Queens of the Hill

Artiom Alhazov Sergiu Ivanov
David Orellana-Martín

Institutul de Matematică și Informatică Université d'Évry
Universidad de Sevilla

CMC 2023

Core Wars

Core Wars is a programming game in which two or more programs run in a simulated computer with the goal of terminating every other program and surviving as long as possible.

corewars.org

MARS: the Core Wars environment

Common memory space

Only instructions

↳ data is part of the instructions

Language: Redcode

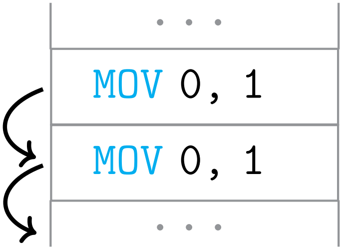
. . .
`MOV 0, 1`
. . .
`ADD #4, 3`
`MOV 2, @2`
`JMP -2`
`DAT #0, #0`
. . .

The Imp: the simplest warrior

`MOV 0, 1`

Relative memory addresses:
↪ 0 current, 1 next.

Program: copy the current instruction to the next address.



Result: The Imp copies itself all over memory.

The Imp does not win

The Imp is good at survival, but bad at killing.
↪ it kills no processes.

The `DAT` instruction kills the current process.

To kill a process, insert `DAT` and make it execute it.

The Dwarf

Bomb the memory with regularly spaced DAT.

```
0: ADD #4, 3
```

Add 4 to instruction 3.
↪ its 2nd argument

```
1: MOV 2, @2
```

Move instruction 1+2 to the value of its @2nd argument.

```
2: JMP -2
```

Jump back 2 instructions.

```
3: DAT #0, #0
```

The Dwarf

Bomb the memory with regularly spaced DAT.

```
0:  ADD  #4,  3
1:  MOV  2,  @2
2:  JMP  -2
3:  DAT  #0,  #4
4:  ...
5:  ...
6:  ...
7:  DAT  #0,  #4
8:  ...
```

Add 4 to instruction 3.
↪ its 2nd argument

Move instruction 1+2 to the value of its @2nd argument.

Jump back 2 instructions.

DAT at addresses not dangerous to the Dwarf.

Core Wars Tournaments

King of the hill mode:

- 10–30 warriors
- sequentially interleaving runs
- score $= f($ number of killed rivals $)$

Highest score: current king of the hill

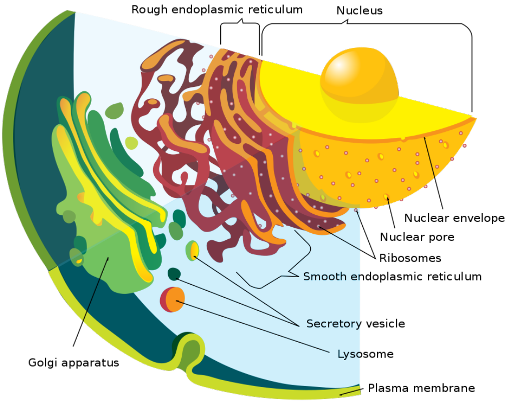
Lowest score:

1. push off the hill
2. replace by the next contestant

So what?

P systems vs. Life

Inspired by the eukaryotic cell | Decentralized computing



$a \to aa$
$a \to (a, \mathrm{out})$

a

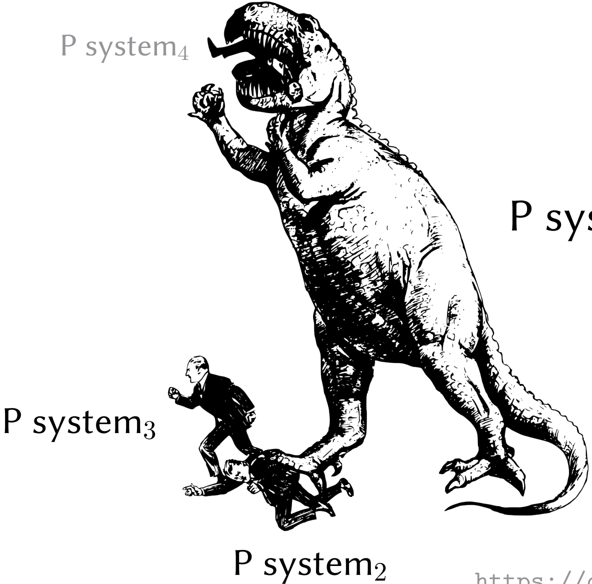
1

$a \to b$
$b \to c\,(c, \mathrm{in})$

0

Use P systems as a tool for thinking about Life.

P systems vs. Evolution



https://openclipart.org/

Queens of the Hill

Run tournaments between P systems

Valkyries

Valkyries: expectations of the formalism

- Ease of interaction in a group of valkyries.
- Ease of programming individual valkyries.

$$\Downarrow$$

Transition P systems with communication and anti-matter.

Valkyrie P systems

$$\Pi = (O, \mu, w_1, \dots, w_n, R_1, \dots, R_n)$$

- $O = \Sigma \cup \Delta_k$: the finite alphabet of objects
- $\Delta_k = \{\delta_t, \bar{\delta}_t \mid 1 \leq t \leq k\} \cup \{\delta\}, k \in \mathbb{N}$: the dissolution timers
- μ: the hierarchical membrane structure
- w_i: the initial multiset in membrane i
- R_i: the finite set of rules in membrane i

Rule types

1. Full cooperation: $abc \to xyz$
2. Target indications: $Tar = \{in, here, out\}$:

   $ab \to (c, out)(d, in)$
3. Membrane dissolution: $ab \to \delta$
   - all objects and the inner membranes fall through to the parent membrane
4. Dissolution timers: $\delta_t \to \delta_{t-1}, \delta_1 \to \delta$
5. Anti-matter annihilation for Δ_k: $\delta_t\bar{\delta}_t \to \lambda$.
   - weak priority for annihilation
   - Δ_{k} forbidden in normal LHS.

Computations and not halting

Usual semantics:

1. Apply the rules in the maximally parallel manner.
2. Perform the dissolutions.

Halting configurations: no more applicable rules.

Don't care about halting: continuous computation

↪ like in VAS, Lindenmayer systems, and Core Wars

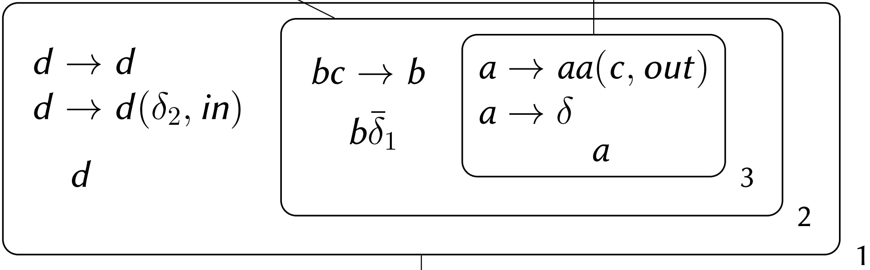
Example

annihilation rules not shown

1

- double some a and eject c
- dissolve the membrane

- b progressively erases c



- maintain d
- maintain d and inject δ_2

Example annihilation rules not shown

2

$d \to d$

$d \to d(\delta_2, in)$

d

$bc \to b$

$b\bar{\delta}_1$

$a \to aa(c, out)$

$a \to \delta$

a

3

2

1

- First δ_2 in membrane 2 will become δ_1 and annihilate with $\bar{\delta}_1$.
- Second δ_2 will dissolve membrane 2.

Example

annihilation rules not shown

3

$$d \to d$$
$$d \to d(\delta_2, in)$$
$$d$$

$$bc \to b$$
$$b\bar{\delta}_1$$

$$a \to aa(c, out)$$
$$a \to \delta$$
$$a$$

3

2

1

Possible evolutions:

dissolved	w_1	w_2	w_3
$\emptyset$	d	$b\bar{\delta}_1 c^*$	a^{2^k}
2	$dbc^*\{\delta_2, \delta_1, \lambda\}$		a^{2^k}
3	d	$b\bar{\delta}_1 c^* a^*$	
$2, 3$	$dbc^* a^*\{\delta_2, \delta_1, \lambda\}$		

Tournaments

Formal definition   inspired by P colonies

$$\mathcal{Q} = \Big[\; \big[\, \Pi_1 \, \big]_{(1,1)} \; \cdots \; \big[\, \Pi_m \, \big]_{(m,1)} \quad \{a \to (a, in) \mid a \in O\} \; \Big]_0$$

- Common alphabet $O = \Sigma \cup \Delta_k$ for all Π_i.
- Membrane i in Π_j $\rightsquigarrow$ membrane (j, i).

Tournament semantics

Same semantics for $\mathcal{Q}$ as for individual valkyries.

Dissolving the skin of a valkyrie is allowed $\Rightarrow$ killing.

The skin bounces all symbols back.
$\hookrightarrow$ including δ_k and $\bar{\delta}_k$

The symbols may end up in another valkyrie
$\Rightarrow$ communication by non-determinism.

Tournament organization

1. Run all valkyries in *max* mode for N steps, resolving non-determinism probabilistically.
2. Repeat 1 M times.
3. Compute the score of the valkyrie Π_j based on how many of its membranes were dissolved.

$$\text{score}(\Pi_j) = \frac{1}{|\Pi_j|}\left(|\Pi_j| - \frac{1}{M}\sum_{i=1}^{M} \text{diss}_i(\Pi_j)\right)$$

Tournament scoring

$$\text{score}(\Pi_j) = \frac{1}{|\Pi_j|}\left(|\Pi_j| - \frac{1}{M}\sum_{i=1}^{M}\text{diss}_i(\Pi_j)\right)$$

- $\text{diss}_i(\Pi_j)$: number of membranes of Π_j that were dissolved during i-th run of 1
- $|\Pi_j|$: total number of membranes in Π_j

Computing the score: an example

$$\mathrm{score}(\Pi_j) = \frac{1}{|\Pi_j|}\left(|\Pi_j| - \frac{1}{M}\sum_{i=1}^{M} \mathrm{diss}_i(\Pi_j)\right)$$

Let $|\Pi_j| = 5$, $M = 3$, and suppose 2, 3, and 4 membranes were dissolved:

$$\mathrm{diss}_1(\Pi_j) = 2 \qquad \mathrm{diss}_2(\Pi_j) = 3 \qquad \mathrm{diss}_3(\Pi_j) = 4$$

$$\mathrm{score}(\Pi_j) = \frac{1}{5}\left(5 - \frac{2+3+4}{3}\right) = \frac{2}{5}$$

Tournament parameters

m	10–20	The number of entrants.
N	1000	The length of a computation.
M	50	The total number of computations.
k	5	The maximal value of the index t in δ_t.
$\lvert\Sigma\rvert$	10	The number of working symbols.

Values derived from similar experiments with multi-agent systems $\Rightarrow$ Test and improve!

Computational complexity

Quickly.

Valkyries are computationally complete

Simulate $(p, \mathrm{ADD}(r), q)$ by $p \to q a_r$.

Simulate $(p, \mathrm{SUB}(r), q, s)$:

	Decrement	*Zero test*
1.	$p \to \bar{p}_1 \hat{p}_1$	$p \to \tilde{p}_1 \dot{p}_1$
2.	$\bar{p}_1 a_r \to \bar{p}_2,\ \hat{p}_1 \to \hat{p}_2$	$\dot{p}_1 a_r \to \#,\ \tilde{p}_1 \to \tilde{p}_2$
3.	$\hat{p}_2 \bar{p}_2 \to q,\ \hat{p}_2 \bar{p}_1 \to \#$	$\tilde{p}_2 \dot{p}_1 \to s$

The language is rich enough.

Tournaments are not computations

The proof relies on non-determinism and halting.

No halting in tournaments.

The non-determinism is resolved probabilistically.

$\Rightarrow$ Partial biased coverage of the computation tree.

Efficiency $>$ Expressive power

First valkyries

The Bomber

Bomb around with δ_t.

$$\left[\begin{array}{c} \{a \to a(\delta_t, out) \mid 1 \le t \le k\} \\ a \end{array}\right]_1$$

Number of valkyries $\downarrow$ Efficiency $\downarrow$

- With 2 valkyries, δ_t may come back into the Bomber.

The Bar Bomber

Bomb around with δ_t, but also stock up $\bar{\delta}_t$.

$$\boxed{\begin{array}{c}\{a \to a\,\bar{\delta}_t\,(\delta_t, out) \mid 1 \le t \le k\} \\ a\end{array}}_{\,1}$$

Protects against other Bombers.

Overwhelmed when too many Bombers around.

The Anti-Bomber

Bomb around with δ_t, but also eject $\bar{\delta}_t$.

$$\left[\begin{array}{c} \{a \to a\,(\bar{\delta}_t, out)\,(\delta_t, out) \mid 1 \le t \le k\} \\ a \end{array}\right]_1$$

Protects this valkyrie, but also the other valkyries.

The Delta Wall

If the number of entrants m is known:

$\hookrightarrow$ or the upper bound on m

$$\boxed{\begin{array}{c} a \to a\bar{\delta}_1^{r(m-1)} \\ a\bar{\delta}_1^{r(m-1)} \end{array}}_1$$

Stock up "enough" copies of $\bar{\delta}_1$.

$\hookrightarrow$ r defines "enough"

The 2-layer Onion

Wrap the core valkyrie in layers.

$$a \to a(b, out)$$

$$a$$

2

$$\{a \to (a, out) \mid a \in \Sigma\}$$

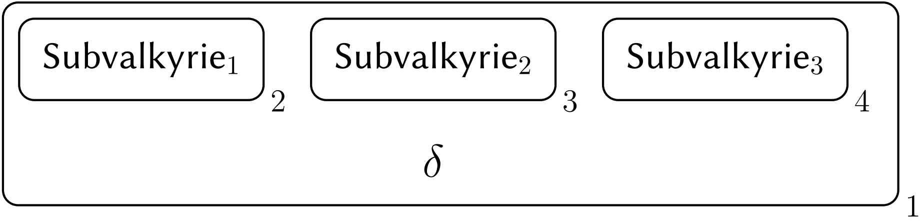
1

How to emit δ_t without dissolving membrane 1?

Destabilize other valkyries by pushing other symbols.

The Bombshell

Release multiple valkyries from a common skin.
↪ multiple charges



Brings potential cooperation.

Costs a membrane dissolution on the score.

Discussion and future work

Tournaments in class

1. Students design valkyries.

   ↪ group work

2. We run tournaments.
3. Students get grades.

A classic in teaching multi-agent systems.

Teaching and research benefits

1. Design P systems.
2. Revise probabilities.
3. Think about survival in an adversarial environment.
   ↪ Potential for thinking about the origins of Life.
4. Promote various formalisms and simulation engines.
   ↪ cP systems, kernel P systems, numerical P systems, spiking neural P systems, etc.

Scoring

$$\mathrm{score}(\Pi_j) = \frac{1}{|\Pi_j|}\left(|\Pi_j| - \frac{1}{M}\sum_{i=1}^{M}\mathrm{diss}_i(\Pi_j)\right)$$

Other scoring functions?

- Better capture the results.
- Avoid trivial edge cases.
- Measure the production of certain symbols?

  ↪ forget dissolution

Core Wars vs. Queens of the Hill

	Core Wars	Queens of the Hill
Data	Secondary	Important
Erasure	Instruction-based	Dissolution
Programs	Mutable	Immutable (fixed rules)
Determinism	Deterministic	Non-deterministic (probabilistic)

$$\mathcal{Q} = \boxed{\boxed{\Pi_1} \quad \cdots \quad \boxed{\Pi_m}}$$

$$\boxed{\begin{array}{c} a \to a(\delta_t, out) \\ a \end{array}} \qquad \boxed{\begin{array}{c} a \to a\,\color{blue}{\bar{\delta}_t}\color{black}\,(\delta_t, out) \\ a \end{array}}$$

$$\boxed{\begin{array}{c} a \to a\bar{\delta}_1^{\color{blue}{r}\color{black}(\color{magenta}{m}\color{black}-1)} \\ a\bar{\delta}_1^{\color{blue}{r}\color{black}(\color{magenta}{m}\color{black}-1)} \end{array}} \qquad \color{blue}{\boxed{\begin{array}{c} \color{black}{\boxed{\begin{array}{c} a \to a(b, out) \\ a \end{array}}} \\ \{a \to (a, out) \mid a \in \Sigma\} \end{array}}}$$

$$\boxed{\delta \quad \boxed{\text{Subvalkyrie}_1} \quad \boxed{\text{Subvalkyrie}_2} \quad \boxed{\text{Subvalkyrie}_3}}$$

Thank you BWMC organizers!