Eco-Grammar Systems, Decidability and the Tiling Problem*

Petr Sosík

Institute of Computer Science, Silesian University, Opava,
Czech Republic, E-mail:Petr.Sosik@fpf.slu.cz

Technical report No. 1998/1

Abstract

Some new results about families of languages generated by the (extended) conditional tabled eco-grammar systems are presented. Undecidability or decidability of the finiteness and emptiness problem of some $CTEG$ language families is shown. The tiling problem is used as a tool for the proof of restricted non-extended eco-grammar systems. It is shown that forbidding restricted CTEG systems with a forbidding context of the length 2 can generate all possible tilings of the plane by Wang tiles (dominoes).

1 Introduction

Eco-grammar systems have been studied since 1994 as a grammatical formalism suitable for modelling the interactions of an eco-system and the organisms living in it. They have been presented first at [6], and in the past works ([7, 2] and others) several variants have been proposed, particularly *conditional tabled eco-grammar systems.*

The basic model does not consider nonterminal symbols; if we allow them, we obtain *extended* systems, motivated by the possibility that sometimes we are interested in some stages of the development only.

In the first part of the paper we prove that the hierarchy of ECTEG systems with the p-type conditions according to various length of permitting and forbidding string collapses, which was so far known for the *restricted* version only.

Moreover, even the *restricted* version of ECTEG systems generates all context-sensitive language, which statement has several consequences including collapsing of some hierarchies of $CTEG$ family.

In the second part we prove the (un)decidability of the finiteness problem for some of $CTEG$ families. For some other families the (un)decidability

*Research supported by the Grant Agency of Czech Republic, grant No. 201/98/P047.

follows from the previously published results, and the same holds for the emptiness problem. The known fact of (un)decidability of these two problems for CTEG systems are summarized in two tables.

2 Basic definitions

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. For a set $W \subseteq V$, $|x|_W = \sum_{a \in W} |x|_a$. $\Psi_V(x) = (|x|_{a_1}, \ldots, |x|_{a_n})$ 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 [11].

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 $z_1 = a_1 a_2 \ldots a_r$, $z_2 = x_1 x_2 \ldots x_r$, 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 (e.g. [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.

2.1 Conditional tabled eco-grammar systems

First, a preliminary definition.

Definition 2.1 *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 (of two parameters, the second parameter being always a generated string) *is true* for c_i as the first parameter, and certain predicate *is not true* for d_i. The predicates we consider here will be of the following types:

Definition 2.2 *Given an alphabet V, we define the following three predicates over $V^* \times V^*$:*

$$\begin{aligned}
\pi_b(x,y) = 1 \quad &\text{iff} \quad y = y_1 x y_2, \\
\pi_s(x,y) = 1 \quad &\text{iff} \quad y = y_1 x_1 y_2 x_2 \ldots y_r x_r y_{r+1}, \\
&\phantom{\text{iff}} \quad x = x_1 x_2 \ldots x_r, \;\; x_i, y_i \in V^* \text{ for all } i, \\
\pi_p(x,y) = 1 \quad &\text{iff} \quad \Psi_V(x) \leq \Psi_V(y), \;\; \text{componentwise.}
\end{aligned}$$

Therefore, $\pi_b(x,y)$ is true when x is a subword of y (a *block* of it), $\pi_s(x,y)$ is true when x is a *scattered* subword of y and $\pi_p(x,y)$ 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 is given by words, not by symbols.

Definition 2.3 *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.

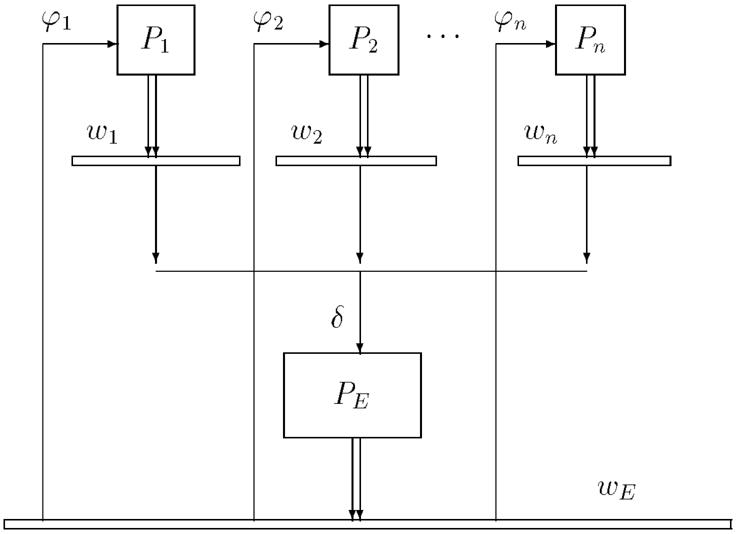


Figure 1: A model of conditional tabled eco-grammar system.

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 2.2. We shall specify this in a formal way below. One can see that we have a system with the structure as represented in figure 1.

Definition 2.4 *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 2.5 *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* $\pi_\alpha(c_{ji}, w_i) = 1$, $\pi_\alpha(d_{ji}, w_i) = 0$*, for all* $1 \leq i \leq n$*, and* $w_E \Longrightarrow_{P_j} w'_E$ *(remind that* c_j, d_j *are* n*-ary conditions).*

(2) Every $A_i, 1 \leq i \leq n$*, has a table* $(e_{ij}, f_{ij} : P_{ij})$ *for some* $j, 1 \leq j \leq r_i$*, such that* $\pi_\alpha(e_{ij}, w_E) = 1$, $\pi_\alpha(f_{ij}, w_E) = 0$*, 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 there is not a possibility of misunderstanding, we describe also a CTEG system Σ by notation $\mathrm{CTEG}_n(i, j; \alpha)$ (not slashed) for brevity with n, i, j, α as above.

When we are interested only in strings over some subalphabet T of V_E, hence in the language $L_\alpha(\Sigma, \sigma_0) \cap T^*$, then we speak about *extended conditional tabled eco-grammar system* (ECTEG system, for short) and about an *extended* environmental language. An ECTEG system will be presented as an $(n+2)$-tuple

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

and the family of the extended environmental languages will be denoted by $ECTEG_n(i, j; \alpha)$, n, i, j, α as above.

If in a CTEG system as above we ignore all permitting contexts (for instance, we replace all of them by λ – note that $\pi_\alpha(\lambda, y) = 1$ 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.

In [8] a subfamily of ECTEG systems, so-called *restricted* ECTEG systems are defined. In them the sets of the productions in the tables of the agents (that is P_{ij}, for all i, j,) are only of the form $A \to B$. We denote them as $\mathrm{ECTEG}^{\mathrm{r}}$ systems and the corresponding family of the extended environmental languages will be denoted by $ECTEG_n^r(i, j; \alpha)$, n, i, j, α as above.

Unless stated otherwise, we consider only λ-free (E)CTEG systems in this paper.

3 Family relationship results

Note that due to the details of the proofs, all the results in this section hold as well for *restricted* ECTEG systems. It suggests an open question, whether there is any difference between restricted and unrestricted ECTEG systems.

The following relations directly follow from definitions (symbol E must be used either in all parts of the statements or nowhere):

Lemma 3.1

(i) $(E)CTEG_n(i,j;\alpha) \subseteq (E)CTEG_m(i',j';\alpha)$, *for all* $\alpha \in \{b,s,p\}$, *for* $n \leq m$ *and* $i \leq i', j \leq j'$.

(ii) $(E)CTEG_n(i,j;b) = (E)CTEG_n(i,j;s) = (E)CTEG_n(i,j;p)$, *for* $n \geq 1, i,j \in \{0,1\}$.

(iii) $(E)CTEG_\infty(0,0;\alpha) = (E)CTEG_1(0,0;\alpha) = (E)T0L,\ \ \alpha \in \{b,s,p\}$.

(iv) $(E)CTEG^r_n(i,j;\alpha) \subseteq (E)CTEG_n(i,j;\alpha)$, *for any* $i,j \geq 0$, $n \geq 1$, $\alpha \in \{b,s,p\}$.

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

The following results are shown in [2]:

Lemma 3.2

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

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

(ii) $FIN - CTEG_\infty(\infty,\infty;\alpha) \neq \emptyset$, $\alpha \in \{b,s,p\}$.

(iii) $CTEG_\infty(0,1) \subset ET0L$, *strict inclusion.*

For ECTEG systems, the relations expressed in the following three lemmas are shown in [8]:

Lemma 3.3 $ECTEG_n(i,j;\alpha) = ECTEG_1(i,j;\alpha)$ *for any* $i,j \geq 0$, $n \geq 1$, $\alpha \in \{b,s,p\}$.

Lemma 3.4 $ECTEG_\infty(\infty,\infty;\alpha) \subseteq CS$, $\alpha \in \{b,s,p\}$.

Lemma 3.5

(i) $ECTEG_n(i,j;p) \subseteq ECTEG_n(i,j;s)$ *for all* $n \geq 1$, $i,j \geq 0$.

(ii) $ECTEG^r_\infty(i,j;p) = MAT_{ac} \subseteq ECTEG^r_\infty(i,j;s)$ *for any* $i \geq 1$, $j \geq 2$.

As an immediate consequence of Lemmas 3.1 and 3.2 we have

Lemma 3.6 $ECTEG_\infty(0,0) = ECTEG_\infty(0,1) = ET0L$.

The following two theorems show that also for every non-restricted ECTEG system with p-type conditions their lengths can be reduced to (1,2):

Theorem 3.7 *$ECTEG_n(i,j;p) = ECTEG_n(1,j;p)$ for any $i,j \geq 0, n \geq 1$.*

Proof. Since the number of agents is irrelevant (Lemma 3.3), it is enough to consider the case $n = 1$. Consider an arbitrary one-agent ECTEG$_1(i,j;p)$ system $\Sigma = (E, T, A_1)$,

- $E = (V_E, (c_1, d_1 : P_1), \ldots, (c_m, d_m : P_m))$ for some $m \geq 1$;
- $A_1 = (V_1, (e_1, f_1 : Q_1), \ldots, (e_r, f_r : Q_r))$ for some $r \geq 1$.

We will construct an equivalent ECTEG$_1(1,j;p)$ system $\Sigma' = (E', T, A_1')$. As a basic principle we simulate every condition $\pi_p(x,y)$ for $x = a_1 \ldots a_i$ with i subsequent conditions $\pi_p(a_k, y)$ for $k = 1, \ldots, i$. The case $|x|_{a_k} > 1$ for some k is resolved by indexing of the symbols.

System Σ' starts by choosing nondeterministically a lower index t describing a table of A_1, whose derivation step will be simulated. Simultaneously it assigns to symbols of w_E (again nondeterministically) another lower index k, $1 \leq k \leq i$, describing their position in permitting string e_t. Forbidding string f_t is applied when transforming symbols $B \in w_1$ to the form $B^{(t1)}$.

Then the system tests (via symbols $B^{(t1)} \ldots B^{(t|e_t|)}$) whether the symbols of e_t are in w_E. During this testing the environment string cannot change, as only the completing rules of table $(\emptyset, \emptyset : Q_t')$ for some t can be applied to w_E. If the testing is successful, the agent performs a derivation step (otherwise it remains in an infinite loop of completing rules) and the whole cycle is repeated symmetrically for the environment derivation step.

In the following construction, all completing rules in the tables of E' have the form $A \to \$$ for any $A \in V_E'$, and completing rules in the tables of A_1' have the form $B \to \#$ for any $B \in V_1'$, where $\$,\#$ are new nonterminals (see below).

- $E' = (V_E', (c_1', d_1' : P_1'), \ldots, (c_{m'}', d_{m'}' : P_{m'}'))$ for some $m' \geq 1$, where
  - $V_E' = V_E \cup \{A_{tk} | A \in V_E,\ 1 \leq t \leq r,\ 1 \leq k \leq i\} \cup \{A^{(sk)} | A \in V_E,\ 1 \leq s \leq m,\ 1 \leq k \leq i\} \cup \{\$\}$;
  - E' contains tables $(\emptyset, \emptyset : P_t')$ for every t, $1 \leq t \leq r$, containing rules $A \to A_{tk}$ and $A_{tk} \to A_{tk}$ for every $A \in V_E$ and every k, $1 \leq k \leq i$;
  - E' moreover contains tables $(B, d_s : P_{Bs}')$ for every $B \in V_1$ and for every s, $1 \leq s \leq m$, containing rules $A_{tk} \to A^{(s1)}$ for every $A \in V_E$ and every t, k $1 \leq t \leq r$, $1 \leq k \leq i$;
  - for every s, $1 \leq s \leq m$ such that $c_s = \emptyset$ there is a table $(\emptyset, \emptyset : P_{s1}')$ in E' containing rules $A^{(s1)} \to \alpha$, for every rule $A \to \alpha$ in P_s;
  - for every s, $1 \leq s \leq m$ such that $c_s \neq \emptyset$ there is a sequence of tables $(c_{s1}', \emptyset : P_{s1}'), \ldots, (c_{s|c_s|}', \emptyset : P_{s|c_s|}')$ in E' simulating P_s, where

* c'_{sk} is k-th symbol of the string c_s, indexed $_{sk}$;
* tables P'_{sk} for every k, $1 \leq k < |c_s|$ contain rules $A^{(sk)} \rightarrow A^{(s(k+1))}$ for every $A \in V_E$;
* table $P'_{s|c_s|}$ contain rules $A^{(s|c_s|)} \rightarrow \alpha$, for every rule $A \rightarrow \alpha$ in P_s.

- $A'_1 = (V'_1, (e'_1, f'_1 : Q'_1), \ldots, (e'_{r'}, f'_{r'} : Q'_{r'}))$ for some $r' \geq 1$, where
  - $V'_1 = V_1 \cup \{B_{sk} | B \in V_1,\ 1 \leq s \leq m,\ 1 \leq k \leq i\} \cup \{B^{(tk)} | B \in V_1,\ 1 \leq t \leq r,\ 1 \leq k \leq i\} \cup \{\#\}$;
  - A'_1 contains tables $(\emptyset, \emptyset : Q'_s)$ for every s, $1 \leq s \leq m$, containing rules $B \rightarrow B_{sk}$ and $B_{sk} \rightarrow B_{sk}$ for every $B \in V_1$ and every k, $1 \leq k \leq i$;
  - A'_1 moreover contains tables $(A, f_t : Q'_{At})$ for every $A \in V_E$ and for every t, $1 \leq t \leq r$, containing rules $B_{sk} \rightarrow B^{(t1)}$ for every $B \in V_1$ and every s, k $1 \leq s \leq m$, $1 \leq k \leq i$;
  - for every t, $1 \leq t \leq r$ such that $e_t = \emptyset$ there is a table $(\emptyset, \emptyset : Q'_{t1})$ in A'_1 containing rules $B^{(t1)} \rightarrow \alpha$, for every rule $B \rightarrow \alpha$ in Q_t;
  - for every t, $1 \leq t \leq r$ such that $e_t \neq \emptyset$ there is a sequence of tables $(e'_{t1}, \emptyset : Q'_{t1}), \ldots, (e'_{t|e_t|}, \emptyset : Q'_{t|e_t|})$ in A'_1 simulating Q_t, where
    * e'_{tk} is k-th symbol of the string e_t, indexed $_{tk}$;
    * tables Q'_{tk} for every k, $1 \leq k < |e_t|$ contain rules $B^{(tk)} \rightarrow B^{(t(k+1))}$ for every $B \in V_1$;
    * table $Q'_{t|e_t|}$ contain rules $B^{(t|e_t|)} \rightarrow \alpha$, for every rule $B \rightarrow \alpha$ in Q_t.

The structure of the environment and the agent is symmetric. The starting configuration of Σ' is $\sigma'_0 = (w_{E0}, w'_{10})$ for some $w_{E0} \in V_E^*$ and $w'_{10} = B_{11} \ldots Z_{11}$, where $\sigma_0 = (w_{E0}, w_{10})$ for $w_{10} = B \ldots Z$ is a starting configuration of Σ.

It follows from the given description, that $\sigma_1 \Longrightarrow_{\Sigma}^{*} \sigma_2$ *iff* $\sigma'_1 \Longrightarrow_{\Sigma'}^{*} \sigma'_2$ for any $\sigma_1 = (w_{E1}, w_{11})$, $\sigma_2 = (w_{E2}, w_{12})$, where $\sigma'_1 = (w_{E1}, w'_{11})$ and $\sigma'_2 = (w_{E2}, w'_{12})$ are identical with σ_1, σ_2 except that the symbols of w'_{11}, w'_{12} have assigned lower indexes $_{11}$ as above (see example 3.8).

Since the terminal alphabet of Σ and Σ' are identical and all the new symbols of Σ' are nonterminals, it can be concluded that $L(\Sigma) = L(\Sigma')$. $\square$

Example 3.8 Consider the following $\mathrm{ECTEG}_1(2, 0; p)$ system $\Sigma = (E, T, A_1)$, where

- $E = (\{a, b\}, (\emptyset, \emptyset : \{a \rightarrow bb, b \rightarrow aa\}), (\emptyset, \emptyset : \{a \rightarrow a/b, b \rightarrow b\}))$;
- $T = \{a, b\}$;
- $A_1 = (\{c\}, (ab, \emptyset : \{c \rightarrow c\}))$.

According to the previous proof it can be simulated by the $\mathrm{ECTEG}_1(1,0;p)$ system $\Sigma' = (E', T, A_1')$, where

- $E' = (\{a, a_{11}, a_{12}, a^{(11)}, a^{(12)}, a^{(21)}, a^{(22)}, b, b_{11}, b_{12}, b^{(11)}, b^{(12)}, b^{(21)}, b^{(22)}\}$,
  $(\emptyset, \emptyset : \{a \to a_{11}/a_{12}, b \to b_{11}/b_{12}, a_{11} \to a_{11}, a_{12} \to a_{12}, b_{11} \to b_{11}, b_{12} \to b_{12}\})$,
  $(c, \emptyset : \{a_{11} \to a^{(11)}, a_{12} \to a^{(11)}, b_{11} \to b^{(11)}, b_{12} \to b^{(11)}\})$,
  $(c, \emptyset : \{a_{11} \to a^{(21)}, a_{12} \to a^{(21)}, b_{11} \to b^{(21)}, b_{12} \to b^{(21)}\})$,
  $(\emptyset, \emptyset : \{a^{(11)} \to bb, b^{(11)} \to aa\}), (\emptyset, \emptyset : \{a^{(21)} \to a/b, b^{(21)} \to b\}))$;

- $A_1' = (\{c, c_{11}, c_{12}, c_{21}, c_{22}, c^{(11)}, c^{(12)}\}$,
  $(\emptyset, \emptyset : \{c \to c_{11}/c_{12}, c_{11} \to c_{11}, c_{12} \to c_{12})$,
  $(\emptyset, \emptyset : \{c \to c_{21}/c_{22}, c_{21} \to c_{21}, c_{22} \to c_{22})$,
  $(a, \emptyset : \{c_{11} \to c^{(11)}, c_{12} \to c^{(11)}, c_{21} \to c^{(11)}, c_{22} \to c^{(11)}\})$,
  $(b, \emptyset : \{c_{11} \to c^{(11)}, c_{12} \to c^{(11)}, c_{21} \to c^{(11)}, c_{22} \to c^{(11)}\})$,
  $(a_{11}, \emptyset : \{c^{(11)} \to c^{(12)}\}), (b_{12}, \emptyset : \{c^{(12)} \to c\}))$.

For instance the derivation $(ab, c) \Longrightarrow_\Sigma (bbaa, c)$ can be simulated by the derivation $(ab, c_{11}) \Longrightarrow_{\Sigma'} (a_{11}b_{12}, c^{(11)}) \Longrightarrow_{\Sigma'} ((a_{11}b_{12}, c^{(12)}) \Longrightarrow_{\Sigma'} (a_{11}b_{12}, c) \Longrightarrow_{\Sigma'} (a^{(11)}b^{(11)}, c_{11}) \Longrightarrow_{\Sigma'} (bbaa, c_{11})$.

Theorem 3.9 *$ECTEG_n(i,j;p) = ECTEG_n(i,2;p)$ for any $i, j \geq 0,\ n \geq 1$.*

Proof. The scheme of the proof is analogous as in the previous theorem. We again cut the forbidding string into symbols and test them separately. Contrary, the testing is not in AND style, but rather in OR style, since the forbidding condition $\pi_p(x,y)$ for $x = a_1 \dots a_i$ does not hold if *any* of the conditions $\pi_p(a_k, y)$ for $k = 1, \dots, i$ do not hold. Again the case $|x|_{a_k} > 1$ needs special attention: by assigning indexes to the symbols we check whether there is less then $|x|_{a_k}$ symbols in the string y. Only for this case we need $|x|_{a_k} - 1$ forbidding contexts of length 2, which are subsequently tested for symbols a_k with various indexes.

Formally, consider an arbitrary one-agent $\mathrm{ECTEG}_1(i,j;p)$ system $\Sigma = (E, T, A_1)$, where

- $E = (V_E, (c_1, d_1 : P_1), \dots, (c_m, d_m : P_m))$ for some $m \geq 1$;

- $A_1 = (V_1, (e_1, f_1 : Q_1), \dots, (e_r, f_r : Q_r))$ for some $r \geq 1$.

We will construct an equivalent $\mathrm{ECTEG}_1(i,2;p)$ system $\Sigma' = (E', T, A_1')$. In the following construction, all completing rules in the tables of E' have the form $A \to \$$ for any $A \in V_E'$, and completing rules in the tables of A_1' have the form $B \to \#$ for any $B \in V_1'$, where $\$,\#$ are new nonterminals (see below).

- $E' = (V_E', (c_1', d_1' : P_1'), \dots, (c_{m'}', d_{m'}' : P_{m'}'))$ for some $m' \geq 1$, where
  - $V_E' = V_E \cup \{A_{tk} | A \in V_E,\ 1 \leq t \leq r,\ 1 \leq k < j\} \cup \{A^{(sk)} | A \in V_E,\ 1 \leq s \leq m,\ 1 \leq k < j\} \cup \{\$\}$;

- E' contains tables $(\emptyset, \emptyset : P'_t)$ for every t, $1 \le t \le r$, containing rules $A \to A_{tk}$ and $A_{tk} \to A_{tk}$ for every $A \in V_E$ and every k, $1 \le k \le \max(|f_t|_A - 1, 1)$;
- for every s, $1 \le s \le m$ such that $c_s = \emptyset$ there is a set of tables $(B, \emptyset : P'_{Bs})$ for every $B \in V_1$ in E', containing rules $A_{tk} \to A^{(s1)}$ for every $A \in V_E$ and every t, k, $1 \le t \le r$, $1 \le k < j$;
- for every s, $1 \le s \le m$ such that $c_s \ne \emptyset$ there is a table $(c_s, \emptyset : P''_s)$ in E' containing rules $A_{tk} \to A^{(s1)}$ for every $A \in V_E$ and every t, k, $1 \le t \le r$, $1 \le k < j$;
- for every s, $1 \le s \le m$ such that $d_s = \emptyset$ there is a table $(\emptyset, \emptyset : P'_{s1})$ in E' containing rules $A^{(s1)} \to \alpha$, for every rule $A \to \alpha$ in P_s.
- for every s, $1 \le s \le m$ such that $d_s \ne \emptyset$ there is the following set of tables in E' simulating P_s :
  - for every $B \in V_1$ such that $|d_s|_B = 1$ there is a table $(\emptyset, B_{s1} : P'_{sB})$ containing rules $A^{(s1)} \to \alpha$, for every rule $A \to \alpha$ in P_s.
  - for every $B \in V_1$ such that $|d_s|_B \ge 2$ there is a sequence of tables $(\emptyset, B_{s1}B_{s1} : P'_{s1}), \ldots, (\emptyset, B_{sq}B_{sq} : P'_{sq})$ in E', where $q = |d_s|_B - 1$ and
    - tables P'_{sk} for every k, $1 \le k < q$ contain rules $A^{(sk)} \to A^{(s(k+1))}$ for every $A \in V_E$;
    - table P'_{sq} contains rules $A^{(sq)} \to \alpha$, for every rule $A \to \alpha$ in P_s.

- $A'_1 = (V'_1, (e'_1, f'_1 : Q'_1), \ldots, (e'_{r'}, f'_{r'} : Q'_{r'}))$ for some $r' \ge 1$, where
  - $V'_1 = V_1 \cup \{B_{sk} | B \in V_1,\ 1 \le s \le m,\ 1 \le k < j\} \cup \{B^{(tk)} | B \in V_1,\ 1 \le t \le r,\ 1 \le k < j\} \cup \{\#\}$;
  - A'_1 contains tables $(\emptyset, \emptyset : Q'_s)$ for every s, $1 \le s \le m$, containing rules $B \to B_{sk}$ and $B_{sk} \to B_{sk}$ for every $B \in V_1$ and every k, $1 \le k \le \max(|d_t|_B - 1, 1)$;
  - for every t, $1 \le t \le r$ such that $e_t = \emptyset$ there is a set of tables $(A, \emptyset : Q'_{At})$ for every $A \in V_E$ in A'_1, containing rules $B_{sk} \to B^{(t1)}$ for every $B \in V_1$ and every s, k, $1 \le s \le m$, $1 \le k < j$;
  - for every t, $1 \le t \le r$ such that $e_t \ne \emptyset$ there is a table $(e_t, \emptyset : Q''_t)$ in A'_1 containing rules $B_{sk} \to B^{(t1)}$ for every $B \in V_1$ and every s, k, $1 \le s \le m$, $1 \le k < j$;
  - for every t, $1 \le t \le r$ such that $f_t = \emptyset$ there is a table $(\emptyset, \emptyset : Q'_{t1})$ in A'_1 containing rules $B^{(t1)} \to \alpha$, for every rule $B \to \alpha$ in Q_t.
  - for every t, $1 \le t \le r$ such that $f_t \ne \emptyset$ there is a following set of tables in A'_1 simulating Q_t :
    - for every $A \in V_E$ such that $|f_t|_A = 1$ there is a table $(\emptyset, A_{t1} : Q'_{tA})$ containing rules $B^{(t1)} \to \alpha$, for every rule $B \to \alpha$ in Q_t.

* for every $A \in V_E$ such that $|f_s|_A \geq 2$ there is a sequence of tables $(\emptyset, A_{t1}A_{t1} : Q'_{t1}), \ldots, (\emptyset, A_{tq}A_{tq} : Q'_{tq})$ in V'_1, where $q = |f_t|_A - 1$ and
    - tables Q'_{tk} for every k, $1 \leq k < q$ contain rules $B^{(tk)} \rightarrow B^{(t(k+1))}$ for every $B \in V_1$;
    - table Q'_{tq} contains rules $B^{(tq)} \rightarrow \alpha$, for every rule $B \rightarrow \alpha$ in Q_t.

Again the starting configuration of Σ' is $\sigma'_0 = (w_{E0}, w'_{10})$ for some $w_{E0} \in V_E$ and $w'_{10} = B_{11} \ldots Z_{11}$, where $\sigma_0 = (w_{E0}, w_{10})$ for $w_{10} = B \ldots Z$ is a starting configuration of Σ.

It follows from the given description, that $\sigma_1 \Longrightarrow^*_\Sigma \sigma_2$ *iff* $\sigma'_1 \Longrightarrow^*_{\Sigma'} \sigma'_2$ for any $\sigma_1 = (w_{E1}, w_{11})$, $\sigma_2 = (w_{E2}, w_{12})$, where $\sigma'_1 = (w_{E1}, w'_{11})$ and $\sigma'_2 = (w_{E2}, w'_{12})$ are identical with σ_1, σ_2 except that the symbols of w'_{11}, w'_{12} have assigned lower indexes $_{11}$ as above (see example 3.10).

It can be concluded that $L(\Sigma) = L(\Sigma')$. □

Example 3.10 Consider the ECTEG$_1(0,3;p)$ system $\Sigma = (E, T, A_1)$ identical with the one of example 3.8, except that the agent $A_1 = (\{c\}, (\emptyset, aab : \{c \rightarrow c\}))$.

The proper simulating ECTEG$_1(0,2;p)$ system $\Sigma' = (E', T, A'_1)$ is again similar to the one of example 3.8, except that the first table of E' does not contain the rules with b_{12} and the last two tables of the agent A'_1 are replaced by the tables $(\emptyset, a_{11}a_{11} : \{c^{(11)} \rightarrow c\})$, $(\emptyset, b_{11} : \{c^{(11)} \rightarrow c\}))$.

The derivation $(ab, c) \Longrightarrow_\Sigma (bbaa, c)$ then can be simulated by $(ab, c_{11}) \Longrightarrow_{\Sigma'} (a_{11}b_{11}, c^{(11)}) \Longrightarrow_{\Sigma'} (a_{11}b_{11}, c) \Longrightarrow_{\Sigma'} (a^{(11)}b^{(11)}, c_{11}) \Longrightarrow_{\Sigma'} (bbaa, c_{11})$.

Corollary 3.11 *$ECTEG_n(i,j;p) = ECTEG_n(1,2;p)$ for every $i, j \geq 0$, $n \geq 1$.*

It is shown in [8] that the $CTEG$ families are contained in CS. The following theorem shows that even for restricted systems with block forbidding conditions of length 2 the inclusion becomes equality.

Theorem 3.12 *$ECTEG^r_1(0,2;b) = CS$.*

Proof. Consider an arbitrary CS grammar $G = (N, T, P, S)$ in the one-sided normal form [12], i.e. every rule is of one of the types (1) $A \rightarrow BC$, (2) $A \rightarrow a$, (3) $AB \rightarrow AC$, for some $A, B, C \in N$ and $a \in T$.

Let us denote $n = |N|$, let Γ be the set of left sides of type 3 rules in P, let $m = |\Gamma|$, let $r = m(n-1) + m^2$.

Let $\Sigma = (E, T, A_1)$ be a one-agent restricted ECTEG system, where

- $E = (V_E, (\emptyset, \# : P_E))$ is a conditional T0L scheme, for which

- V_E contains the following symbols:
  - all symbols in $N \cup T$;
  - special symbols $B_A^{(1)}, \dots, B_A^{(r)}$ for every $AB \in \Gamma$;
  - special symbols $A^{(1)} \dots A^{(r)}$ for every $A \in N$;
- $\#$ is a symbol from V_1 (see below);
- P_E contains the following rules:
  - $A \to B^{(1)}C^{(1)}$ for every type 1 rule $A \to BC$ in P;
  - all type 2 rules in P;
  - $a \to a$ for every $a \in T$;
  - $A \to A^{(1)}$ for every $A \in N$;
  - $B \to B_A^{(1)}$ for every $AB \in \Gamma$;
  - $X^{(i)} \to X^{(i+1)}$ for every i, $1 \leq i < r$ and for every symbol $X^{(i)} \in V_E$;
  - $A^{(r)} \to A$ for every $A \in N$;
  - $B_A^{(r)} \to C$ for every type 3 rule $AB \to AC$ in P;

- $A_1 = (V_1, (\emptyset, \emptyset : P_0), (\emptyset, f_1 : P_1), \dots, (\emptyset, f_r : P_r))$ is a conditional T0L scheme. In all the tables P_i, all completion rules are of the form $x \to \#$ for every symbol $x \in V_1$ not contained in the following description.
  - $V_1 = \{\#, S^{(0)}, S^{(1)}, \dots, S^{(r)}\}$;
  - table P_i for every i, $0 \leq i < r$ contains rules $S^{(i)} \to S^{(i+1)}$;
  - table P_r contains rules $S^{(r)} \to S^{(0)}$;
  - forbidding string f_i for every i, $1 \leq i \leq m(n-1)$ is equal to $C^{(i)}B_A^{(i)}$, where (C, AB) is i-th member of the set $\{(C, AB) | (C, AB) \in N \times \Gamma, C \neq A\}$;
  - forbidding string $f_{m(n-1)+i}$ for every i, $1 \leq i \leq m^2$ is equal to $B_A^{(i)} D_C^{(i)}$, where (AB, CD) is i-th member of $\Gamma \times \Gamma$.

Starting configuration of Σ is $\sigma_0 = (S, S^{(0)})$. The system works cyclically with $r + 1$ steps period, which is governed by the agent rules. Due to the described agent tables, at i-th step of the cycle the i-th table of agent must be used, otherwise the agent generates the symbol $\#$, which blocks the derivation of the environment.

The whole cycle, which simulates one or more derivation steps of G, can be described as follows:

- In the first step the use of an arbitrary number of rules in P are simulated; in the case of the type 3 rules, the symbols B_A means "there is B with the left context A."

- In the following $m(n-1)$ steps the agent tests whether or not there is a "false" substring $C, B_A \in V_E$ such, that $C \neq A$ in the environment string. During this and the following phase the environment does not change; it is only synchronized via superscripts of all nonterminals.

- In the following m^2 steps the agent tests whether or not the environment tries to use type 3 rules in "overlapping" manner, i.e. whether or not there is a substring B_A, D_C for some $B_A, D_C \in V_E$. In the case that this or the previous condition is violated, the derivation is blocked.

- In the last step of the cycle the type 3 derivations of the environment are finished and the whole cycle starts over.

It follows from the previous description that $x \Longrightarrow_G y$ if and only if $(x, S_0) \Longrightarrow_\Sigma^* (y, S_0)$ and during this derivation no new terminal symbols can appear. Hence $L(G) = L(\Sigma, (S, S_0))$. □

Together with Lemma 3.4 we obtain

Corollary 3.13

$$ECTEG_n(i, j; b) = CS \textit{ for all } n \geq 1, i \geq 0, j \geq 2.$$

The Lemma 3.2 can be now completed as follows:

Corollary 3.14 *All families $CTEG_n(i, j; \alpha)$, for any $n \geq 1$ and $\alpha \in \{b, s, p\}$, are:*

(a) strictly included in $ET0L$ for $i = 0, j \leq 1$;

(b) incomparable with $ET0L$ otherwise (i.e. for either $i > 0$ or $j > 1$ or both).

Proof. Due to Theorem 3.12, one can obtain via intersection of family $CTEG_1(0, 2; b)$ with T^+ all CS languages for some terminal alphabet T. Since the $ET0L$ family is AFL, clearly the families $CTEG_n(i, j; b)$ are not contained in $ET0L$ for any $i \geq 0, j \geq 2, n \geq 1$. The statements of Lemma 3.2 conclude the proof. □

Corollary 3.15

$$ECTEG_\infty(i, j; p) \subseteq ECTEG_\infty(i, j; s) \subseteq ECTEG_\infty(i, j; b) \textit{ for } i \leq 1, j \geq 0.$$

Proof. The statement follows directly from the Lemmas 3.1, 3.4, 3.5 and from the Theorem 3.12. □

Lemma 3.5 can be extended by the use of Theorem 3.7 as follows:

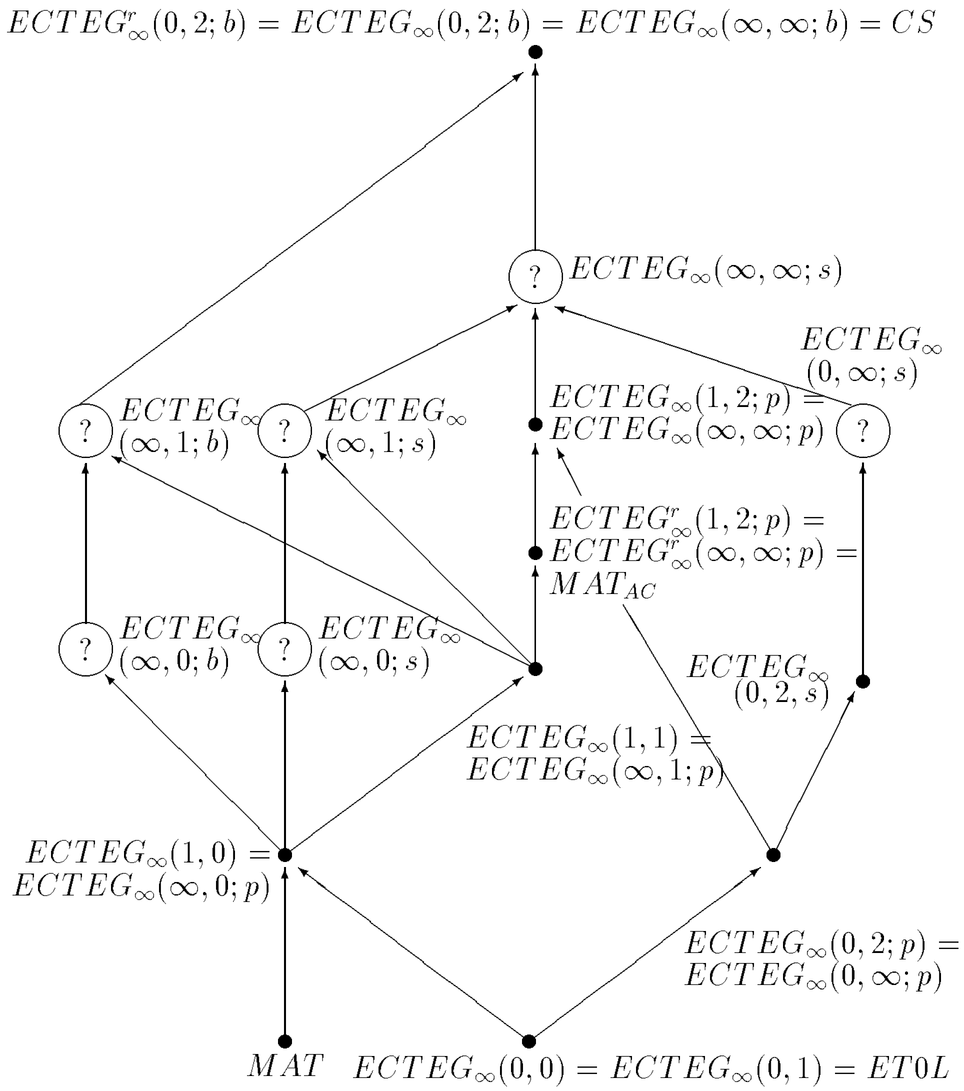


Figure 2: The hierarchy of classes of λ-free ECTEG languages.

Corollary 3.16

$$ECTEG_n(i,j;p) \subseteq ECTEG_n(i,j;s) \textit{ for all } n \geq 1,\ \ i,j \geq 0.$$

$$ECTEG_n(i,j;p) \subseteq ECTEG_n(i,j;b) \textit{ for all } n \geq 1,\ \ i,j \geq 0.$$

A relation of languages $ECTEG_n(i,j;s)$ and $ECTEG_n(i,j;b)$ for the case $i \geq 2, j \leq 1$ and their properties remain an *open problem.*

The known results concerning ECTEG language classes are sumarized in Figure 2. The arrows denote the known subset relation of classes. Note that *only* the subset relations outgoing from the root nodes of the diagram are known to be strict. The circles with quotation mark denote possible hierarchy of classes depending on the length of context conditions. Small dot nodes denote single classes of languages (e.g. in some cases collapsing hierarchies). Note that three important classes of languages are characterized by λ-free ECTEG systems: the class $ET0L$, MAT_{ac} and CS.

4 Decidability results

Recall first the following basic facts [4, 10]:

Lemma 4.1 *The emptiness and the finiteness problem are:*

(i) decidable for ET0L systems;

(ii) decidable for λ-free matrix grammars without appearance checking;

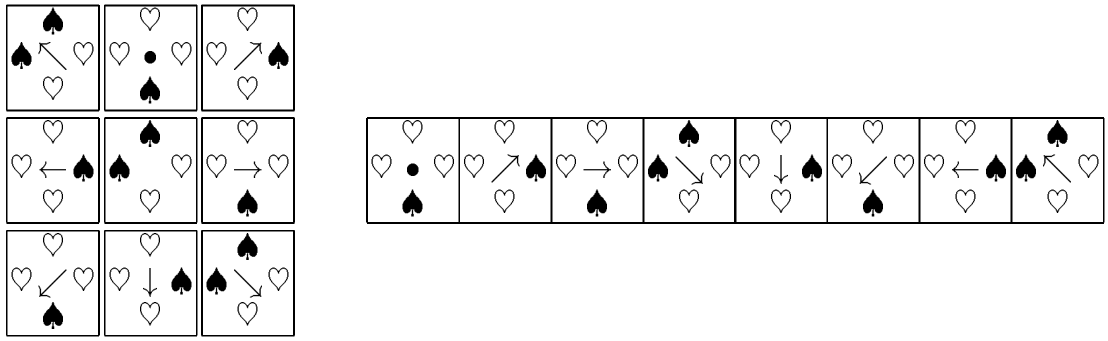
(iii) undecidable for λ-free matrix grammars with appearance checking;

(iv) undecidable for CS grammars.

The following theorem shows an interesting connection between CTEG systems and the tiling problem. Recall that Wang tiles (dominoes) are square tiles with coloured edges (from some finite colour set). The tiles can be put together only so that the neighbouring edges have the same colour. Neither rotation nor reflection of the tiles is allowed. Having a given set of tiles, each tile is understood as a prototype, i.e. it can be used (infinitely) many times in the tiling. The following result can be found in [1]:

Lemma 4.2 *Given a finite set of Wang tiles, it is undecidable, whether there exists a tiling of the plane by these tiles or not.*

Theorem 4.3 *It is undecidable for a given $CTEG_1^{\pi}(0,2;b)$ system Σ and a starting configuration σ_0, whether $L(\Sigma,\sigma_0)$ is finite or not.*



(a) (b)

Figure 3: An example of a tiling (a) and the corresponding environment string (b).

Proof. Consider an arbitrary finite set $\mathcal{S}$ of Wang tiles (dominoes). For every such set we will construct a $\mathrm{CTEG}_1^{\mathrm{r}}(0,2;b)$ system Σ with starting configuration σ_0 such that there is a tiling of the plane by the set $\mathcal{S}$ if and only if $L(\Sigma,\sigma_0)$ is infinite.

We give an informal description of the CTEG system Σ first. We denote the colours of edges of the tiles by symbols. The alphabet V_E of Σ consists of all tiles from the set $\mathcal{S}$, each contained in $4n^2$ exemplars, distinguished by an upper index (where $n = |\mathcal{S}|$). The reason is that the environment must be synchronized, similarly as in the proof of Theorem 3.12.

The tiling is built in the form of square of tiles, growing one by one peripheral layer of the tiles. The environment string denotes the cyclical peripheral layer of the tiles. The tiles are marked with an arrow, specifying the side of the square, where they lie (and thus also direction of their "expansion." The starting tile of the string is to the right of the upper-left corner one and it is marked with a $\bullet$. The situation is depicted at fig. 3.

Formally, let us denote $n = |\mathcal{S}|$, $r = 4n^2$. Let $\Sigma = (E, T, A_1)$ be a one-agent restricted CTEG system, where

- $E = (V_E, (\emptyset, \# : P_{E1}), \ldots, (\emptyset, \# : P_{En}))$ is a conditional T0L scheme, for which
  - V_E contains tiles $\boxed{\begin{array}{ccc} & a & \\ d & x & b \\ & c & \end{array}}^{(0)} \ldots \boxed{\begin{array}{ccc} & a & \\ d & x & b \\ & c & \end{array}}^{(r)}$ for every tile $\boxed{\begin{array}{ccc} & a & \\ d & & b \\ & c & \end{array}} \in \mathcal{S}$ and for every $x \in \{\rightarrow, \leftarrow, \downarrow, \uparrow, \nearrow, \searrow, \swarrow, \nwarrow, \bullet\}$;
  - $\#$ is a symbol from V_1 (see below);

- P_{Ei} for every i, $1 \le i \le n$, contains the following rules (common for all P_{Ei}) for every $\boxed{\begin{array}{ccc} & a & \\ d & & b \\ & c & \end{array}} \in \mathcal{S}$:
  - $\boxed{\begin{array}{ccc} & a & \\ d & \uparrow & b \\ & c & \end{array}}^{(0)} \to \boxed{\begin{array}{ccc} & e & \\ g & \uparrow & f \\ & a & \end{array}}^{(1)}$ for every $\boxed{\begin{array}{ccc} & e & \\ g & & f \\ & a & \end{array}} \in \mathcal{S}$;
  - $\boxed{\begin{array}{ccc} & a & \\ d & \nearrow & b \\ & c & \end{array}}^{(0)} \to \boxed{\begin{array}{ccc} & e & \\ g & \uparrow & f \\ & a & \end{array}}^{(1)} \boxed{\begin{array}{ccc} & h & \\ f & \nearrow & m \\ & j & \end{array}}^{(1)} \boxed{\begin{array}{ccc} & j & \\ b & \to & k \\ & l & \end{array}}^{(1)}$ for every triple $\boxed{\begin{array}{ccc} & e & \\ g & & f \\ & a & \end{array}}, \boxed{\begin{array}{ccc} & h & \\ f & & m \\ & j & \end{array}}, \boxed{\begin{array}{ccc} & j & \\ b & & k \\ & l & \end{array}} \in \mathcal{S}^3$;
  - $\boxed{\begin{array}{ccc} & a & \\ d & \to & b \\ & c & \end{array}}^{(0)} \to \boxed{\begin{array}{ccc} & e & \\ b & \to & f \\ & g & \end{array}}^{(1)}$ for every $\boxed{\begin{array}{ccc} & e & \\ b & & f \\ & g & \end{array}} \in \mathcal{S}$;

  $\vdots$
  - $\boxed{\begin{array}{ccc} & a & \\ d & \leftarrow & b \\ & c & \end{array}}^{(0)} \to \boxed{\begin{array}{ccc} & e & \\ g & \leftarrow & d \\ & f & \end{array}}^{(1)}$ for every $\boxed{\begin{array}{ccc} & e & \\ g & & d \\ & f & \end{array}} \in \mathcal{S}^2$ (direction $\nwarrow$ is intentionally omitted in this list);
  - $\boxed{\begin{array}{ccc} & a & \\ d & x & b \\ & c & \end{array}}^{(i)} \to \boxed{\begin{array}{ccc} & a & \\ d & x & b \\ & c & \end{array}}^{(i+1)}$ for every $x \in \{\to, \leftarrow, \downarrow, \uparrow, \nearrow, \searrow, \swarrow, \nwarrow, \bullet\}$ and every i, $1 \le i < r$;
  - $\boxed{\begin{array}{ccc} & a & \\ d & x & b \\ & c & \end{array}}^{(r)} \to \boxed{\begin{array}{ccc} & a & \\ d & x & b \\ & c & \end{array}}^{(0)}$ for every $x \in \{\to, \leftarrow, \downarrow, \uparrow, \nearrow, \searrow, \swarrow, \nwarrow, \bullet\}$;

P_{Ei} for every i, $1 \le i \le n$ moreover corresponds to i-th tile $\boxed{\begin{array}{ccc} & a_i & \\ d_i & & b_i \\ & c_i & \end{array}}$ from $\mathcal{S}$ and contains the following specific rules:

  - $\boxed{\begin{array}{ccc} & c_i & \\ f & \nwarrow & l \\ & e & \end{array}}^{(0)} \to \boxed{\begin{array}{ccc} & g & \\ j & \leftarrow & f \\ & h & \end{array}}^{(1)} \boxed{\begin{array}{ccc} & k & \\ m & \nwarrow & d_i \\ & g & \end{array}}^{(1)}$ for every triple $\boxed{\begin{array}{ccc} & c_i & \\ f & & l \\ & e & \end{array}}, \boxed{\begin{array}{ccc} & g & \\ j & & f \\ & h & \end{array}}, \boxed{\begin{array}{ccc} & k & \\ m & & d_i \\ & g & \end{array}} \in \mathcal{S}^3$;
  - $\boxed{\begin{array}{ccc} & f & \\ j & \bullet & g \\ & h & \end{array}}^{(0)} \to \boxed{\begin{array}{ccc} & a_i & \\ d_i & \bullet & b_i \\ & c_i & \end{array}}^{(1)} \boxed{\begin{array}{ccc} & d_i & \\ b_i & \uparrow & e \\ & f & \end{array}}^{(1)}$ for every double $\boxed{\begin{array}{ccc} & f & \\ j & & g \\ & h & \end{array}}, \boxed{\begin{array}{ccc} & d_i & \\ b_i & & e \\ & f & \end{array}} \in \mathcal{S}^2$.

- $A_1 = (V_1, (\emptyset, \emptyset : P_0), (\emptyset, f_1 : P_1), \ldots, (\emptyset, f_r : P_r))$ is a conditional T0L scheme. In all the tables P_i all completion rules are of the form $x \to \#$ for every symbol $x \in V_1$ not contained in the following description:
  - $V_1 = \{\#, S^{(0)}, S^{(1)}, \ldots, S^{(r)}\}$;
  - table P_0 contains rule $S^{(0)} \to S^{(1)}$;
  - tables P_i for every i, $1 \le i < r$ contain rules $S^{(i)} \to S^{(i+1)}$;
  - table P_r contains rule $S^{(r)} \to S^{(0)}$;
  - forbidding string f_i for every i, $0 < i \le n^2$ is equal to some $\boxed{\begin{matrix} & a & \\ d & \uparrow & b \\ & c & \end{matrix}}^{(i)} \boxed{\begin{matrix} & e & \\ h & \uparrow & f \\ & g & \end{matrix}}^{(i)}$ if $b \ne h$, and $\emptyset$ otherwise, so that all doubles from $\mathcal{S}^2$ are used;
  - forbidding string f_i for every i, $n^2 < i \le 2n^2$ is equal to some $\boxed{\begin{matrix} & a & \\ d & \to & b \\ & c & \end{matrix}}^{(i)} \boxed{\begin{matrix} & e & \\ h & \to & f \\ & g & \end{matrix}}^{(i)}$ if $c \ne e$, and $\emptyset$ otherwise, so that all doubles from $\mathcal{S}^2$ are used;
  - forbidding string f_i for every i, $2n^2 < i \le 3n^2$ is equal to some $\boxed{\begin{matrix} & a & \\ d & \downarrow & b \\ & c & \end{matrix}}^{(i)} \boxed{\begin{matrix} & e & \\ h & \downarrow & f \\ & g & \end{matrix}}^{(i)}$ if $d \ne f$, and $\emptyset$ otherwise, so that all doubles from $\mathcal{S}^2$ are used;
  - forbidding string f_i for every i, $3n^2 < i \le 4n^2$ is equal to some $\boxed{\begin{matrix} & a & \\ d & \leftarrow & b \\ & c & \end{matrix}}^{(i)} \boxed{\begin{matrix} & e & \\ h & \leftarrow & f \\ & g & \end{matrix}}^{(i)}$ if $a \ne g$, and $\emptyset$ otherwise, so that all doubles from $\mathcal{S}^2$ are used.

In the first step of the cycle the systems generates a new peripheral layer of the tiles. Then the environment string waits for r steps, until the tests of correctness are done by the agent. (One can be sure that the new leftmost and rightmost tile of the string are coloured well, due to their synchronized generation by environment tables.)

The tables P_i for every i, $1 \le i \le r$ allow further derivation if and only if there are no tiles neighbouring with differently coloured sides. If the colours do not correspond, the derivation is blocked. The testing is performed via forbidding conditions of the agent's tables including all possible "bad" neighbourhoods of the tiles.

Axioms of the system are all configurations of the form $\sigma_0 = (w_{E0}, S^{(0)})$, where w_{E0} can be any peripheral 8-tile-long layer of a correct 3×3 tiling. An example of a derivation steps is in figure 4.

It follows from the given description, that $L(\Sigma, \sigma_0)$ is infinite, if and only if a tiling of the plane exists with a given set of tiles. $\square$

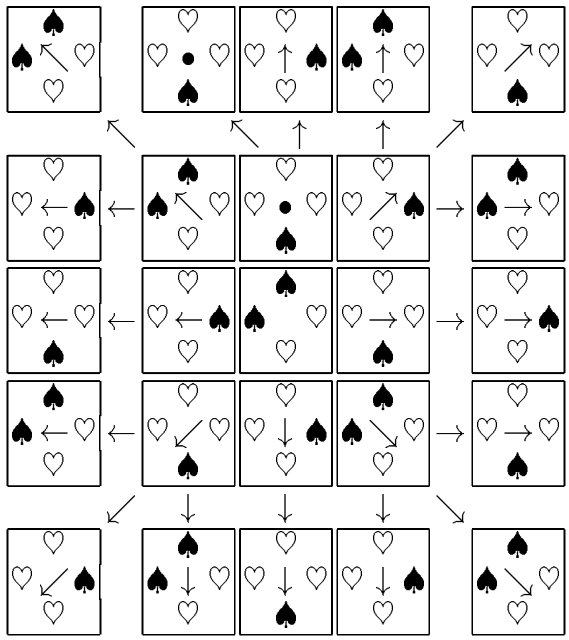


Figure 4: An example of a derivation of another peripheral layer from the preceding one.

The previous result (and also Lemma 3.5) suggests that the forbiding conditions of the length at least 2 are strong enough to make the finiteness problem undecidable. Now we show that this does not hold for the permitting conditions, at least in the case of p-type predicates.

Be reminded that for arbitrary strings x, y over an alphabet $V = \{a_1, \ldots, a_n\}$ we denote

$$\Psi_V(x) \leq \Psi_V(y) \; iff \; |x|_{a_i} \leq |y|_{a_i} \text{ for every } i, 1 \leq i \leq n;$$

$$\Psi_V(x) < \Psi_V(y) \; iff \; \Psi_V(x) \leq \Psi_V(y) \text{ and } |x|_{a_i} < |y|_{a_i} \text{ for some } i, \; 1 \leq i \leq n.$$

Lemma 4.4 *Let $\left\{x^{(i)}\right\}_{i=0}^{\infty}$ be an infinite sequence of strings over an alphabet $V = \{a_1, \ldots, a_n\}$ such that $\Psi_V(x^{(i)}) \neq \Psi_V(x^{(j)})$ for $i \neq j$. Moreover, let there be some $p \in \mathcal{N}$ such that $|x^{(i)}| \leq |x^{(i+1)}| \leq p|x^{(i)}|$ for every $i \geq 0$.*

Then there exists a number $m \in \mathcal{N}$ such that $\Psi_V(x^{(j)}) < \Psi_V(x^{(m)})$ for some $j < m$.

Proof. We will show, that there is an upper bound of m, depending only on n and on the length of string $x^{(0)}$. In other words, there is a function $f : \mathcal{N} \times \mathcal{N} \longrightarrow \mathcal{N}$ such that $m \leq f(n, |x^{(0)}|)$.

Assume that $f(n, s)$ is a nondecreasing function of s for a fixed n. We give the proof by induction on n.

For the case $n = 1$ we can define $f(1, s) = 1$ for every $s \geq 0$, since necessarily $\Psi_V(x^{(0)}) < \Psi_V(x^{(1)})$ and hence $m = 1$.

Assume the statement holds for some $n \geq 1$ and assume by contradiction that there is no upper bound of m for the case $n+1$. Then there must be an i, $1 \leq i \leq n+1$ such that $|x^{(1)}|_{a_i} < |x^{(0)}|_{a_i}$. Denote $V_i = \{a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{n+1}\}$ and $s_0 = |x^{(0)}|$. By induction hypothesis there is at most $f(n, |x^{(1)}|_{V_i}) \leq f(n, ps_0)$ different strings $x^{(k)}$ (for $1 < k \leq 1 + f(n, ps_0)$) with the fixed $|x^{(k)}|_{a_i} = |x^{(1)}|_{a_i}$ such that $\Psi_{V_i}(x^{(j)}) < \Psi_{V_i}(x^{(k)})$ does not hold for some $j < k$. (We simply ignore the symbols a_i, since their number in $x^{(k)}$'s is fixed, and take into the account only the remaining n symbols of V.) Denote s_1 the maximal length of string $x^{(1)}$, m_1 the maximal number of these strings $x^{(k)}$:

$$\begin{aligned} s_1 &= ps_0, \\ m_1 &= f(n, s_1). \end{aligned}$$

Now, after generating at most m_1 different strings, we cannot continue with the chosen value of $|x|_{a_i}$. For the string $x^{(m_1+1)}$, either a different number of a_i's or a new i, $1 \leq i \leq n+1$ must be chosen such that $|x^{(m_1+1)}|_{a_i} < |x^{(0)}|_{a_i}$. By induction hypothesis again, there is at most $f(n, |x^{(m_1+1)}|_{V_i}) \leq f(n, p^{m_1+1}s_0)$ other strings $x^{(k)}$ with the fixed $|x^{(k)}|_{a_i} = |x^{(m_1+1)}|_{a_i}$ such that $\Psi_{V_i}(x^{(j)}) < \Psi_{V_i}(x^{(k)})$ does not hold for any $j < k$. Denote

$$\begin{aligned} s_2 &= p^{m_1+1}s_0 = p^{m_1}s_1, \\ m_2 &= f(n, s_2). \end{aligned}$$

After another choice of $x^{(m_2+m_1+1)}$ such that $|x^{(m_1+1)}|_{a_i} < |x^{(0)}|_{a_i}$ for some i, $1 \leq i \leq n+1$ (as before) we obtain

$$\begin{aligned} s_3 &= p^{m_2+m_1+1}s_0 = p^{m_2}s_2, \\ m_3 &= f(n, s_3). \end{aligned}$$

Since there are only s_0 such different choices, after applying all of them we have

$$\begin{aligned} s_{s_0} &= p^{m_{s_0-1}+\ldots+m_2+m_1+1}s_0 = p^{m_{s_0-1}}s_{s_0-1}, \\ m_{s_0} &= f(n, s_{s_0}). \end{aligned}$$

Now, for any choice of new $x^{(m)}$, where $m = m_{s_0} + m_{s_0-1} + \ldots + m_2 + m_1 + 1$, there is a $j < m$ such that $x^{(j)} < x^{(m)}$, and hence we have found the upper bound of m.

The function f can now be defined in the following way: Let's define first recurrent sequences s_i, m_i :

$$\begin{aligned} s_0 &= |x^{(0)}|, \\ m_0 &= 1, \\ \\ s_j &= p^{m_{j-1}}s_{j-1}, \qquad \text{for } j > 0, \\ m_j &= f(n, s_j), \qquad \text{for } j > 0, \end{aligned}$$

where $f(n, s_j)$ is nondecreasing by the induction hypothesis. Then

$$f(n+1, s_0) = \sum_{j=0}^{s_0} m_j$$

is also a nondecreasing function of s_0 for a fixed n. $\square$

Theorem 4.5 *It is decidable, given a $CTEG_1(i, 0; p)$ system Σ, for $i \geq 0$ and a configuration σ_0, whether $L(\Sigma, \sigma_0)$ is finite.*

Proof. We can assume $V_E \cap V_1 = \emptyset$ without loss of generality. Denote $V = V_E \cup V_1 = \{a_1, \ldots, a_n\}$. We will represent each configuration $\sigma = (w_E, w_1)$ of Σ by its Parikh vector $\Psi(\sigma) = \Psi(w_E)\Psi(w_1)$ associated to V.

Note that the development of the system in configuration $\sigma = (w_E, w_1)$ depends only on $\Psi(\sigma)$ and not on the order of symbols in w_E, w_1 due to the type p of conditions. Hence $L(\Sigma, \sigma_0)$ is infinite if and only if there is an infinite sequence of subsequent configurations $\{\Psi(\sigma_i)\}_{i=0}^{\infty}$ such that their Parikh vectors are pairwise different. We will show, that this holds if and only if there is $m \in \mathcal{N}$ such that $\Psi(\sigma_j) < \Psi(\sigma_m)$ for some $j,\ 0 \leq j < m$.

The sequence of Parikh vectors satisfies the prerequisites of Lemma 4.4, for $p = \max\{|x|; (a \to x) \in P_E \cup P_1\}$, and hence we have the sufficient condition of our statement. For the necessary condition, assume that for some $j, m,\ 0 \leq j < m$ we have $\Psi(\sigma_j) < \Psi(\sigma_m)$. Then in subsequent $m - j$ steps of the system the same tables and rules can be used as in the derivation $\sigma_j \Longrightarrow^+ \sigma_m$, since all the permitting conditions remain valid. Hence there are configuration steps $\sigma_m \Longrightarrow \sigma_{m+1} \Longrightarrow \ldots \Longrightarrow \sigma_{m+(m-j)}$ such that $\Psi(\sigma_{j+i}) < \Psi(\sigma_{m+i})$ for every $i,\ 1 \leq i \leq m - j$. Then there is infinite sequence $\Psi(\sigma_m) < \Psi(\sigma_{m+(m-j)}) < \Psi(\sigma_{m+2(m-j)}) < \ldots$ and thus $L(\Sigma, \sigma_0)$ is infinite. $\square$

We can summarize the current state of the art of the finiteness and emptiness problem of (E)CTEG systems in the following tables. Here single rows correspond to different types of condition b, s, p, columns correspond to the different lengths of permitting and forbidding strings. The number of agents is irrelevant for the decidability of these problems. We denote + as decidable, - as undecidable, ? as an open problem.

Corollary 4.6 *Tables 1 and 2 hold.*

Proof. The first column in both tables follows from Lemmas 3.2 and 3.6. "+" in the second column of table 1 follows from Theorem 4.5. "$-$" in the last column of table 1 follows from Theorem 4.3.

"$-$" in the third column of table 2 are a consequence of Theorem 3.12, the last column follows from Theorem 3.12 and Lemmas 3.5, 3.1 and 4.1. $\square$

Note: This paper deals only with the nonerasing version of CTEG systems and therefore the *membership* problem is decidable for all of them, since all the families are contained in CS.

PROBLEM	$i,j \rightarrow$ $\alpha \downarrow$	$i=0$ $j \leq 1$	$i \geq 1$ $j=0$	$i \geq 1$ $j=1$	$i \geq 0$ $j \geq 2$
FINITENESS	b	$+$	?	?	$-$
	s	$+$	?	?	?
	p	$+$	$+$	?	?
EMPTINESS	b	T R I V I A L			
	s				
	p				

Table 1: Decidability results of the finiteness and the emptiness problem for $CTEG_\infty(i,j;\alpha)$ families.

PROBLEM	$i,j \rightarrow$ $\alpha \downarrow$	$i=0$ $j \leq 1$	$i \geq 1$ $j \leq 1$	$i=0$ $j \geq 2$	$i \geq 1$ $j \geq 2$
FINITENESS	b	$+$	?	$-$	$-$
	s	$+$	?	?	$-$
	p	$+$	?	?	$-$
EMPTINESS	b	$+$	?	$-$	$-$
	s	$+$	?	?	$-$
	p	$+$	?	?	$-$

Table 2: Decidability results of the finiteness and the emptiness problem for $ECTEG_\infty(i,j;\alpha)$ families.

5 Acknowledgements

I would like to thank to Alica Kelemenová and Jozef Kelemen for their continuous support, to Kristína Smítalová, Václav Studený, Valeria Mihalache and Gheorge Păun for valuable discussion and/or information. Finally to Leoš Štýbnar for the famous inspiring tiling-puzzles in his university office.

References

[1] R. L. Berger: The undecidability of the domino problem. *Memoires of American Mathematical Society* **66**, (1966)

[2] E. Csuhaj-Varjú, Gh. Păun, A. Salomaa: Conditional tabled eco-grammar systems. In: [9], 227–239.

[3] J. Dassow, V. Mihalache: Eco-grammar systems, matrix grammars and E0L systems. In: [9], 210–226.

[4] J. Dassow, Gh. Păun: *Regulated Rewriting in Formal Language Theory.* Springer-Verlag, Berlin, Heidelberg, 1989.

[5] J. Gruska: *Foundations of Computing.* International Thomson Computer Press, 1997.

[6] J. Kelemen, E. Csuhaj-Varjú, A. Kelemenová, Gh. Păun: Eco-grammar systems. A preview. In: R. Trappl (ed.): *Proc.* 12$^{\text{th}}$ *European Meeting on Cybernetics and System Research,* World Scientific, Singapore, 1994, 941–948.

[7] A. Kelemenová, J. Kelemen: From colonies to (eco)grammar systems–an overview. In: J. Karhumaki, H. Maurer, G. Rozenberg (eds.): *Results and Trends in Theoretical Computer Science*, Springer-Verlag, Berlin 1994, 213–231.

[8] V. Mihalache: Extended conditional-tabled eco-grammar systems. *Journal of Information Processing and Cybernetics* **30**, 213–229 (1994)

[9] Gh. Păun (ed.): *Artificial Life: Grammatical Models.* The Black Sea University Press, Bucharest, 1995.

[10] G. Rozenberg, A. Salomaa: *The Mathematical Theory of L systems.* Academic Press, New York, 1980.

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

[12] A. Salomaa, G. Rozenberg (eds.): *Handbook of Formal Languages.* Springer-Verlag, 1997.