On place invariants of nested Petri nets.

Speaker: Leonid Dworzanski
Scientific advisor: Irina A. Lomazova

National Research University Higher School of Economics

April 11, 2015

Outline

What's the plan?
What's the plan?
P-invariants for classical Petri nets
Problem Statement
Problem Statement

What's the plan?

What is a P-invariant of a classical Petri net?

What other types of P-invariants exist?

How to define them for NP-nets?

What's the plan?

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

P-invariants for classical Petri nets

- What are P-invariants for classical Petri nets?

  `http://www.informatik.uni-hamburg.de/TGI/PetriNets/introductions/aalst/`

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)

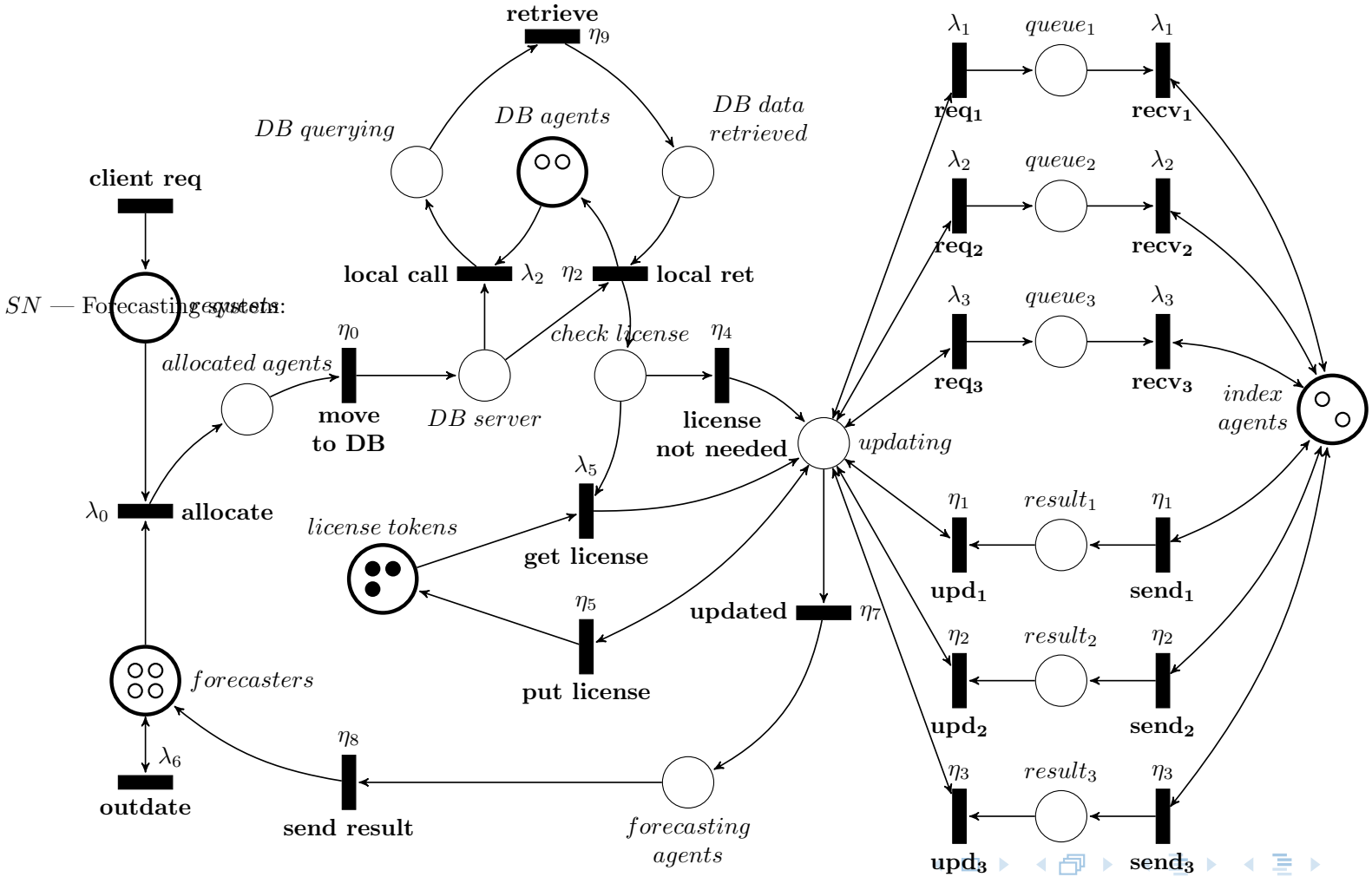
Problem Statement

Invariants have to be defined for NP-nets.

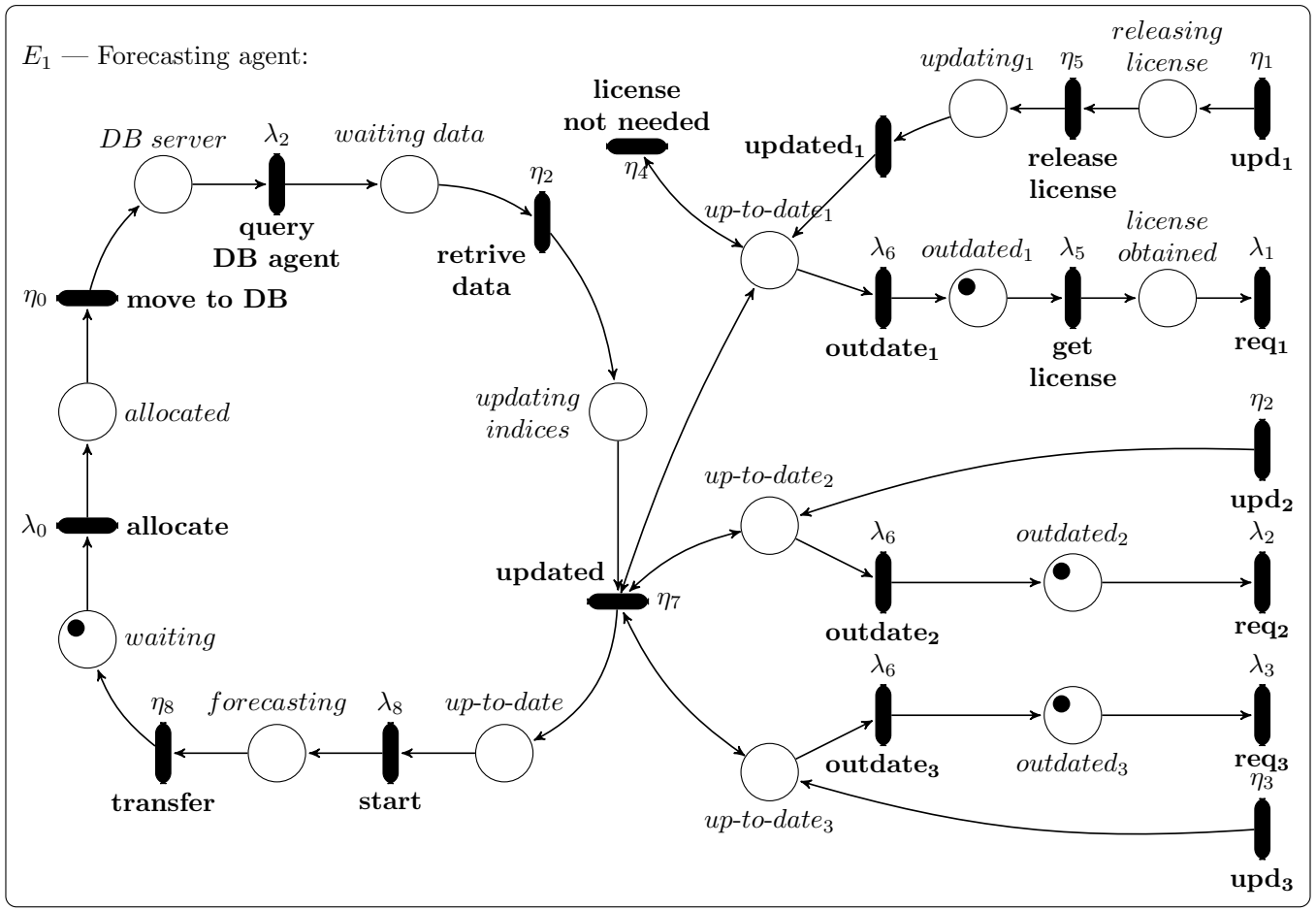
- How to defined them?
- Shall we take into account different levels?
- Will the invariants of NP-net components be compositional?
- Are such compositions are only possible invariants?
- Shall they be number- or term-valued functions?
- etc

Outline

Forecasting system

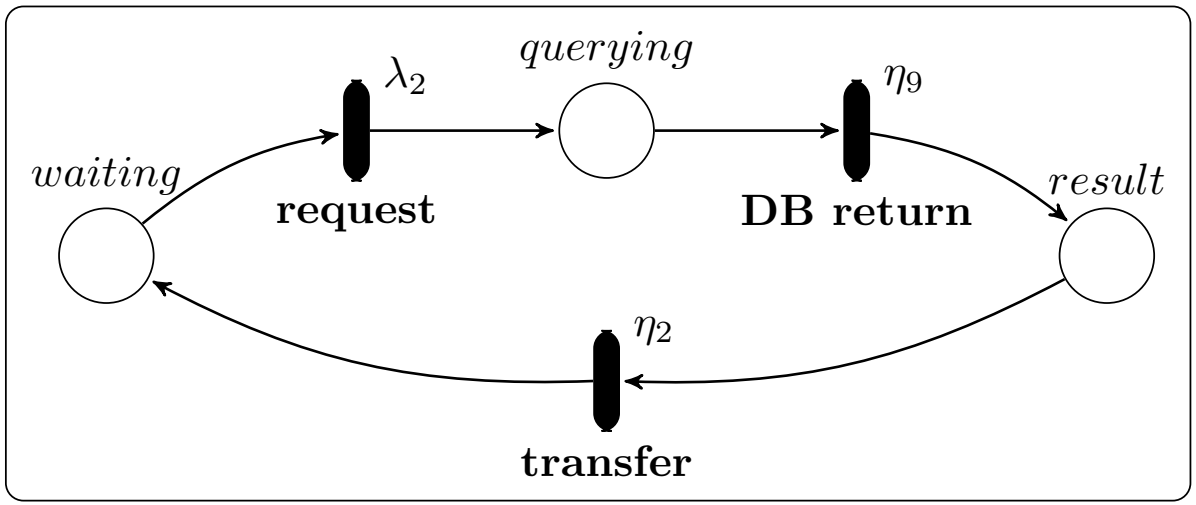


Example

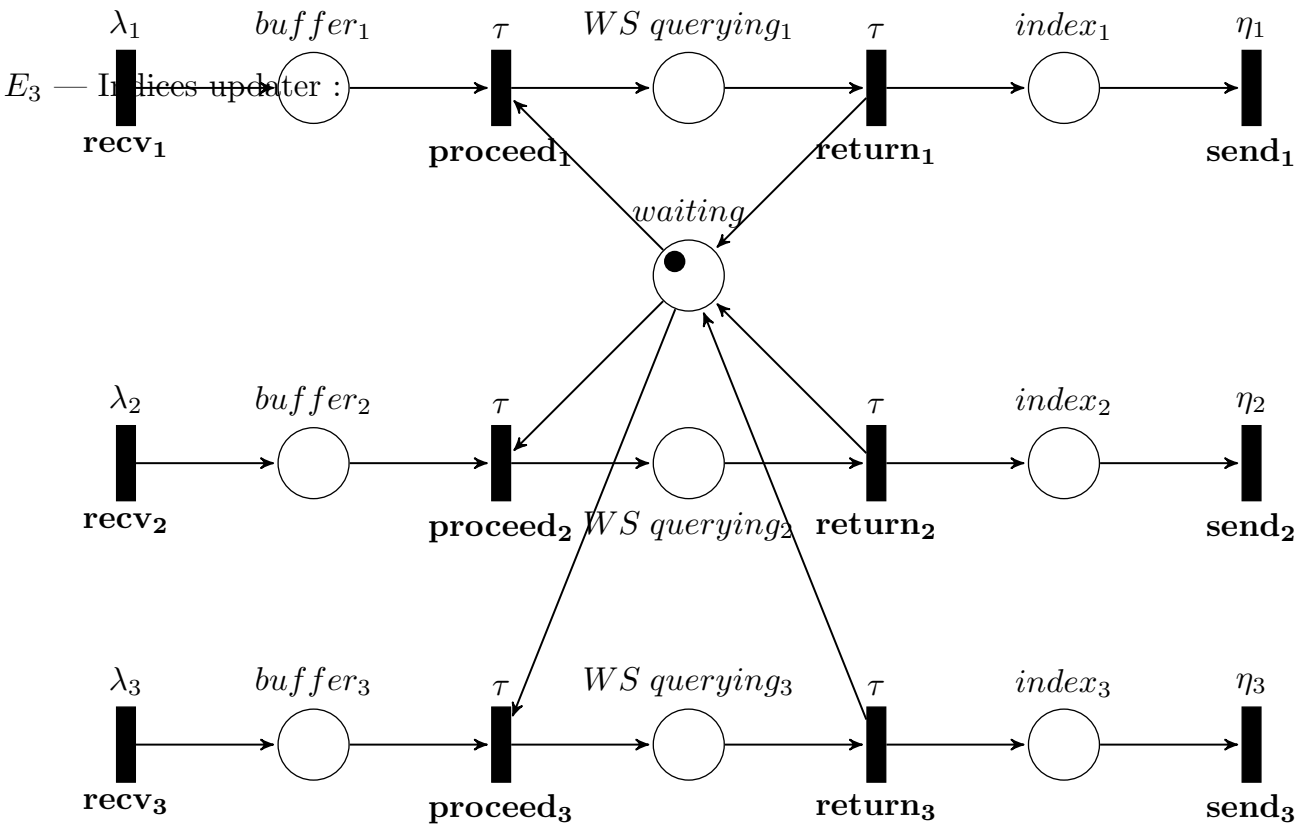

E_1 — Forecasting agent:
DB server
λ_2
waiting data
query
DB agent
η_2
retrive
data
license
not needed
η_4
η_0
move to DB
allocated
λ_0
allocate
waiting
η_8
forecasting
λ_8
up-to-date
transfer
start
updating
indices
updated
η_7
$updating_1$
η_5
releasing
license
η_1
$updated_1$
release
license
upd_1
$up\text{-}to\text{-}date_1$
λ_6
$outdated_1$
λ_5
license
obtained
λ_1
$outdate_1$
get
license
req_1
η_2
upd_2
$up\text{-}to\text{-}date_2$
λ_6
$outdated_2$
λ_2
$outdate_2$
req_2
λ_6
λ_3
$outdate_3$
$outdated_3$
req_3
η_3
$up\text{-}to\text{-}date_3$
upd_3

Example

E_2 — Database midlet :



Example


λ_1
$buffer_1$
τ
$WS\ querying_1$
τ
$index_1$
η_1
E_3 — Indices updater :
$\mathbf{recv_1}$
$\mathbf{proceed_1}$
$\mathbf{return_1}$
$\mathbf{send_1}$
$waiting$
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

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 defined 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.
$w_t : P_{SN} \to \mathbb{Z}$ and $w_m : P_{SN} \to \mathbb{Z}$ are weight functions.
For each place $p \in P_{SN}$ typed with the element net N, we assign the weight function w_p, that maps P_N into $\mathbb{Z}$.
Let $\hat{w}_p : \mathcal{M}(N) \to \mathbb{Z}$ be the linear extension of w_p to the markings of N defined in the standard way
$\forall m_p \in \mathcal{M}(N) : \hat{w}_p(m_p) = \sum_{q \in P_N} (m_p(q) w_p(q))$ (i.e. weigthed sum).
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

Outline

Problem Statement

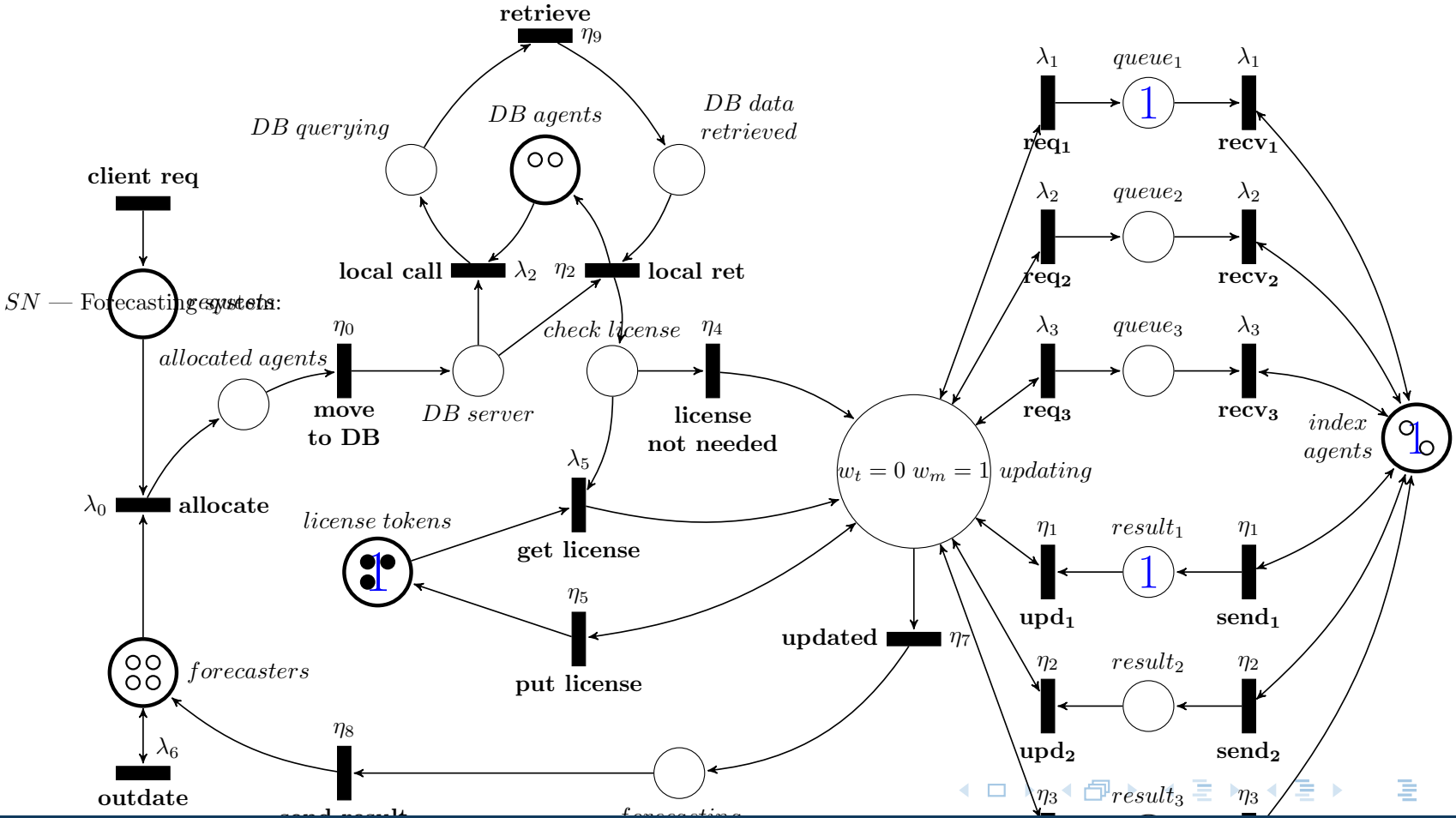
- How to defined them?
- What will we gain from the definition?
- How to calculate them?
- What properties do these things have?

Properties

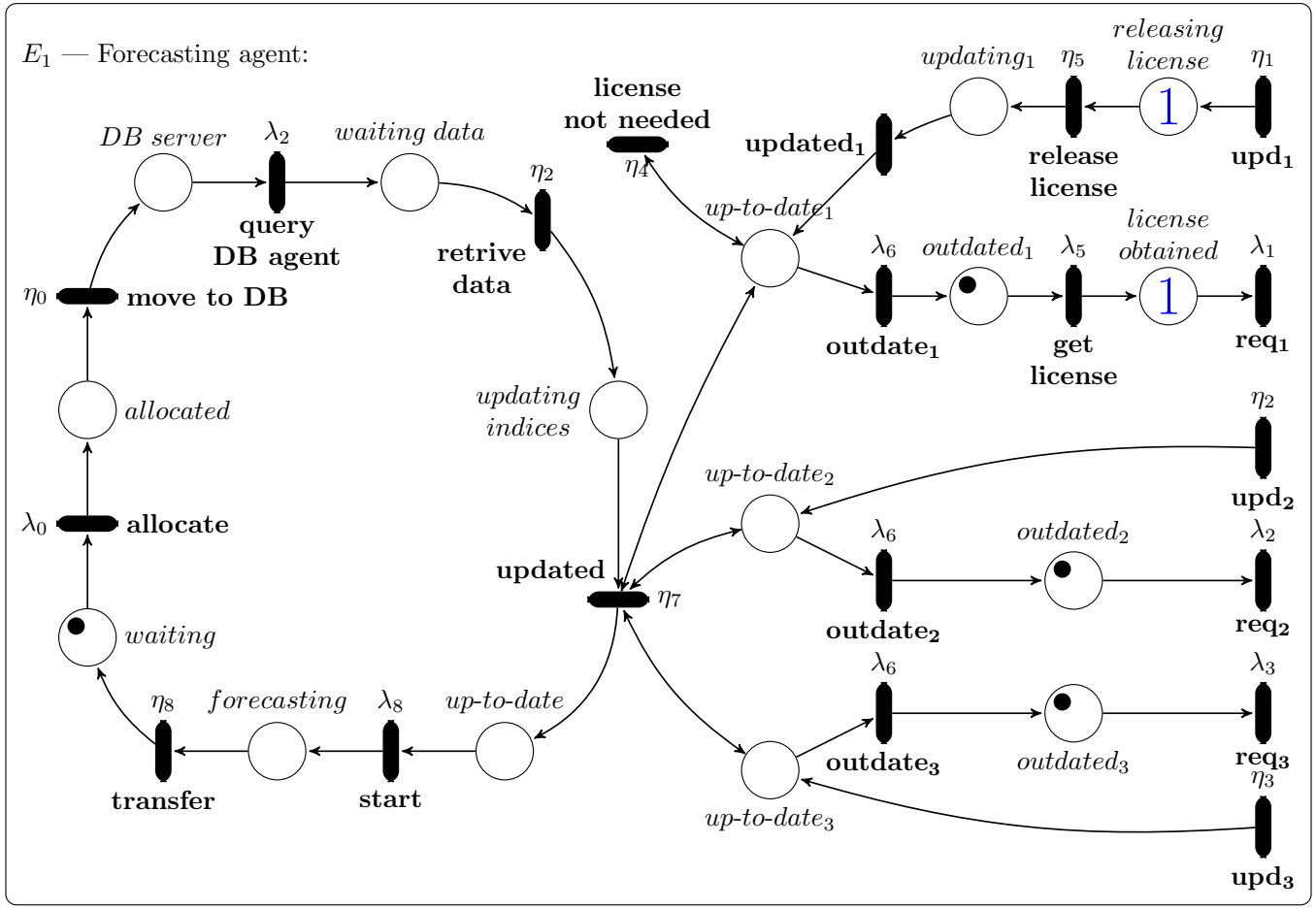
Following properties can be inferred from the invariants of the NP-net:

- 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

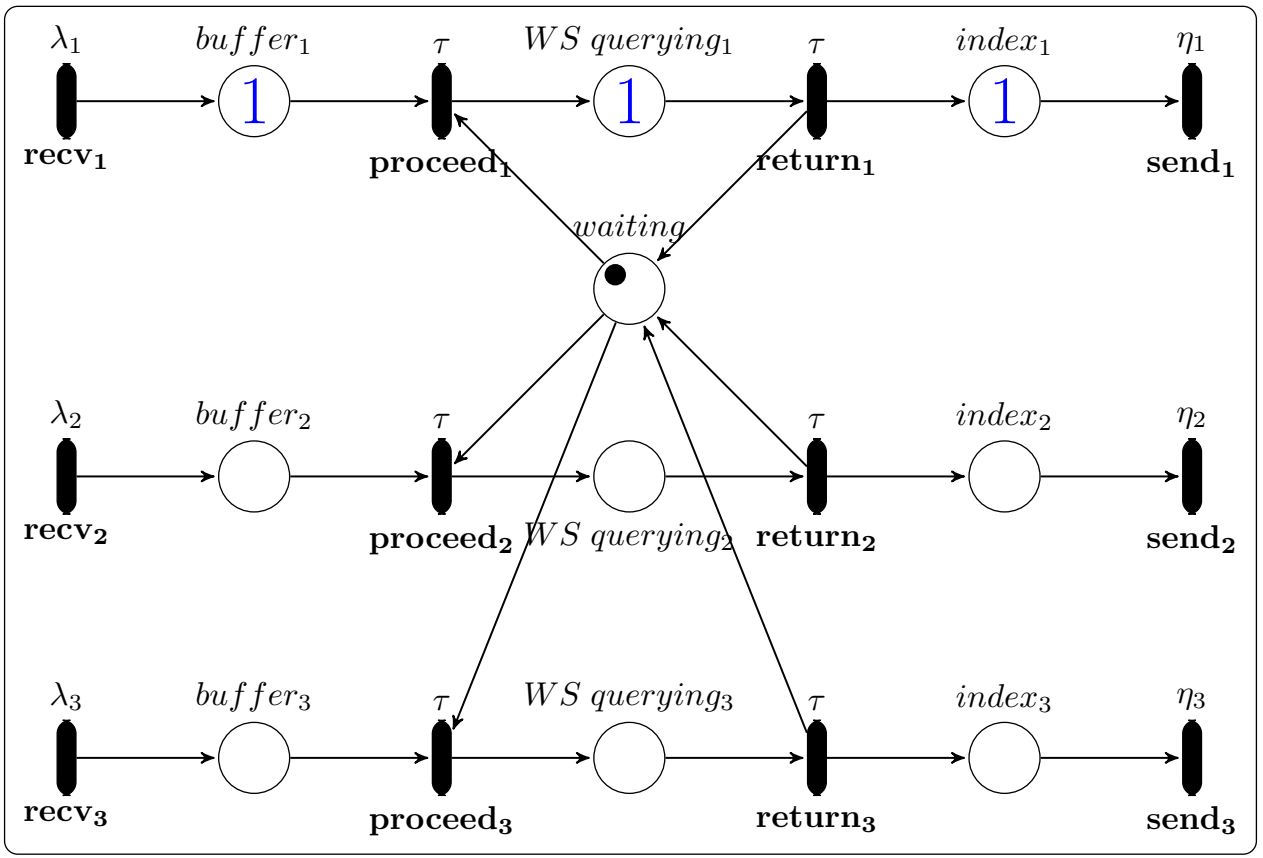


Example


E_1 — Forecasting agent:
DB server
λ_2
waiting data
query
DB agent
η_2
retrive
data
η_0
move to DB
allocated
λ_0
allocate
waiting
η_8
forecasting
λ_8
up-to-date
transfer
start
updating
indices
updated
η_7
license
not needed
η_4
$updating_1$
η_5
releasing
license
η_1
$updated_1$
release
license
upd_1
$up\text{-}to\text{-}date_1$
λ_6
$outdated_1$
λ_5
license
obtained
λ_1
$outdate_1$
get
license
req_1
η_2
upd_2
$up\text{-}to\text{-}date_2$
λ_6
$outdated_2$
λ_2
$outdate_2$
req_2
λ_6
λ_3
$outdate_3$
$outdated_3$
req_3
η_3
$up\text{-}to\text{-}date_3$
upd_3

Example

E_3 — Indices updater :



Interesting points

- all weightening functions are not necessary invariants;
- if they are invariants their composition is not necessary an invariant;
- invariants are distributed among the structure of an NP-net;
- a flow of tokens is “distributed” among levels.

Problem Statement

End of Part 1...

Outline

Conclusion

- NP-invariants are weighten sum of weighten sum of ...;
- NP-invariants capture flow among levels;