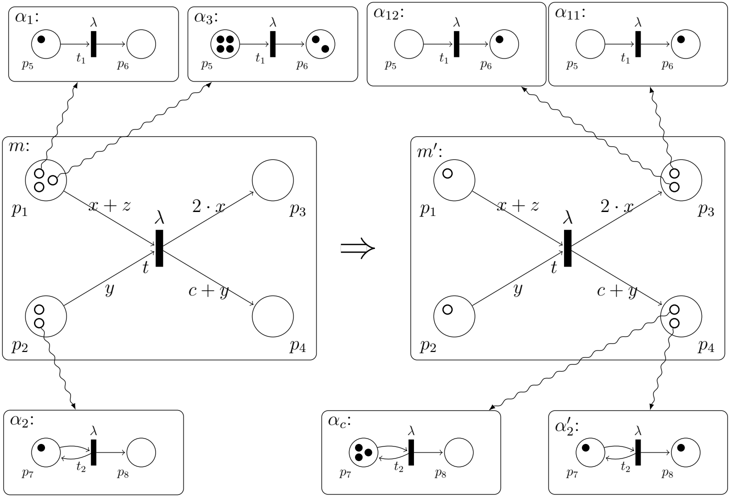
On place invariants of nested Petri nets.

Speaker: Leonid Dworzanski
Scientific advisor: Irina A. Lomazova

National Research University Higher School of Economics

December 8, 2015

Example



α_1:
λ
p_5 t_1 p_6
α_3:
λ
p_5 t_1 p_6
α_{12}:
λ
p_5 t_1 p_6
α_{11}:
λ
p_5 t_1 p_6
m:
p_1 $x + z$ λ $2 \cdot x$ p_3
t
y $c + y$
p_2 p_4
$\Rightarrow$
m':
p_1 $x + z$ λ $2 \cdot x$ p_3
t
y $c + y$
p_2 p_4
α_2:
λ
p_7 t_2 p_8
α_c:
λ
p_7 t_2 p_8
α'_2:
λ
p_7 t_2 p_8

Outline

1. Intro
   - Whats the plan?
   - P-invariants for classical Petri nets
   - Problem Statement
2. Example
3. Definitions
4. Application
5. Application
6. Application

Outline

1 Intro
- Whats the plan?
- P-invariants for classical Petri nets
- Problem Statement

2 Example

3 Definitions

4 Application

5 Application

6 Application

Whats the plan?

- What are P-invariants of classical Petri nets?
- What other types of P-invariants do exist?
- How to define them for NP-nets?

P-invariants for classical Petri nets

- What are P-invariants of classical Petri nets?
  `Interactive tutorials by Wil v.d.Aalst`
  `www.informatik.uni-hamburg.de/TGI/PetriNets/`

Problem Statement

Invariants are defined for almost every Petri net calculus.

- Classical P/T nets;
- CP-nets (M. Schiffer);
- Coloured Petri nets (K. Jensen);
- Algebraic Petri nets (W. Reisig, K. Schmidt);
- NP-nets (not yet)

Need for analysis methods

Theoretical issues:

- Covering problem is decidable (NP-nets are well structured transition systems)

Need for analysis methods

Theoretical issues:

- Covering problem is decidable (NP-nets are well structured transition systems)
- It is possible to model Petri nets w/ reset arcs via NP-nets
- NP-nets are strictly more expressive than Petri nets

Need for analysis methods

Theoretical issues:

- Covering problem is decidable (NP-nets are well structured transition systems)
- It is possible to model Petri nets w/ reset arcs via NP-nets
- NP-nets are strictly more expressive than Petri nets
- Boundedness is undecidable
- Reachability is undecidable
- Liveness is undecidable

What can we do?

Problem Statement

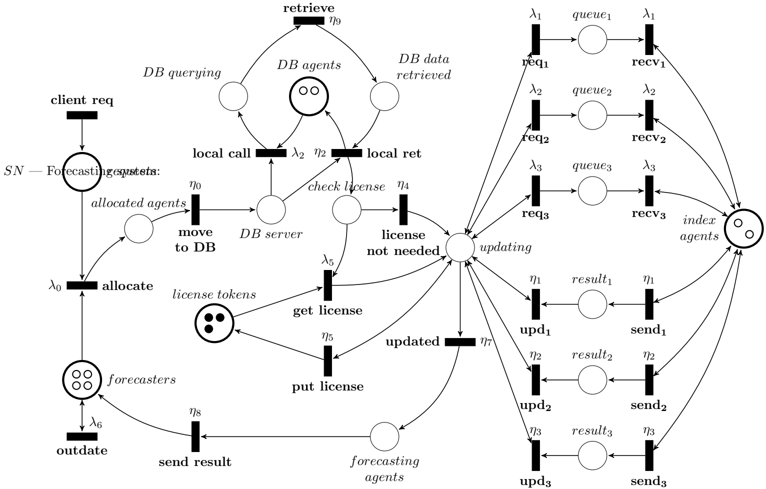
Invariants have to be defined for NP-nets.

- How to define them?
- Shall we take into account different levels?
- Will the invariants of NP-net components be compositional?
- Are such compositions only possible invariants?
- Shall they be number- or term-valued functions?
- etc

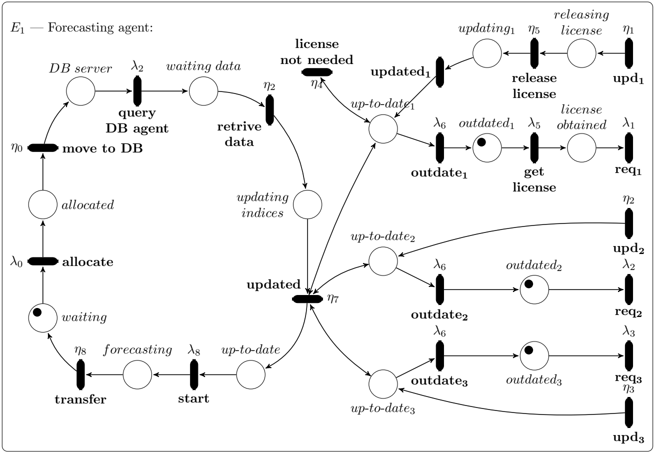
Outline

1. Intro
   - Whats the plan?
   - P-invariants for classical Petri nets
   - Problem Statement
2. Example
3. Definitions
4. Application
5. Application
6. Application

Forecasting system

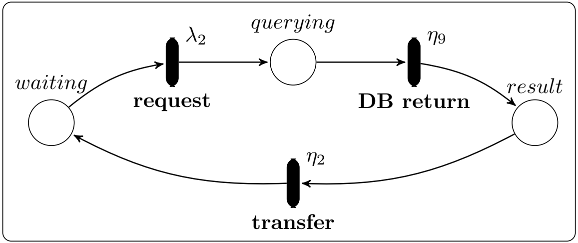


Example

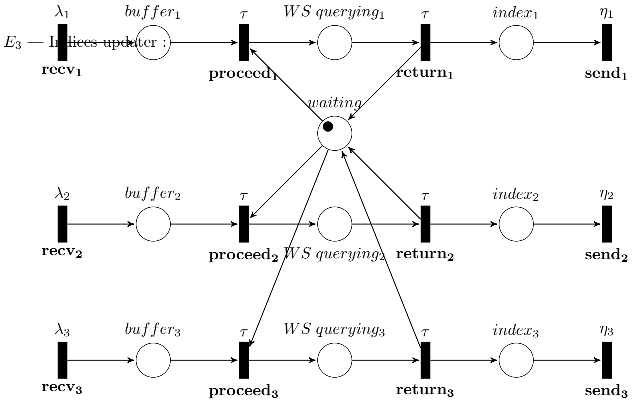

E_1 — Forecasting agent:
DB server
λ_2
waiting data
query DB agent
η_2
retrive data
license not needed
η_4
updating$_1$
η_5
releasing license
η_1
updated$_1$
release license
upd$_1$
up-to-date$_1$
λ_6
outdated$_1$
λ_5
license obtained
λ_1
outdate$_1$
get license
req$_1$
η_0
move to DB
allocated
updating indices
η_2
upd$_2$
up-to-date$_2$
λ_6
outdated$_2$
λ_2
λ_0
allocate
updated
η_7
outdate$_2$
req$_2$
waiting
η_8
forecasting
λ_8
up-to-date
λ_6
λ_3
transfer
start
outdate$_3$
outdated$_3$
req$_3$
η_3
up-to-date$_3$
upd$_3$

Example

E_2 — Database midlet :



Example


E_3 — Indices updater :
λ_1
$buffer_1$
τ
$WS\ querying_1$
τ
$index_1$
η_1
$\mathbf{recv_1}$
$\mathbf{proceed_1}$
$\mathbf{return_1}$
$\mathbf{send_1}$
$waiting$
λ_2
$buffer_2$
τ
τ
$index_2$
η_2
$\mathbf{recv_2}$
$\mathbf{proceed_2}$
$WS\ querying_2$
$\mathbf{return_2}$
$\mathbf{send_2}$
λ_3
$buffer_3$
τ
$WS\ querying_3$
τ
$index_3$
η_3
$\mathbf{recv_3}$
$\mathbf{proceed_3}$
$\mathbf{return_3}$
$\mathbf{send_3}$

Outline

1 Intro
- Whats the plan?
- P-invariants for classical Petri nets
- Problem Statement

2 Example

3 Definitions

4 Application

5 Application

6 Application

NP-net

Definition

Lab is a set of transition labels. $N_1, \ldots, N_k$ are CPNs, where all transitions are labeled with $Lab \cup \{\tau\}$.
NP-net is a tuple $NP = \langle N_1, \ldots, N_k, SN \rangle$, where $N_1, \ldots, N_k$ - element nets, and SN - system net. $SN = \langle P_{SN}, T_{SN}, F_{SN}, \gamma, \Lambda \rangle$

1. constants or multiple instances of the same variable are not allowed in input arc expressions of t;
2. each variable in an output arc expression for t occurs in one of the input arc expressions of t.

Problem Statement

- How to define them?
- What will we gain from the definition?
- How to calculate them?
- What properties do these things have?

NP-net invariant

Definition

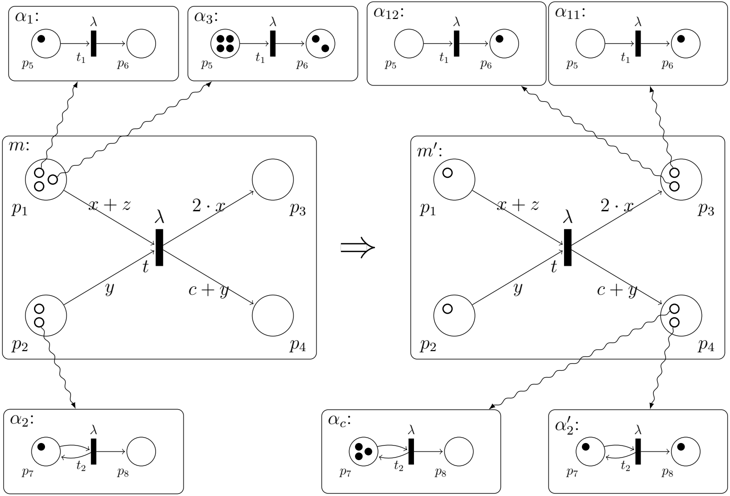
$NP = \langle N_1, \ldots, N_k, SN \rangle$ is an NP-net.
$w_t : P_{SN} \rightarrow \mathbb{Z}$ and $w_m : P_{SN} \rightarrow \mathbb{Z}$ are weight functions.
$\forall p \in P_{SN}$ typed with N, we assign the weight function w_p, that maps P_N into $\mathbb{Z}$.
$\hat{w}_p : \mathcal{M}(N) \rightarrow \mathbb{Z}$ is the linear extension of w_p to the markings of N
$\forall m_p \in \mathcal{M}(N) : \hat{w}_p(m_p) = \sum_{q \in P_N} (m_p(q) w_p(q))$.

The weight function

$$W_{NP}(m) = \sum_{p \in P_{SN}} \sum_{\alpha \in m(p)} (w_t(p) + w_m(p)\hat{w}_p(m_\alpha))$$

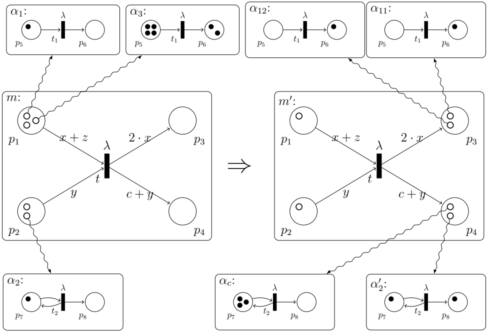
is an invariant of NP-net NP iff
$\forall m \in \mathcal{R}_{NP}(m_0) : W_{NP}(m) = W_{NP}(m_0)$.

Example



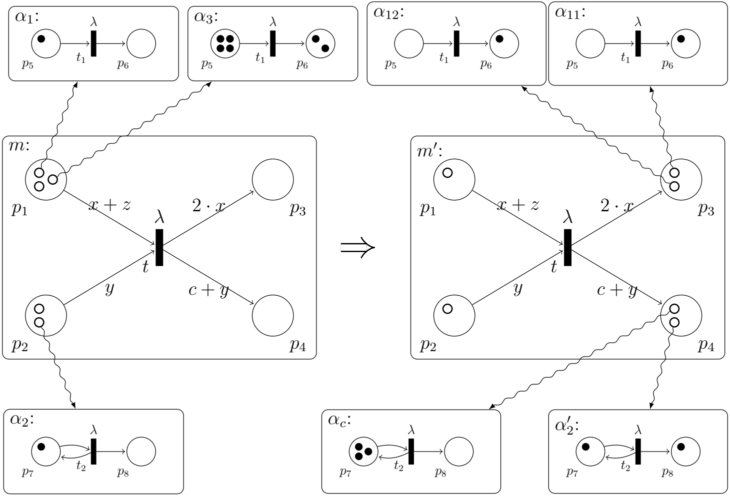
α_1:
λ
p_5 t_1 p_6
α_3:
λ
p_5 t_1 p_6
α_{12}:
λ
p_5 t_1 p_6
α_{11}:
λ
p_5 t_1 p_6
m:
p_1
$x + z$
λ
$2 \cdot x$
p_3
t
y
$c + y$
p_2
p_4
$\Rightarrow$
m':
p_1
$x + z$
λ
$2 \cdot x$
p_3
t
y
$c + y$
p_2
p_4
α_2:
λ
p_7 t_2 p_8
α_c:
λ
p_7 t_2 p_8
α'_2:
λ
p_7 t_2 p_8

Example

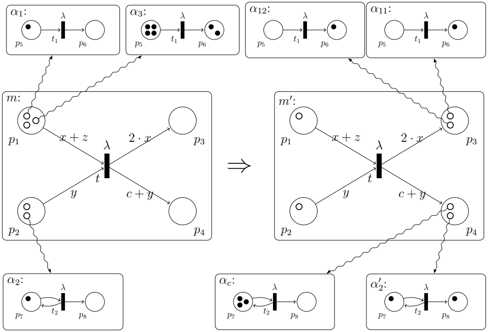


$$w_m(p_{SN})W(p_{SN})(p) = \sum_{q \in t_{SN}^{\bullet}} w_m(q)\|Var(\langle t_{SN}, q\rangle)\|_x W(q)(p) \quad (1)$$

Example



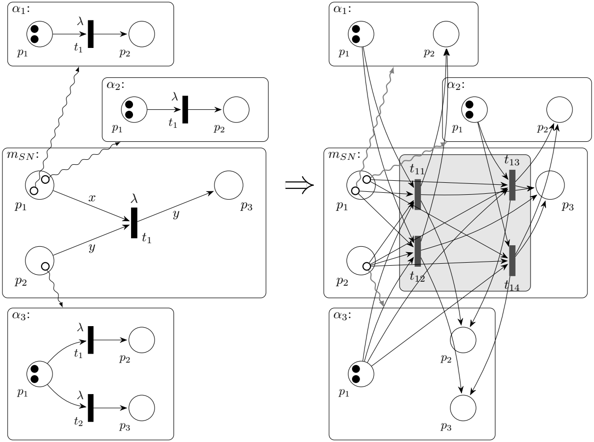
Example



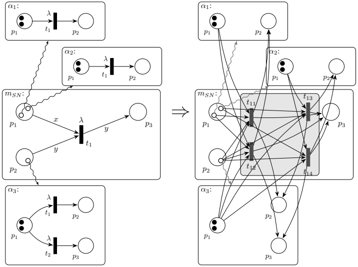
$$\sum_{p\in {}^\bullet t_{SN}} w_t(p)|Var(\langle p, t_{SN}\rangle)| = \sum_{p\in t_{SN}^\bullet} w_t(p)\|Var(\langle t_{SN}, p\rangle)\|$$

$$+w_m(p)\cdot \sum_{c\in Con(\langle t_{SN},p\rangle)} W(p)(c) \quad (2)$$

Example


α1:
λ
p1
t1
p2
α2:
λ
p1
t1
p2
mSN:
p1
x
λ
y
p3
t1
y
p2
α3:
λ
t1
p2
p1
λ
t2
p3
⇒
α1:
p1
p2
α2:
p1
p2
mSN:
t13
t11
p1
p3
p2
t12
t14
α3:
p2
p1
p3

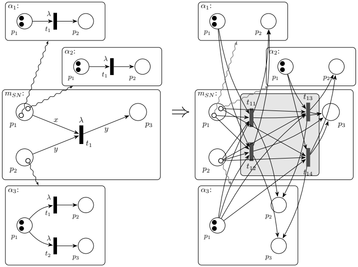
Example



$$\sum_{p \in {}^\bullet t_{SN}} w_t(p) \| Var(\langle p, t_{SN} \rangle) \| +$$

$$\sum_{t_i \in \{t_1, \dots, t_q\}} \sum_{p \in {}^\bullet t_i} w_m(p_{\alpha_i}) W(p_{\alpha_i})(p) \gamma(p, t_i) = \quad (3)$$

Example



$$= \sum_{p \in t_{SN}^{\bullet}} w_t(p) \| Var(\langle t_{SN}, p \rangle) \| + w_m(p) \cdot \sum_{c \in Con(\langle t_{SN}, p \rangle)} W(p)(c) +$$

$$\sum_{t_i \in \{t_1, \ldots, t_q\}} \sum_{p \in t_i^{\bullet}} w_m(p_{\alpha_i}) W(p_{\alpha_i})(p) \gamma(p, t_i) \quad (4)$$

Outline

1. Intro
   - Whats the plan?
   - P-invariants for classical Petri nets
   - Problem Statement
2. Example
3. Definitions
4. Application
5. Application
6. Application

Problem Statement

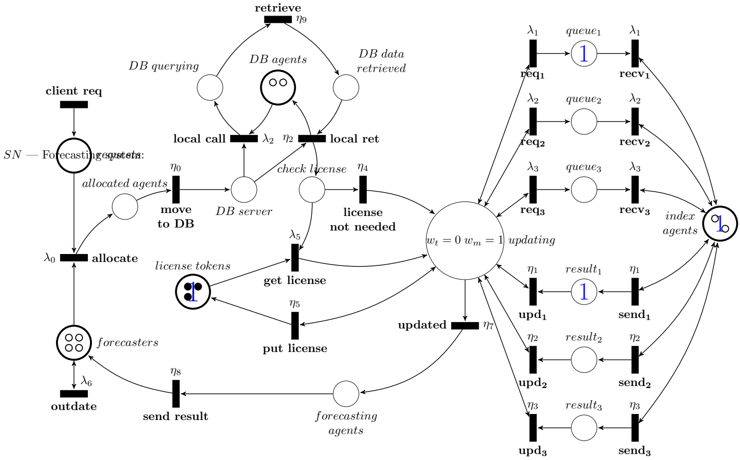
- How to defined them?
- What will we gain from the definition?
- How to calculate them?
- What properties do these things have?

Properties

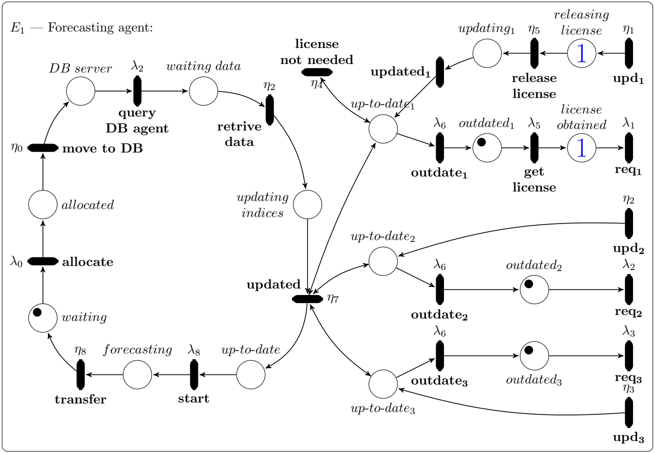
Following properties can be inferred from the invariants of the NP-net:

- the system can simultaneously send no more than 3 requests to the external index service;
- we always return license token before forecasting;
- queues sizes are finite and = number of license tokens;
- system is bounded (except user's query);
- there won't be dangling request in updating indices section;
- if the index 1 is outdated, then forecasting agent may enter updating state only with license token.

Forecasting system

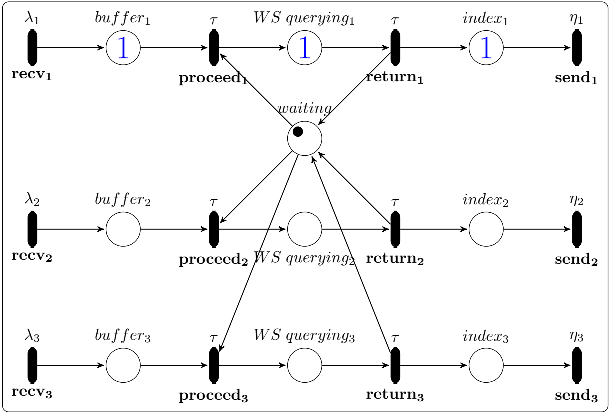


Example


E_1 — Forecasting agent:
DB server
λ_2
waiting data
η_2
query DB agent
retrive data
η_0
move to DB
allocated
λ_0
allocate
waiting
updating indices
updated
η_7
η_8
forecasting
λ_8
up-to-date
transfer
start
license not needed
η_4
updating$_1$
η_5
releasing license
η_1
1
updated$_1$
release license
upd$_1$
up-to-date$_1$
λ_6
outdated$_1$
λ_5
license obtained
λ_1
outdate$_1$
get license
req$_1$
η_2
upd$_2$
up-to-date$_2$
λ_6
outdated$_2$
λ_2
outdate$_2$
req$_2$
λ_6
λ_3
outdate$_3$
outdated$_3$
req$_3$
η_3
up-to-date$_3$
upd$_3$

Example

E_3 — Indices updater :



Outline

1 Intro
  - Whats the plan?
  - P-invariants for classical Petri nets
  - Problem Statement

2 Example

3 Definitions

4 Application

5 Application

6 Application

Problem Statement

- How to defined them?
- What will we gain from the definition?
- How to calculate them?
- What properties do these things have?

Algorithm for finding invariants

Algorithm (Place invariants finding). The algorithm is mostly straightforward. For each transition of the NP-net, it extracts the corresponding equations. The solutions of the constructed system of equations $\mathcal{E}$ determines all possible invariants.

- Step 1. [Elementary autonomous steps] For each elementary autonomous transition of every elementary net, extract a firing equation and add it to $\mathcal{E}$.
- Step 2. [External weights of system transitions] For each variable of every system autonomous transition, extract a firing equation and add it to $\mathcal{E}$.

Algorithm for finding invariants

- Step 3. [Internal weights of system transitions] For each variable of every system autonomous transition, extract a firing equation and add it to $\mathcal{E}$.
  For each possible vertical synchronization step of every vertical synchronization transition extract a firing equation and add it to $\mathcal{E}$
- Step 4. [Solution of equation system] Find solutions of $\mathcal{E}$. Each solution of $\mathcal{E}$ corresponds to an inductive invariant of NP.

Outline

1. Intro
   - Whats the plan?
   - P-invariants for classical Petri nets
   - Problem Statement
2. Example
3. Definitions
4. Application
5. Application
6. Application

Problem Statement

- How to defined them?
- What will we gain from the definition?
- How to calculate them?
- What properties do these things have?

Interesting points

- weight functions are not necessary invariants of NP-net components;
- the composition of NP-net components invariants is not necessary an NP-net invariant;
- invariants are distributed among the structure of an NP-net;
- a flow of tokens is "distributed" among levels.