Unbounded Product-Form Petri Nets

Patricia Bouyer-Decitre[1], Serge Haddad[1] & Vincent Jugé[1,2]

1: LSV (CNRS, ENS Paris-Saclay & Inria) — 2: LIGM (Univ. Paris-Est Marne-la-Vallée)

07/09/2017

This work has been supported by

ERC project EQualIS (FP7-308087)

In the last decades, interest in quantitative systems rose dramatically. *Efficient* quantitative analysis of *high-level,* expressive systems with *infinite* underlying state space was *a major* difficulty.

From the performance evaluation *community* arose the idea of *studying* "product forms"...

Contents

Discrete-time & continuous-time Markov chains

Discrete-time Markov chain

Stochastic dynamical system evolving at integer time-points: $(X_n)_{n \in \mathbb{Z}_{\geqslant 0}}$

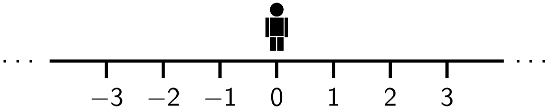
Example: Random walk

Discrete-time & continuous-time Markov chains

Discrete-time Markov chain

Stochastic dynamical system evolving at integer time-points: $(X_n)_{n \in \mathbb{Z}_{\geqslant 0}}$

Example: Random walk — $(X_n)_{n \geqslant 0} = 0$

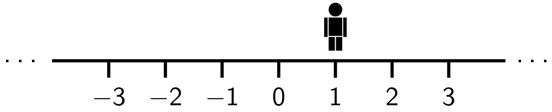


Discrete-time & continuous-time Markov chains

Discrete-time Markov chain

Stochastic dynamical system evolving at integer time-points: $(X_n)_{n \in \mathbb{Z}_{\geqslant 0}}$

Example: Random walk — $(X_n)_{n \geqslant 0} = 0, 1$

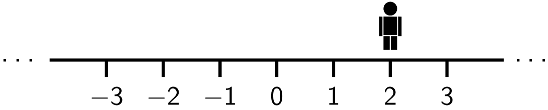


Discrete-time & continuous-time Markov chains

Discrete-time Markov chain

Stochastic dynamical system evolving at integer time-points: $(X_n)_{n \in \mathbb{Z}_{\geqslant 0}}$

Example: Random walk — $(X_n)_{n \geqslant 0} = 0, 1, 2$

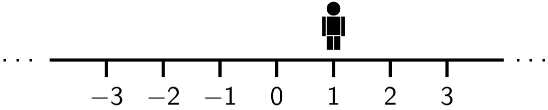


Discrete-time & continuous-time Markov chains

Discrete-time Markov chain

Stochastic dynamical system evolving at integer time-points: $(X_n)_{n \in \mathbb{Z}_{\geqslant 0}}$

Example: Random walk — $(X_n)_{n \geqslant 0} = 0, 1, 2, 1, \ldots$



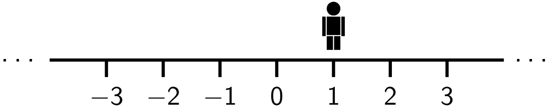
Discrete-time & continuous-time Markov chains

Discrete-time Markov chain

Stochastic dynamical system evolving at integer time-points: $(X_n)_{n \in \mathbb{Z}_{\geqslant 0}}$

Example: Random walk — $(X_n)_{n \geqslant 0} = 0, 1, 2, 1, \ldots$

Markov Property: $\{(X_k)_{k \geqslant n} \mid (X_k)_{k \leqslant n}\} = \{(X_k)_{k \geqslant n} \mid X_n\}$



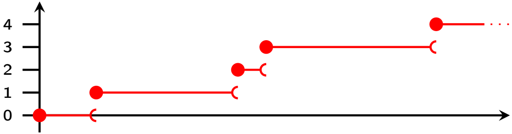
Discrete-time & continuous-time Markov chains

Discrete-time Markov chain

Stochastic dynamical system evolving at integer time-points: $(X_n)_{n \in \mathbb{Z}_{\geqslant 0}}$

Example: Random walk — $(X_n)_{n \geqslant 0} = 0, 1, 2, 1, \ldots$

Markov Property: $\{(X_k)_{k \geqslant n} \mid (X_k)_{k \leqslant n}\} = \{(X_k)_{k \geqslant n} \mid X_n\}$



Continuous-time Markov chain

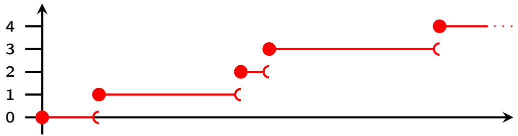
Random dynamical system evolving at **real** time-points: $(X_t)_{t \in \mathbb{R}_{\geqslant 0}}$

Example: Counting bus arrivals

Discrete-time & continuous-time Markov chains

Discrete-time Markov chain

Stochastic dynamical system evolving at integer time-points: $(X_n)_{n \in \mathbb{Z}_{\geqslant 0}}$
Example: Random walk — $(X_n)_{n \geqslant 0} = 0, 1, 2, 1, \ldots$
Markov Property: $\{(X_k)_{k \geqslant n} \mid (X_k)_{k \leqslant n}\} = \{(X_k)_{k \geqslant n} \mid X_n\}$



Continuous-time Markov chain

Random dynamical system evolving at **real** time-points: $(X_t)_{t \in \mathbb{R}_{\geqslant 0}}$
Example: Counting bus arrivals
Markov Property: $\{(X_s)_{s \geqslant t} \mid (X_s)_{s \leqslant t}\} = \{(X_s)_{s \geqslant t} \mid X_t\}$

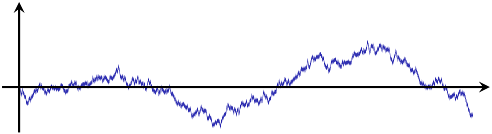
Discrete-time & continuous-time Markov chains

Discrete-time Markov chain

Stochastic dynamical system evolving at integer time-points: $(X_n)_{n \in \mathbb{Z}_{\geqslant 0}}$

Example: Random walk — $(X_n)_{n \geqslant 0} = 0, 1, 2, 1, \ldots$

Markov Property: $\{(X_k)_{k \geqslant n} \mid (X_k)_{k \leqslant n}\} = \{(X_k)_{k \geqslant n} \mid X_n\}$



Continuous-time Markov chain ≠ Markov process

Random dynamical system evolving at **real** time-points: $(X_t)_{t \in \mathbb{R}_{\geqslant 0}}$

Example: Counting bus arrivals

Markov Property: $\{(X_s)_{s \geqslant t} \mid (X_s)_{s \leqslant t}\} = \{(X_s)_{s \geqslant t} \mid X_t\}$

Ergodic Markov chains

What does a Markov chain look like after a long time?

The Markov chain is **ergodic** if it has a **limit** probability measure $\mu = \lim_{t \to +\infty} X_t$.

Examples: Random walk on $\mathbb{Z}$: **not ergodic**

Random walk on $\mathbb{Z}/7\mathbb{Z}$: **ergodic**

Communication networks: ?

Ergodic Markov chains

What does a Markov chain look like after a long time?

The Markov chain is **ergodic** if it has a **limit** probability measure $\mu = \lim_{t \to +\infty} X_t$: μ is the unique **invariant probability measure**.

Examples: Random walk on $\mathbb{Z}$: **not ergodic**

Random walk on $\mathbb{Z}/7\mathbb{Z}$: **ergodic**

Communication networks: ?

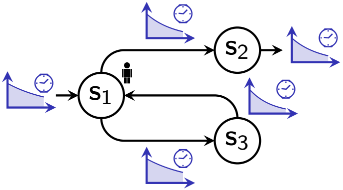
Our goal: Check ergodicity & compute μ **efficiently** in relevant cases.

(undecidable in general)

Product-form Markov chains

Fixed-rate queuing systems (≈ Jackson networks)

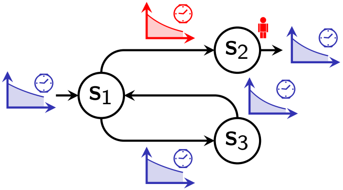
Clients follow edges when fixed-rate **exponential clocks** ring.



Product-form Markov chains

Fixed-rate queuing systems (≈ Jackson networks)

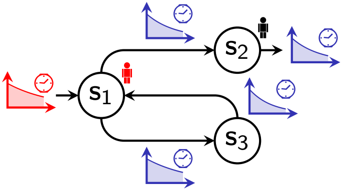
Clients follow edges when fixed-rate **exponential clocks** ring.



Product-form Markov chains

Fixed-rate queuing systems (≈ Jackson networks)

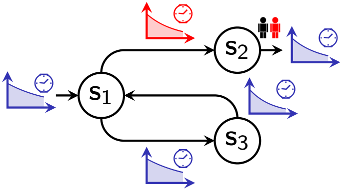
Clients follow edges when fixed-rate **exponential clocks** ring.



Product-form Markov chains

Fixed-rate queuing systems (≈ Jackson networks)

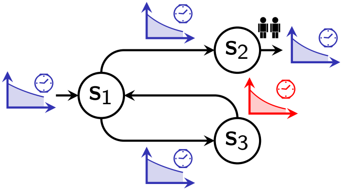
Clients follow edges when fixed-rate **exponential clocks** ring.



Product-form Markov chains

Fixed-rate queuing systems (≈ Jackson networks)

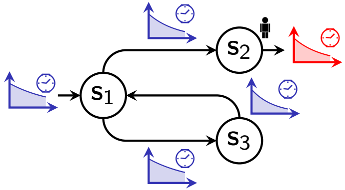
Clients follow edges when fixed-rate **exponential clocks** ring.



Product-form Markov chains

Fixed-rate queuing systems (≈ Jackson networks)

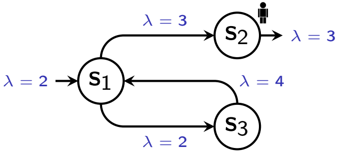
Clients follow edges when fixed-rate **exponential clocks** ring.



Product-form Markov chains

Fixed-rate queuing systems (≈ Jackson networks)

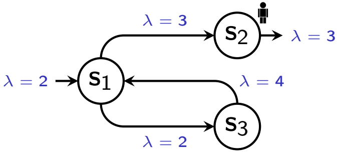
Clients follow edges when fixed-rate **exponential clocks** ring.



Product-form Markov chains

Fixed-rate queuing systems (≈ Jackson networks)

Clients follow edges when fixed-rate **exponential clocks** ring.
Such systems have **product-form** invariant measures.



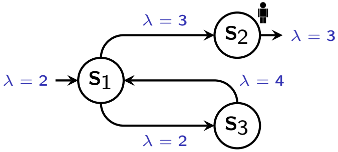
$\mathbf{n}_i = \#$ 🧍 in $\mathbf{s}_i$

$$\mu(\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3) = \mathbf{K}\mathbf{p}_1^{\mathbf{n}_1}\mathbf{p}_2^{\mathbf{n}_2}\mathbf{p}_3^{\mathbf{n}_3}$$

Product-form Markov chains

Fixed-rate queuing systems (≈ Jackson networks)

Clients follow edges when fixed-rate **exponential clocks** ring.
Such systems have **product-form** invariant measures.



$\mathbf{n}_i = \#$ clients in $\mathbf{s}_i$ **Ergodic** Markov chain!

$$\mu(\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3) = \mathbf{K}\mathbf{p}_1^{\mathbf{n}_1}\mathbf{p}_2^{\mathbf{n}_2}\mathbf{p}_3^{\mathbf{n}_3}$$

$$\mathbf{p}_1 = \mathbf{p}_2 = 2/3,\ \mathbf{p}_3 = 1/3,\ \mathbf{K} = 27/2$$

Product-form invariant measures

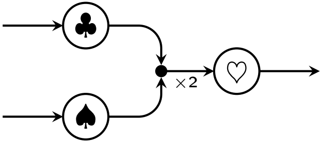
Linear algebra gives us the coefficients $\mathbf{p}_i$ in polynomial time.
It remains to compute the normalising constant $\mathbf{K}$ without enumerating the reachability set. (easy in queuing systems, hard in general)

Beyond queuing systems

What about concurrency?

What if people interact with each other?

Example: Marrying people

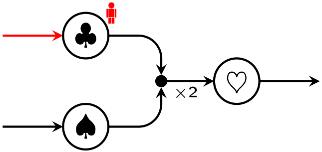


Beyond queuing systems

What about concurrency?

What if people interact with each other?

Example: Marrying people

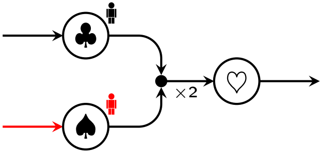


Beyond queuing systems

What about concurrency?

What if people interact with each other?

Example: Marrying people

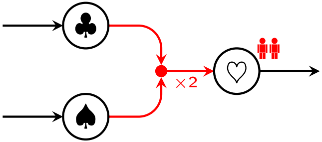


Beyond queuing systems

What about concurrency?

What if people interact with each other?

Example: Marrying people

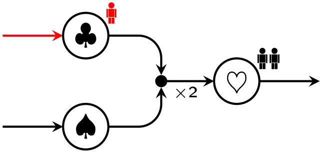


Beyond queuing systems

What about concurrency?

What if people interact with each other?

Example: Marrying people

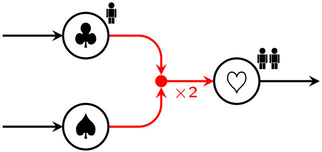


Beyond queuing systems

What about concurrency?

What if people interact with each other?

Example: Marrying people



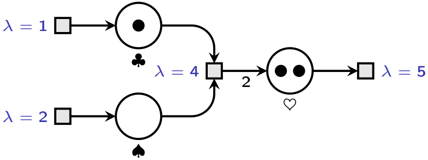
Beyond queuing systems

What about concurrency?

What if people interact with each other?

Example: Marrying people

Use **stochastic Petri nets**!



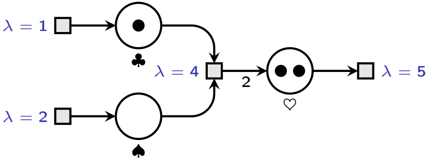
Beyond queuing systems

What about concurrency?

What if people interact with each other?

Example: Marrying people

Use **stochastic Petri nets**!



Next step: Investigate qualitative & quantitative properties, e.g.:

- boundedness,
- liveness,
- ergodicity,
- reachability,
- coverability,
- invariant measure. . .

Stochastic Petri nets & Π^2-nets

In spite of their greater complexity...

Some classes of stochastic Petri nets are product form!

- Coleman et al.'s condition (depends on firing rates)
- Π^2-nets (independent of firing rates)

Stochastic Petri nets & Π^2-nets

In spite of their greater complexity...

Some classes of stochastic Petri nets are product form!

- Coleman et al.'s condition (depends on firing rates)
- Π^2-nets (independent of firing rates)

Complexity results in Π^2-nets

- Checking that a net is a Π^2-net: **PTime** ☺
- Coverability: **ExpSpace-hard** ☹
- Liveness: **PSpace-hard** ☹ (even in safe nets)
- Reachability: **PSpace-hard** ☹ (even in safe nets)

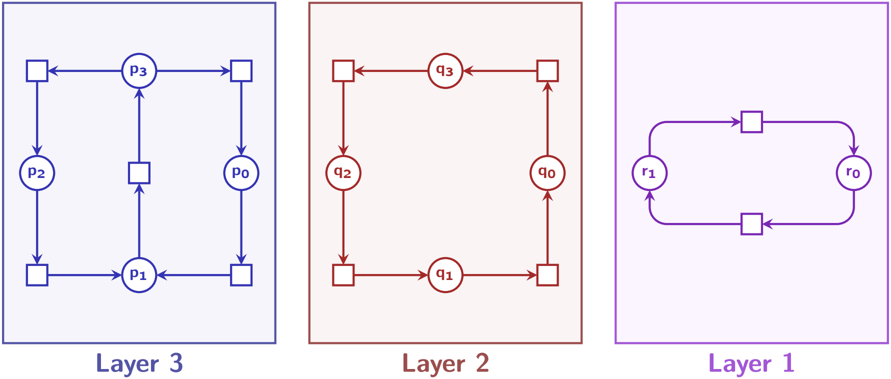
Efficiently computing normalising constants seems out of reach!

Π^2-nets & closed Π^3-nets ($\approx$ nested queuing systems)

Π^2-nets are too general: let us focus on (closed) Π^3-nets!

Π^2-nets & closed Π^3-nets ($\approx$ nested queuing systems)

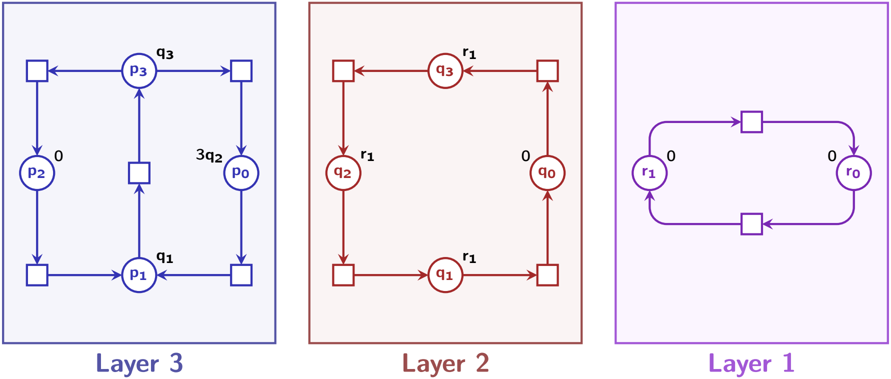
Π^2-nets are too general: let us focus on (closed) Π^3-nets!



- Layers = Strongly connected finite-state machines

Π^2-nets & closed Π^3-nets (≈ nested queuing systems)

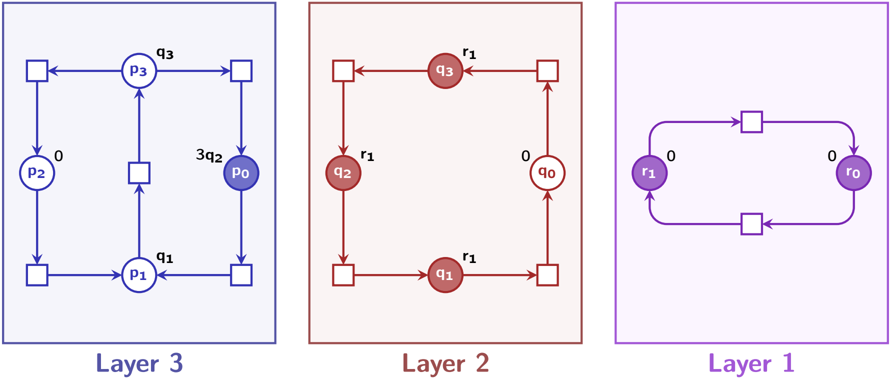
Π^2-nets are too general: let us focus on (closed) Π^3-nets!



- Layers = Strongly connected finite-state machines
- Every place requires **resources** in the previous layer

Π^2-nets & closed Π^3-nets (≈ nested queuing systems)

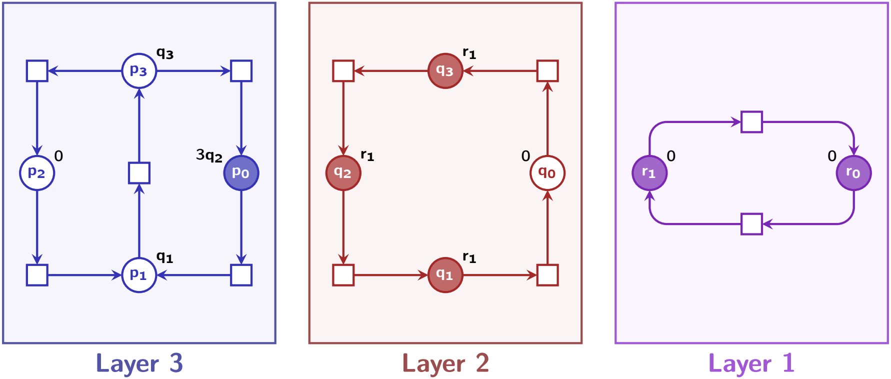
Π^2-nets are too general: let us focus on (closed) Π^3-nets!



- Layers = Strongly connected finite-state machines
- Every place requires **resources** in the previous layer
- Resources must be the most **resource-intensive** within their layer

Π^2-nets & closed Π^3-nets ($\approx$ nested queuing systems)

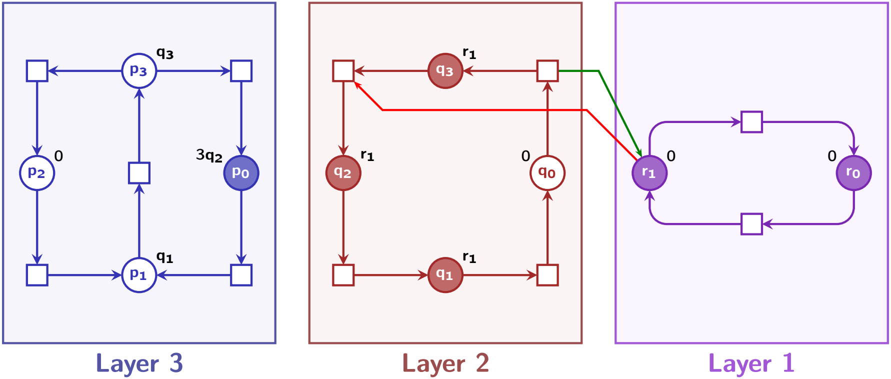
Π^2-nets are too general: let us focus on (closed) Π^3-nets!



- Layers = Strongly connected finite-state machines
- Every place requires **resources** in the previous layer
- Resources must be the most **resource-intensive** within their layer
- Creating/deleting a token releases/consumes its resources

Π^2-nets & closed Π^3-nets ($\approx$ nested queuing systems)

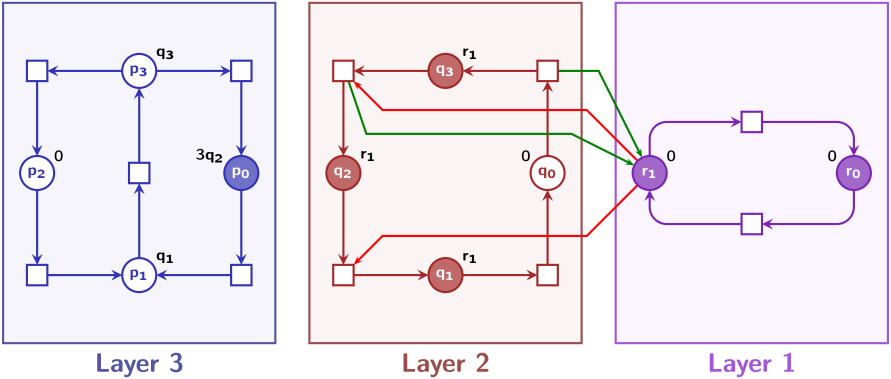
Π^2-nets are too general: let us focus on (closed) Π^3-nets!



- Layers = Strongly connected finite-state machines
- Every place requires **resources** in the previous layer
- Resources must be the most **resource-intensive** within their layer
- Creating/deleting a token releases/consumes its resources

Π^2-nets & closed Π^3-nets ($\approx$ nested queuing systems)

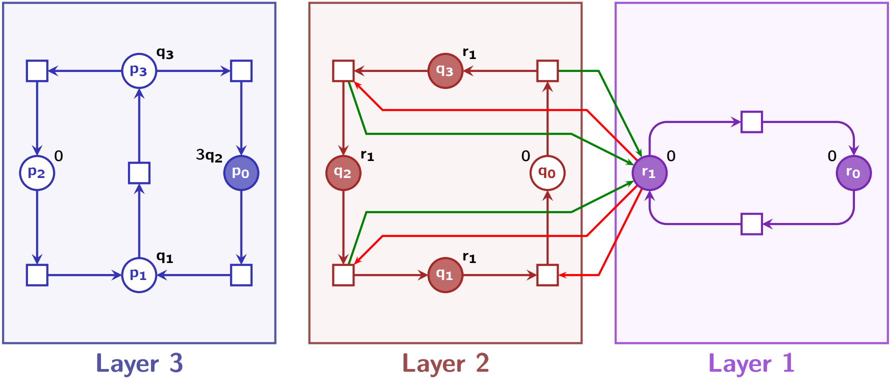
Π^2-nets are too general: let us focus on (closed) Π^3-nets!



- Layers = Strongly connected finite-state machines
- Every place requires **resources** in the previous layer
- Resources must be the most **resource-intensive** within their layer
- Creating/deleting a token releases/consumes its resources

Π^2-nets & closed Π^3-nets (≈ nested queuing systems)

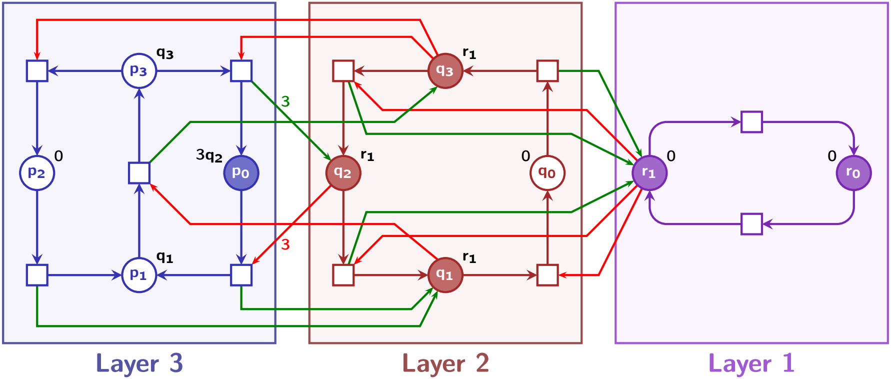
Π^2-nets are too general: let us focus on (closed) Π^3-nets!



- Layers = Strongly connected finite-state machines
- Every place requires **resources** in the previous layer
- Resources must be the most **resource-intensive** within their layer
- Creating/deleting a token releases/consumes its resources

Π^2-nets & closed Π^3-nets ($\approx$ nested queuing systems)

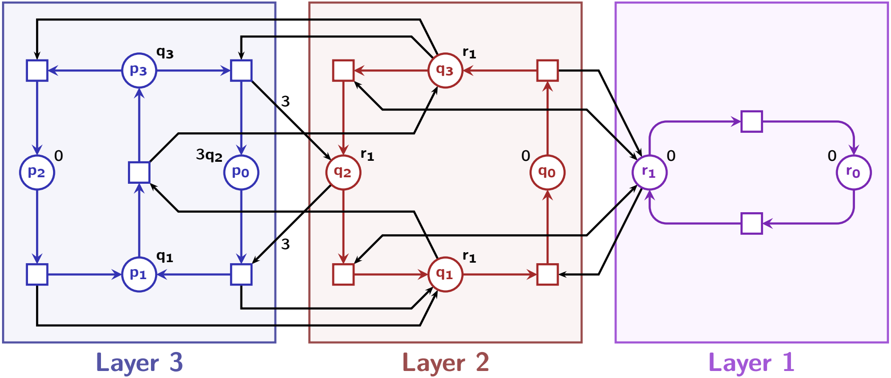
Π^2-nets are too general: let us focus on (closed) Π^3-nets!



- Layers = Strongly connected finite-state machines
- Every place requires **resources** in the previous layer
- Resources must be the most **resource-intensive** within their layer
- Creating/deleting a token releases/consumes its resources

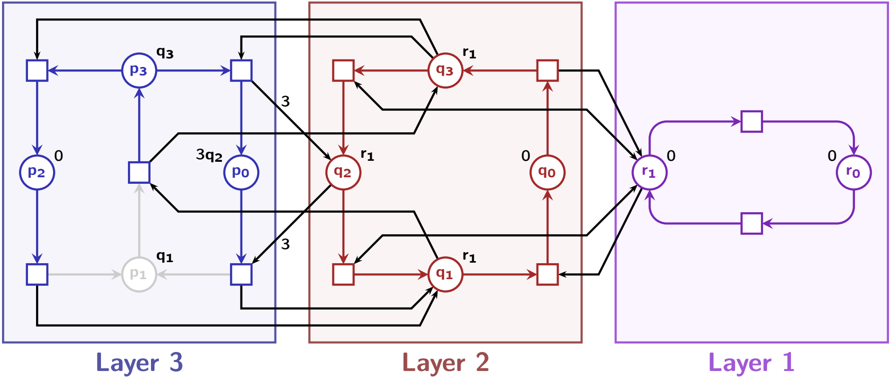
Closed Π^3-nets & open Π^3-nets

Closed Π^3-nets cannot model open systems: can we handle them?



Closed Π^3-nets & open Π^3-nets

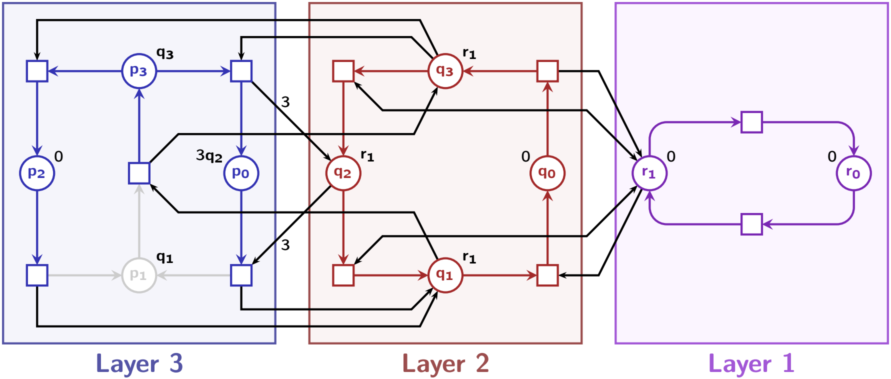
Closed Π^3-nets cannot model open systems: can we handle them?



Delete **one** place in the top layer and make your net **open**!

Closed Π^3-nets & open Π^3-nets

Closed Π^3-nets cannot model open systems: can we handle them?

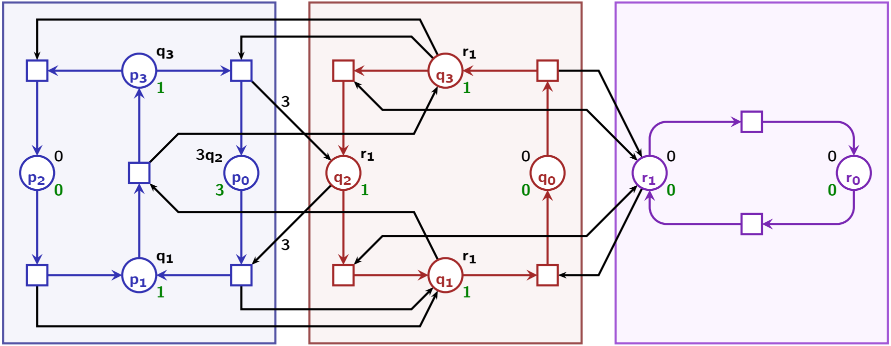


Delete **one** place in the top layer and make your net **open**!

1-layer open/closed Π^3-nets = open/closed (connected) Jackson networks

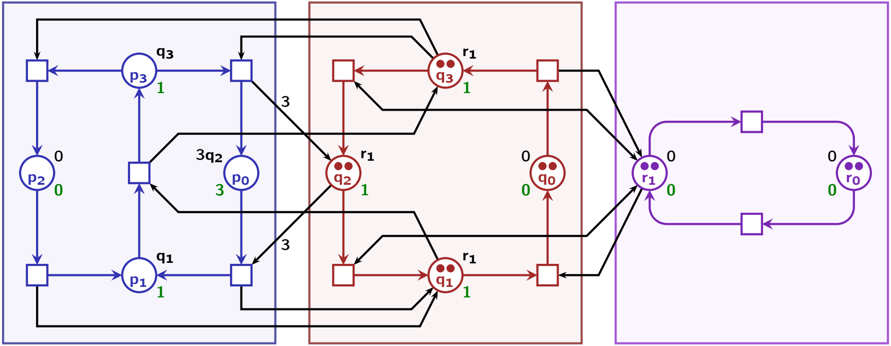
Qualitative properties in (closed or open) Π^3-Nets

- Liveness: linear constraints — involves **potential**



Qualitative properties in (closed or open) Π^3-Nets

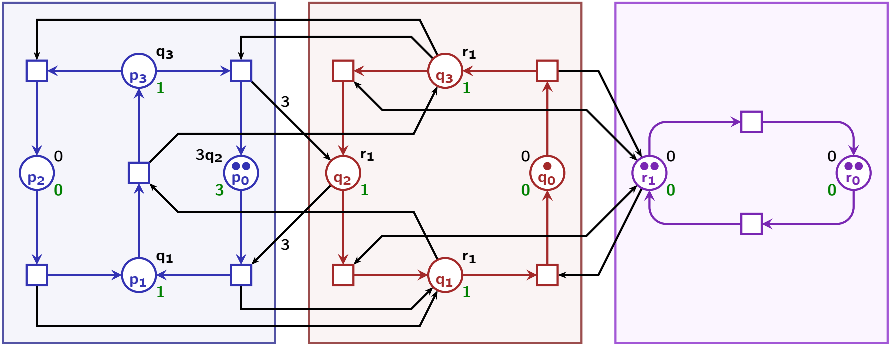
- Liveness: linear constraints — involves **potential**



Not live ✗ (Layer 3 is dead)

Qualitative properties in (closed or open) Π^3-Nets

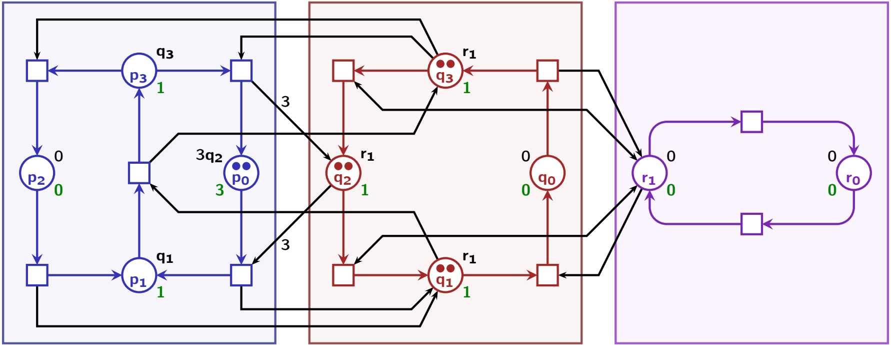
- Liveness: linear constraints — involves **potential**



Not live ✗ (Layer 3 is dead)

Qualitative properties in (closed or open) Π^3-Nets

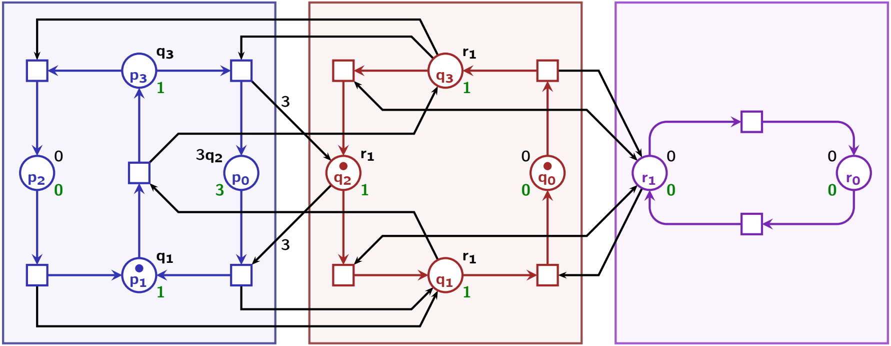
- Liveness: linear constraints — involves potential



Not live ✗ (Layer 2 is dead)

Qualitative properties in (closed or open) Π^3-Nets

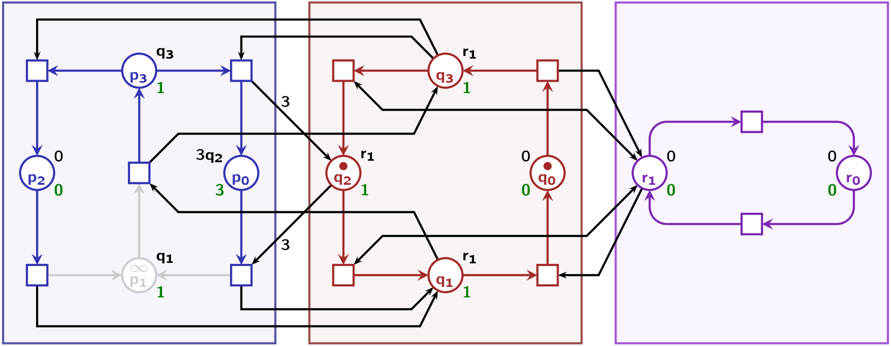
- Liveness: linear constraints — involves **potential**



Live ✓

Qualitative properties in (closed or open) Π^3-Nets

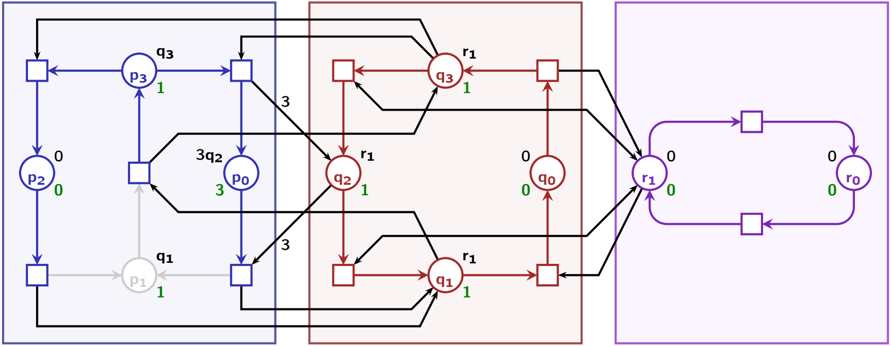
- Liveness: linear constraints — involves **potential**



Live ✓

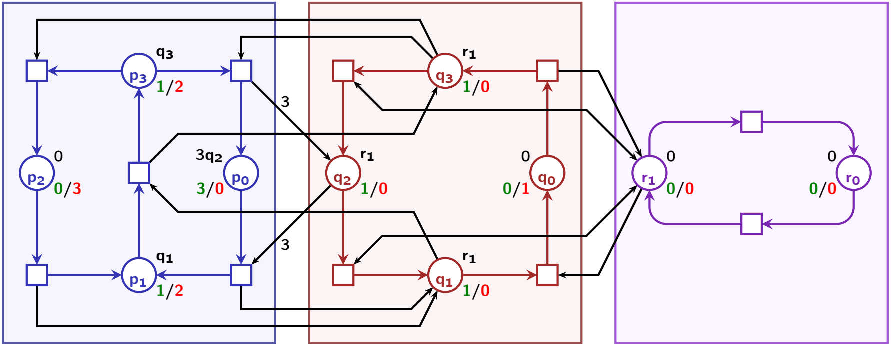
Qualitative properties in (closed or open) Π^3-Nets

- Liveness: **PTime** ☺



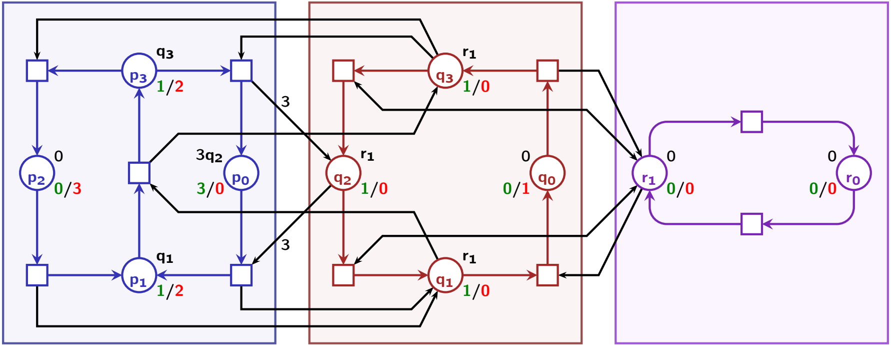
Qualitative properties in (closed or open) Π^3-Nets

- Liveness: **PTime** ☺
- Reachability in **live** nets: linear constraints — involves potential **gap**



Qualitative properties in (closed or open) Π^3-Nets

- Liveness: **PTime** ☺
- Reachability in **live** nets: linear constraints — involves potential **gap**

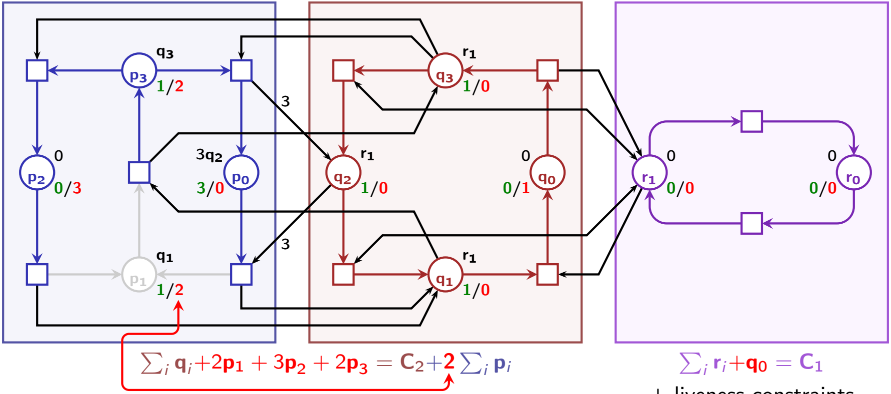


$\sum_i \mathbf{p}_i = \mathbf{C}_3$ $\sum_i \mathbf{q}_i + 2\mathbf{p}_1 + 3\mathbf{p}_2 + 2\mathbf{p}_3 = \mathbf{C}_2$ $\sum_i \mathbf{r}_i + \mathbf{q}_0 = \mathbf{C}_1$

+ liveness constraints

Qualitative properties in (closed or open) Π^3-Nets

- Liveness: **PTime** ☺
- Reachability in **live** nets: linear constraints — involves potential **gap**



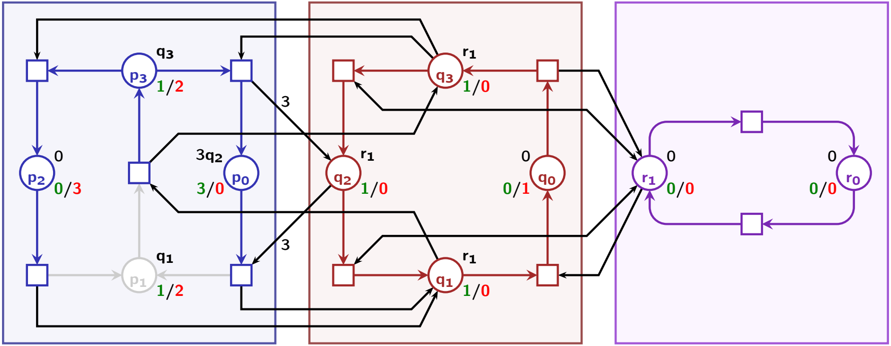
$\sum_i q_i + 2p_1 + 3p_2 + 2p_3 = C_2 + 2\sum_i p_i$

$\sum_i r_i + q_0 = C_1$

\+ liveness constraints

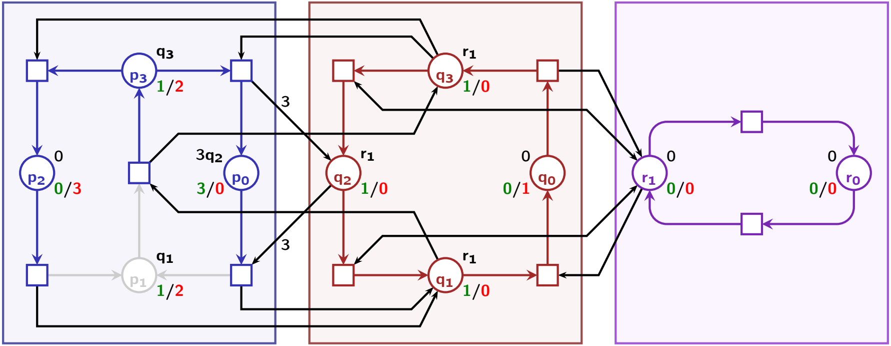
Qualitative properties in (closed or open) Π^3-Nets

- Liveness: **PTime** ☺
- Reachability in **live** nets: **PTime** ☺ (NP hard in general ☹)



Qualitative properties in (closed or open) Π^3-Nets

- Liveness: **PTime** ☺
- Reachability in **live** nets: **PTime** ☺ (NP hard in general ☹)
- Coverability in **live** nets: **PTime** ☺ (NP hard in general ☹)
- Boundedness in **live** nets: **PTime** ☺ (co-NP hard in general ☹)



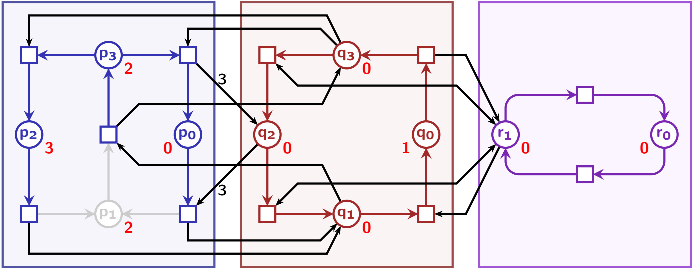
Ergodicity

All (live or not) **closed Π^3-nets are bounded, hence ergodic!**

Ergodicity

All (live or not) **closed Π^3-nets are bounded, hence ergodic!**

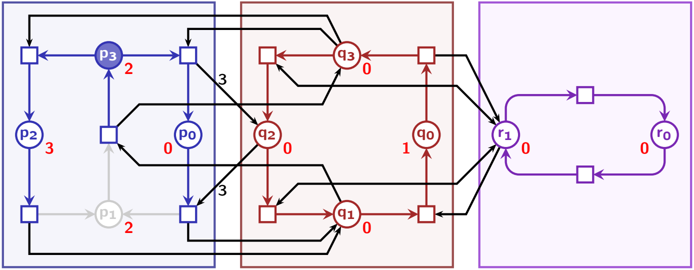
This **open** Π^3-net is ergodic if:



Ergodicity

All (live or not) **closed Π^3-nets are bounded, hence ergodic!**

This **open** Π^3-net is ergodic if: $(\mu = \mathbf{K} \prod_{i \neq 1} \mu_i^{\mathbf{p}_i} \prod_k \nu_k^{\mathbf{q}_k} \prod_\ell \rho_\ell^{\mathbf{r}_\ell})$

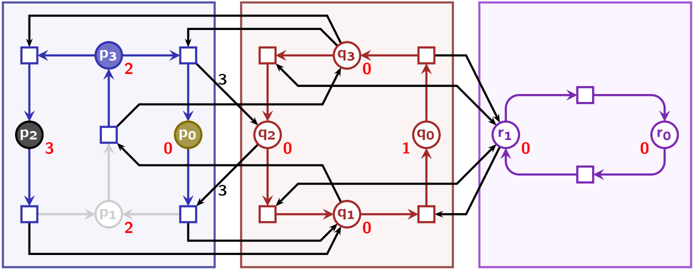


- if $\mathbf{gap}(\mathbf{p}_i) = 2$, then $\mu_i < 1$;

Ergodicity

All (live or not) **closed Π^3-nets are bounded, hence ergodic!**

This **open** Π^3-net is ergodic if: $\left(\mu = \mathbf{K} \prod_{i \neq 1} \mu_i^{\mathbf{p}_i} \prod_k \nu_k^{\mathbf{q}_k} \prod_\ell \rho_\ell^{\mathbf{r}_\ell}\right)$

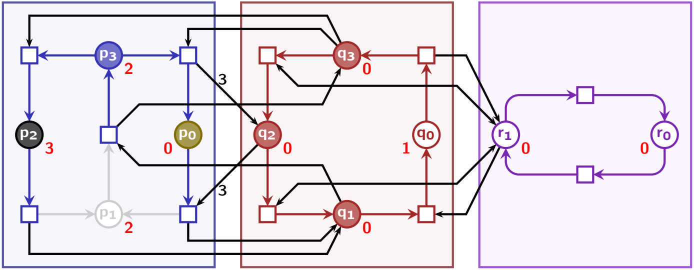


- if $\mathbf{gap}(\mathbf{p}_i) = 2$, then $\mu_i < 1$;
- if $\mathbf{gap}(\mathbf{p}_i) > 2 > \mathbf{gap}(\mathbf{p}_j)$, then ${\mu_j}^{\mathbf{gap}(\mathbf{p}_i)-2} {\mu_i}^{2-\mathbf{gap}(\mathbf{p}_j)} < 1$;

Ergodicity

All (live or not) **closed Π^3-nets are bounded, hence ergodic!**

This **open** Π^3-net is ergodic if: $\left(\mu = \mathbf{K} \prod_{i \neq 1} \mu_i^{\mathbf{p}_i} \prod_k \nu_k^{\mathbf{q}_k} \prod_\ell \rho_\ell^{\mathbf{r}_\ell}\right)$

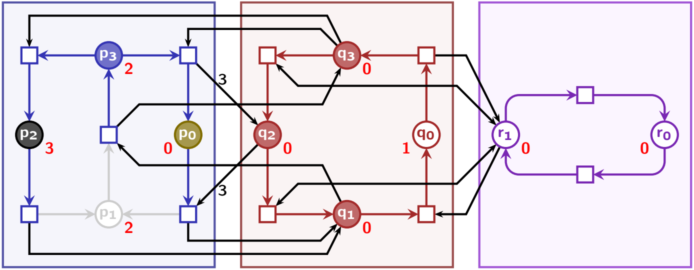


- if $\mathbf{gap}(\mathbf{p}_i) = 2$, then $\mu_i < 1$;
- if $\mathbf{gap}(\mathbf{p}_i) > 2 > \mathbf{gap}(\mathbf{p}_j)$, then ${\mu_j}^{\mathbf{gap}(\mathbf{p}_i)-2} {\mu_i}^{2-\mathbf{gap}(\mathbf{p}_j)} < 1$;
- if $2 > \mathbf{gap}(\mathbf{p}_j)$ & $\mathbf{gap}(\mathbf{q}_k) = 0$, then ${\nu_k}^{2-\mathbf{gap}(\mathbf{p}_j)} \mu_j < 1$.

Ergodicity

All (live or not) **closed Π^3-nets are bounded, hence ergodic!**

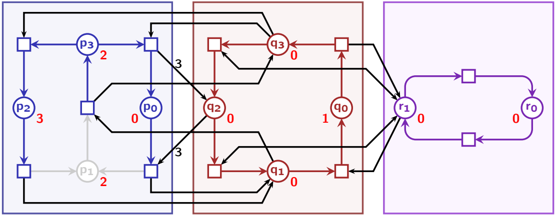
This **open** Π^3-net is ergodic if and only if: $\left(\mu = \mathbf{K} \prod_{i \neq 1} \mu_i^{\mathbf{p}_i} \prod_k \nu_k^{\mathbf{q}_k} \prod_\ell \rho_\ell^{\mathbf{r}_\ell}\right)$



- if $\mathbf{gap}(\mathbf{p}_i) = 2$, then $\mu_i < 1$;
- if $\mathbf{gap}(\mathbf{p}_i) > 2 > \mathbf{gap}(\mathbf{p}_j)$, then ${\mu_j}^{\mathbf{gap}(\mathbf{p}_i)-2} {\mu_i}^{2-\mathbf{gap}(\mathbf{p}_j)} < 1$;
- if $2 > \mathbf{gap}(\mathbf{p}_j)$ & $\mathbf{gap}(\mathbf{q}_k) = 0$, then ${\nu_k}^{2-\mathbf{gap}(\mathbf{p}_j)} \mu_j < 1$.

Computing the invariant measure

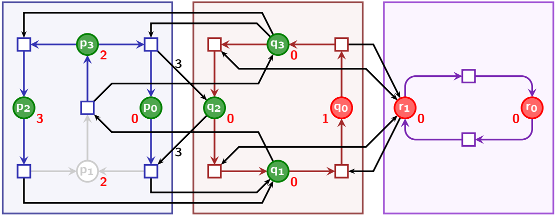
$$\mu = \mathsf{K} \prod_{\text{place } p} \mu_p^{\text{tokens in } p}$$



1. Computing μ_p: **PTime** using linear algebra
2. Computing **K**: key ideas

Computing the invariant measure

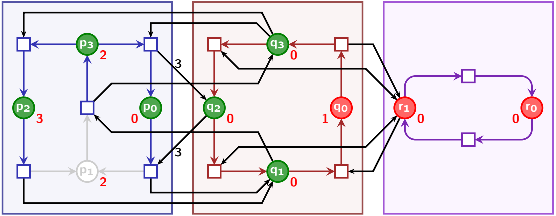
$$\mu = \mathsf{K} \prod_{\text{place } p} \mu_p^{\text{tokens in } p}$$



1. Computing μ_p: **PTime** using linear algebra
2. Computing **K**: key ideas
    - Consider unbounded places & bounded places;

Computing the invariant measure

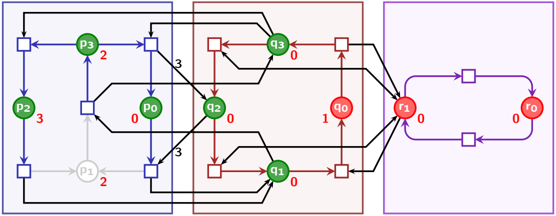
$$\mu = \mathbf{K} \prod_{\text{place } p} \mu_p^{\text{tokens in } p}$$



1. Computing μ_p: **PTime** using linear algebra
2. Computing $\mathbf{K}$: key ideas
   - Consider unbounded places & bounded places;
   - Split $\mathbf{K}$ into a finite sum $\sum_i \mathbf{K}_{\text{unbounded},i} \mathbf{K}_{\text{bounded},i}$; (≈ convolution)

Computing the invariant measure

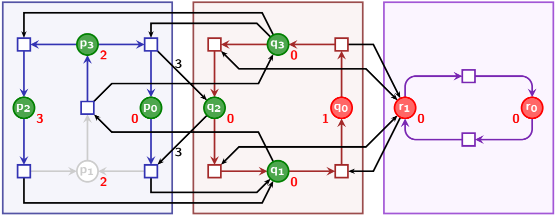
$$\mu = \mathbf{K} \prod_{\text{place } p} \mu_p^{\text{tokens in } p}$$



1. Computing μ_p: **PTime** using linear algebra
2. Computing $\mathbf{K}$: key ideas
   - Consider unbounded places & bounded places;
   - Split $\mathbf{K}$ into a finite sum $\sum_i \mathbf{K}_{\text{unbounded},i} \mathbf{K}_{\text{bounded},i}$; (≈ convolution)
   - Reachability set = lattice points in a finite-dimensional polytope;

Computing the invariant measure

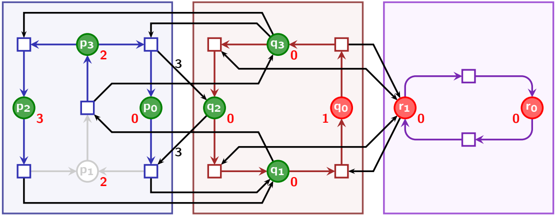
$$\mu = \mathbf{K} \prod_{\text{place } p} \mu_p^{\text{tokens in } p}$$



1. Computing μ_p: **PTime** using linear algebra
2. Computing **K**: key ideas
   - Consider unbounded places & bounded places;
   - Split **K** into a finite sum $\sum_i \mathbf{K}_{\text{unbounded},i} \mathbf{K}_{\text{bounded},i}$; (≈ convolution)
   - Reachability set = lattice points in a finite-dimensional polytope;
   - Use induction on the polytope's dimension & size.

Computing the invariant measure

$$\mu = \mathbf{K} \prod_{\text{place } p} \mu_p^{\text{tokens in } p}$$



1. Computing μ_p: **PTime** using linear algebra
2. Computing **K**: key ideas
   - Consider unbounded places & bounded places;
   - Split **K** into a finite sum $\sum_i \mathbf{K}_{\text{unbounded},i} \mathbf{K}_{\text{bounded},i}$; (≈ convolution)
   - Reachability set = lattice points in a finite-dimensional polytope;
   - Use induction on the polytope's dimension & size.

Theorem (Bouyer, Haddad & Jugé 2017)

One can check if a Π^3-net is ergodic in polynomial time.
If yes, one can compute the invariant measure in pseudo-polynomial time.

Conclusion & Future Work

A **new** class of stochastic Petri nets: (open) Π^3-nets

- allows handling resources in open queuing systems;
- product-form Petri nets;
- efficient characterisation of live Π^3-nets and
  of their qualitative/quantitative properties.

Conclusion & Future Work

A **new** class of stochastic Petri nets: (open) Π^3-nets

- allows handling resources in open queuing systems;
- product-form Petri nets;
- efficient characterisation of live Π^3-nets and
  of their qualitative/quantitative properties.

Further goals:

- handling multiple open layers; (done)
- tight complexity bounds for general Π^3-nets;
- other generalisations?

Conclusion & Future Work

A **new** class of stochastic Petri nets: (open) Π^3-nets

- allows handling resources in open queuing systems;
- product-form Petri nets;
- efficient characterisation of live Π^3-nets and of their qualitative/quantitative properties.

Further goals:

- handling multiple open layers; (done)
- tight complexity bounds for general Π^3-nets;
- other generalisations?

A short bibliography

- J. Jackson, *Jobshop-like queueing systems*, Management Science, 1963.
- W. Gordon & G. Newell, *Closed queuing systems with exponential servers*, Operations Research, 1967.
- A. Lazar & T. Robertazzi, *Markovian Petri net protocols with product form solution*, Performance Evaluation, 1991.
- J. Coleman et al., *Product form equilibrium distributions and a convolution algorithm for stochastic Petri nets*, Performance Evaluation, 1996.
- S. Haddad et al., *Product-form and stochastic Petri nets: A structural approach*, Performance Evaluation, 2005.
- J. Mairesse & H.-T. Nguyen, *Deficiency zero Petri nets and product form*, Fundamenta Informaticae, 2010.
- S. Haddad et al., *Synthesis and analysis of product-form Petri nets*, Fundamenta Informaticae, 2013.
- P. Bouyer et al., *Unbounded product-form Petri nets*, CONCUR, 2017.