Non-interference assessment in bounded Petri nets via Integer Linear Programming

American Control Conference 2018 (ACC 2018)

Francesco Basile and Gianmaria De Tommasi

Milwaukee - 28 June 2018

DIETI UNINA Università degli Studi di Napoli Federico II
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione

Outline

Non-interference

- In system security it is important **to prevent information leaks**

Non-interference

- In system security it is important **to prevent information leaks**
- **Objective:** to prevent to an **intruder** to access to *secret* information

Non-interference

- In system security it is important **to prevent information leaks**
- **Objective:** to prevent to an **intruder** to access to *secret* information
- DES have been used to model different information flow properties
  - opacity (the secret is a state or a sequence)

Non-interference

- In system security it is important **to prevent information leaks**
- **Objective:** to prevent to an **intruder** to access to *secret* information
- DES have been used to model different information flow properties
  - opacity (the secret is a state or a sequence)
  - non-interference

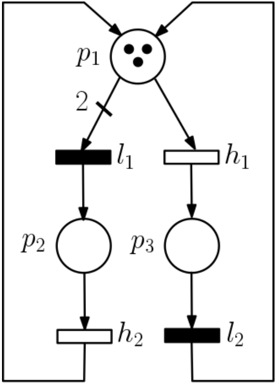
Non-interference

- In system security it is important **to prevent information leaks**
- **Objective:** to prevent to an **intruder** to access to *secret* information
- DES have been used to model different information flow properties
  - opacity (the secret is a state or a sequence)
  - non-interference

Y.-C. Wu and S. Lafortune,
Comparative analysis of related notions of opacity in centralized and coordinated architectures,
Discrete Event Dyn. Syst., vol. 23, no. 3, pp. 307–339, 2013

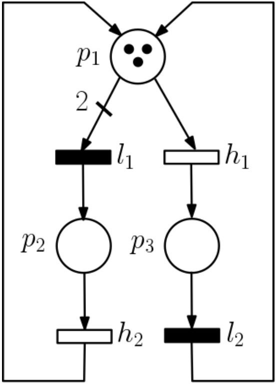
N. Busi and R. Gorrieri,
A survey on non-interference with Petri nets,
Lectures on Concurrency and Petri Nets, pp. 328–344, 2004

Non-interference in PN systems



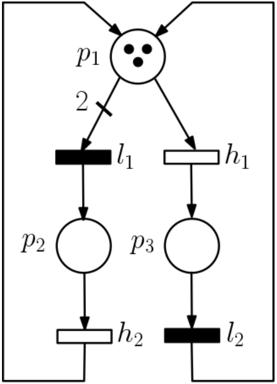
- Two classes of users: **high-level** and **low-level** users

Non-interference in PN systems



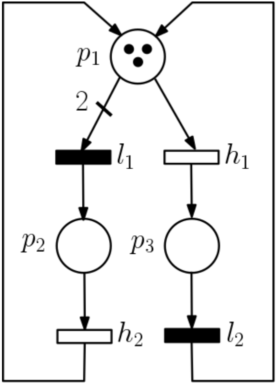
- Two classes of users: **high-level** and **low-level** users
- A leak of information occurs when a low-level user (the **intruder**) obtains information meant to be visible only to high-level users

Non-interference in PN systems



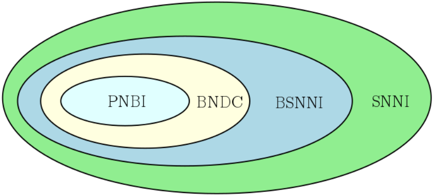
- Two classes of users: **high-level** and **low-level** users
- A leak of information occurs when a low-level user (the **intruder**) obtains information meant to be visible only to high-level users
- Both high-level and low-level users know the system structure, but they interact with the system in two different ways (*views*)

Non-interference in PN systems



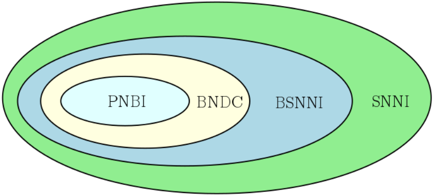
- Two classes of users: **high-level** and **low-level** users
- A leak of information occurs when a low-level user (the **intruder**) obtains information meant to be visible only to high-level users
- Both high-level and low-level users know the system structure, but they interact with the system in two different ways (*views*)
- If the high-level view of the system *interferes* with the low-level one, information leaks may occur

Non-interference properties



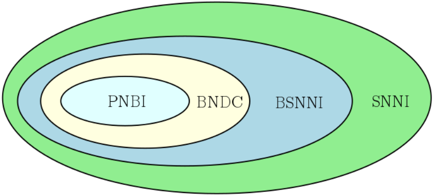
- In a **strong non-deterministic non-interference (SNNI)** the firings of a high-level transition cannot enable any *additional* firing of any low-level transition

Non-interference properties



- In a **strong non-deterministic non-interference (SNNI)** the firings of a high-level transition cannot enable any *additional* firing of any low-level transition
- In a **Bisimulation SNNI (BSNNI)** the firing of a low-level transition cannot disable the firing of any high-level transition

Non-interference properties



- In a **strong non-deterministic non-interference (SNNI)** the firings of a high-level transition cannot enable any *additional* firing of any low-level transition
- In a **Bisimulation SNNI (BSNNI)** the firing of a low-level transition cannot disable the firing of any high-level transition
- More restrictive non-interference properties exist
  - Bisimulation non-deducibility on composition (BNDC)
  - Place-based non-interference (PBNI)

Contribution of this work

- **Two necessary and sufficient conditions** are provided

Contribution of this work

- **Two necessary and sufficient conditions** are provided
  - **to check SNNI** in bounded PNs
  - **to check BSNNI** in bounded PNs

Contribution of this work

- **Two necessary and sufficient conditions** are provided
  - **to check SNNI** in bounded PNs
  - **to check BSNNI** in bounded PNs
- The proposed approach relies on the algebraic representation of the PN dynamic
- The proposed conditions are based on the solution of Integer Linear Programming (ILP) problems

Contribution of this work

- **Two necessary and sufficient conditions** are provided
  - **to check SNNI** in bounded PNs
  - **to check BSNNI** in bounded PNs
- The proposed approach relies on the algebraic representation of the PN dynamic
- The proposed conditions are based on the solution of Integer Linear Programming (ILP) problems
  - *Off-the-shelf* commercial software can be used (e.g., CPLEX, FICO-Xpress)

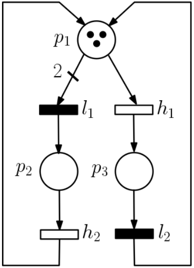
Main assumptions

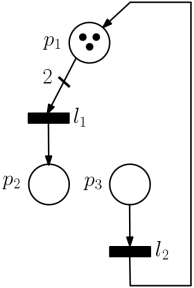
- **Main assumptions**
  - The net system is bounded

Main assumptions

Main assumptions

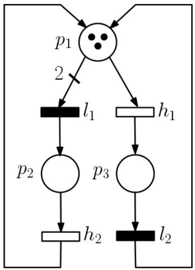
- The net system is bounded
- The *low-level* subnet (subnet *induced* by the low-level transitions) is acyclic

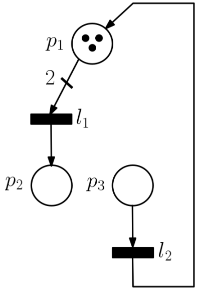




Main assumptions

- **Main assumptions**
  - The net system is bounded
  - The *low-level* subnet (subnet *induced* by the low-level transitions) is acyclic





- *Unnecessary assumptions*
  - the net does not need to belong to any special class (ordinary or safe)

Notation

- The P/T net: $N = (P, L, H, \mathbf{Pre}, \mathbf{Post})$, with $L \cap H = \emptyset$
- The incidence matrix: $\boldsymbol{C} = \mathbf{Post} - \mathbf{Pre}$
- The net system $\mathcal{S} = \langle N, \boldsymbol{m}_0 \rangle$
- Projection of a string on the set of low-view transitions L
  - $Pr_L(\varepsilon) = \varepsilon$
  - $Pr_L(\sigma t) = \begin{cases} Pr_L(\sigma)t & \text{if } t \in L \\ Pr_L(\sigma) & \text{otherwise} \end{cases}$

Notation

- The P/T net: $N = (P, L, H, \mathbf{Pre}, \mathbf{Post})$, with $L \cap H = \emptyset$
- The incidence matrix: $\boldsymbol{C} = \mathbf{Post} - \mathbf{Pre}$
- The net system $\mathcal{S} = \langle N, \boldsymbol{m}_0 \rangle$
- Projection of a string on the set of low-view transitions L
  - $Pr_L(\varepsilon) = \varepsilon$
  - $Pr_L(\sigma t) = \begin{cases} Pr_L(\sigma)t & \text{if } t \in L \\ Pr_L(\sigma) & \text{otherwise} \end{cases}$

  The projection $Pr_L(\cdot)$ can be extended in the usual way to sets of sequences, i.e., if $\Sigma \subseteq (L \cup H)^*$ then

$$Pr_L(\Sigma) = \{Pr_L(\sigma) \mid \sigma \in \Sigma\} .$$

SNNI

Low-view trace equivalence

Two net systems $\mathcal{S}_1$ and $\mathcal{S}_2$ are said to be **low-view trace equivalent**, denoting it by

$$\mathcal{S}_1 \overset{Pr}{\approx}_{tr} \mathcal{S}_2 \,,$$

if and only if

$$Pr_{L_1}\left(\mathcal{L}(N_1\,, \boldsymbol{m}_{0_1})\right) = Pr_{L_2}\left(\mathcal{L}(N_2\,, \boldsymbol{m}_{0_2})\right) \,,$$

where $\mathcal{L}(N_i\,, \boldsymbol{m}_{0_i})$ is the language generate by the i-th net system.

SNNI

SNNI

Let $\mathcal{S} = \langle N \, , \boldsymbol{m}_0 \rangle$ be a net system and $\mathcal{S}_L = \langle N_L \, , \boldsymbol{m}_0 \rangle$ the system defined on the corresponding low-level subnet N_L. $\mathcal{S}$ is said to be **strong non-deterministic non-interference** if and only if

$$\mathcal{S} \overset{Pr}{\approx}_{tr} \mathcal{S}_L \, .$$

SNNI

SNNI

Let $\mathcal{S} = \langle N\, , \boldsymbol{m}_0 \rangle$ be a net system and $\mathcal{S}_L = \langle N_L\, , \boldsymbol{m}_0 \rangle$ the system defined on the corresponding low-level subnet N_L. $\mathcal{S}$ is said to be **strong non-deterministic non-interference** if and only if

$$\mathcal{S} \overset{Pr}{\approx}_{tr} \mathcal{S}_L \, .$$

In a SNNI system, the firings of a high-level transition cannot enable any *additional* firing of any low-level transition

Low-view bisimilarity

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two net systems. A *low-view bisimulation* from $\mathcal{S}_1$ to $\mathcal{S}_2$ is a relation $\mathcal{R}$ on $R(N_1, \boldsymbol{m}_{0_1}) \times R(N_2, \boldsymbol{m}_{0_2})$ such that if $(\boldsymbol{m}_1, \boldsymbol{m}_2) \in \mathcal{R}$, then for all $t \in \bigcup_{i=1,2} L_i \cup H_i$ it is:

1. if $\boldsymbol{m}_1[t\rangle\boldsymbol{m}_1'$ then there exist τ and $\boldsymbol{m}_2'$ such that $\boldsymbol{m}_2[\tau\rangle\boldsymbol{m}_2'$, with $Pr_{L_1}(t) = Pr_{L_2}(\tau)$ and $(\boldsymbol{m}_1', \boldsymbol{m}_2') \in \mathcal{R}$;
2. if $\boldsymbol{m}_2[t\rangle\boldsymbol{m}_2'$ then there exist τ and $\boldsymbol{m}_1'$ such that $\boldsymbol{m}_1[\tau\rangle\boldsymbol{m}_1'$, with $Pr_{L_2}(t) = Pr_{L_1}(\tau)$ and $(\boldsymbol{m}_1', \boldsymbol{m}_2') \in \mathcal{R}$.

$\mathcal{S}_1$ and $\mathcal{S}_2$ are said to be **low-view bisimilar**, denoting it by

$$\mathcal{S}_1 \overset{Pr}{\approx}_{bis} \mathcal{S}_2 ,$$

if and only if there exists a low-level bisimulation $\mathcal{R}$ from $\mathcal{S}_1$ and $\mathcal{S}_2$ such that $(\boldsymbol{m}_{0_1}, \boldsymbol{m}_{0_2}) \in \mathcal{R}$.

BSNNI

Let $\mathcal{S} = \langle N \, , \boldsymbol{m}_0 \rangle$ be a net system and $\mathcal{S}_L = \langle N_L \, , \boldsymbol{m}_0 \rangle$ the system defined on the corresponding low-level subnet N_L. $\mathcal{S}$ is said to be **bisimulation strong non-deterministic non-interference** if and only if

$$\mathcal{S} \overset{Pr}{\approx}_{bis} \mathcal{S}_L \, .$$

BSNNI

Let $\mathcal{S} = \langle N\,, \boldsymbol{m}_0 \rangle$ be a net system and $\mathcal{S}_L = \langle N_L\,, \boldsymbol{m}_0 \rangle$ the system defined on the corresponding low-level subnet N_L. $\mathcal{S}$ is said to be **bisimulation strong non-deterministic non-interference** if and only if

$$\mathcal{S} \overset{Pr}{\approx}_{bis} \mathcal{S}_L\,.$$

- The class of SNNI systems includes the class of BSNNI systems, but the two classes are not equivalent
- In a BSNNI system the firing of a low-level transition cannot disable the firing of any high-level transition

SNNI for bounded net systems

A bounded net system $\mathcal{S} = \langle N, \boldsymbol{m}_0 \rangle$ is SNNI **if and only if** the set of constraints

$$\begin{cases} \boldsymbol{m}_0 + \boldsymbol{C}_L \cdot \hat{\boldsymbol{\sigma}} \geq \mathbf{Pre} \cdot \boldsymbol{s}_1 \\ \boldsymbol{m}_0 + \boldsymbol{C}_L \cdot \hat{\boldsymbol{\sigma}} + \boldsymbol{C} \cdot \boldsymbol{s}_1 \geq \mathbf{Pre} \cdot \boldsymbol{s}_2 \\ \dots \\ \boldsymbol{m}_0 + \boldsymbol{C}_L \cdot \hat{\boldsymbol{\sigma}} + \boldsymbol{C} \cdot \sum_{i=1}^{J-1} \boldsymbol{s}_i \geq \mathbf{Pre} \cdot \boldsymbol{s}_J \\ \boldsymbol{m}_0 + \boldsymbol{C}_L \cdot \hat{\boldsymbol{\sigma}} + \boldsymbol{C} \cdot \sum_{i=1}^{J} \boldsymbol{s}_i \geq \mathbf{0} \\ \sum_{i=1}^{J} \boldsymbol{s}_i(t) = 1 \end{cases} \tag{1}$$

does not admit any solution $\boldsymbol{s}_1, \dots, \boldsymbol{s}_J \in \mathbb{N}^{n_L+n_H}$ **for all** $t \in L$, with $J \geq \mathcal{J}_{\min}$ and $\hat{\boldsymbol{\sigma}}$ being equal to the solution of the ILP problem

$$\begin{aligned} & \max \boldsymbol{\sigma}(t) \\ & \text{s.t.} \\ & \begin{cases} \boldsymbol{m}_0 + \boldsymbol{C}_L \cdot \boldsymbol{\sigma} \geq \mathbf{0} \\ \boldsymbol{\sigma} \in \mathbb{N}^{n_L} \end{cases} \end{aligned} \tag{2}$$

BSNNI for bounded net systems

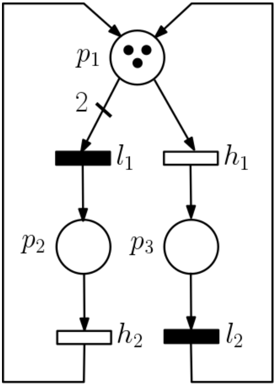
A SNNI bounded net system $\mathcal{S} = \langle N, \mathbf{m}_0 \rangle$ is BSNNI **if and only if** the set of constraints

$$\begin{cases} \mathbf{m}_0 \geq \mathbf{Pre} \cdot \mathbf{s}_1 \\ \mathbf{m}_0 + \mathbf{C} \cdot \mathbf{s}_1 \geq \mathbf{Pre} \cdot \mathbf{s}_2 \\ \dots \\ \mathbf{m}_0 + \mathbf{C} \cdot \sum_{i=1}^{J-1} \mathbf{s}_i \geq \mathbf{Pre} \cdot \mathbf{s}_J \\ \mathbf{m}_0 + \mathbf{C} \cdot \sum_{i=1}^{J} \mathbf{s}_i \geq \mathbf{0} \\ \sum_{i=1}^{J} \mathbf{s}_i(t_L) = \hat{\boldsymbol{\sigma}}(t_L) \\ \sum_{i=1}^{J} \mathbf{s}_i(t_H) = \sum_{i=1}^{J} \bar{\boldsymbol{\sigma}}_i(t_H) \end{cases} \tag{3}$$

admits a solution $\mathbf{s}_1, \dots, \mathbf{s}_J \in \mathbb{N}^{n_L+n_H}$ **for all** $t_L \in L$ **and** $t_H \in H$, with $J \geq \mathcal{J}_{\min}$, $\hat{\boldsymbol{\sigma}}$ being equal to the solution of (2), and $\bar{\boldsymbol{\sigma}}_1, \dots, \bar{\boldsymbol{\sigma}}_J$ equal to the solution of the ILP problem

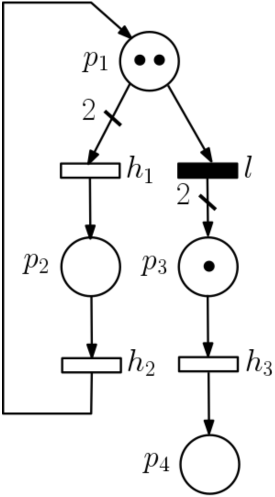
$$\begin{aligned} &\max \sum_{i=1}^{J} \boldsymbol{\sigma}_i(t_H) \\ &\text{s.t.} \\ &\begin{cases} \mathbf{m}_0 \geq \mathbf{Pre} \cdot \boldsymbol{\sigma}_1 \\ \mathbf{m}_0 + \mathbf{C} \cdot \boldsymbol{\sigma}_1 \geq \mathbf{Pre} \cdot \boldsymbol{\sigma}_2 \\ \dots \\ \mathbf{m}_0 + \mathbf{C} \cdot \sum_{i=1}^{J-1} \boldsymbol{\sigma}_i \geq \mathbf{Pre} \cdot \boldsymbol{\sigma}_J \\ \mathbf{m}_0 + \mathbf{C} \cdot \sum_{i=1}^{J} \boldsymbol{\sigma}_i \geq \mathbf{0} \\ \boldsymbol{\sigma}_i \in \mathbb{N}^{n_L+n_H}, \quad i = 1, 2, \dots, J \end{cases} \end{aligned} \tag{4}$$

Example (I)



- By setting $J = 5$, the solution of (2) for the transition $l_1 \in L$ returns $\hat{\sigma}(l_1) = 1$
- Given the solution $\hat{\sigma}$, also the feasibility problem (1) admits a solution
- The net system is NOT SNNI
- The time needed to solve a single instance of the ILP problem (2) and of the feasibility problem (1) is less the 500 μs using GLPK on a MacBook Pro equipped with an Intel® i5 at 3.1 GHz and with 16 GB of RAM

Example (II)



- By setting $J = 5$, in this case (1) does not admit any solution for any $t \in L$,
- The net system is SNNI
- The feasibility problem (3) does not admit a solution as well
- The firing of l prevents the firing of the two high level transitions
- The net system is NOT BSNNI
- About 2 ms are needed to check both SNNI and BSNNI on the considered hardware

Conclusions

- The mathematical representation of Petri nets has been exploited to provide necessary and sufficient conditions to check both SNNI and BSNNI in bounded systems

Conclusions

- The mathematical representation of Petri nets has been exploited to provide necessary and sufficient conditions to check both SNNI and BSNNI in bounded systems
- Possible extensions:
  - **relaxation of the acyclicity assumption on the low-level subnet** (submitted to the next CDC)
  - labeled net systems
  - **non-interference enforcing** (submitted to the next CDC)

Conclusions

- The mathematical representation of Petri nets has been exploited to provide necessary and sufficient conditions to check both SNNI and BSNNI in bounded systems
- Possible extensions:
  - **relaxation of the acyclicity assumption on the low-level subnet** (submitted to the next CDC)
  - labeled net systems
  - **non-interference enforcing** (submitted to the next CDC)
  - **algebraic characterization of opacity in PNs** (WODES 2018)

F. Basile and G. De Tommasi,
An algebraic characterization of language-based opacity in labeled Petri nets,
WODES'18, Sorrento Coast, Italy, May 2018

Questions?