SILESIAN
UNIVERSITY
FACULTY OF PHILOSOPHY
AND SCIENCE IN OPAVA

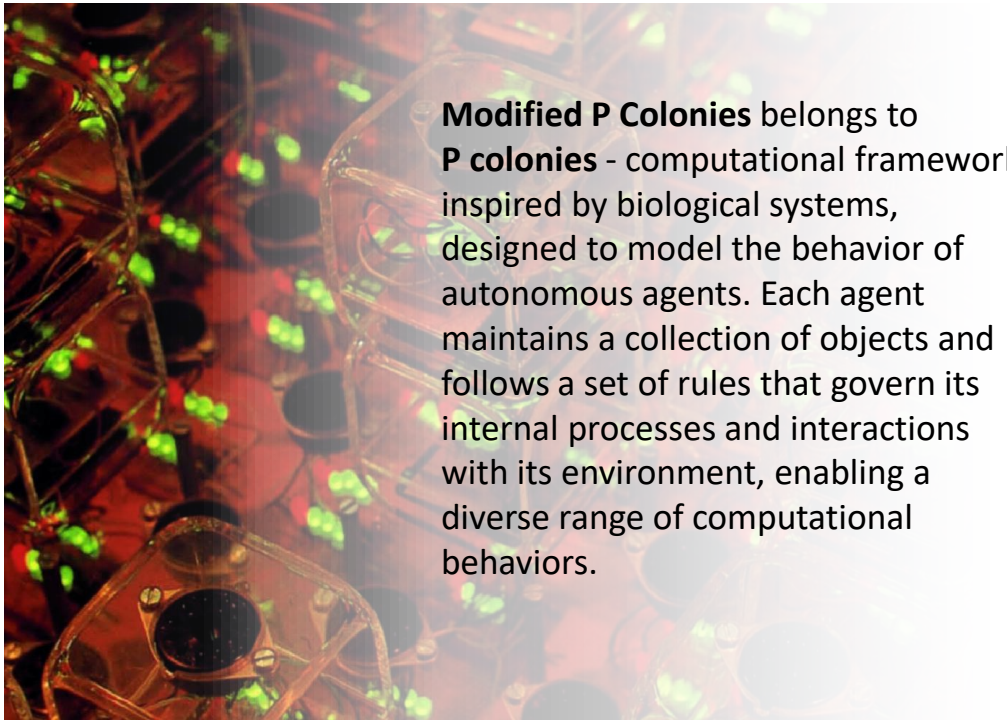
P COLONIES WITH RULE APPLICATION INTENSITY

Lucie Ciencialová, Luděk Cienciala

Institute of Computer Science,
Faculty of Philosophy and Science in Opava,
Silesian University in Opava, Czech Republic

Sevilla, 14. 1. 2026

MODIFIED P COLONIES



Modified P Colonies belongs to **P colonies** - computational frameworks inspired by biological systems, designed to model the behavior of autonomous agents. Each agent maintains a collection of objects and follows a set of rules that govern its internal processes and interactions with its environment, enabling a diverse range of computational behaviors.

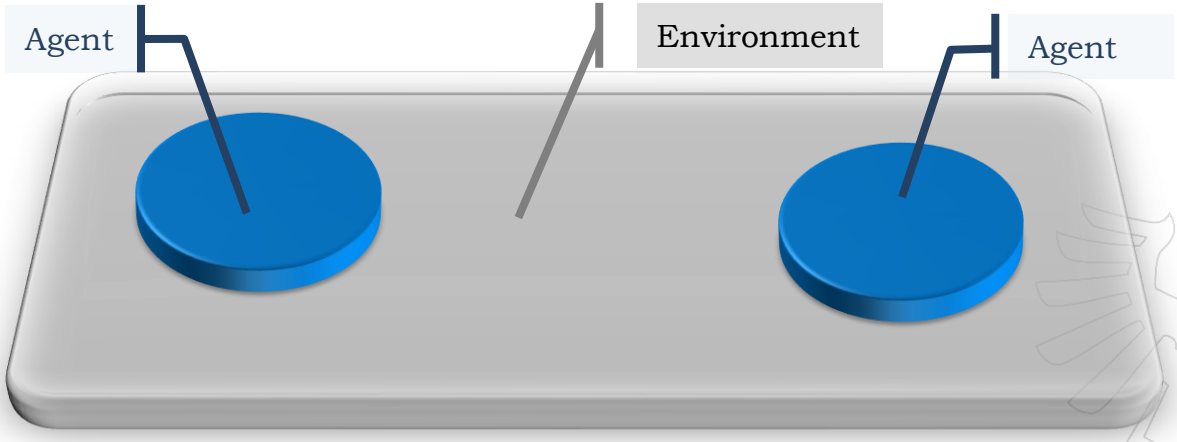
- Introduction of Modified P Colonies
- Key Features of Modified P Colonies
- Understanding RAI
- Results
- Simulating Modified P Colonies
- Conclusion

MODIFIED P COLONIES

Structure – one membrane agents in a shared environment.

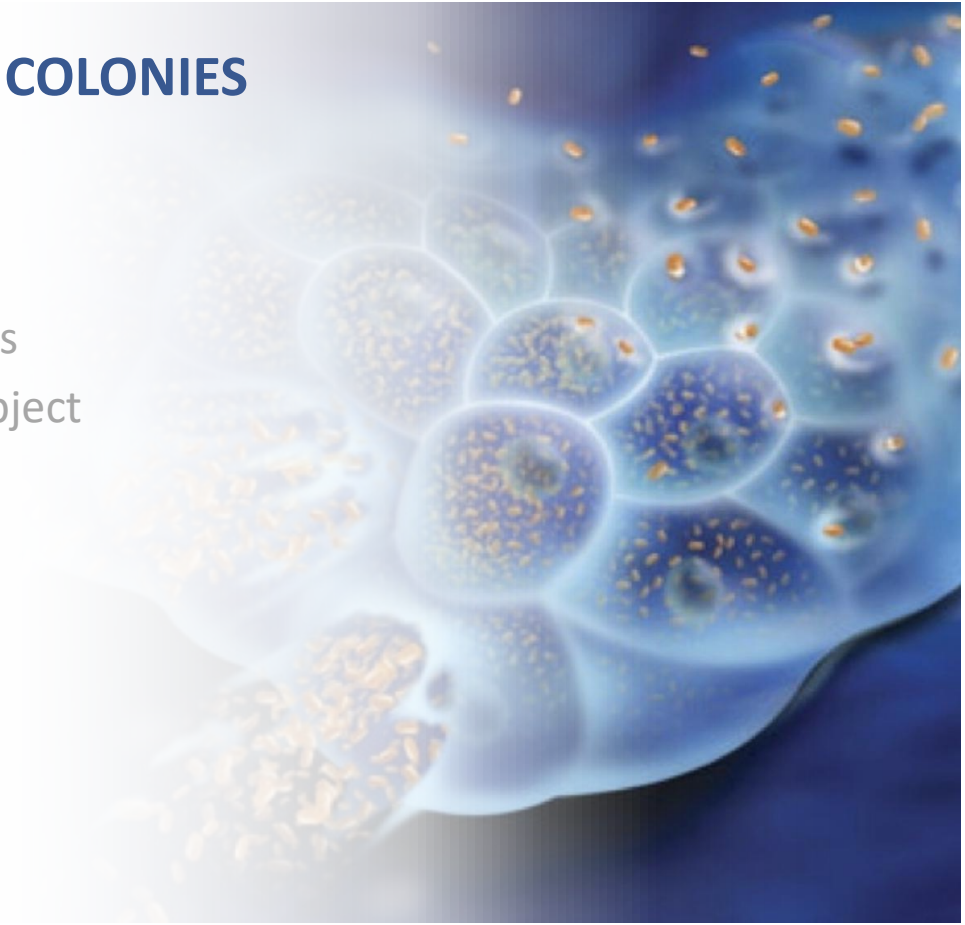
Material – alphabet - set of objects.

Tools – set of rules associated with each agent.



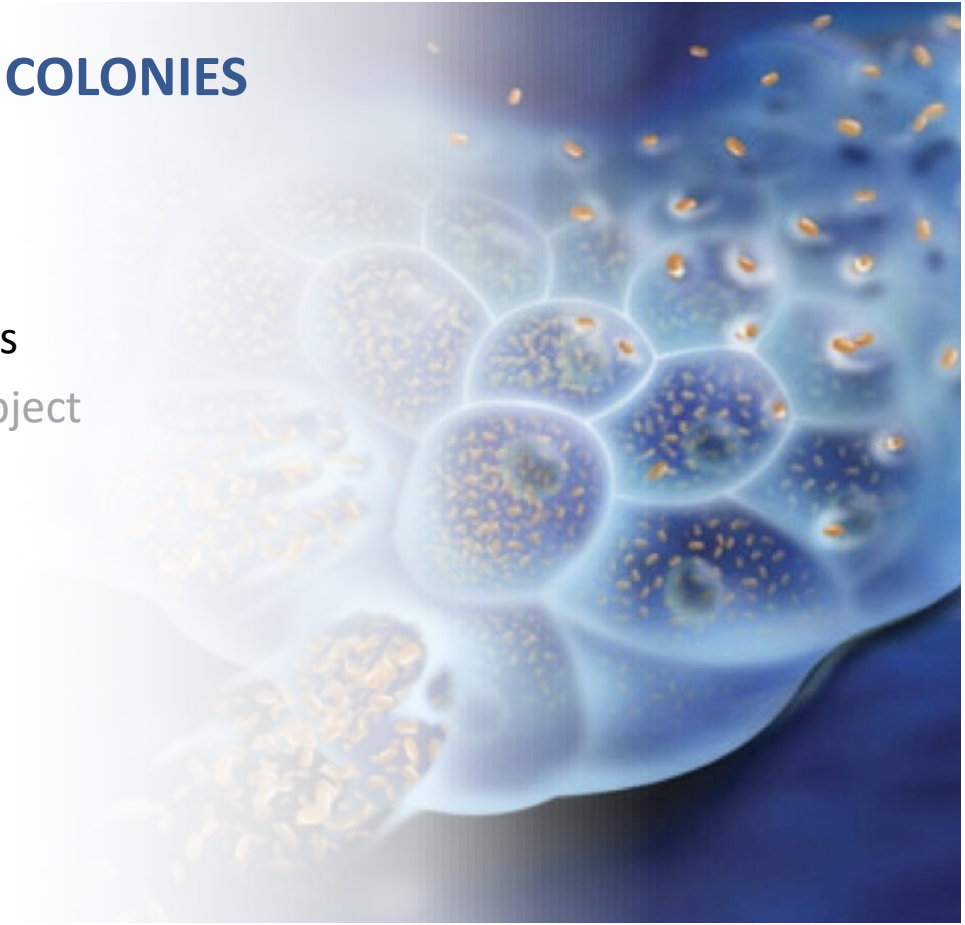
KEY FEATURES OF MODIFIED P COLONIES

- Unbounded agent capacity
- Evolution and communication rules
- λ in rules – creating and erasing object
- Maximally parallel execution



KEY FEATURES OF MODIFIED P COLONIES

- Unbounded agent capacity
- Evolution and communication rules
- λ in rules – creating and erasing object
- Maximally parallel execution



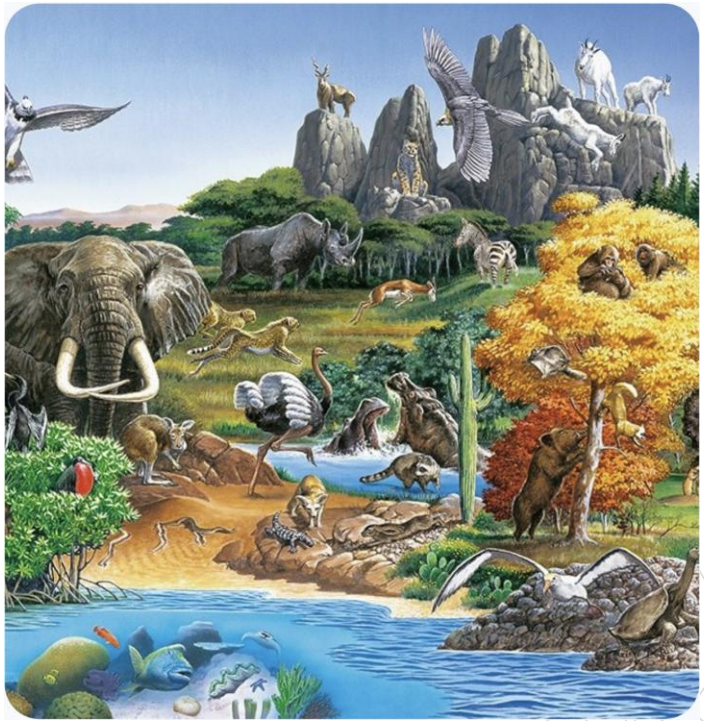
RULES

Evolution rule: **(evo, *a*, *b*, 1.0)**

- agent rewrites object ***a*** to object ***b***

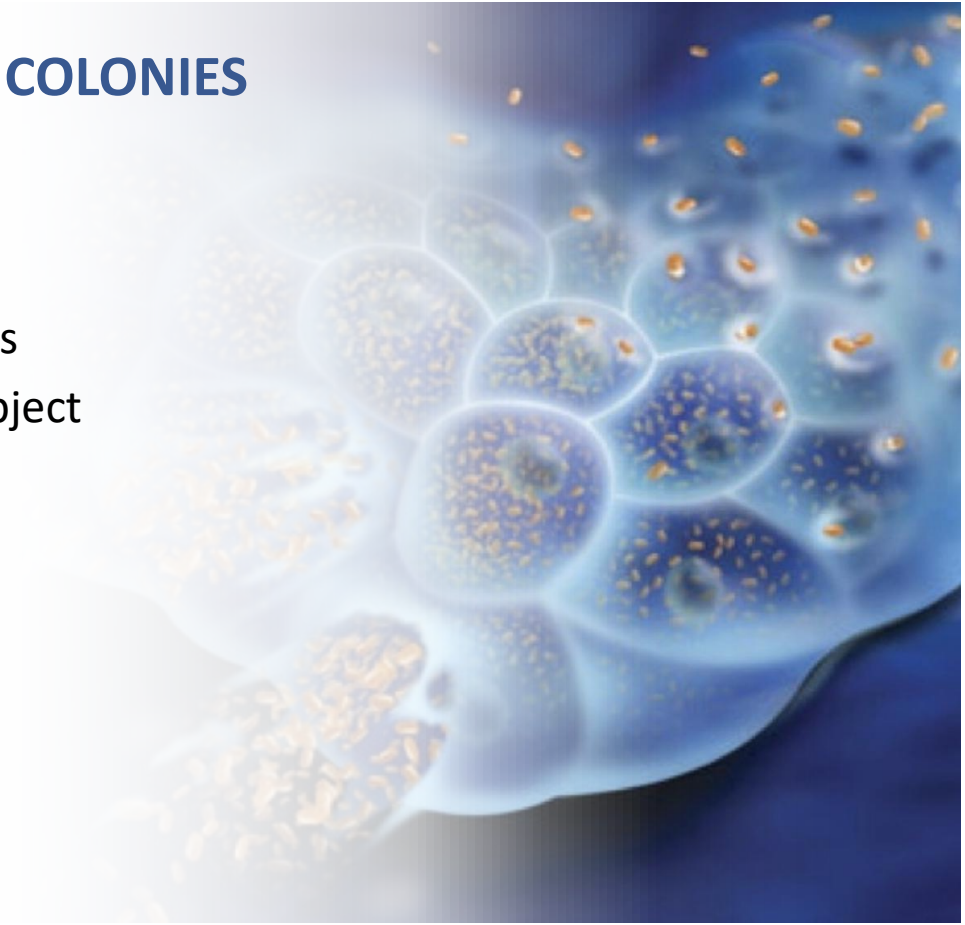
Communication rule: **(com, *c*, *d*, 1.0)**

- agent changes object ***c*** for object ***d*** from environment



KEY FEATURES OF MODIFIED P COLONIES

- Unbounded agent capacity
- Evolution and communication rules
- λ in rules – creating and erasing object
- Maximally parallel execution



λ IN RULES

Evolution rule: (evo, λ, b, 1.0)

- agent creates object b

(evo, a, λ, 1.0)

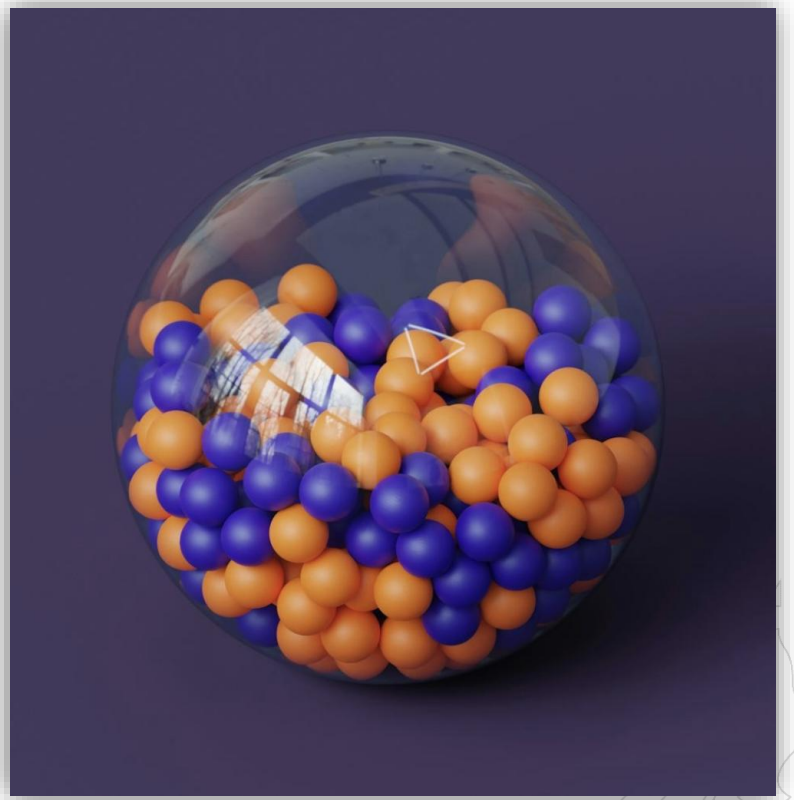
- agent deletes object a

Communication rule: (com, λ, d, 1.0)

- agent consumes object d from the environment

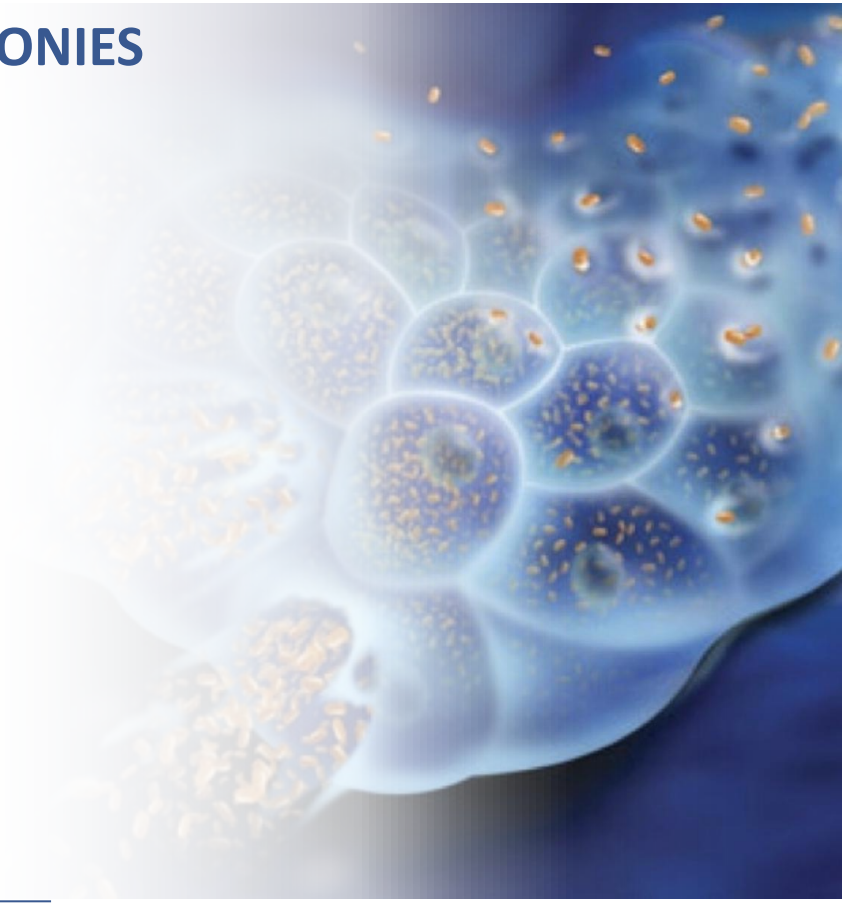
(com, c, λ, 1.0)

- agent puts object c to the environment



KEY FEATURES OF MODIFIED P COLONIES

- Unbounded agent capacity
- Evolution and communication rules
- λ in rules – creating and erasing object
- Maximally parallel execution



UNDERSTANDING RULE APPLICATION INTENSITY (RAI)

Rule Application Intensity (RAI) is a critical aspect of Modified P Colonies. It is defined as a positive real number associated with each rule, dictating the precise number of times a rule must be applied during a computational step.

At each step, the RAI is recalculated based on whether the associated rule was applied, applicable, or neither. There is positive and negative RAI adjustment in the definition of P Colony.

$$\alpha' = \begin{cases} \alpha \oplus \delta^{+} & \text{- rule was executed} \\ \alpha & \text{- rule was applicable but not used} \\ \max(1, \alpha \ominus \delta^{-}) & \text{- rule was not applicable} \end{cases}$$

RESULT OF COMPUTATION

Result is set by definition:

The result is located either within the chosen agent or in the environment.

If a final object is specified, then the result is represented by a number (or numbers), corresponding to the count of these objects in the selected destination. Otherwise, the result is a multiset of objects or set of multisets.

The result can be obtained at the end of the halting computation (option ***hr***), or - ***emr*** - it can be formed gradually by adding partial results after each step into a single outcome, or - ***epr*** - in each computation step of the P Colony, where one result is produced; thus, a single computation generates a subset of the final set of results.

RESULTS

- The class of languages generated by Modified P Colonies with RAI in the emr output mode contains at least one language that is not context-free.

$$\Pi_1 = (A, e, B_1, \omega_E, \oplus\delta^+, \ominus\delta^-, F),$$

With alphabet A={a,e} and one agent having only one rule: (evo, λ, a, 1), RAI adjustments are set to ×2 and ÷2, F={o_1, f=a, emr}, subset of result is obtained during one computation, result is set of numbers – the number of objects a inside the agent B_1.

Result is set of numbers - $\{2^n - 1 \mid n \geq 0\}$

RESULTS

- The class of languages generated by Modified P Colonies with RAI in the emr output mode contains at least one language that is not context-free.

$r = (\text{evo}, \lambda, a, 1)$

Step	w_E (Environment)	w_1 (Agent Content)	ρ_1 (RAI)
0	λ	λ	1
1	λ	a	2
2	λ	aaa	4
3	λ	$aaaaaaa$	8
4	λ	$aaaaaaaaaaaaaaa$	16
5	λ	$aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa$	32

RESULTS

- The class of sets of natural numbers generated by λ-free Modified P Colonies with constant RAI in the emr output mode contains all finite sets of natural numbers.

For example, let $S = \{6, 5, 2, 0\}$ be the finite set of natural numbers. Consider ordering a set from largest to smallest number. We construct P Colony with (6-0) agents. The initial content of the environment contain 6 objects *a*. In each step the number of a in the environment is decreased by execution of rules of type (com, h_i, a, 1), where i corresponds to i-th step of computation.

The number of agents that can use this rule corresponds to difference between two consecutive numbers.

RESULTS

- The class of sets of natural numbers generated by λ-free Modified P Colonies with constant RAI in the emr output mode contains all finite sets of natural numbers.

For example, let $S = \{6, 5, 2, 0\}$ be the finite set of natural numbers. Consider ordering a set from largest to smallest number. We construct P Colony with (6-0) agents. The initial content of the envir
the environment is decrea
corresponds to i-th step of

The number of agents that
consecutive numbers.

Step	Environment w_E	Contents of Agents
0	$aaaaaa$	$[h_1, h_1, h_1, h_1, h_1, h_1]$
1	$aaaaa$	$[a, h_2, h_2, h_2, h_2, h_2]$
2	aa	$[a, a, a, a, h_3, h_3]$
3		$[a, a, a, a, a, a]$

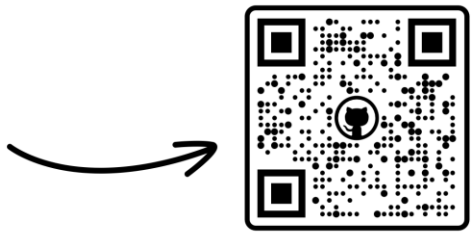
RESULTS

- The class of sets of natural numbers generated by λ-free Modified P Colonies with constant Rule Application Intensity in the hr output mode equals NRE.

We constructed Modified P Colony wit constant RAI and with $r + 2|\text{sub-instructions}|+1$ agents simulating any computation of register machine.

SOFTWARE

- We are developing software that will allow us to run simulations of computations in a Modified P Colony working in all final modes.
- This will allow us to present the effect of various adjustments on RAI and use the software in teaching,
- and to create and validate simulations of real processes and make prediction of their future development.



CONCLUSION

- We explored computational properties of Modified P Colonies with RAI
- We discussed various forms of result
- Future work will focus on
  - computational power and descriptional complexity of proposed variants of final properties
  - Real-life application

SILESIAN
UNIVERSITY
FACULTY OF PHILOSOPHY
AND SCIENCE IN OPAVA

THANK YOU
FOR YOUR ATTENTION