Structural Methods for Performance Analysis of PNs

S. Haddad
LAMSADE CNRS-UPRESA 7024 / University Paris Dauphine

What does “structural” mean ?

A tool for structural analysis : partial flows

What is product-form analysis ?

Revisiting product-form stochastic PNs

Other applications of partial flows

A first case study for structural analysis : the law of Little

<u>Long-run analysis parameters</u>

λ the mean number of arrivals per unit of time

W the mean service time of a client

L the mean number of clients in service

The law of Little

Under general conditions *(J. Little 61)*,

- The limits λ , **W** , **L** exist almost surely
- and the following relation holds :

$$\mathbf{L} = \lambda \,.\, \mathbf{W}$$

<u>Intuitive justification</u>

If some stationnarity is reached,

the input throughput λ should be equal to

the output throughput **L . (1 / W)**

Hypotheses for the law of Little

- Different stochastic hypotheses such like renewal conditions on the system behaviour
- Important structural result : as soon as the limits exist, the equality holds *(S. Stidham Jr 72)*
- First interpretation

Structural

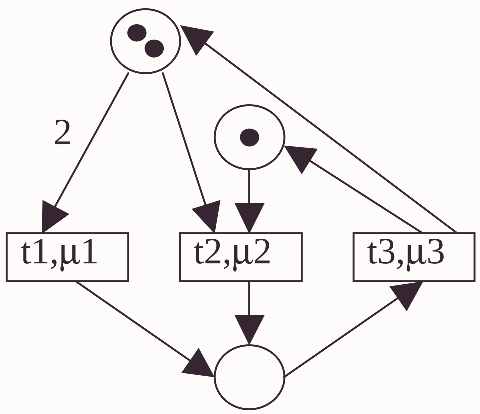
$\Leftrightarrow$

Independent of stochastic parameters

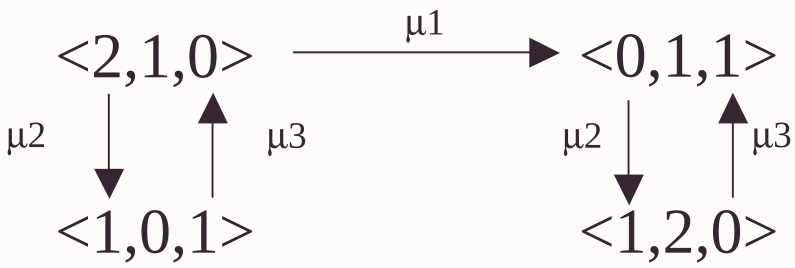
Another case study for structural analysis : Ergodicity

A stochastic process is ergodic iff
it ultimately reaches a stationnary state distribution
independent of its initial distribution

A (finite state) SPN

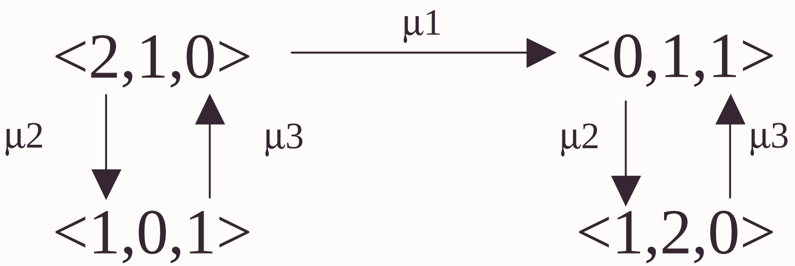


Its reachability graph
(a Markov chain)



The equilibrium distribution

The Markov chain



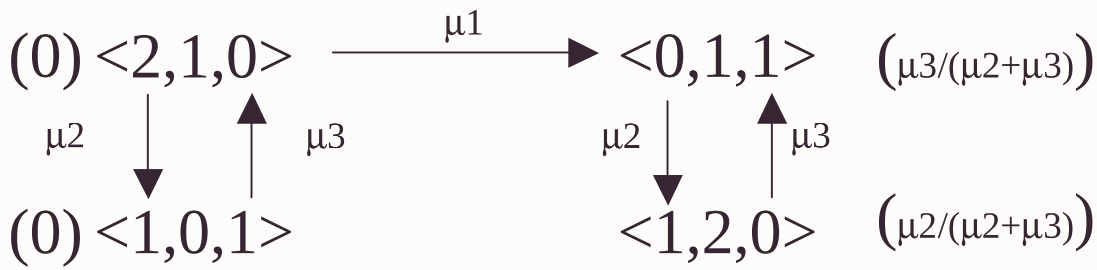
Its infinitesimal generator Q

$$\begin{pmatrix} -\mu1-\mu2 & \mu2 & \mu1 & 0 \\ \mu3 & -\mu3 & 0 & 0 \\ 0 & 0 & -\mu2 & \mu2 \\ 0 & 0 & -\mu3 & \mu3 \end{pmatrix}$$

The equations

$$\pi.Q = 0$$
$$\pi.1^T = 1$$

The solution



Ergodicity

• A finite Markov chain is ergodic iff it has a single sink strongly connected component.

• A structural property of the Markov chain but …

• a behavioural property of the SPN

• Second interpretation

Structural

$\Leftrightarrow$

Independent of the net semantics

Flows of a Petri Net

- The incidence matrix C = Post-Pre measures the token changes induced by a sequence σ

- $m[\sigma>m' \Rightarrow m' = m + C . \vec{\sigma}$ where $\vec{\sigma}$ is the vector of occurrences of transitions in σ

- Let v be a vector on places such that $v . C = 0$ (a flow)

- Then v.m' = v.m for any sequence, thus v.m is invariant and equal to $v.m_0$

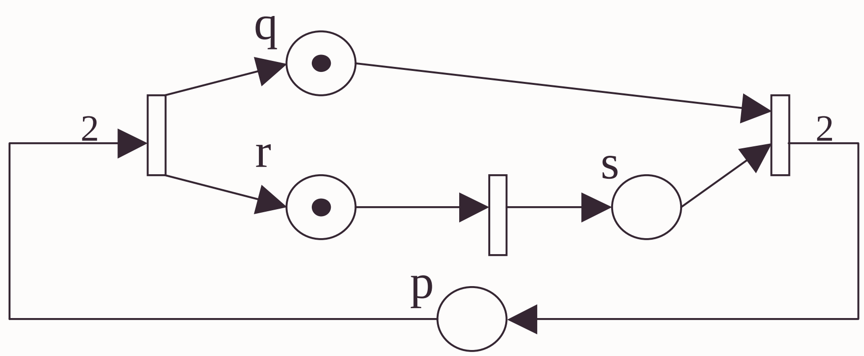
Applications of Flows

Bounds on places

$m(p)+2.m(q)=2$
$\Rightarrow m(p) \leq 2$

Search of implicit places

q - r - s and m_0
$\Rightarrow$ q is an implicit place



Efficient characterization of a superset of the reachable states

$m(p) + 2.m(q) = 2$ and $m(p) + 2.m(r) + 2.m(s) = 2$

Computation of Flows

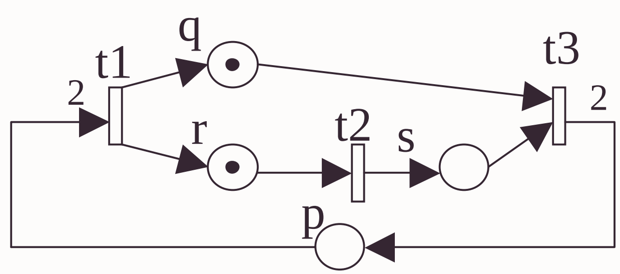
- Computing flows means computing a generative family of flows
- The algorithm depends on the choice of the coefficients:
  - Rational coefficients lead to a polynomial time Gaussian elimination
  - Positive rational coefficients lead to Farkas algorithm (the size of the family is non polynomial). In practice, the algorithms behave variously but time and space complexities are neglectible w.r.t. the state space generation when the initial marking grows

- Generalization of computations to high-level nets

Partial Flows

A partial flow is given by:

- a vector v on places
- a subset of transitions T'
- a vector r on T'
- such that $v.C_{T'} = r$



v = r + s

T' = {t2,t3}

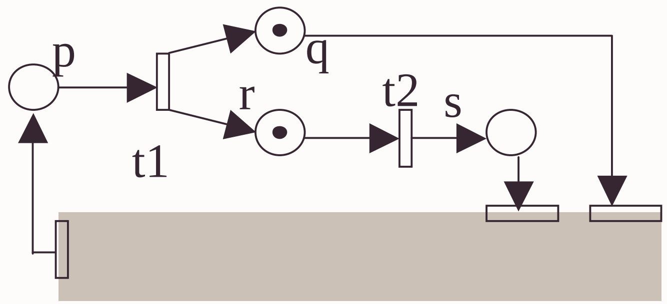
r = -t3

Interpretation

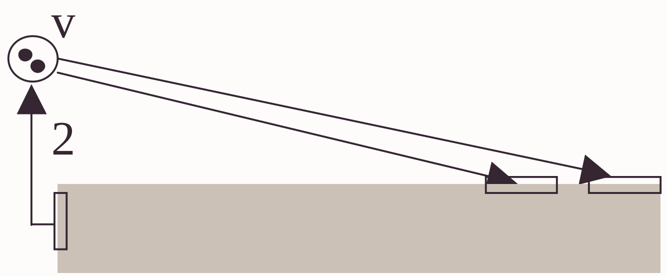
- the subset T' is the space of constraints
- r is the value of the constraints
- v is the solution of $\exists ?\ v$ s.t. $v.C_{T'} = r$

Why looking for partial flows ?

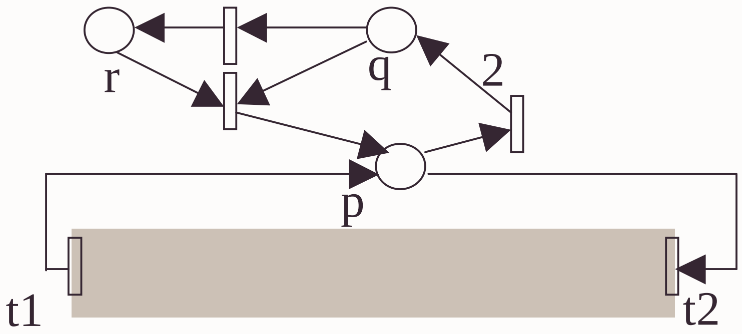
Abstraction of a component



v = 2.p + q + r + s
T’ = {t1,t2}
r = 0



Activities witness



v = p +1/2. q +1/2. r
T’ = T
r = t1 - t2

Computation of partial flows

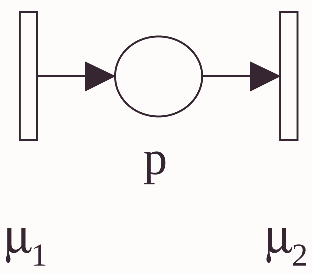
- Computing partial flows means also computing a generative family of flows
- In case of rational coefficients this generative family is obtained by the sum of:
  - the generative family of the homogeneous equation
  - any solution of the original equation
- In case of positive rational coefficients this generative family is obtained by the sum of:
  - the generative family of the homogeneous equation
  - a solution of the equation per minimal support
- Same complexity as for flows computation

Context of the product-form analysis

- A finite-state Markiovan system (queuing network, SPN,...) can be studied by analysing the generated Markov chain …
- However the size of the Markov chain is at least as big as the size state space which prohibits the study of large systems
- An alternative for “simple” systems is to derive the stationnary distribution from *the structure of the model* and *the stochastic parameters*
- The product-form analysis is based on a decomposition of the model

The key ideas of the product-form analysis (1)

Analogy with the simple queue



$$\pi(n.p) = K \, . \, (\mu_1 / \mu_2)^n$$

General form of the solution

$$\pi(m) = K \, . \prod_{p \in P} (f_p(\mu_1,\ldots,\mu_{nt}))^{m(p)}$$

The key ideas of the product-form analysis (2)

Decomposing the subset of transitions

T_a	T_b	$Q = Q_a + Q_b$

Q_a obtained from Q by cancelling the rates of T_b

Solving a more constrained system

(local balance equations)

$$\pi \,.\, Q_a = 0 \text{ and } \pi \,.\, Q_b = 0 \Rightarrow \pi \,.\, Q = 0$$

The key ideas of the product-form analysis (3)

Combining the ideas

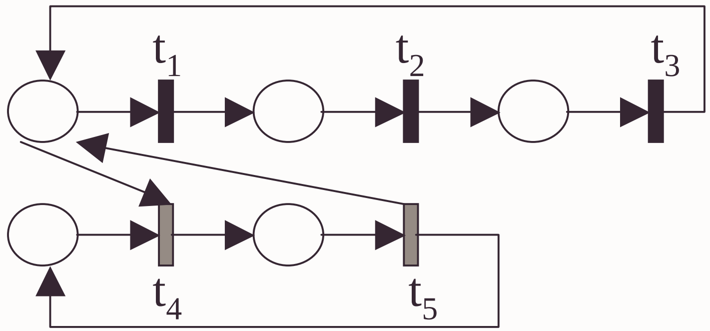
- $\pi(m) = K \,.\, f_a(m,\mu_1,\ldots,\mu_{na}) \,.\, f_b(m,\mu_{na+1},\ldots,\mu_{na+nb})$
- $f_a(m,\mu_1,\ldots,\mu_{na}) = \prod_{p \in P} (f_{a,p}(\mu_1,\ldots,\mu_{na}))^{m(p)}$
- $t \in T_b \; m[t>m' \Rightarrow f_a(m,\mu_1,\ldots,\mu_{na}) = f_a(m',\mu_1,\ldots,\mu_{na})$

Problem How to ensure such a form for the distribution ?

Product-form stochastic Petri nets:

Activities *(Henderson et al 89)*

- Subsets of transitions model concurrent client activities *(synchronised by resources)*
- Each component is a set of rhythms *(T-invariants)*



<u>Job</u>

t_1 : compute

t_2 : print

t_3 : end job

<u>Interactive task</u>

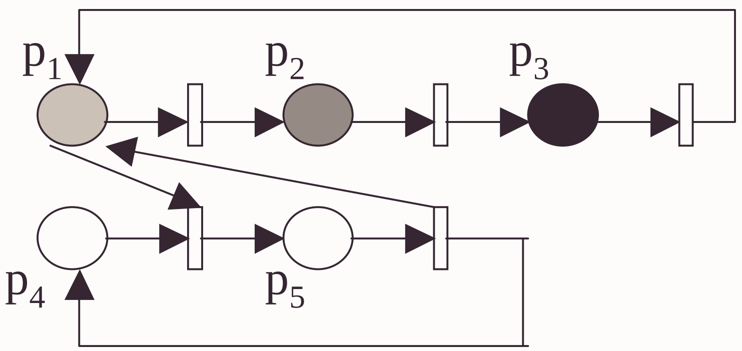
t_4 : begin dialog

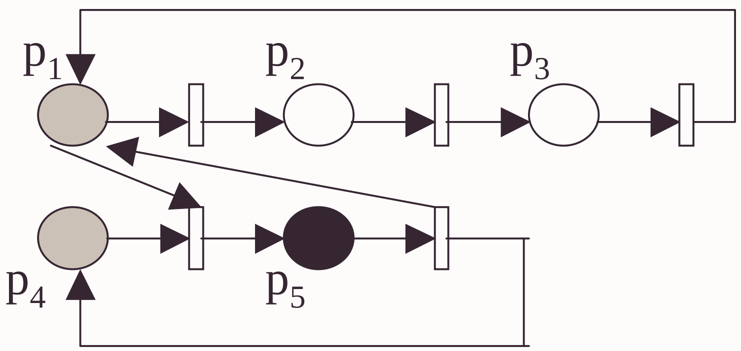
t_5 : end dialog

Product-form stochastic Petri nets:
Resources

- Places are resources
- To each "state" of an activity, is associated a bag of resources

p_1 idle processors	p_4 idle users
p_2 computation	p_5 windows
p_3 printing	





Product-form stochastic Petri nets:

The additional condition

Compatibility between:

- the relative throughput of transitions in an activity
- the stationnarity of the marking distribution

leads to a rank equation involving logarithms of rates

$$\text{Rank}(C) = \text{rank}(C')$$

$$\begin{pmatrix} -1 & 0 & 1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \quad \begin{pmatrix} -1 & 0 & 1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \\ \ln(\mu2/\mu1) & \ln(\mu3/\mu1) & \ln(\mu1/\mu3) & \ln(\mu5/\mu4) & \ln(\mu4/\mu5) \end{pmatrix}$$

Product-form stochastic Petri nets:
Discussion

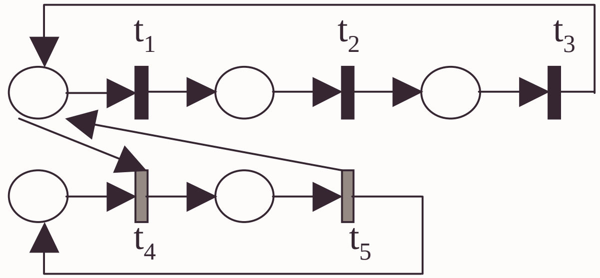
- Technical condition without intuitive interpretation
- Existence of product-form depending on rates
  (opposite to the typical cases)
- Sensitive to numerical approximations
- Theoretical occurrence of condition satisfaction neglictible
(but it often occurs in practice)

↳ need of a structural characterization

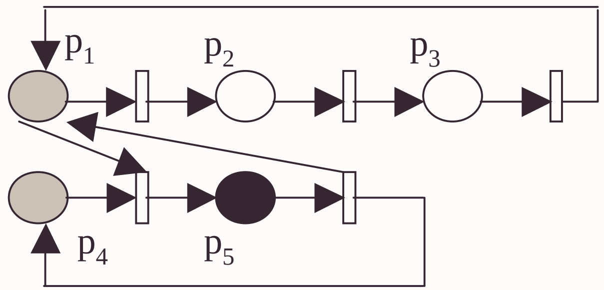
Revisiting product-form stochastic PNs *(Joint work Paris9 - Saragossa -Torino 2000)*

Presence in the model of :

- the activities



- the resources



But where are the client states ?

Looking for the client states

Product-form does not require to identify the absolute number of clients in any “state” …

but only a relative number *(up to an additive constant)*

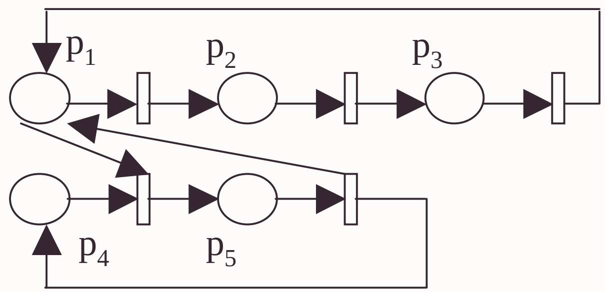
<u>First attempt</u> Client states are represented by places

<u>Job activity</u>

p_2 : job computing

p_3 : job printing

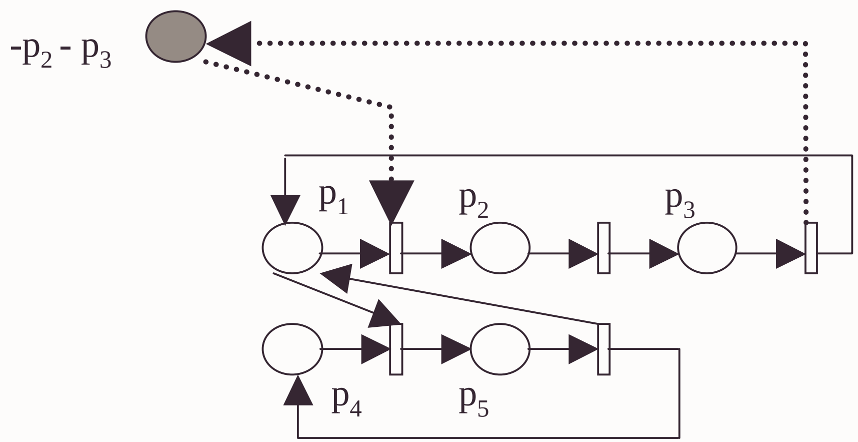
?? : idle job



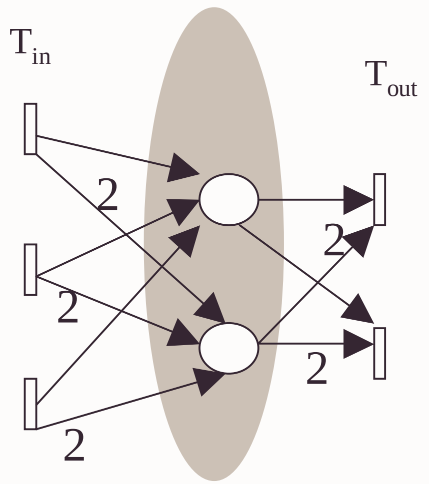
Client states are partial flows

Number of idle jobs :

- increases with the firing of t_3
- decreases with the firing of t_1
- left invariant by the firing of another transition



Client states are partial flows (2)



Partial flow associated to a state

- v vector of places s.t. v.C = r
- with $r = T_{in} - T_{out}$

Structural characterisation of product-form SPNs

Any client state of any activity
(i.e. an input-output bag)
must be represented by a partial flow

- Existence of product-form whatever the rates may be
- Sufficient and <u>necessary</u> condition for local balance equations
- Intuitive interpretation
- Polynomial time checking algorithm
- Extension of possible rates dependency

  (e.g. global dependency of rates of one activity
  on the "number" of clients in a state of another activity)

Aggregation and Tensorial Product *(S. Haddad , P. Moreaux 95,96)*

Performance analysis of high-level Petri nets
(Stochastic Well-formed Nets)

- Reducing the state-based analysis by combining :
  - Aggregation *(symbolic reachability graph)*
  - Tensorial decomposition in synchronous and asynchronous contexts
- Asynchronous decomposition require abstraction of components
  - ↳ partial flows
- Ensuring weak dependency between components
  - ↳ partial flows

Conclusion

- Refined notion of structure
- Performance analysis : an application field for structural methods
- Structural analysis useful for combining behavioural methods
- Interplay between verification and evaluation