Extended Spiking Neural P Systems Generating Strings and Vectors of Non-Negative Integers

Artiom Alhazov

Rovira i Virgili University and

Academy of Sciences of Moldova

Rudolf Freund

Marion Oswald

Marija Slavkovik

Vienna University of Technology

Extended Spiking Neural P Systems
ESNP Systems

$$\Pi = (m, S, r)$$

- m   number of cells
  (neurons) uniquely identified by a number between 1 and m
- S   initial configuration
  initial value (of spikes) in each neuron
- R   finite set of rules (i,E,k,d;r)

ESNP Systems: Rules

Rules: (i,E/k,d;r) where

- i $\in$ [1..m]   specifies cell i this rule is assigned to
- E $\in$ REG(N) checking set
- k $\geq$ 0 "number of spikes" consumed by this rule
- d $\geq$ 0 delay (omitted when being 0)
- r   set of productions of the form (l,w,t)
  where
  - l $\in$ [1..m] specifies the target cell
  - w $\geq$ 0 weight of energy sent along the axon from neuron i to neuron l,
  - t delay along the axon (omitted when being 0)

ESNP Sytems - Configuration

- for each neuron, the actual number of spikes in the neuron is specified;
- in each neuron i, an "activated rule" (i,E/k,d′;r) may wait to be executed where d′ is the remaining time until the neuron spikes;
- in each axon to a neuron l, we may find pending packages of the form (l,w,t′) where t′ is the remaining time until w spikes have to be added to neuron l provided it is not closed for input at the time this package arrives.

ESNP Sytems – Transition 1

- for each neuron i

  first check whether there is an "activated rule" (i,E/k,d′;r) waiting to be executed;

  if d′=0, then neuron i "spikes", i.e.,
  for every production (l,w,t) in r the corresponding package (l,w,t) is put on the axon from neuron i to neuron l, and after that, this "activated rule" (i,E/k,0;r) is eliminated;

ESNP Sytems – Transition 2

- for each neuron l,
  now consider all packages (l,w,t') on axons leading to neuron l;
  provided the neuron is not closed, all weights w in such packages where $t'=0$ are summed up and added to the corresponding number of spikes in neuron l;

  in any case, the packages with $t'=0$ are eliminated from the axons, whereas for all packages with $t'>0$, t' is decremented by one;

ESNP Sytems – Transition 3

- for each neuron i,

  check again whether there is an "activated rule" (i,E/k,d′;r) (with d′>0) or not;

  - NO: any rule (i,E/k,d;r) from R for which the current number of spikes in the neuron is in E can be applied (and then put a copy of this rule as "activated rule" for this neuron into the description of the current configuration);
  - YES: replace d′ by d′-1 and keep (i,E/k,d′-1;r) as the "activated rule" in neuron i in the description of the configuration for the next step of the computation.

ESNP Sytems – SNP Systems

In the original model, in the productions (l,w,t) of a rule (i,E/k,d;{ (l,w,t) }), only w=1 (for spiking rules) or w=0 (for forgetting rules) as well as t=0 was allowed (and for forgetting rules, the checking set E had to be finite and disjoint from all other sets E in neuron i). Moreover, reflexive axons, i.e., leading from neuron i to neuron i, were not allowed.

Yet the most important extension is that different rules for neuron i may affect different axons leaving from it whereas in the original model the structure of the axons (called synapses there) was fixed.

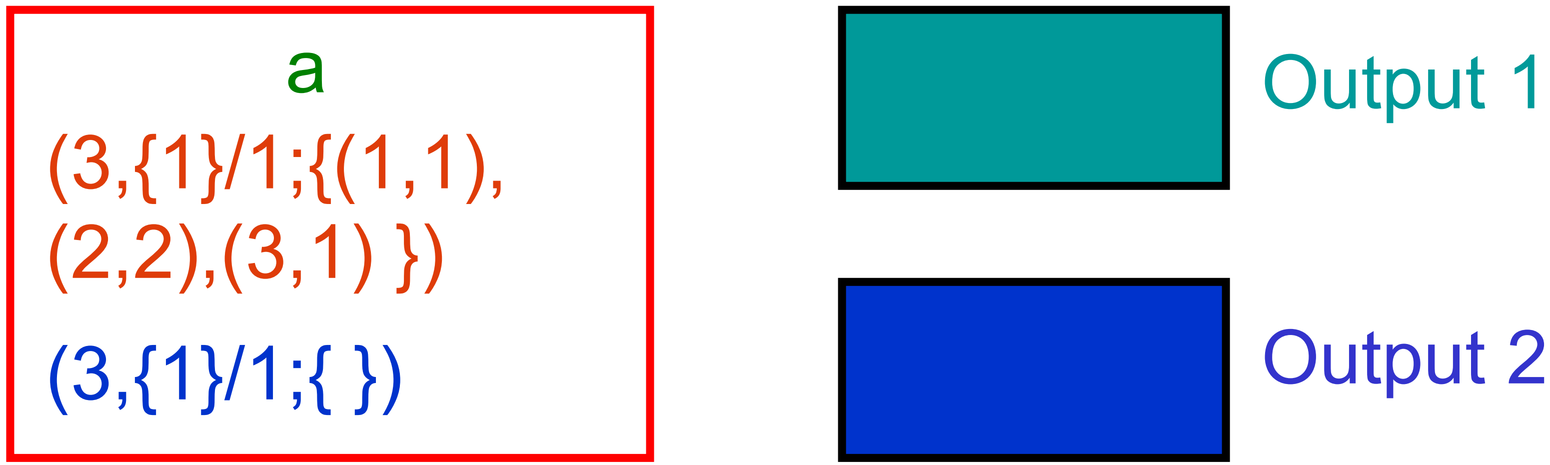
A First Example

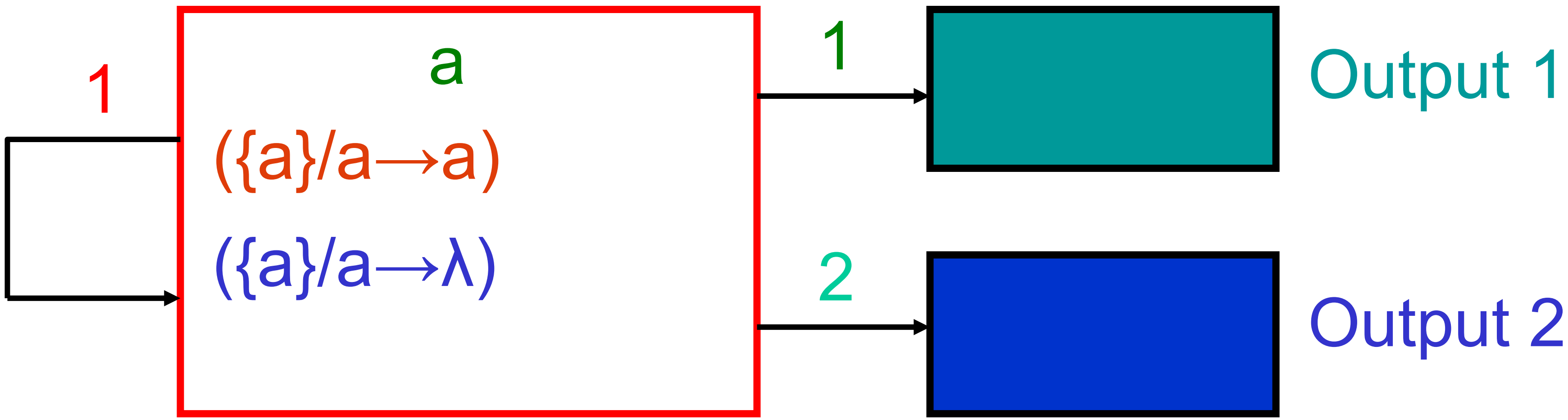
$\{(m,2m) \mid m \geq 0\}$ can easily be generated by an ESNP system *in real time* with only one *actor* neuron and two ouput neurons.

Other than in SNP systems

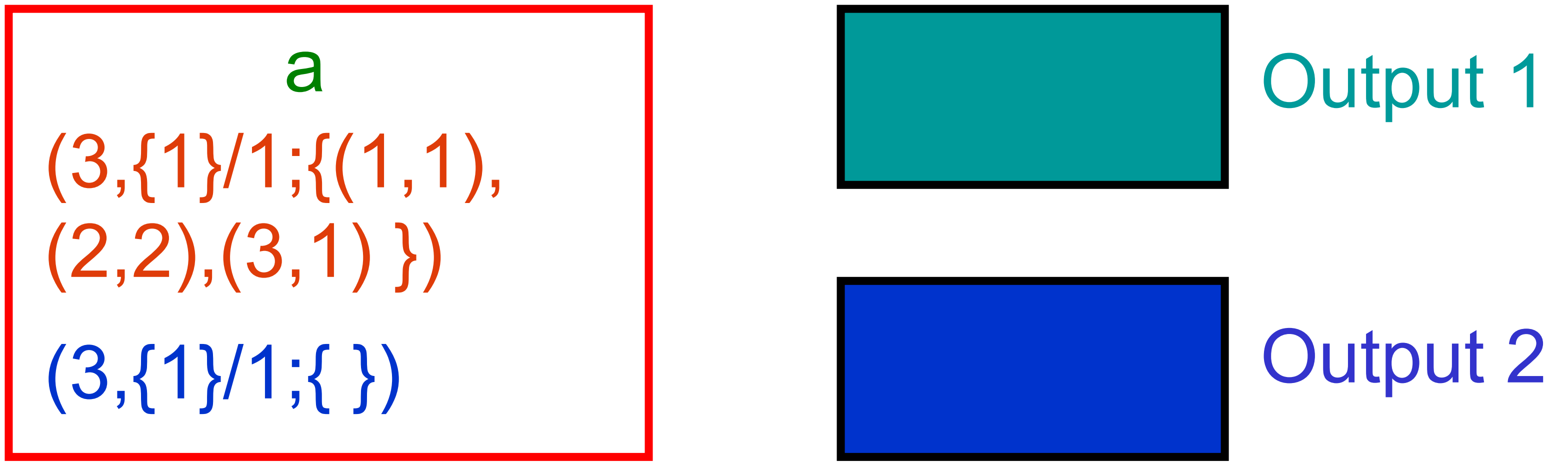
- at different times different numbers of spikes can be sent to various neurons.
- the output is taken as the number of spikes present in the output neurons at the end of a halting computation (and not as the time between the first two spikes in each output neuron).

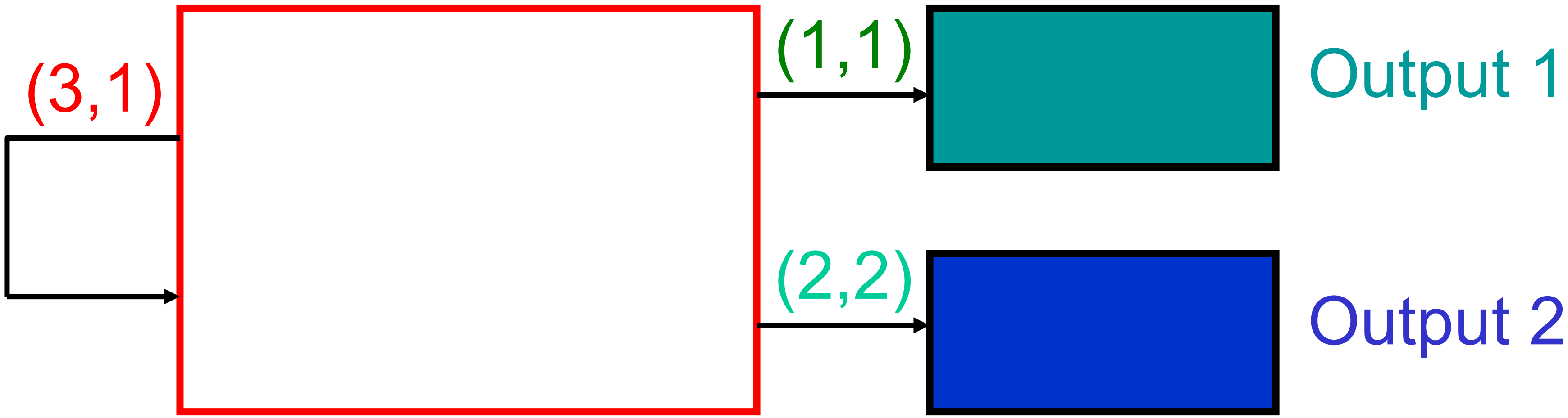
A First Example – ESNP System





A First Example – Computation





A First Example – Computation

a

(3,{1}/1;{(1,1),
(2,2),(3,1) })

(3,{1}/1;{ })

Output 1

Output 2

a

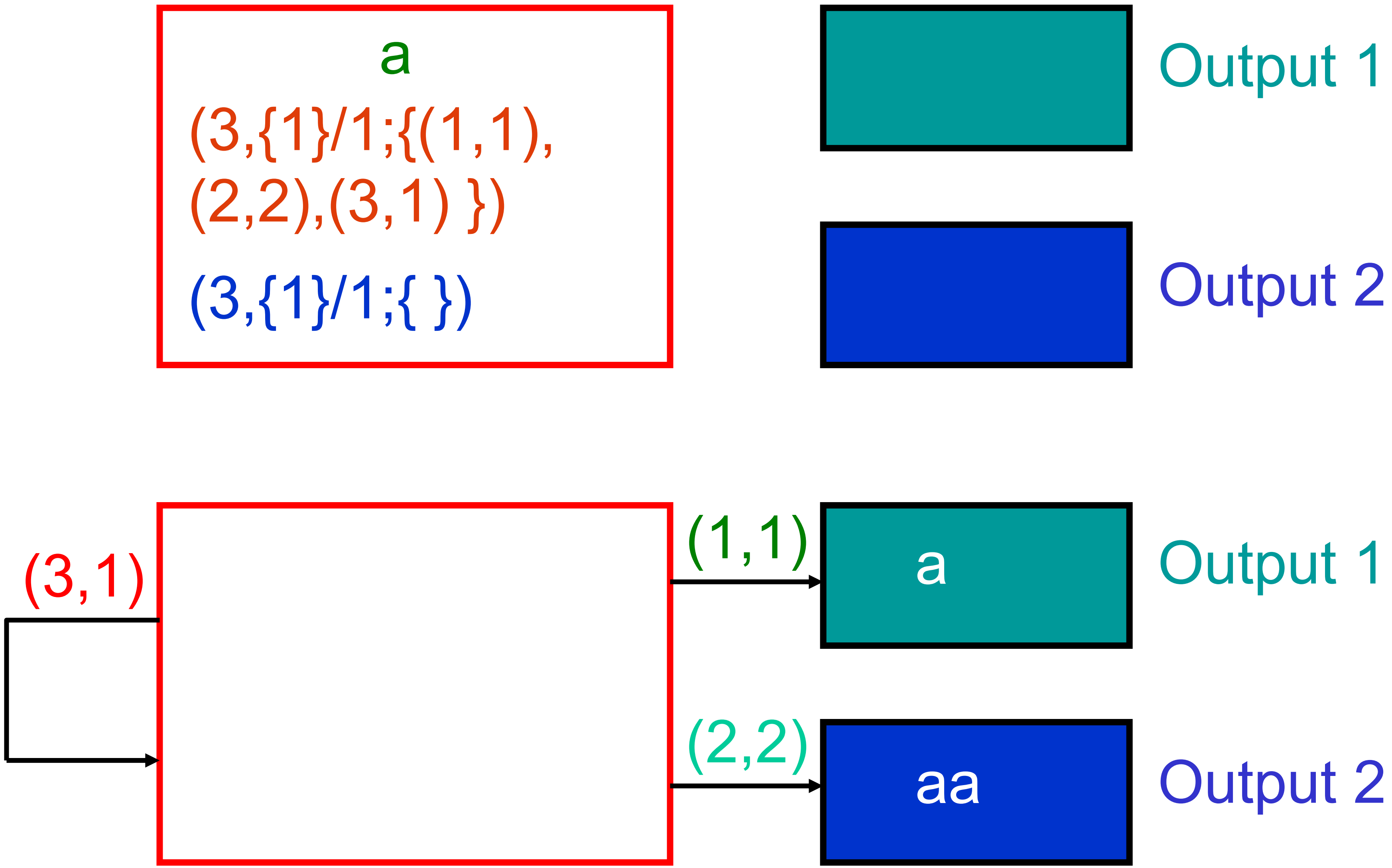
a

Output 1

aa

Output 2

A First Example – Computation



A First Example – Computation

a

(3,{1}/1;{(1,1),
(2,2),(3,1) })

(3,{1}/1;{ })

Output 1

Output 2

a

aa

Output 1

aaaa

Output 2

A First Example – Computation

a

(3,{1}/1;{(1,1),
(2,2),(3,1) })

(3,{1}/1;{ })

Output 1

Output 2

aa

Output 1

aaaa

Output 2

Example : Characterization of Regular Sets of Non-negative Integers

A set of non-negative integers M is semilinear if and only if it can be generated by a finite (or bounded) ESNP system with only two neurons.

Let M be a semilinear set of non-negative integers generated by the regular grammar $G = (N,\{a\},A_1,P)$, $N = \{A_i \mid 1 \leq i \leq m\}$, the start symbol A_1, and P the set of regular productions of the forms $B \rightarrow aC$ and $A \rightarrow \lambda$ with $A,B,C \in N$.
We now construct the finite ESNP system $\Pi = (2,S,R)$ that generates an element of M by the number of spikes contained in the output neuron 1 at the end of a halting computation:

Example : Characterization of Regular Sets of Non-negative Integers

We start with one spike (representing the start symbol A_1 in neuron 2 (which keeps track of the actual non-terminal symbol) and no spike in the output neuron 1, i.e., $S = \{(1,0),(2,1)\}$. The production $A_i \rightarrow aA_j$ is simulated by the rule $(2,\{i\}/i;\{(1,1),(2,j)\})$ and $A_i \rightarrow \lambda$ is simulated by the rule $(2,\{i\}/i;\{\ \})$, i.e., in sum we obtain

$$\Pi = (2,\{(1,0),(2,1)\},R),$$

$$R = \{(2,\{i\}/i;\ \{(1,1),(2,j)\}) \mid 1 \leq i,j \leq m,\ A_i \rightarrow aA_j \in P\}$$

$$\cup\ \{(2,\{i\}/i;\{\ \}) \mid 1 \leq i \leq m,\ A_i \rightarrow \lambda \in P\}.$$

Example : Characterization of Regular Sets of Non-negative Integers

We can also generate the numbers in M as the difference between the (first) two spikes arriving in the output neuron by the following ESNP system Π':

$$\Pi' = (2,\{(1,0),(2,m+1)\},R'),$$

$$R' = \{(2,\{m+1\}/m;\{(1,1),(2,1)\})\}$$

$$\cup \{(2,\{i\}/i;\{(2,j)\}) \mid 1 \leq i,j \leq m,\ A_i \rightarrow aA_j \in P\}$$

$$\cup \{(2,\{i\}/i;\{(1,1)\}) \mid 1 \leq i \leq m,\ A_i \rightarrow \lambda \in P\}.$$

Results: Bounded/Finite ESNP Systems

Lemma: For any ESNP system where during a computation only a bounded number of spikes occurs in the actor neurons, the generated language is regular.

The question whether in an ESNP system during a computation only a bounded number of spikes occurs in the actor neurons is not decidable, but:

Theorem: A language L with $L \subseteq T^*$ for a terminal alphabet T with $card(T)=n$ is regular if and only if it can be generated by a finite ESNP system with $n+1$ neurons.

Results: Bounded/Finite ESNP Systems

Corollary: A set of n-dimensional vectors is semilinear if and only if it can be generated by a finite ESNP system with n+1 neurons.

The preceding results depend on the way the output is taken – as the number of spikes present in the output neurons at the end of a halting computation and not as the time between the first two spikes in each output neuron.

Results: Unbounded ESNP Systems

Theorem: Any recursively enumerable language L with $L \subseteq T^*$ for a terminal alphabet T with card(T)=n can be generated by an ESNP system with n+3 neurons.

Corollary: Any recursively enumerable set of n-dimensional vectors can be generated by an ESNP system with n+3 neurons.

Ideas for Future Research

- investigate variants of definitions and/or restrictions allowing for the characterization of families (of vectors of non-negative integers or strings) between the families of regular and recursively enumerable sets
- consider inhibiting spikes, i.e., allow negative values for the weights w in the productions (l,w,t); when this package arrives at the target cell l, then it causes this cell to be closed for t time steps (all such negative values are summed up at the moment they arrive at the cell provided it is not closed)

Thank You for Your Attention !