

# Invariant checking for rewriting systems over nested words with data

Ahmed Bouajjani, **Cezara Drăgoi**, Yan Jurski, and Mihaela Sighireanu

LIAFA , CNRS & University Paris 7, France  
175, rue du Chevaleret, 75013 Paris, France  
E-mail: [cezara.dragoi@liafa.jussieu.fr](mailto:cezara.dragoi@liafa.jussieu.fr)

# Dynamic Networks of Processes

$$P_1 \quad || \quad P_2 \quad || \quad \dots \quad || \quad P_n$$

- $n$  not fixed: dynamic creation and deletion of processes,
- $P_i$  are infinite state processes that:
  - manipulate data over unbounded complex domains (stacks, lists, multi dimensional arrays...),
  - recursive procedure calls
- $P_i$  synchronize: rendez-vous, broadcast, shared variables.

Challenge: two sources of infinity

- number of processes
- data domains

Verification approach:

- pre/post condition reasoning
- invariants checking

Check that:

- $post(Inv) \subseteq Inv$
- $Inv \subseteq Good$

Verification approach:

- pre/post condition reasoning
- invariants checking

Check that:

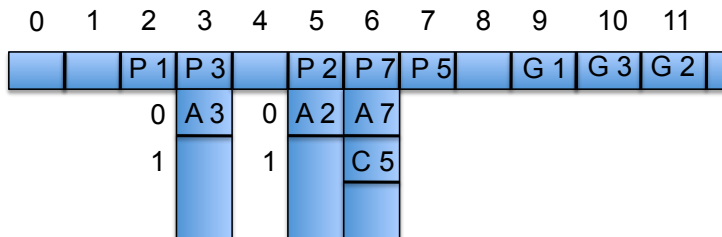
- $post(Inv) \subseteq Inv$
- $Inv \subseteq Good$

We introduce a logic

- **N**ested **D**ata **W**ord **L**ogic  
to represent sets of states
- **N**ested **D**ata **W**ord **R**ewriting **S**ystem  
to model the transitions of the network

- The domain of Nested Data Words
- Nested Data Word Logic
- Decidability result
- Rewriting Systems over Nested Words
- Application to verification

# Representing configurations as Nested Data Words



A word in  $NDW_2$

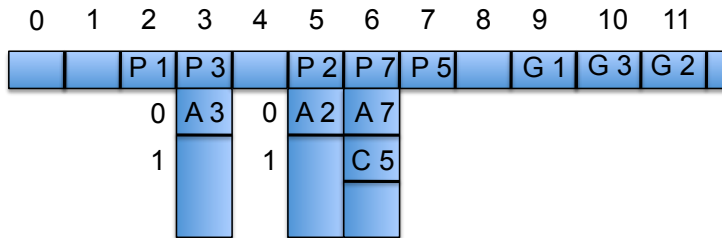
$$\Sigma = \{P, M, A, B, C, G\} \quad \mathbb{D} = \mathbb{Z} \quad NDW_1$$

$$w[6] = (P, 7, [0 \mapsto (A, 7), 1 \mapsto (C, 5)])$$

$$w[6, 0] = (A, 7)$$

$$w[9] = (G, 1, \epsilon)$$

# Representing configurations as Nested Data Words



A word in  $\text{NDW}_2$

- assume  $\mathbb{D}$  the data domain for the values of the (local/global) variables

We define  $\text{NDW}$  to be  $\bigcup_{k \geq 0} \text{NDW}_k$  where:

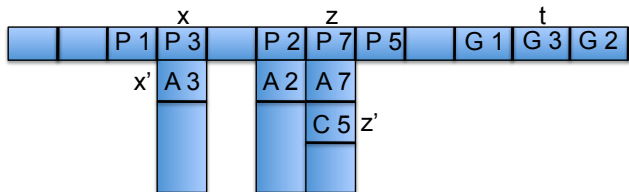
- $\text{NDW}_1 : \{w : \mathbb{N} \rightarrow \Sigma \times \mathbb{D}\}$ ,
- $\text{NDW}_k = \{w : \mathbb{N} \rightarrow \Sigma \times \mathbb{D} \times \text{NDW}_{k-1}\}$  if  $k > 0$ .

- The domain of Nested Data Words
- **Nested Data Word Logic**
- Decidability result
- Rewriting Systems over Nested Words
- Application to verification

- assume  $\text{FO}(\mathbb{D}, \mathbb{O}, \mathbb{P})$  a first order logic on  $\mathbb{D}$ , with operations in  $\mathbb{O}$  and predicates in  $\mathbb{P}$

NDWL is a **logic on NDW** parametrized by  $\text{FO}(\mathbb{D}, \mathbb{O}, \mathbb{P})$

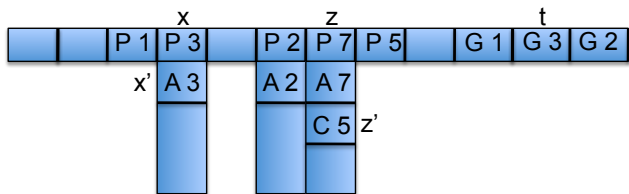
# Label predicates



$w \in \text{NDW}_2$

$P(\gamma[x]) \wedge G(\gamma[t])$

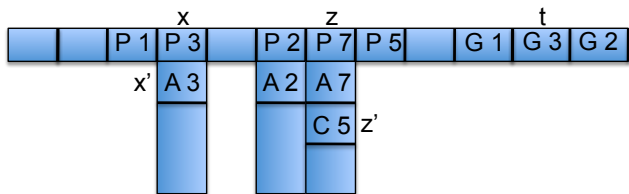
# Label predicates



$w \in \text{NDW}_2$

$$P(\gamma[x]) \wedge G(\gamma[t])$$
$$A(\gamma[x, x']) \wedge C(\gamma[z, z'])$$

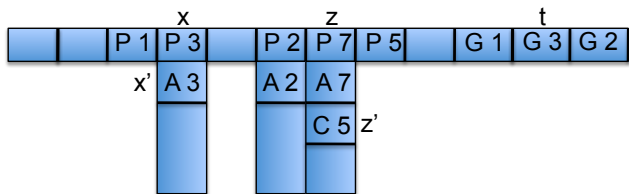
# Label predicates and index constraints



$w \in \text{NDW}_2$

$$P(\gamma[x]) \wedge G(\gamma[t]) \wedge x < z$$
$$A(\gamma[x, x']) \wedge C(\gamma[z, z'])$$

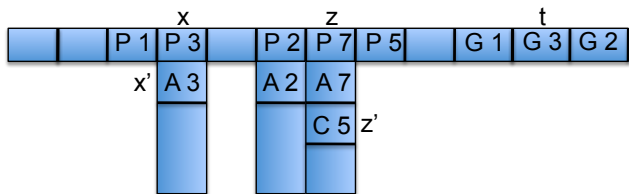
# Label predicates and index constraints



$w \in \text{NDW}_2$

$$P(\gamma[x]) \wedge G(\gamma[t]) \wedge x < z$$
$$A(\gamma[x, x']) \wedge C(\gamma[z, z']) \wedge x' < z'$$

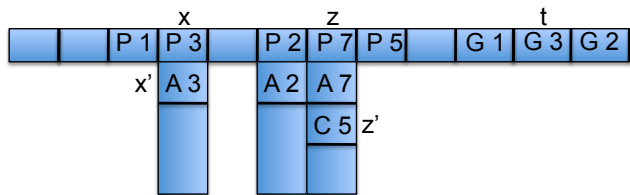
# Label predicates and index constraints



$w \in \text{NDW}_2$

$$\begin{aligned} & \text{idx}(x, \gamma) \wedge P(\gamma[x]) \wedge G(\gamma[t]) \wedge x < z \\ & \text{idx}(x', \gamma[x]) \wedge A(\gamma[x, x']) \wedge C(\gamma[z, z']) \wedge x' < z' \end{aligned}$$

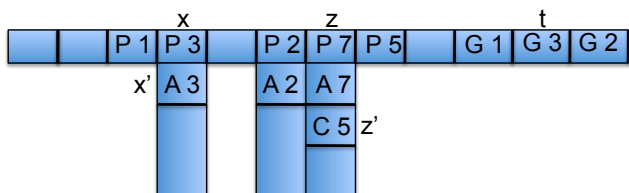
# Data constraints



$w \in NDW_2$

$$\begin{aligned} & idx(x, \gamma) \wedge P(\gamma[x]) \wedge G(\gamma[t]) \wedge x < z \\ & idx(x', \gamma[x]) \wedge A(\gamma[x, x']) \wedge C(\gamma[z, z']) \wedge x' < z' \\ & v(\gamma[x]) = 3 \wedge v(\gamma[z, z']) = 5 \\ & v(\gamma[x]) + d + v(\gamma[z, z']) \geq 17 \end{aligned}$$

# Data constraints



$w \in \text{NDW}_2$

$$\begin{aligned} & \text{idx}(x, \gamma) \wedge P(\gamma[x]) \wedge G(\gamma[t]) \wedge x < z \\ & \text{idx}(x', \gamma[x]) \wedge A(\gamma[x, x']) \wedge C(\gamma[z, z']) \wedge x' < z' \\ & v(\gamma[x]) = 3 \wedge v(\gamma[z, z']) = 5 \\ & v(\gamma[x]) + d + v(\gamma[z, z']) \geq 17 \\ & \delta(\gamma[x]) \neq \delta(\gamma[z]) \end{aligned}$$

- The domain of Nested Data Words
- Nested Data Word Logic
- **Decidability result**
- Rewriting Systems over Nested Words
- Application to verification

The satisfiability problem of NDWL is **undecidable**

- even if we restrict to  $\text{NDW}_1$  for very simple data logics such as  $(\mathbb{N}, =)$ , the fragment  $\forall^* \exists^*$  is undecidable

We define  $\Sigma_2$  as the smallest set of formulas closed under disjunction and conjunction, which contains all the closed formulas of the form:

$$\exists_{\leq k}^* \forall_k^* \exists_{\leq k-1}^* \forall_{k-1}^* \cdots \exists_1^* \forall_1^* \{\exists_d, \forall_d\}^* \cdot \phi$$

$\phi$  is a quantifier-free formula in NDWL

We define  $\Theta_1$  the smallest set of formulas in  $\Sigma_2$  closed under disjunction and conjunction that contains

$$\{\exists_k^*, \forall_k^*\} \{\exists_{k-1}^*, \forall_{k-1}^*\} \cdots \{\exists_1^*, \forall_1^*\} \{\exists_d, \forall_d\}^* \cdot \phi$$

$\phi$  is a quantifier-free formula in NDWL.

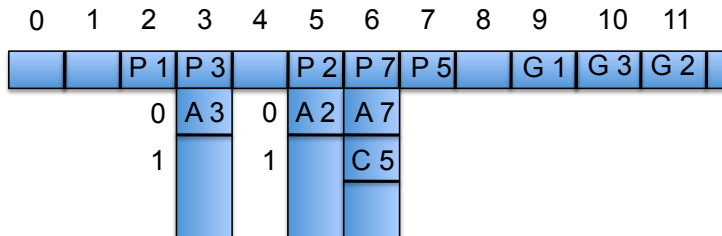
## Theorem

*The satisfiability of NDWL( $\text{FO}(\mathbb{D}, \Omega, \Xi)$ ) formulas in the fragment  $\Sigma_2$  can be reduced to the satisfiability of a formula in  $\text{FO}(\mathbb{D}, \Omega, \Xi)$ .*

## Remark

*The complexity of the reduction procedure is NP when the number of universally quantified variables is fixed.*

# Properties of Dynamic Networks



**Stack:** “in successive calls of some procedure the values of its parameters decreases”

$$\forall y \forall y', z'. P(\gamma[y]) \wedge y' < z' \wedge \text{idx}(y', \gamma[x]) \wedge \text{idx}(z', \gamma[y]) \Rightarrow \nu(\gamma[y, y']) > \nu(\gamma[y, z'])$$

**Data:** “some local process variable is greater then 2 ”

$$\exists x. \text{idx}(x, \gamma) \wedge \nu(\gamma[x]) \geq 2$$

**Control structure:** “all processes are running”

$$\forall y \exists x'. \text{idx}(y, \gamma) \wedge P(\gamma[y]) \Rightarrow \text{idx}(x', \gamma[y])$$

- Nested Data Words
- Nested Data Word Logic
- Decidability result
- **Rewriting Systems over Nested Words**
- Application to verification

## Transitions of the network:

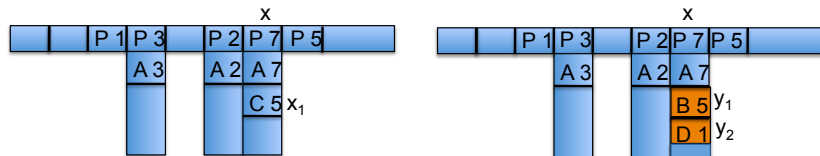
- create/delete process
- local behavior of processes
- communication and synchronization between processes
  - locks
  - shared variables
  - rendez-vous, e.g. *notify()*
  - broadcast, e.g. *notifyAll()*

# Rewriting rules



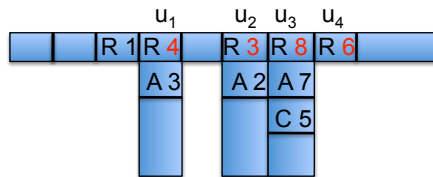
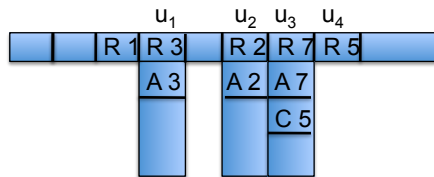
$$R1 : P \hookrightarrow_m P \quad P : \quad v(\gamma[x_1]) = 2 \quad / \quad v(\gamma'[y_1]) = v(\gamma[x_1]) \wedge \delta(\gamma'[y_1]) = \delta(\gamma[x_1]) \wedge v(\gamma'[y_2]) = 2v(\gamma[x_1])$$

# Rewriting rules



$$R2 : C \hookrightarrow_s D : P(\gamma[x]) \wedge v(\gamma[x]) = 7/v(\gamma[x, x_1]) = v(\gamma'[x, y_1]) \wedge v(\gamma'[x, y_2]) = 1$$

# Rewriting rules



$$R3: P \mapsto P' : v(\gamma[u]) \geq 2 / v(\gamma'[u]) = v(\gamma[u]) + 1$$

$S=(\Sigma, \Delta)$  is a NDW-RS where

- $\Sigma$  is a finite set of labels,
- $\Delta$  is a finite set of rewriting rules

$$(\vec{x}, \vec{A}) \hookrightarrow_{\#} (\vec{y}, \vec{B}) : \varphi_g / \varphi_a \quad | \quad (\vec{u}, \vec{C}) \mapsto \vec{D} : \psi_g / \psi_a$$

For every rule  $R \in \Delta$  its semantics is given by a NDW formula:

$$reach_R(\gamma, \gamma') = \varphi_g \wedge \varphi_a \wedge order(\vec{x}, \vec{y}) \wedge \psi_g \wedge \psi_a$$

- Nested Data Words
- Nested Data Word Logic
- Decidability result
- Rewriting Systems over Nested Words
- **Application to verification**

# Invariance checking

We denote by  $\text{NDW-RS}[\Sigma_2]$  the class of NDW-RS s.t. for every rule  $\varphi_g, \varphi_a \in \Sigma_2$  and  $\psi_g, \psi_a$  in  $\Theta_1$ .

## Proposition

*For every rule of a rewriting system in  $\text{NDW-RS}[\Sigma_2]$ , the associated NDWL formula is in the fragment  $\Sigma_2$ .*

## Theorem

*The problem whether a closed formula  $\varphi(\gamma) \in \Theta_1$  is an inductive invariant of  $(S, \varphi_{init})$ , where  $S = (\Sigma, \Delta)$  is in  $\text{NDW-RS}[\Sigma_2]$  and  $\varphi_{init}$  is a closed formula in  $\Sigma_2$  is decidable.*

Using NDW–RS we verified:

- mutual exclusion algorithms : Burns, Richard - Agrawala
- Java-like code with recursive procedure calls and synchronization by monitors

Using NDW–RS we verified:

- mutual exclusion algorithms : Burns, Richard - Agrawala
- Java-like code with recursive procedure calls and synchronization by monitors

Thank You!