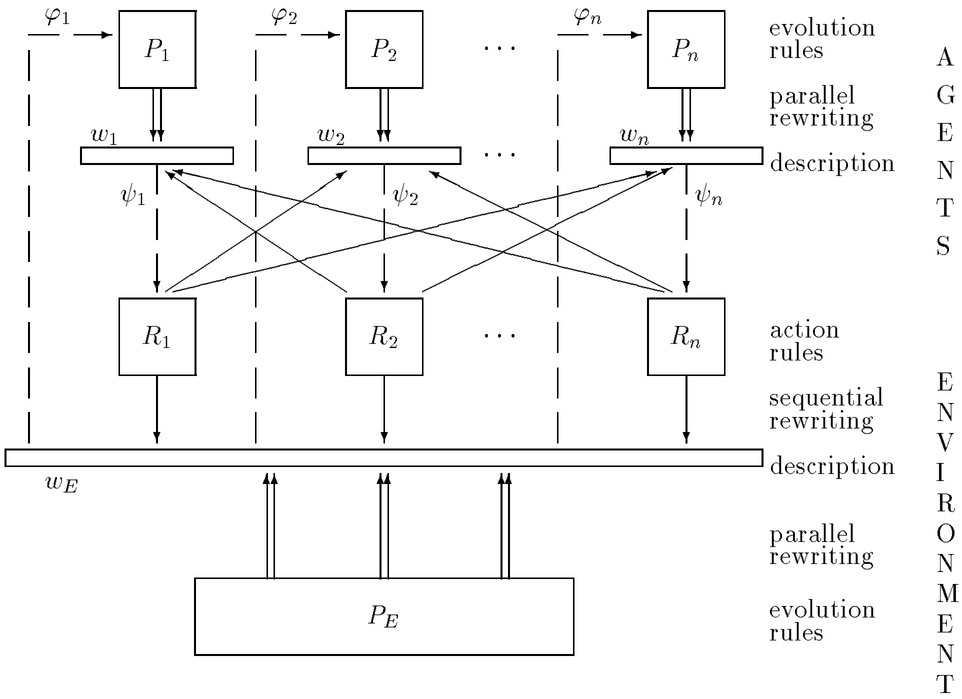

Fig. 1

At every moment, the system is described by a *configuration*, an $(n+1)$-tuple of strings, $(w_E, w_1, \ldots, w_n)$, describing the environment and the agents. Starting from an initial configuration, sequences of configurations will be obtained, describing the evolution of the system. Collecting only the strings w_E, we can associate a language to such a system, the set of environment descriptions. In this way, an eco-grammar system can be viewed as a generative mechanism. Results about the power of such systems can be found in [3], [4], as well as in several contributions to [11]. In one of these contributions to [11], we have considered a variant of eco-grammar systems, both with practical and theoretical motivation, [5].

On the one hand, the model in [3], [4] assumes that the environment evolves independently of the agents. This does not cover such cases as those when the agents are strongly polluting sources, volcanos, damaged nuclear power plants and so on. Such powerful agents act no longer locally. On the other hand, the model in [3], [4] contains the selection mappings which are only assumed to be computable. It is just natural to consider simplified versions, with particular mappings specifying the active evolution rules in any given time unit.

Following these requests, in the model considered in [5] under the name of *conditional tabled eco-grammar systems* (CTEG systems), we have removed the local action of agents on the environment, but we have introduced a dependence of the environment evolution on the states of agents. More precisely, this dependence is given by condition strings – permitting and forbidding: a prescribed string must be present, another one must be not, in order to can apply a given subset of the set of 0L rules (hence a table). We have started the study of the power of such systems in [5], without going into too many details. For instance, we have proved that for certain cases (only permitting condition strings, which must appear as scattered or as permuted scattered subwords of the current strings) CTEG systems with n agents can be simulated by systems with one agent only.

We shall continue here the study of CTEG systems, comparing them with their natural counterparts in L systems theory: regulated T0L systems and ET0L systems. We again find that systems with one agent only are surprisingly powerful (non-ET0L languages can be generated, even with permitting contexts of length one or with forbidding contexts of length two).

2. L systems prerequisites

As usual, we denote by V^* the free monoid generated by an alphabet V; λ is the empty string, $|x|$ is the length of $x \in V^*$, $|x|_a$ is the number of occurrences of the symbol a in the string x, $\Psi_V(x) = (|x|_{a_1}, \ldots, |x|_{a_n})$ is the Parikh vector associated to $x \in V^*$, for $V = \{a_1, \ldots, a_n\}$. The families of finite, regular, context-free, context-sensitive, and recursively enumerable languages are denoted by FIN, REG, CF, CS, RE, respectively. Basic elements of formal language theory which we use here can be found in [13].

A 0L system is a triple $G = (V, w, P)$, where V is an alphabet, $w \in V^*$ 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 necessary, we specify this by $\Longrightarrow_G$) if $z_1 = a_1a_2\ldots a_r$, $z_2 = x_1x_2\ldots x_r$, for $a_i \to x_i \in P, 1 \leq i \leq r$. The language generated by G is $L(G) = \{z \in V^* \mid w \Longrightarrow^* z\}$, where $\Longrightarrow^*$ is the reflexive and transitive closure of $\Longrightarrow$.

An E0L (*extended* 0L) system is a quadruple $G = (V, T, w, P)$, where $G' = (V, w, P)$ is a 0L system and $T \subseteq V$. The language generated by G is defined by $L(G) = L(G') \cap T^*$.

A T0L (*tabled* 0L) system is a construct $G = (V, w, P_1, \ldots, P_n), n \geq 1$, where each $G_i = (V, w, P_i), 1 \leq i \leq n$, is a 0L system. The generated language is

$L(G) = \{z \in V^* \mid w \Longrightarrow_{G_{i_1}} w_1 \Longrightarrow_{G_{i_2}} \ldots \Longrightarrow_{G_{i_m}} w_m = z,\ m \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq m\} \cup \{w\}$.

An ET0L (*extended* T0L) system is a construct $G = (V, T, w, P_1, \ldots, P_n), n \geq 1$, where $G' = (V, w, P_1, \ldots, P_n)$ is a T0L system and $T \subseteq V$. We define $L(G) = L(G') \cap T^*$.

The families of languages generated by 0L, E0L, T0L and ET0L systems are denoted by $0L, E0L, T0L, ET0L$, respectively.

Pairs of the form (V, P) (0L systems without axiom) are called 0L *schemes* and $(n+1)$-tuples $(V, P_1, \ldots, P_n)$ (T0L systems without axiom) are called T0L schemes.

The following relations are known (see, for instance, [10], [12]):

1. $0L \subset T0L \subset ET0L,\ 0L \subset E0L \subset ET0L$,
2. $CF \subset E0L$,
3. $T0L$ is incomparable with each of the families $FIN, REG, CF, E0L$; $0L$ is incomparable with FIN, REG, CF.

Following the model of regulated context-free grammars, regulated T0L systems were considered (we refer to [6] for details). Because the family of languages generated by *matrix* T0L systems is (strictly) included in the family of languages generated by *programmed* T0L systems, which, in turn, is incomparable with the family of languages generated by *random context* T0L systems (Theorem 8.4 in [6]), we present only these latter classes.

A *programmed* T0L system is a construct $G = (V, w, (b_1 : P_1, E_1), \ldots, (b_n : P_n, E_n))$, where $G' = (V, w, P_1, \ldots, P_n)$ is a T0L system, $b_1, \ldots, b_n$ are *labels* associated to tables and $E_i \subseteq Lab, 1 \leq i \leq n$, for $Lab = \{b_1, \ldots, b_n\}$. For $(b_i, x), (b_j, y) \in Lab \times V^*$ we write $(b_i, x) \Longrightarrow (b_j, y)$ if $x \Longrightarrow y$ using the table P_i and $b_j \in E_i$ (after using the table P_i, with the label b_i, we use a table with the label in the set E_i). The language generated by G is $L(G) = \{x \in V^* \mid (b_{i_0}, w) \Longrightarrow (b_{i_1}, w_1) \Longrightarrow \ldots \Longrightarrow (b_{i_m}, w_m) = (b_{i_m}, x),\ m \geq 0, b_{i_0} \in Lab, b_{i_j} \in E_{i_{j-1}}, 1 \leq i \leq m\}$.

The family of such languages is denoted by $(P)T0L$. (The letter P is parenthesized in order to avoid confusion with the family of propagating T0L languages.)

A *random context* T0L system is a construct $G = (V, w, (Q_1 : P_1), \ldots, (Q_n : P_n))$, where $G' = (V, w, P_1, \ldots, P_n)$ is a T0L system and $Q_i \subseteq V, 1 \leq i \leq n$. A table P_i can be applied to a string $x \in V^*$ only when $|x|_a > 0$ for all $a \in Q_i$.

The family of languages generated in this way is denoted by $(RC)T0L$.

Note that in neither case we have appearance checking features, that is forbidding contexts in random context T0L systems and failure fields in programmed T0L systems.

When the above regulating mechanisms are added to ET0L systems we denote by $E(P)T0L, E(RC)T0L$ the corresponding families.

Proofs of the following results can be found in [6] (Theorems 8.3, 8.4):

1. $T0L \subset (P)T0L \subset ET0L,\ T0L \subset (RC)T0L$,
2. $ET0L = E(P)T0L \subset E(RC)T0L$,
3. $(P)T0L$ and $(RC)T0L$ are incomparable and the same is true for $ET0L$ and $(RC)T0L$.

3. The new class of eco-grammar systems

First, a preliminary definition.

Definition 1. A *conditional T0L scheme* with k-ary context conditions, $k \geq 1$, is a construct

$$G = (V, (c_1, d_1 : P_1), \ldots, (c_n, d_n : P_n)),$$

where V is an alphabet, $P_1, \ldots, P_n$ are (complete) tables of 0L rules over V, and $c_i = (c_{i1}, \ldots, c_{ik}), d_i = (d_{i1}, \ldots, d_{ik})$, $1 \leq i \leq n$, with $c_{ij}, d_{ij} \in V^*$ for all i, j.

Informally speaking, c_i is used as a *permitting* condition parameter and d_i as a *forbidding* condition parameter, the table P_i being used only when certain predicate *is true* for c_i and certain predicate *is not true* for d_i. The predicates we consider here will be of the following types:

Definition 2. Given an alphabet V, we define the following three predicates over $V^* \times V^*$:

$$\begin{aligned} P_b(x, y) &= 1 \ iff \ y = y_1 x y_2, \\ P_s(x, y) &= 1 \ iff \ y = y_1 x_1 y_2 x_2 \ldots y_r x_r y_{r+1}, \\ &\qquad\qquad x = x_1 x_2 \ldots x_r, \ x_i, y_i \in V^* \ for \ all \ i, \\ P_p(x, y) &= 1 \ iff \ \Psi_V(x) \leq \Psi_V(y), \ componentwise. \end{aligned}$$

Therefore, $P_b(x, y)$ is true when x is a subword of y (a *block* of it), $P_s(x, y)$ is true when x is a *scattered* subword of y and $P_p(x, y)$ is true when a *permutation* of x is a scattered subword of y.

Using such predicates we can define the derivation relation in a conditional T0L scheme. This corresponds to random context T0L systems, but here the context are given by words, not by symbols.

Definition 3. A *conditional tabled eco-grammar* (CTEG, for short) system of degree $n, n \geq 1$, is a construct

$$\Sigma = (E, A_1, \ldots, A_n),$$

where

(i) $E = (V_E, (c_1, d_1 : P_1), \ldots, (c_m, d_m : P_m))$ is a conditional T0L scheme with n-ary context conditions, $c_i, d_i \in V_1^* \times \ldots \times V_n^*, 1 \leq i \leq m$,
(ii) $A_i = (V_i, (e_{i1}, f_{i1} : P_{i1}), \ldots, (e_{ir_i}, f_{ir_i} : P_{ir_i})), 1 \leq i \leq n$, is a conditional T0L scheme with 1-ary context conditions, $e_{ij}, f_{ij} \in V_E^*$ for all i, j.

The component E corresponds to the environment and $A_i, 1 \leq i \leq n$, correspond to the agents.

Therefore, the evolution of the environment (the table to be used) depends on the states of the n agents, whereas the evolution of each agent depends on the state of the environment, in the sense specified by the context condition strings c_i, d_i (in case of the environment) and e_{ij}, f_{ij} (in case of the agents) via predicates as in Definition 2. We shall specify this in a formal way below. One can see that we have a system with the structure as represented in figure 2.

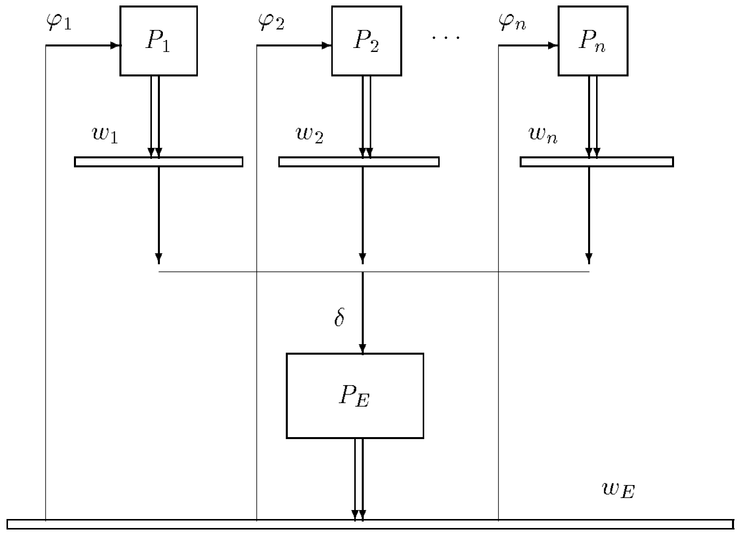

Fig. 2

Definition 4. For a CTEG system as above, a *configuration* is an $(n+1)$-tuple

$$\sigma = (w_E, w_1, \ldots, w_n),$$

with $w_E \in V_E^*, w_i \in V_i^*, 1 \leq i \leq n$.

Definition 5. For a CTEG system Σ as above, $\alpha \in \{b, s, p\}$, and two configurations $\sigma = (w_E, w_1, \ldots, w_n), \sigma' = (w'_E, w'_1, \ldots, w'_n)$, we write $\sigma \Longrightarrow_\alpha \sigma'$ if and only if:

(1) There is a table $(c_j, d_j : P_j)$ in E such that $P_\alpha(c_{ji}, w_i) = 1$, $P_\alpha(d_{ji}, w_i) = 0$, for all $1 \leq i \leq n$, and $w_E \Longrightarrow_{P_j} w'_E$.
(2) Every $A_i, 1 \leq i \leq n$, has a table $(e_{ij}, f_{ij} : P_{ij}), 1 \leq j \leq r_i$, such that $P_\alpha(e_{ij}, w_E) = 1$, $P_\alpha(f_{ij}, w_E) = 0$, and $w_i \Longrightarrow_{P_{ij}} w'_i$.

As usual, we denote by $\Longrightarrow_\alpha^*$ the reflexive and transitive closure of $\Longrightarrow_\alpha$, $\alpha \in \{b, s, p\}$.

The *environmental language* associated to a system Σ working in the mode α, when starting from a configuration σ_0, is defined by

$$L_\alpha(\Sigma, \sigma_0) = \{w_E \mid \sigma_0 \Longrightarrow_\alpha^* \sigma = (w_E, w_1, \ldots, w_n)\}.$$

We denote by $CTEG_n(i, j; \alpha), n \geq 1, i, j \geq 0, \alpha \in \{b, s, p\}$, the family of languages $L_\alpha(\Sigma, \sigma_0)$, where Σ is a system of degree at most n, with permitting contexts of length at most i and with forbidding contexts of length at most j. When the number of agents, the length of permitting or of forbidding contexts is not bounded, then we replace the corresponding parameter with ∞.

If in a CTEG system as above we ignore all permitting contexts (for instance, we replace all of them by λ – note that $P_\alpha(\lambda, y) = 1$ for all y and α), then we speak about *forbidding* CTEG systems. The corresponding family of languages is denoted by $CTEG_n(0, j; \alpha)$. Symmetrically, if we ignore the forbidding contexts (for instance, we add a new symbol, c, to all alphabets and replace all forbidding contexts by c, which never appears in the configuration components), then we speak about *permitting* CTEG systems. The associated family of languages is denoted by $CTEG_n(i, 0; \alpha)$. For brevity, we write $\emptyset$ instead of the ignored permitting or forbidding contexts.

A possible generalization is to consider multiple contexts, several permitting and forbidding string contexts. We shall not investigate such a variant here.

It is easy to see that all the introduced above variants of CTEG systems are eco-grammar systems with regular selection of evolution rules, in the sense of [3], [4]: we say that a mapping $\rho : V^* \longrightarrow 2^P$, where V is an alphabet and P is a set of rewriting rules, is *regular* when $\rho^{-1}(R)$ is a regular set for all $R \subseteq P$. Here we deal with tables P_i, and $\rho^{-1}(P_i) = \{w \in V^* \mid P_\alpha(c, w) = 1, P_\alpha(d, w) = 0\}$ for $(c, d : P_i)$ a table. Clearly, such languages $\rho^{-1}(P_i)$ are regular.

4. Relationships with regulated T0L systems

The following relations directly follow from definitions:

Lemma 1.

(i) *$CTEG_n(i, j; \alpha) \subseteq CTEG_m(i', j'; \alpha)$, for all $\alpha \in \{b, s, p\}$, for $n \leq m$ and $i \leq i', j \leq j'$.*
(ii) $CTEG_n(i, j; b) = CTEG_n(i, j; s) = CTEG_n(i, j; p)$, for $n \geq 1, i, j \in \{0, 1\}$.
(iii) $CTEG_\infty(0, 0; \alpha) = CTEG_1(0, 0; \alpha) = T0L,\ \alpha \in \{b, s, p\}$.

In view of point (ii) above, when $i, j \in \{0, 1\}$ the specification of α is useless, hence we shall write simply $CTEG_n(i, j)$ instead of $CTEG_n(i, j; \alpha)$.

The following results are proved in [5].

Lemma 2. $CTEG_n(\infty, 0; \alpha) = CTEG_1(\infty, 0; \alpha), n \geq 1$, *for* $\alpha \in \{s, p\}$.

(Therefore, in the case of permitting contexts only, checked in the scattered or in the permuted scattered way, the hierarchy induced by the number of agents collapses to one family.)

Lemma 3. *The languages $\{a^2, a^3\}, \{a^2, b^3\}$ do not belong to $CTEG_\infty(\infty, \infty; \alpha)$, $\alpha \in \{b, s, p\}$.*

Combining this with the relation (iii) in Lemma 1, we get the fact that each $CTEG_n(i, j; \alpha)$, for all possible values of i, j, α, is incomparable with each family F such that $FIN \subseteq F \subseteq CF$ (a similar assertion holds true for the family $T0L$). However,

Lemma 4. *For any finite language L and any symbol a, the language $\{a\} \cup L$ belongs to both families $CTEG_1(1, 0), CTEG_1(0, 1)$.*

In order to have an estimation of the size of families $CTEG_n(i, j; \alpha)$ it is just natural to compare them with $(P)T0L$ and $(RC)T0L$.

Theorem 1. $(RC)T0L \subset CTEG_\infty(1, 0)$.

Proof. Consider a random context T0L system, $G = (V, w, (Q_1 : P_1), \dots, (Q_n : P_n))$, with $Q_i = \{a_{i1}, \dots, a_{im_i}\}$, where $a_{ij} \in V$ for all $1 \le j \le m_i$ and $m_i \ge 0, 1 \le i \le n$.

We construct a CTEG system with $M = \sum_{i=1}^{n} m_i$ agents,

$$\Sigma = (E, A_1, \dots, A_M),$$

where

$$E = (V, (c_1, \emptyset : P_1), \dots, (c_n, \emptyset : P_n), (c_{n+1}, \emptyset : P_{n+1}),$$

with the context conditions $c_i = (c_{i1}, \dots, c_{iM}), 1 \le i \le n$, such that

$$c_{ij} = \emptyset,\ \mathit{for}\ 1 \le j \le \sum_{l=1}^{i-1} m_l,$$

$$c_{ij} = a_{ij},\ \mathit{for}\ \sum_{l=1}^{i-1} m_l + 1 \le j \le \sum_{l=1}^{i} m_l,$$

$$c_{ij} = \emptyset,\ \mathit{for}\ \sum_{l=1}^{i} m_l + 1 \le j \le M.$$

(when $m_i = 0$, hence $Q_i = \emptyset$, we have $c_i = (\emptyset, \dots, \emptyset))$, and

$$P_{n+1} = \{a \to a \mid a \in V\},$$
$$c_{n+1} = (d, d, \dots, d), d\ \mathit{occurs}\ M\ \mathit{times},$$

whereas the agents $A_t, 1 \le t \le M$, are defined by

$$A_t = (\{a_{ij}, d, d'\}, (e_{t1}, \emptyset : P_{t1}), (\emptyset, \emptyset : P_{t2}),\ 1 \le t \le M,$$
$$P_{t1} = \{d \to a_{ij},\ a_{ij} \to d',\ d' \to d'\},\ e_{t1} = a_{ij},$$
$$P_{t2} = \{a_{ij} \to d,\ d \to d,\ d' \to d'\},$$

where $1 \le i \le n, 1 \le j \le m_i$ are such that

$$t = \sum_{l=1}^{i-1} m_l + j.$$

For $\sigma_0 = (w, d, d, \dots, d)$, where d appears M times, we have

$$L(G) = L(\Sigma, \sigma_0).$$

Indeed, in the presence of the symbol d in agents descriptions (this is the case also when starting from σ_0), E can use only tables P_i with $Q_i = \emptyset$ and P_{n+1}, which changes nothing, whereas the agents use either the tables P_{t2} (freely) or P_{t1}, providing that the corresponding symbol a_{ij} is present in the string of E. At the next step, E can use either a table P_i with $Q_i = \emptyset$ or a table P_i whose condition Q_i, although non-empty, is fulfilled. This corresponds to the correct application of the table P_i in the sense of the random context T0L system G. At the same time, all agents will use the tables P_{t2}, returning their strings to d, or the tables P_{t1}. If this table introduces the symbol d', it will never be removed, hence the associated table in E will never be applied from that moment. If P_{t1} introduces a symbol a_{ij}, this again corresponds to fulfilling the condition imposed by the presence of a_{ij} in the string of E. The process can be iterated. Clearly, every derivation in G can be simulated in this way in Σ and, conversely, the evolutions of Σ correspond to correct derivations in G generating the string describing the environment. Consequently, we have the announced equality $L(G) = L(\Sigma, \sigma_0)$, which proves the inclusion $(RC)T0L \subseteq CTEG_\infty(1,0)$.

This inclusion is proper. In [6], page 255, it is proved that the language

$$L = \{a^2, a^4, b^4\}$$

is not in $(RC)T0L$. However, $L \in CTEG_1(1,0)$, because $L = L(\Sigma, \sigma_0)$ for $\Sigma = (E, A_1)$, with

$$\begin{aligned} E &= (\{a,b\}, (c, \emptyset : \{a \to a^2,\ b \to b\}),\ (c', \emptyset : \{a \to b,\ b \to b\})), \\ A_1 &= (\{c, c'\}, (\emptyset, \emptyset : \{c \to c',\ c' \to c'\})), \end{aligned}$$

and $\sigma_0 = (a^2, c)$. □

Corollary. $CTEG_\infty(1,0) - ET0L \neq \emptyset$.

Proof. The assertion follows from the relation $(RC)T0L - ET0L \neq \emptyset$ pointed out in Section 2. □

For the case of programmed T0L systems we obtain a still stronger result.

Theorem 2. $(P)T0L \subset CTEG_1(1,0)$.

Proof. Take a programmed T0L system $G = (V, w, (b_1 : P_1, E_1), \ldots, (b_n : P_n, E_n))$, with $Lab = \{b_1, \ldots, b_n\}$, and construct the CTEG system $\Sigma = (E, A_1)$ with

$$\begin{aligned} E &= (V(b_0, \emptyset : \{a \to a \mid a \in V\}),\ (b_1, \emptyset : P_1), \ldots, (b_n, \emptyset : P_n)), \\ A_1 &= (Lab \cup \{b_0, b_0'\}(\emptyset, \emptyset : \{b_0 \to b_i,\ b_i \to b_i \mid 1 \leq i \leq n\} \cup \{b_0' \to b_0'\}), \\ &\qquad (\emptyset, \emptyset : \{b_i \to b_j \mid b_j \in E_i \cup \{b_0'\}, 1 \leq i \leq n\} \cup \\ &\qquad \cup \{b_0 \to b_0',\ b_0' \to b_0'\})). \end{aligned}$$

For $\sigma_0 = (w, b_0)$ we have $L(G) = L(\Sigma, \sigma_0)$.

At the first step, we change nothing in E but we can change b_0 either for some $b_i, 1 \leq i \leq n$ (using the first table of A_1), or for b_0' (using the second table). In the latter case the work of Σ is blocked. This will happen whenever we introduce b_0' and we have to introduce it when $E_i = \emptyset$; this also corresponds to a blocked derivation in G (no continuation is possible). In the presence of b_i as description of A_1 we can apply the table P_i in E, which corresponds to a derivation step in G: the next description of A_1 will be either a label from E_i, if $E_i \neq \emptyset$, or the blocking symbol b_0'. In conclusion, $L(G) = L(\Sigma, \sigma_0)$, that is $(P)T0L \subseteq CTEG_1(1,0)$.

In order to prove the strictness of this inclusion, let us consider the $(RC)T0L$ system

$$\begin{aligned} G &= (\{a, , b, c, d, e, f, g\}, c, (\emptyset : P_1), (\{f\}, P_2)), \\ P_1 &= \{c \to eca,\ c \to fda\}, \\ P_2 &= \{e \to f,\ f \to g,\ a \to ab\} \end{aligned}$$

(the tables contains also all completion rules $q \to q$ for q not appearing in the left-hand member of a rule specified above).

In [6], page 257, it is proved that the language generated by this system is not in $(P)T0L$. According to the proof of Theorem 1, we can construct a CTEG Σ equivalent with G and having only one agent (M in the proof mentioned will be 1). Consequently, $L(G) \in CTEG_1(1,0)$, which completes the proof. □

5. Relationships with ET0L languages

The family $(P)T0L$ is strictly included in $ET0L$. The family $CTEG_1(1,0)$ not only contains strictly the family $(P)T0L$, but contains also non-ET0L languages (which implies again that $(P)T0L \subset CTEG_1(1,0)$ is a proper inclusion and gives a stronger form of the result in the corollary of Theorem 1).

Theorem 3. $CTEG_1(1,0) - ET0L \neq \emptyset$.

Proof. Let us consider the system $\Sigma = (E, A_1)$ with

$$
\begin{aligned}
E = (&\{s, a, b, c, d, e, f, g, g', g''\},\\
&(a, \emptyset : \{s \to bs\}),\\
&(a, \emptyset : \{s \to bf\}),\\
&(b, \emptyset : \{b \to c,\ f \to g\}),\\
&(c, \emptyset : \{c \to ac,\ c \to ad,\ e \to ae,\ g \to g'\}),\\
&(d, \emptyset : \{d \to e,\ g' \to g''\}),\\
&(e, \emptyset : \{g'' \to g\})),
\end{aligned}
$$

and

$$
\begin{aligned}
A_1 = (&\{a, b, c, d, e, f\},\\
&(s, \emptyset : \{a \to a\}),\\
&(s, \emptyset : \{a \to b\}),\\
&(f, \emptyset : \{b \to c\}),\\
&(g, \emptyset : \{c \to d\}),\\
&(d, \emptyset : \{d \to e\}),\\
&(c, \emptyset : \{e \to c,\ c \to f,\ d \to f\})).
\end{aligned}
$$

(The completion rules $q \to q$, for q not specified above, are not given.)

Then

$$L(\Sigma, (s, a)) \cap (a^+e)^+g'' = \{(a^m e)^n g'' \mid n \geq m \geq 1\},$$

which is not an ET0L language (use Theorem 2.1 in [12]).

Let us examine the work of Σ.

Starting from $\sigma_0 = (s, a)$, using the first two tables of E and of A_1, we generate a configuration $(b^n f, b), n \geq 1$. Only one continuation is possible, $(b^n f, b) \Longrightarrow (c^n g, c)$. Now we have to use the fourth table of E, but for A_1 we can use two tables, the fourth and the sixth ones. Using this latter table we replace c by f and no further step can be taken (no table in E can be used, but the string of E contains the symbol g', hence it is not in $(a^+e)^+g''$). Thus we are led to a configuration of the form $((ah)^n g', d)$, where $h \in \{c, d\}$. Only one table of E is applicable (asked for by d in the string of A_1). The sixth table of A_1 blocks again the system by introducing the symbol f. The only other possibility to continue is by using the table $(d, \emptyset : \{d \to e\})$ of A_1, providing the string of E contains at least one occurrence of d. We obtain $((ah')^n g'', e)$, with $h' \in \{c, e\}$. The only possible continuation is by using the last tables of E and of A_1, but this second table requires the presence of at least one c in the string of E. If the continuation is not possible, then we already have a string in $(a^+e)^+g''$. If we can continue, then we get $((ah')^n g, c)$, and such a configuration, with g present in the string of E and c present in the string of A_1 was already discussed above. Thus, in the presence of the couple (g, c) we can iterate the process, at most a number of times equal to the number of occurrences of c in the initial configuration, that is at most m times. (At every step, at least one c is replaced by d, then by e, and only in the presence of c in the string of E we can continue the work of A_1). In conclusion, the string generated in the first component of configurations, when it is of the form $(a^+e)^+g''$, will be of the form $(a^m e)^n g''$, with $m \leq n$, which completes the proof. □

For the case of CTEG with only forbidding conditions we can obtain a weaker result: conditions of length two are necessary. This, of course, makes relevant the mode of checking the conditions.

Theorem 4. $CTEG_1(0, 2; \alpha) - ET0L \neq \emptyset,\ \alpha \in \{s, p\}$.

Proof. Consider the CTEG system $\Sigma = (E, A_1)$, with

$$\begin{aligned}
E = (&\{s, a, b, c, d, e, f\},\\
&(\emptyset, b : \{s \to bs\}),\\
&(\emptyset, b : \{s \to bf\}),\\
&(\emptyset, b : \{b \to c\}),\\
&(\emptyset, d : \{c \to ac,\ c \to ad,\ e \to ae\}),\\
&(\emptyset, \emptyset : \{d \to e\})),\\
A_1 = (&\{a, b, c, d\},\\
&(\emptyset, f : \{a \to a\}),\\
&(\emptyset, s : \{a \to bc\}),\\
&(\emptyset, dd : \{c \to d\}),\\
&(\emptyset, dd : \{d \to c\})).
\end{aligned}$$

(Each table also contains all completion rules $q \to q$, for symbols q not specified above.)

Then, for $\alpha \in \{s, p\}$, we have

$$L_\alpha(\Sigma, (s, a)) \cap (a^+e)^+f = \{(a^m e)^n f \mid m \geq n \geq 1\},$$

which is not an ET0L language (see [12], Exercise 2.3, page 260, referring to [7]). As the family $ET0L$ is closed under intersection with regular sets, it follows that $L_\alpha(\Sigma, (s, a)) \notin ET0L$.

Let us examine the work of Σ, starting from $\sigma = (s, a)$. Because of s, A_1 can only use the first table; E can use several times its first table, hence we get $(b^n s, a)$. Eventually, E will use the second table. Assume hence that we have obtained $(b^n f, a)$, for some $n \geq 1$. Now A_1 must use the second table, whereas the only applicable table of E is also the second one. We obtain $(b^n f, a) \Longrightarrow_\alpha (c^n f, bc)$. From now on, b remains present in the description of the agent, hence the only applicable tables of E are the last two; d is not present, hence we have to perform a step $(c^n f, bc) \Longrightarrow_\alpha ((ag)^n f, bd)$, where $g \in \{c, d\}$. Because d is present in the description of the agent, the fourth table of E is not applicable. The last table of E replaces each occurrence of d (if any) by e and leaves all other symbols unchanged. From A_1 we can use one of the last two tables, but only when at most one occurrence of d is present in $(ag)^n f$. If we use $(\emptyset, dd : \{c \to d\})$, then nothing is changed, hence either we use $(\emptyset, \emptyset : \{d \to e\})$ and $(\emptyset, dd : \{d \to c\})$, or we use $(\emptyset, \emptyset : \{d \to e\})$ and $(\emptyset, dd : \{c \to d\})$ for a number of times (only at the first use changing the configuration) and eventually we use the first mentioned pairs of tables. Consequently, we eventually get a configuration of the form $((ah)^n f, bc)$, with $h \in \{c, e\}$, and at most one h is equal to e. The process can continue, at every step the number of a occurrences near each symbol h as above being increased by one and the number of e symbols being increased by at most one. Consequently, when all symbols c are replaced (by d and then) by e, we get a string of the form $(a^m e)^n f$ with $m \geq n$. □

Taking into account that $FIN \subset ET0L$ and $FIN - CTEG_\infty(\infty, \infty; \alpha) \neq \emptyset, \alpha \in \{b, s, p\}$ (Lemma 3), from the previous two theorems we obtain:

Corollary. *$ET0L$ is incomparable with all families $CTEG_\alpha(i, j; \alpha), n \geq 1$, with*

1) $i \geq 1, j \geq 0, \alpha \in \{b, s, p\}$,
2) $i \geq 0, j \geq 2, \alpha \in \{s, p\}$.

The question whether $CTEG_1(0, 2; b) - ET0L$ is nonempty or not remains *open*. The next theorem shows that the answer is negative for systems with forbidding contexts of length one, hence Theorem 4 cannot be improved from this point of view.

Theorem 5. $CTEG_\infty(0, 1) \subset ET0L$.

Proof. In view of Lemma 3, we have to prove only the inclusion, the strictness is obvious.

Let $\Sigma = (E, A_1, \ldots, A_n)$ be a CTEG system with

$$E = (V_E, (\emptyset, c_1 : P_1), \ldots, (\emptyset, c_m : P_m)),$$
$$A_i = (V_i, (\emptyset, e_{i1} : P_{i1}), \ldots, (\emptyset, e_{ir_i} : P_{ir_i})), 1 \leq i \leq n,$$

with $c_j = (c_{j1}, \ldots, c_{jn}), c_{ji} \in V_i, 1 \leq i \leq n, 1 \leq j \leq m$, and $e_{ij} \in V_E, 1 \leq i \leq n, 1 \leq j \leq r_i$.

Without loss of generality, we may assume that the alphabets V_i are pairwise disjoint and disjoint from V_E (we can easily achieve that by a systematic change of symbols, in tables and conditions).

We construct an ET0L system G as follows.

For every sequence $j_1, j_2, \ldots, j_n, j$ of integers such that $1 \leq j_i \leq r_i, 1 \leq i \leq n$, and $1 \leq j \leq m$, we consider the symbol $[j_1, \ldots, j_n; j]$ and the set $\{e_{1j_1}, e_{2j_2}, \ldots, e_{nj_n}\}$. Denote by $a_1, \ldots, a_k$ the distinct symbols of this set (some symbols e_{ij_i} might be identical, hence $k \leq n$), and construct the tables

$$P_{(j_1, \ldots, j_n; j)} = \{a_1 \to \#, \ldots, a_k \to \#\} \cup$$

$$
\begin{aligned}
& \cup \{c_{j1} \to \#, \ldots, c_{jn} \to \#\} \cup \\
& \cup \{X \to [j_1, \ldots, j_n; j]\} \cup \\
& \cup \{[j'_1, \ldots, j'_n; j'] \to \# \mid \textit{for all } [j'_1, \ldots, j'_n; j']\}, \\
P'_{(j_1,\ldots,j_n;j)} &= P_{1j_1} \cup P_{2j_2} \cup \ldots \cup P_{nj_n} \cup P_j \cup \\
& \cup \{[j_1, \ldots, j_n; j] \to X\} \cup \\
& \cup \{[j'_1, \ldots, j'_n; j'] \to \# \mid (j'_1, \ldots, j'_n; j') \neq (j_1, \ldots, j_n; j)\} \cup \\
& \cup \{X \to \#\}.
\end{aligned}
$$

(These tables contain also all completion rules $q \to q$, for q not specified above.)

Denote by $\mathcal{P}$ the set of all these tables. Then

$$G = (V, T, w_E w_1 \ldots w_n X, \mathcal{P}),$$

where

$$
\begin{aligned}
T &= V_E \cup \bigcup_{i=1}^{n} V_i \cup \{X\}, \\
V &= T \cup \{\#\} \cup \{[j_1, \ldots, j_n; j] \mid 1 \leq j_i \leq r_i, 1 \leq i \leq n, 1 \leq j \leq m\},
\end{aligned}
$$

and $(w_E, w_1, \ldots, w_n) = \sigma_0$ is a starting configuration for Σ. ¿From the previous construction, it is easy to see that a derivation in G which does not introduce the trap-symbol X consists of alternate use of tables $P_{(j_1,\ldots,j_n;j)}$ and $P'_{(j_1,\ldots,j_n;j)}$, which corresponds to simulating the sequences of tables $P_{1j_1}, P_{2j_2}, \ldots, P_{nj_n}, P_j$, hence to a step in the evolution of Σ ($P_{(j_1,\ldots,j_n;j)}$ checks the non-appearance of the forbidding context both in the agents and in the environment, whereas $P'_{(j_1,\ldots,j_n;j)}$ effectively simulates the sequence of tables in Σ). Consequently,

$$L(G) = \{w'_E w'_1 \ldots w'_n X \mid \sigma_0 \Longrightarrow^*_\alpha (w'_E, w'_1, \ldots, w'_n)\}.$$

Because $ET0L$ is a full AFL (hence it is closed under erasing morphisms), it follows that $L_\alpha(\Sigma, \sigma_0) = \{w'_E \mid w'_E w'_1 \ldots w'_n X \in L(G)\} \in ET0L$. □

Corollary. $CTEG_n(0, 1; \alpha) \subset CTEG_n(0, 2; \alpha),\ n \geq 1, \alpha \in \{s, p\}$.

6. Representations in terms of CTEG languages

Based on languages such as those in Lemma 3, in [5] it is proved that the families $CTEG_\infty(\infty, \infty; \alpha), \alpha \in \{b, s, p\}$, are not closed under union, concatenation, λ-free morphisms, inverse morphisms, and intersection with regular sets. Using operations with languages, we can obtain surprising representations of ET0L languages and even of recursively enumerable languages starting from CTEG languages.

Theorem 6. *For every $ET0L$ language L, there are a language $L' \in CTEG_1(1, 0)$ (or in $CTEG_1(0, 1)$), a $T0L$ language L'' and a regular language R such that $L = L' - L'' = L' \cap R$.*

Proof. We know [10], [11] that each ET0L language is the coding (the image through a length-preserving morphism) of a T0L language. Take $L \in ET0L, L' \in T0L$ with $L \subseteq V_1^*, L' \subseteq V_2^*$, and $h : V_2^* \longrightarrow V_1^*$ such that $L = h(L')$. Consider a T0L system $G = (V_2, w, P_1, \ldots, P_n)$ generating the language L'. We construct the CTEG system $\Sigma = (E, A_1)$ with

$$E = (V_1 \cup V_2', (c, \emptyset : P_1'), \ldots, (c, \emptyset : P_n'), (c', \emptyset : P_{n+1})),$$

for

$$P_{n+1} = \{a' \to h(a) \mid a \in V_2\} \cup \{a \to a \mid a \in V_1\},$$

where $V_2' = \{a' \mid a \in V_2\}$ and $P_i', 1 \leq i \leq n$, are obtained by replacing in rules of P_i each $a \in V_2$ by its primed version a'. (It is assumed that $V_2' \cap V_1 = \emptyset$.) Moreover,

$$A_1 = (\{c, c'\}, (\emptyset, \emptyset : \{c \to c,\ c \to c',\ c' \to c'\})).$$

For $\sigma_0 = (w, c)$ we obtain $L = L(\Sigma, \sigma_0) - L(G) = L(\Sigma, \sigma_0) \cap V_1^*$. This can be easily seen: in the presence of c, E simply simulates the T0L system G, with all symbols primed. Then, in the presence of c', E simulates the morphism h. Removing the strings of primed symbols, we obtain the set L.

The same language $L(\Sigma, \sigma_0)$ is simulated by the following CTEG $\Sigma' = (E', A_1)$, with

$$E' = (V_1 \cup V_2', (\emptyset, c' : P_1'), \ldots, (\emptyset, c' : P_n'), (\emptyset, c : P_{n+1})),$$

where the tables are defined as above. Now the role of c and c' are interchanged, they are used in the forbidding way for controlling the work of E'. $\square$

Of course, because $T0L \subset CTEG_1(0,0)$, we can also represent $ET0L$ languages as morphic images of CTEG languages. Similar representations can be obtained also for recursively enumerable languages.

To this aim, we use the following results from [1], [9].

A context-free grammar *with global forbidding context conditions* is a quadruple $G = (N, T, S, P, Q)$, where $G' = (N, T, S, P)$ is a context-free grammar and $Q \subseteq (N \cup T)^+$. A derivation step $x \Longrightarrow y$ is defined only when x contains no element of Q as a substring, with the exception of the case $S = x$, when no condition is checked. If $Q = \emptyset$, then no checking is made, we have a usual context-free derivation. In [1] it is proved that each context-sensitive language can be generated by a λ-free grammar with global forbidding context conditions of length at most two. When λ-rules are used one obtains a characterization of recursively enumerable languages. A strenghtening of this result has been proved in [9]: for each context-sensitive language L there is a context-free λ-free grammar G with global forbidding context conditions such that G generates L both in the sequential and in the parallel manner. Using this result we can prove

Theorem 7. *For every recursively enumerable language L there is a propagating $CTEG_1(1,2;b)$ system Σ and a morphism h such that $L = h(L_b(\Sigma, \sigma_0))$, for some σ_0.*

Proof. Let $L \in RE, L \subseteq V_1^*$, and consider a morphism h' such that $L = h'(L'')$ for some $L'' \in CS$. Take a (λ-free) context-free grammar G with global forbidding context conditions, $G = (N, T, S, P, Q), Q = \{w_1, \ldots, w_n\}$, satisfying the conditions in [9], $L(G) = L''$, G working in the parallel or in the sequential manner. We construct a CTEG system $\Sigma = (E, A_1)$ such that $L'' = h(L_b(\Sigma, \sigma_0))$, for a certain configuration σ_0.

Let $V = N \cup T, V_0 = V$, and $V_i = \{A^{(i)} \mid A \in V\}, 1 \leq i \leq n$. Assume $N = \{B_1, \ldots, B_r\}$, and let us consider $H = P \cup \{A \to A \mid A \in V\}$. (Clearly, the E0L system (N, T, S, H) generates the same language as the context-free grammar (N, T, S, P).) Let $c_0, c_1, \ldots, c_n, c_{n+1}$ and $d_0, d_1, \ldots, d_r$ be new symbols.

The construction is based on the following idea: First, by changing its state (symbols $c_i, 1 \leq i \leq n$), the agent checks whether the context conditions are satisfied or not. Meantime, the environment only rewrites its state to the corresponding superscript variant (a string consisting of symbols $A^{(i)}$). If the context conditions are observed, then the environment applies some productions and either a new check of context conditions follows, or the agent checks whether the obtained string corresponds to a superscript version of a word over T, and after that the environment rewrites the string to a word over T. If some conditions are not satisfied, then the process is blocked.

The component E will have the following tables:

$$
\begin{aligned}
&H_0 = (c_0, \emptyset : \{S \to \alpha^{(1)} \mid S \to \alpha \in P\}),\\
&H_i = (c_i, \emptyset : \{A^{(i)} \to A^{(i+1)} \mid A \in V\}),\ 1 \leq i \leq n,\\
&H_{n+1} = (c_{n+1}, \emptyset : \{A^{(n+1)} \to \alpha^{(1)} \mid A \to \alpha \in P\} \cup \{A^{(n+1)} \to A^{(1)} \mid A \in V\}),\\
&H'_j = (d_j, \emptyset : \{A^{(1)} \to A^{(1)} \mid A \in V\}),\ 1 \leq j \leq r-1,\\
&H'_r = (d_r, \emptyset : \{A^{(1)} \to A \mid A \in V\}),
\end{aligned}
$$

where $\alpha^{(i)}$ is obtained by replacing in α each symbol $A \in V$ with $A^{(i)}$. Moreover, A_1 has the following tables:

$$
\begin{aligned}
&P_0 = (\emptyset, \emptyset : \{c_0 \to c_1\}),\\
&P_j = (\emptyset, w_j^{(j)} : \{c_j \to c_{j+1}\}),\ 1 \leq j \leq n,\\
&P_{n+1} = (\emptyset, \emptyset : \{c_{n+1} \to d_0,\ c_{n+1} \to c_1\}),\\
&P'_k = (\emptyset, B_k^{(1)} : \{d_{k-1} \to d_k\}),\ 1 \leq k \leq r.
\end{aligned}
$$

In all cases, completion rules $q \to q$ are assumed, for all q not specified above.

Then for each derivation

$$S \Longrightarrow w_1 \Longrightarrow w_2 \Longrightarrow \ldots \Longrightarrow w_t = w \in T^*$$

in G there is a derivation

$$(S, c_0) \Longrightarrow_b^* (w_1^{(1)}, c_1) \Longrightarrow_b^* (w_2^{(1)}, c_1) \Longrightarrow_b^* \ldots \Longrightarrow_b^* (w_t^{(1)}, d_0) \Longrightarrow_b^* (w_t, d_r),$$

in Σ. Moreover, only derivations in Σ associated in this way to derivations in G produce strings over T.

Let us now define the morphism h'' by $h''(a) = a, a \in T$, and $h''(b) = \lambda$ for any other symbol appearing in the alphabet of E. The equality $L(G) = h''(L_b(\Sigma, \sigma_0))$ follows. Composing with the morphism h', we get a representation of L as a morphic image of $L_b(\Sigma, \sigma_0)$. Note that because G is λ-free, Σ was propagating. $\square$

The above representation is nontrivial, in view of the fact that $CTEG_\infty(\infty, \infty; \alpha)$ does not include the family CS (even when using erasing rules: see again Lemma 3).

7. Final remarks

We want to stress here only two ideas: First, the richness of eco-grammar systems both from the point of view of formal language theory issues and from the point of view of applications. Contributions to [11] might be illustrative in this respect. Second, the surprisingly large power of CTEG with only one agent. Such systems can be depicted as in figure 3.

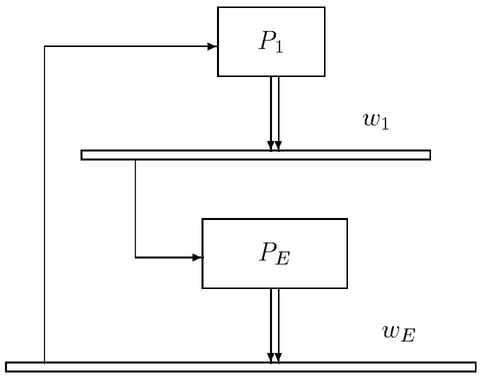

Fig. 3

There is no apparent difference here between the agent and the environment, as in the general case. Systems of this type, consisting of two coupled rewriting devices, each one checking a context condition (in the random context, semi-conditional or conditional sense, in the terminology of [6]) on the string currently generated by the partner device, deserve a deeper investigation (both for Chomsky grammars and L systems). We hope to return to this topic in a forthcoming paper.

References

1. E. Csuhaj-Varju, Grammars with local and global context conditions, *Intern. J. Computer Math.*, 47 (1992), 17 – 27.
2. E. Csuhaj-Varju, J. Dassow, J. Kelemen, Gh. Păun, *Grammar Systems: A Grammatical Approach on Distribution and Cooperation*, Gordon and Breach, London, 1994.
3. E. Csuhaj-Varju, J. Kelemen, A. Kelemenova, Gh. Păun, Eco-grammar systems: A language theoretic model for AL (Artificial Life), manuscript, 1993.
4. E. Csuhaj-Varju, J. Kelemen, A. Kelemenova, Gh. Păun, Eco-grammar systems. A preview, in vol. *Cybernetics and Systems '94* (R. Trappl, ed.), World Sci. Publ., Singapore, 1994, 941 – 949.
5. E. Csuhaj-Varju, Gh. Păun, A. Salomaa, Conditional tabled eco-grammar systems, in vol. *Artificial Life. Grammatical Models* (Gh. Păun, ed.), Proc. of the Workshop on Grammatical Models in Artificial Life, Mangalia, 1994, The Black Sea Univ. Press, Bucharest, 1995.

6. J. Dassow, Gh. Păun, *Regulated Rewriting in Formal Language Theory*, Springer-Verlag, Berlin, Heidelberg, 1989.
7. A. Ehrenfeucht, G. Rozenberg, On proving that certain languages are not ET0L, *Acta Informatica*, 6 (1976), 407 – 415.
8. C. G. Langton, Artificial Life, in vol. *Artificial Life, II* (C. G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, eds.), Proc. of the Workshop on Artificial Life, Santa Fe, 1990, Santa Fe Institute Studies in the Science of Complexity, Proc. vol. X, Addison-Wesley, 1990, 1 – 47.
9. Al. Meduna, A formalization of sequential, parallel and continuous rewriting, *Intern. J. Computer Math.*, 47 (1993), 153 – 161.
10. M. Nielsen, G. Rozenberg, A. Salomaa, S. Skyum, Nonterminals, homomorphisms and codings in different variations of 0L systems, *Acta Informatica*, part I: 4 (1974), 87 – 106, part II: 3 (1974), 357 – 364.
11. Gh. Păun (ed.), *Artificial Life. Grammatical Models*, Proc. of the Workshop on Grammatical Models in Artificial Life, Mangalia, 1994, The Black Sea Univ. Press, Bucharest, 1995.
12. G. Rozenberg, A. Salomaa, *The Mathematical Theory of L Systems*, Academic Press, New York, 1980.
13. A. Salomaa, *Formal Languages*, Academic Press, New York, 1973.