(Very) initial ideas on non-cooperating polymorphic P systems and parallel communicating ET0L systems

György Vaszil

Faculty of Informatics, University of Debrecen

BWMC 2024

As we have seen:

$$\mathcal{L}(NOP(polym\ , ncoo, fin)) = PsET0L.$$

$$\mathcal{L}(NOP(polym\;, ncoo, f(n))) = \text{ ?}$$

What happens if the system is not finitely representable?

$$\mathcal{L}(NOP(polym\ ,ncoo, fin)) = \ ?$$

What happens if the system is not finitely representable?

The righthand sides of rules are “words” of an infinite language →
→ symbols are replaced with “words” of unbounded length →
→ like parallel communicating grammar systems?

Parallel communicating grammar systems

Example with regular components

$$\Gamma = (N, K, T, G_1, G_2, G_3)$$

$N = \{S_1, S_2, S_3\}$	–	nonterminal alphabet
$T = \{a, b, c\}$	–	terminal alphabet
$K = \{Q_1, Q_2, Q_3\}$	–	query symbols

$$G_i = (N \cup K, T, P_i, S_i),\ 1 \leq i \leq 3$$

- $\mathrm{N} \cup \mathrm{K}$ – nonterminals
- T – terminals
- $\mathrm{P_i}$ – set of rewriting rules
- $\mathrm{S_i}$ – $S_i \in N$ start symbol

Example with regular components

$$P_1 : \{\ S_1 \to aS_1,\ S_1 \to aQ_2, \quad P_2 : \{S_2 \to bS_2\} \quad P_3 : \{S_3 \to cS_3\}$$
$$S_2 \to bQ_3,\ S_3 \to c\ \ \}$$

A derivation:

$\boxed{G_1}$ S_1	$\boxed{G_2}$ S_2	$\boxed{G_3}$ S_3
$\vdots$	$\vdots$	$\vdots$
$a..aS_1$	$b..bS_2$	$c..cS_3$
$a..aaQ_2$	$b..bbS_2$	$c..ccS_3$
$a..aab..bbS_2$	S_2	$c..ccS_3$
$a..aab..bbbQ_3$	aS_2	$c..cccS_3$
$a..aab..bbbc..cccS_3$	aS_2	S_3
$\underline{a..aab..bbbc..cccc}$	aaS_2	cS_3

$$L(\Gamma) = \{x \in T^* \quad | \quad (S_1, S_2, \ldots, S_n) \Longrightarrow^* (x, \alpha_2, \ldots, \alpha_n),$$
$$\alpha_i \in V_\Gamma^*, 2 \le i \le n\}.$$

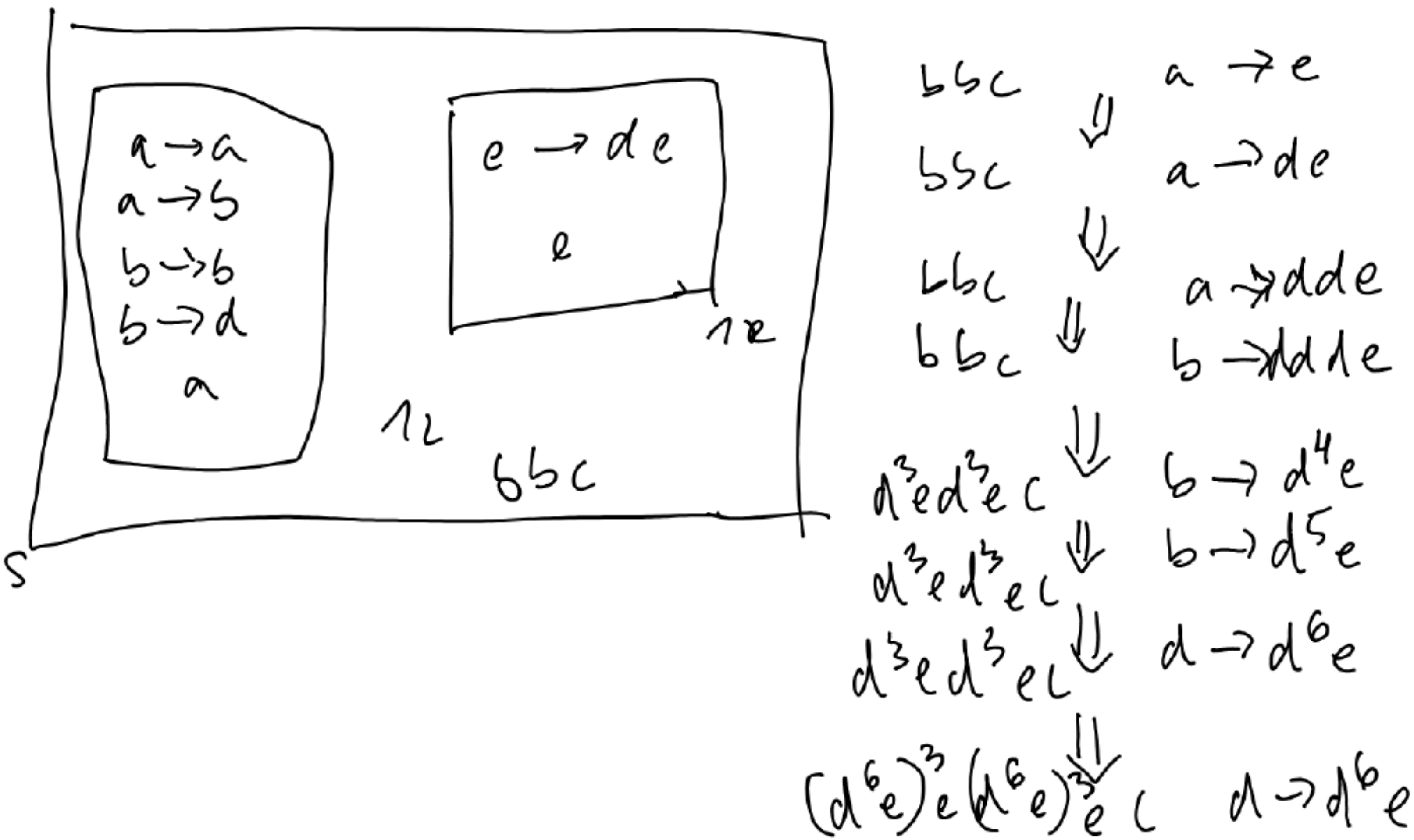
What can PC grammar systems generate?

- With context-free components: RE

- With E(T)0L components?

Theorem 3 *For every recursively enumerable language L, there is a non-returning PC E0L system Γ, a regular language R, and a weak coding h, such that $L = h(L(\Gamma) \cap R)$.*

Polymorphic P systems vs. PC ET0L systems



Polymorphic P systems vs. PC ET0L systems – a system with 2 components

($a_{1L} \to a_{1L}$
$a_{1L} \to b_{1L}$
$b_{1L} \to F$
$d_{1L} \to F$
$b \to b$
$d \to d$)
⋮

($b_{1L} \to b_{1L}$
$b_{1L} \to d_{1L}$
$b \to Q_{1R}$
$a_{1L} \to F$
$c \to c$)
⋮

($d_{1L} \to d_{1L}$
$a_{1L} \to F$
$b_{1L} \to F$
$d \to Q_{1R}$
⋮)

$S - 1L$	$1R$
$bbc\,a_{1L}$	S
$bbc\,a_{1L}$	e
$bbc\,a_{1L}$	de
$bbc\,b_{1L}$	dde
$Q_{1R}Q_{1R}c\,b_{1L}$	$ddde$
$d^3e\,d^3e\,c\,b_{1L}$	
$d^3e\,d^3e\,c\,b_{1L}$	d^4e
$d^3e\,d^3e\,c\,d_{1L}$	d^5e
$(Q_{1R})^3e(Q_{1R})^3e\,c\,d_{1L}$	d^6e
$(d^6e)^3e(d^6e)^3e\,c\,d_{1L}$	

$(e \to de)$

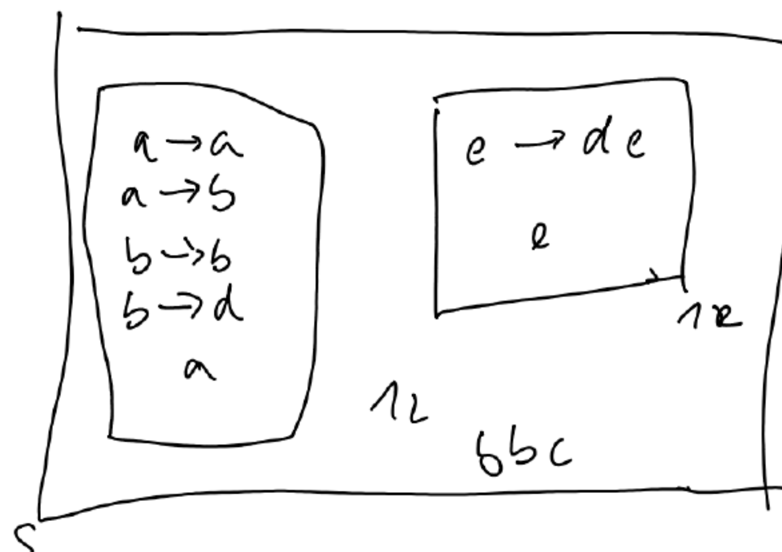
($a_{1L} \to a_{1L}$
$a_{1L} \to b_{1L}$
$b_{1L} \to F$
$d_{1L} \to F$
$b \to b$
$a \to d$)
⋮

($b_{1L} \to b_{1L}$
$b_{1L} \to d_{1L}$
$b \to Q_{1R}$
$a_{1L} \to F$
$c \to c$)
⋮

($d_{1L} \to d_{1L}$
$a_{1L} \to F$
$b_{1L} \to F$
$d \to Q_{1R}$
⋮)

s-1L	1R
$bbc\,a_{1L}$	s
$bbc\,a_{1L}$	e
$bbc\,a_{1L}$	de
$bbc\,b_{1L}$	dde
$Q_{1R}Q_{1R}c\,b_{1L}$	$ddde$
$d^3e\,d^3e\,c\,b_{1L}$	
$d^3e\,d^3e\,c\,b_{1L}$	d^4e
$d^3e\,d^3e\,c\,d_{1L}$	d^5e
$(Q_{1R})^3e(Q_{1R})^3e\,c\,d_{1L}$	d^6e
$(d^6e)^3e(d^6e)^3e\,c\,d_{1L}$	

$(e \to de)$



Conclusion

- Polymorphism in P systems and parallel communication in PC systems seem to have similar consequences.

- It seems that languages of non-cooperating polymorphic P systems are included in the class of Parikh sets of PC ET0L languages.