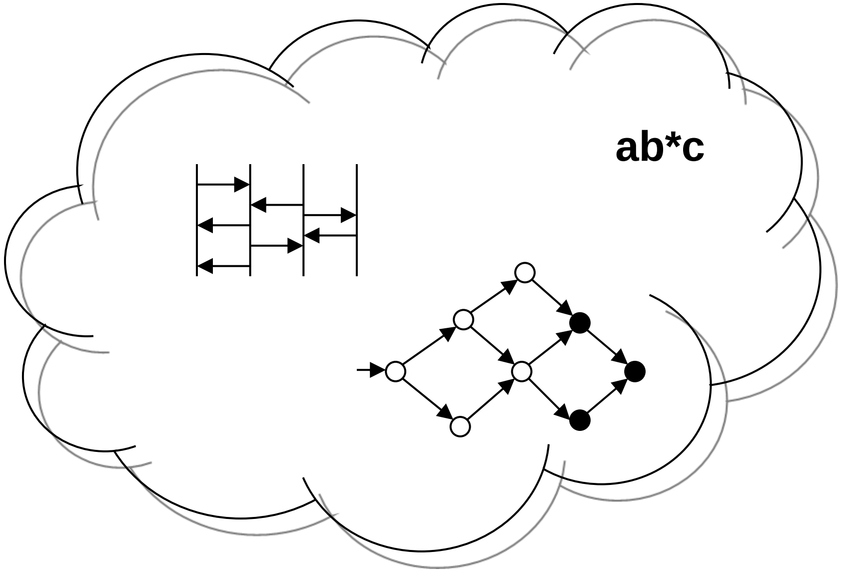
Synthesis of Petri nets from scenarios (pomsets)

Jörg Desel
Robert Lorenz, Robin Bergenthum,
Gabriel Juhás, Sebastian Mauser

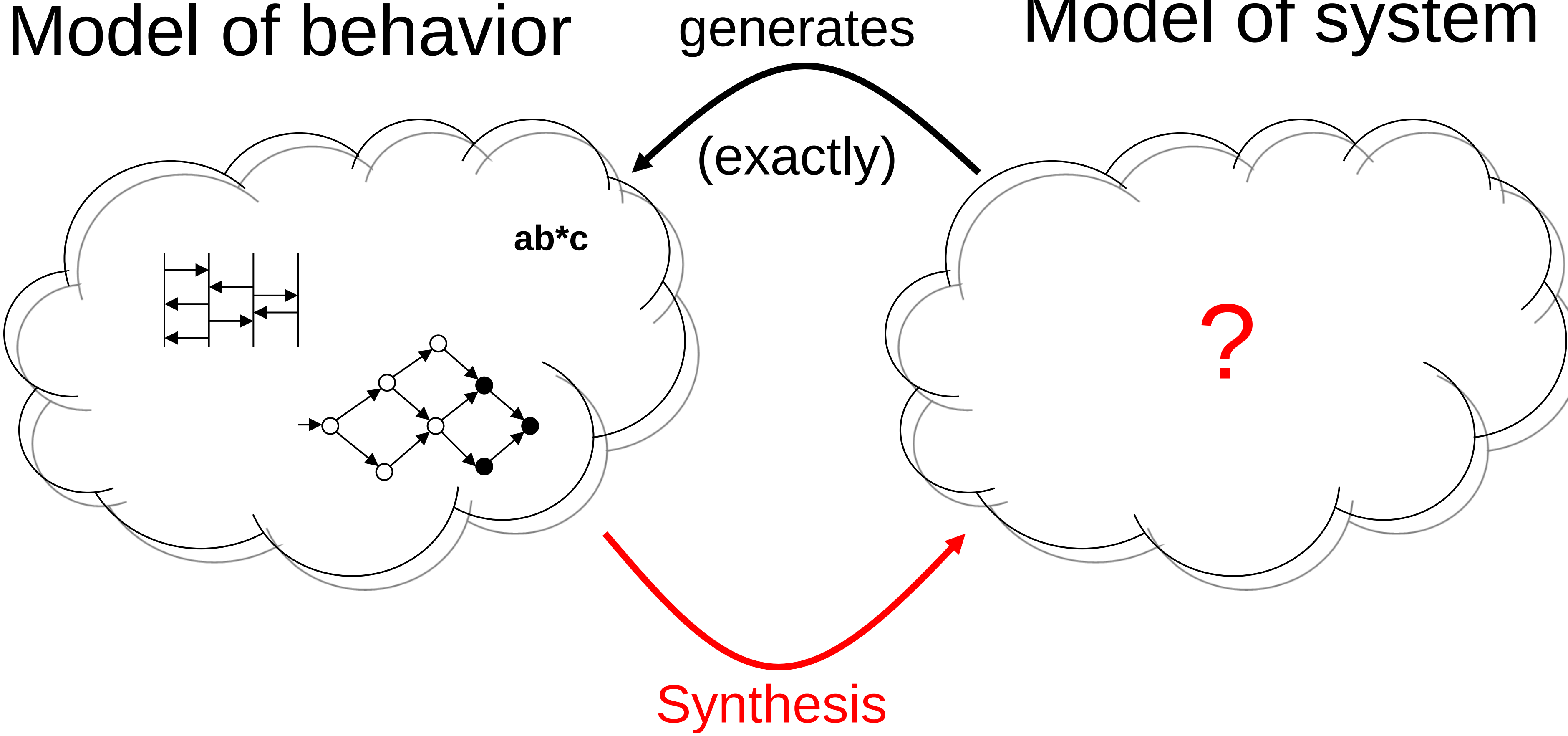
Given:
Model of behavior



Specified/observed behavior

Given:
Model of behavior

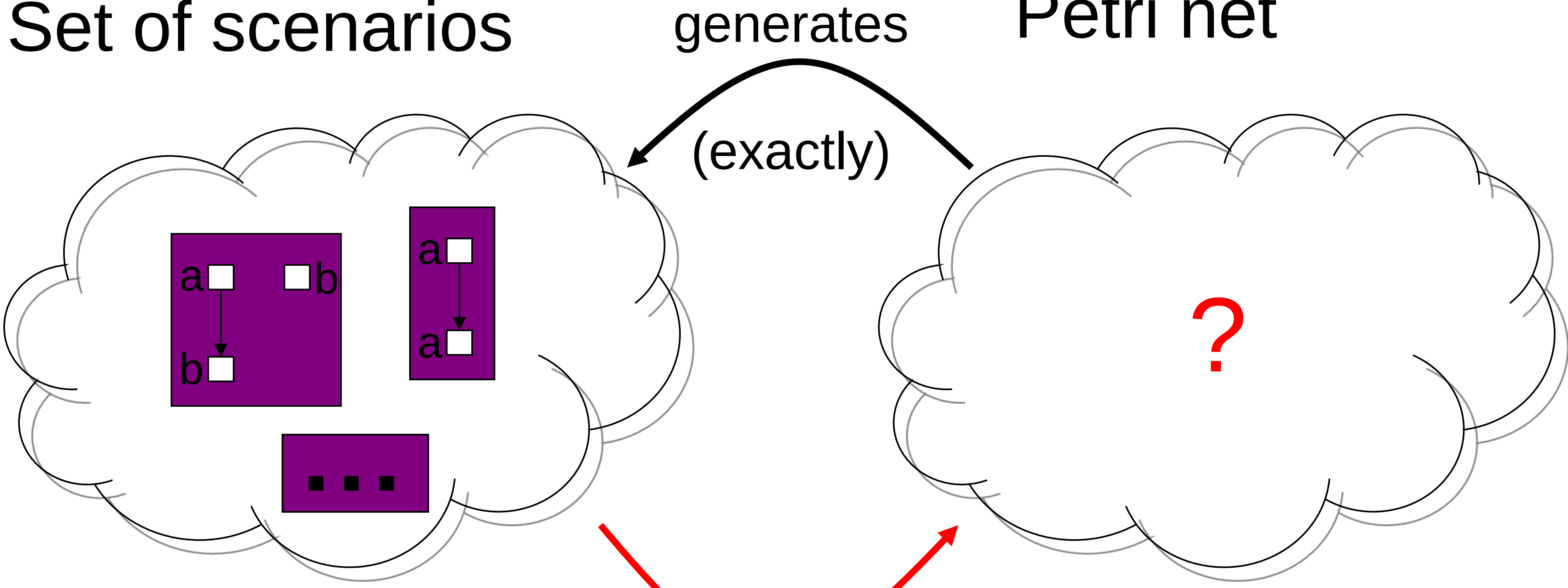
Wanted:
Model of system



Specified/observed behavior

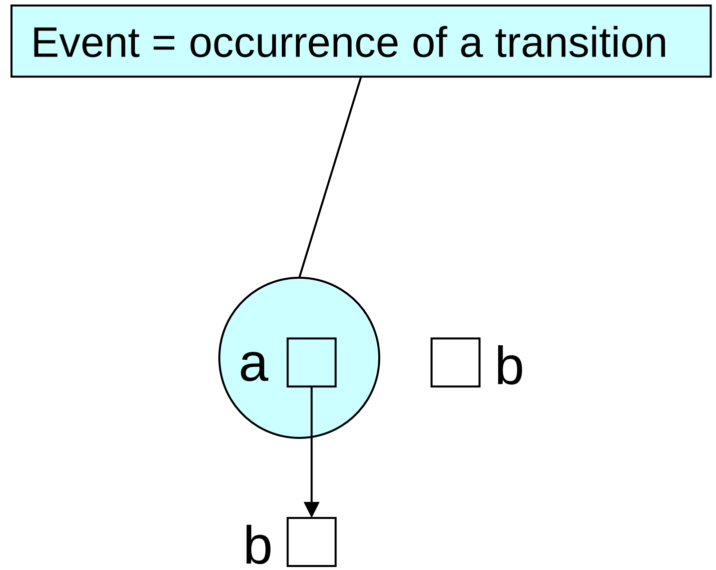
Given:
Set of scenarios

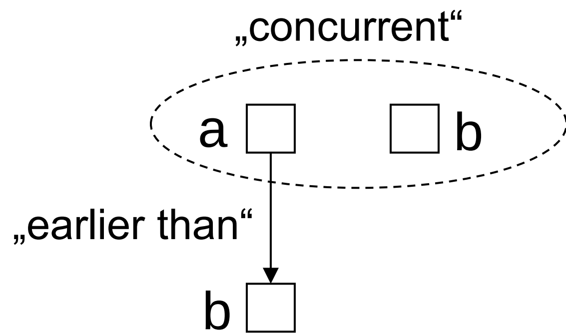
Wanted:
Petri net

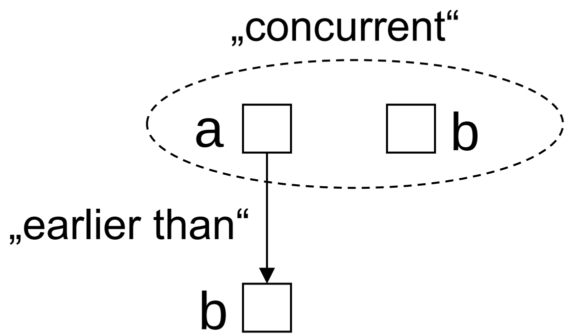


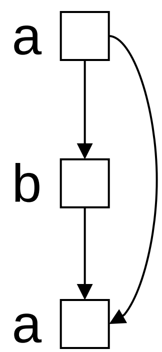
Examples:

- **Sequences**
- **Step sequences**
- **MSCs**
-

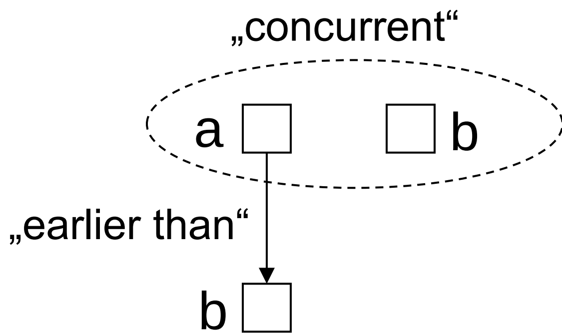

Event = occurrence of a transition
a
b
b


„concurrent“
a
b
„earlier than“
b

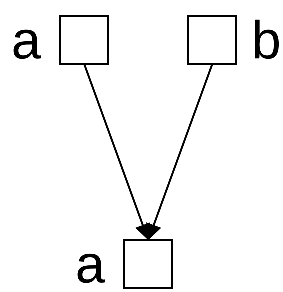


Sequences are pomsets: 

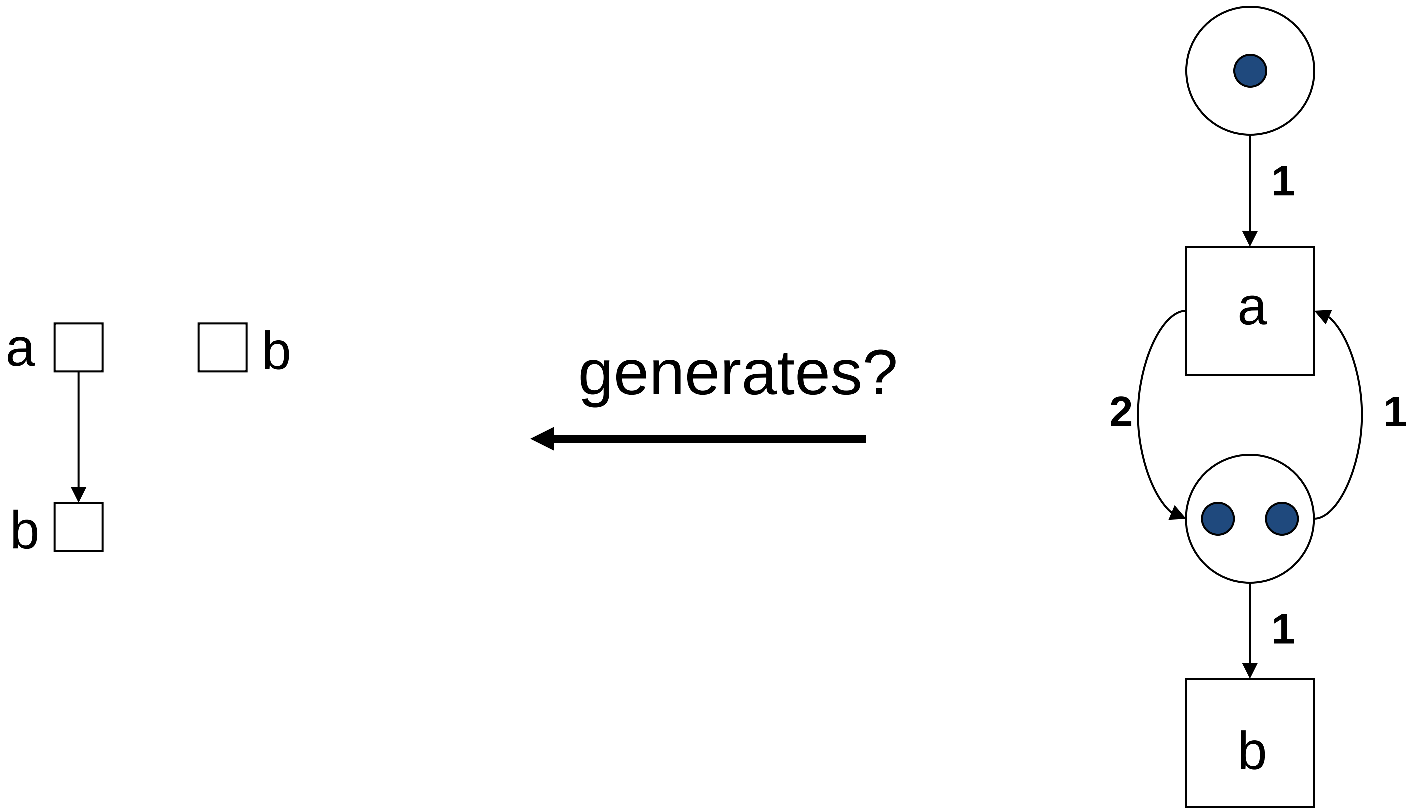
corresponds to aba



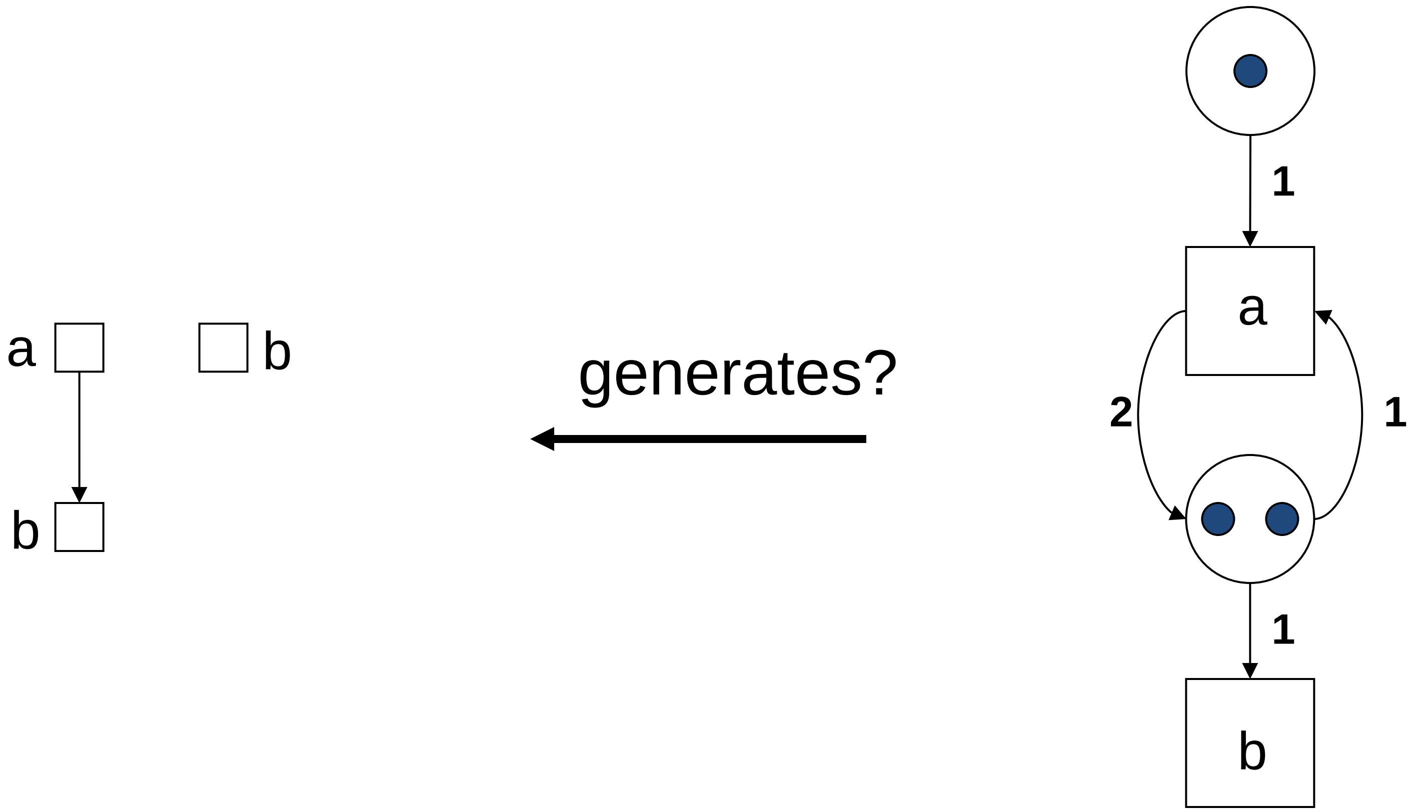
Step sequences are pomsets:



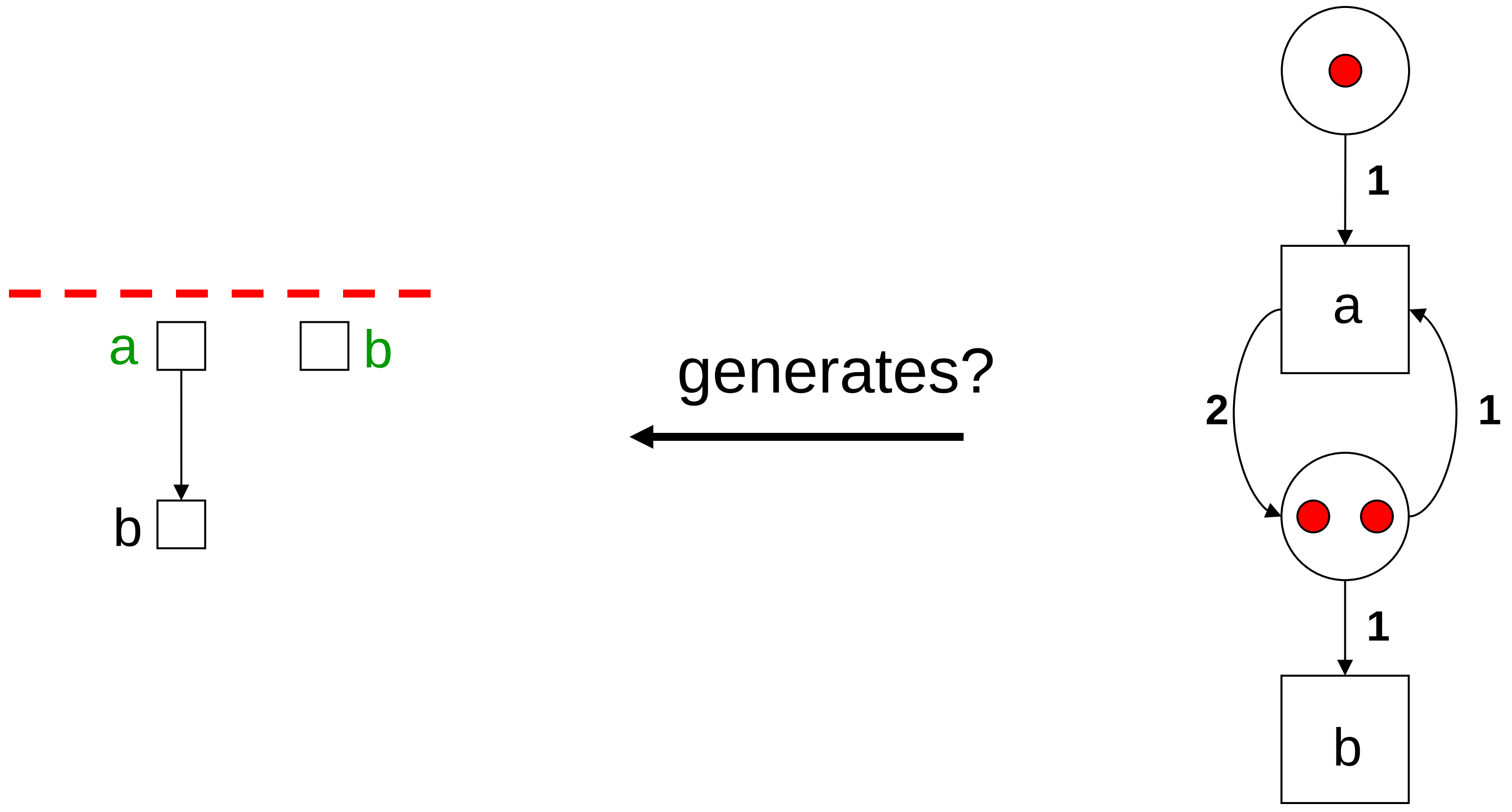
corr. to $(a+b)a$



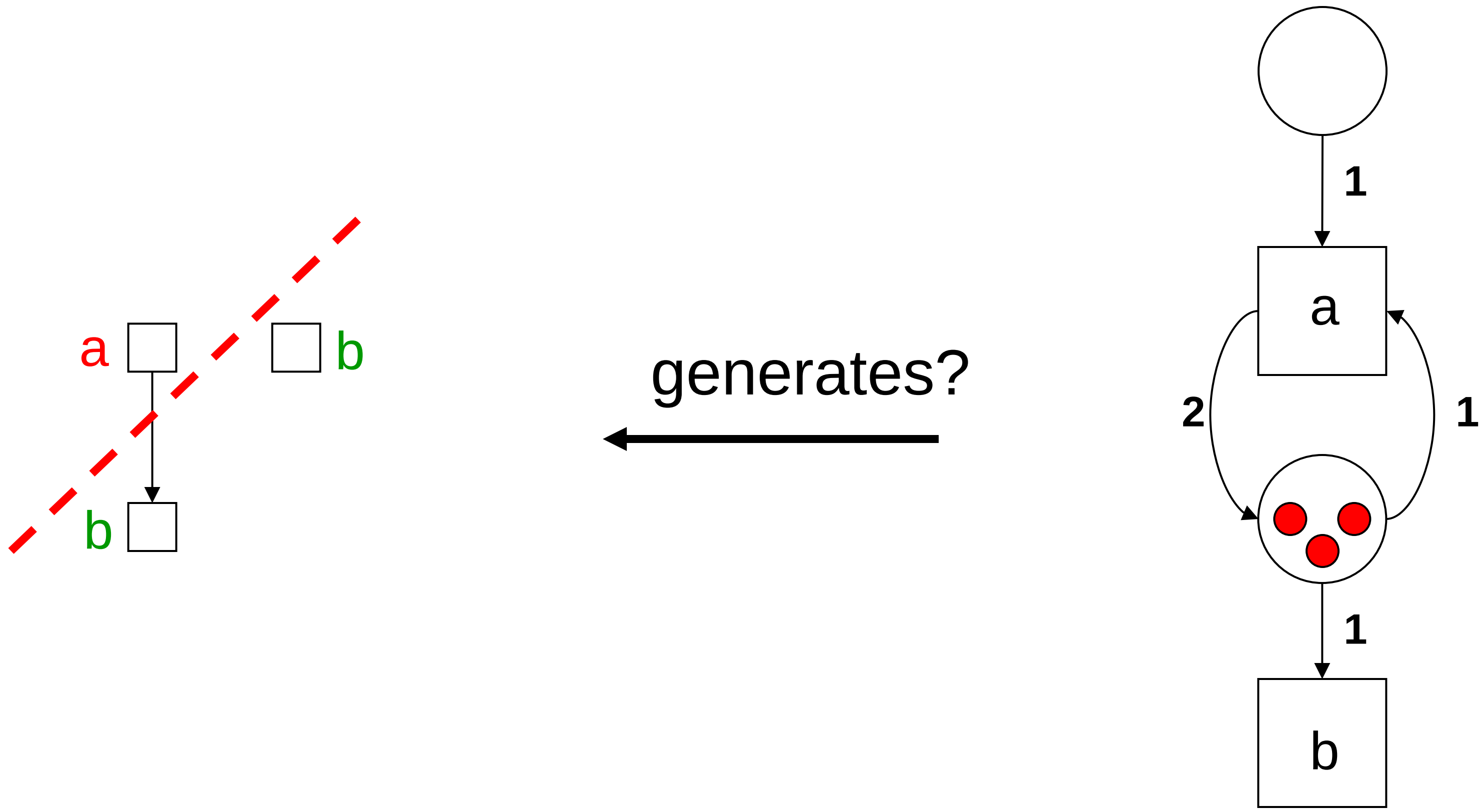
a
b
b
generates?
1
a
2
1
1
b



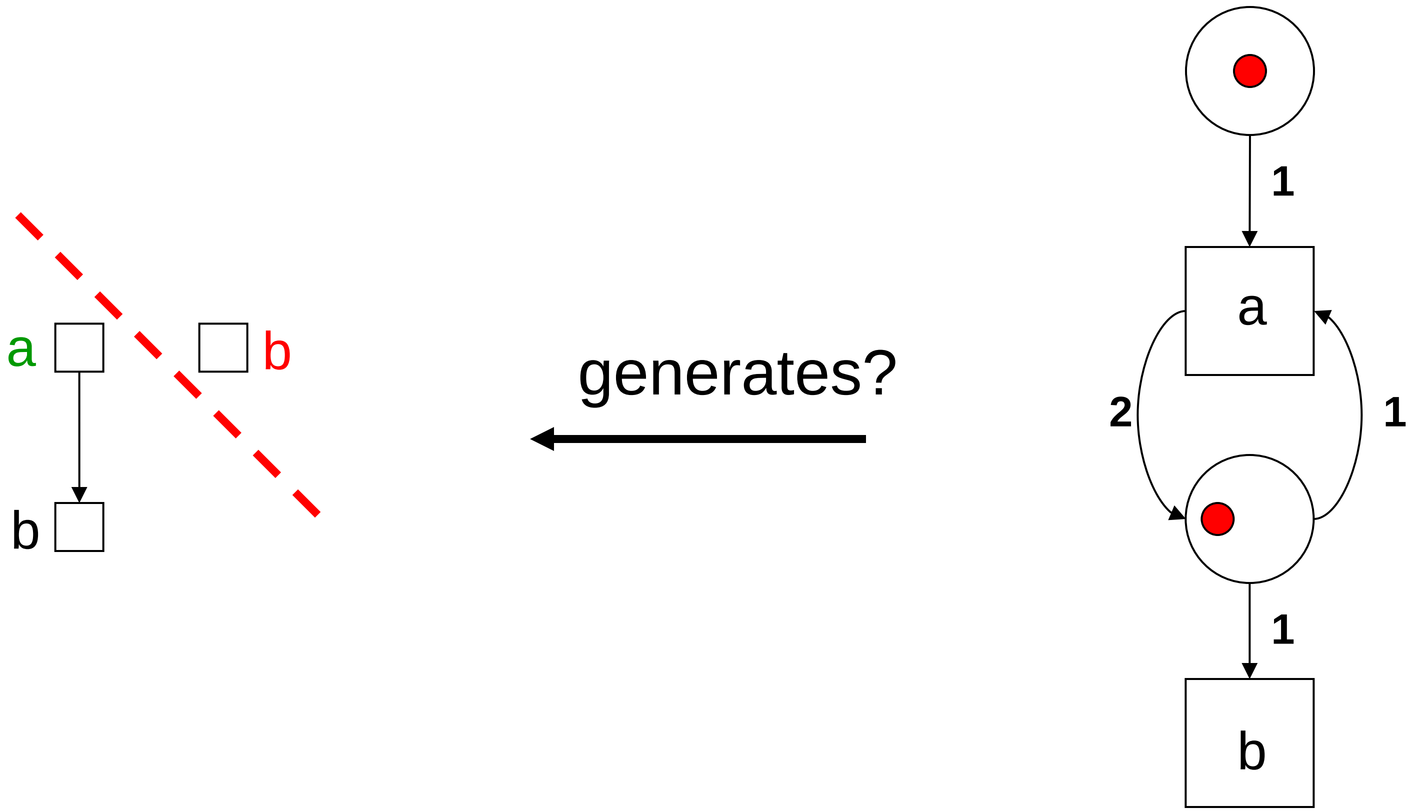
Each prefix enables following step of concurrent transitions



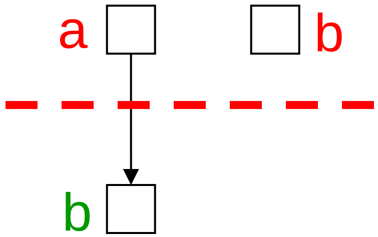
Each prefix enables following step of concurrent transitions



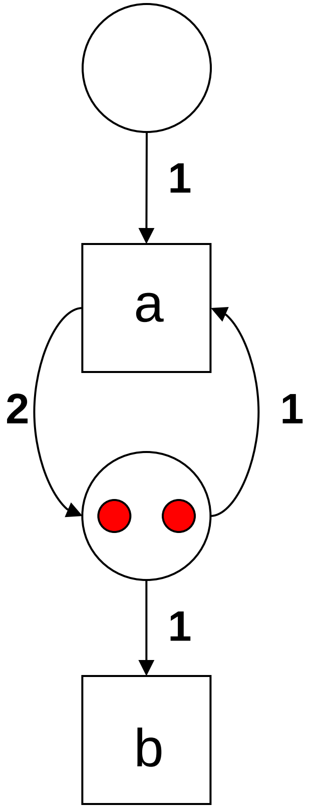
Each prefix enables following step of concurrent transitions



Each prefix enables following step of concurrent transitions



generates?

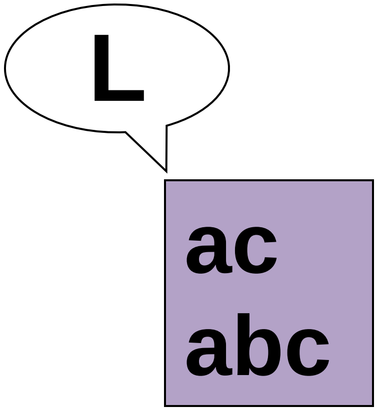


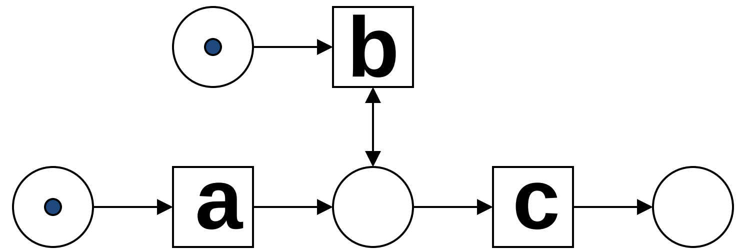
Each prefix enables following step of concurrent transitions

Most simple case: language of sequences

L

ac
abc

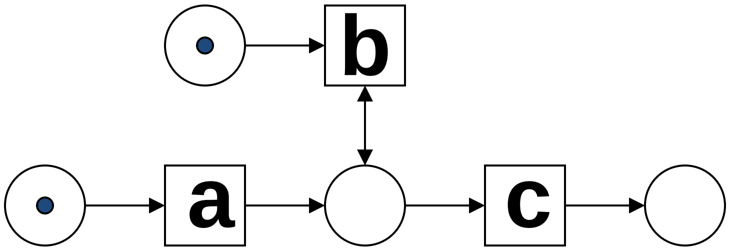




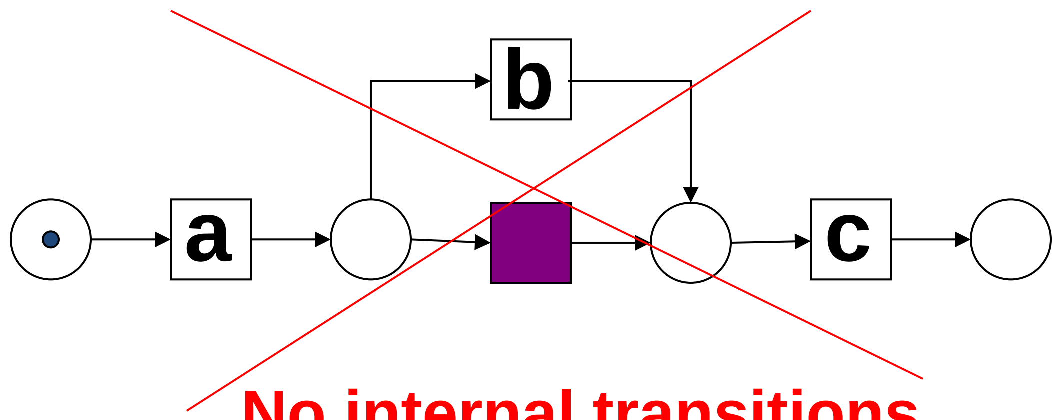
Exact solution (if possible)

L

ac
abc



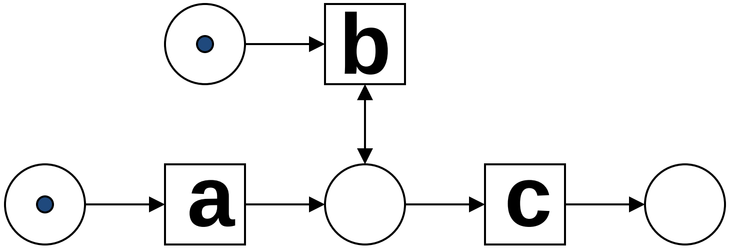
Exact solution (if possible)



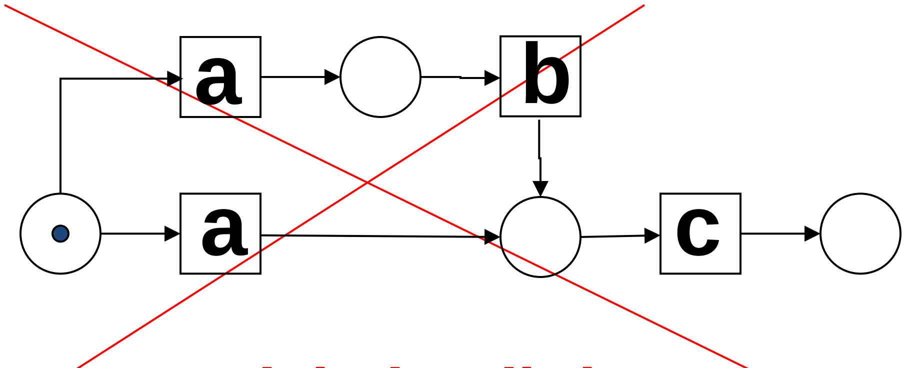
No internal transitions

L

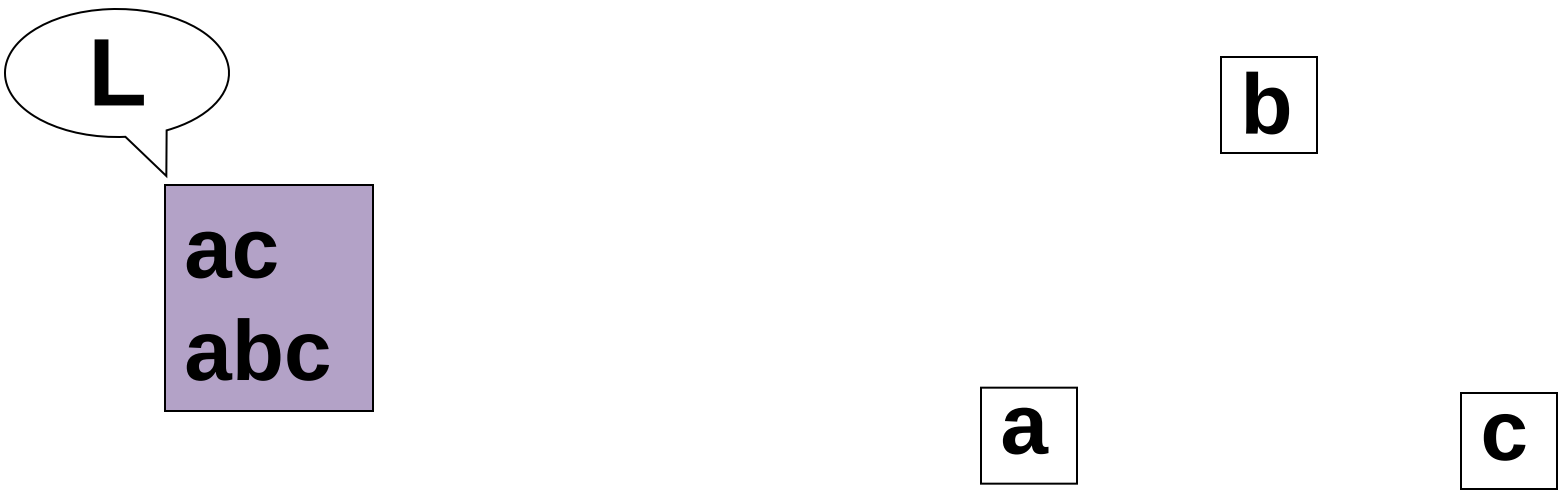
ac
abc



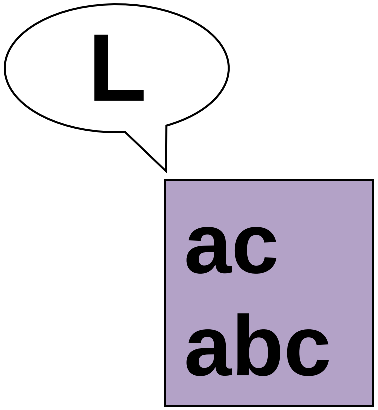
Exact solution (if possible)

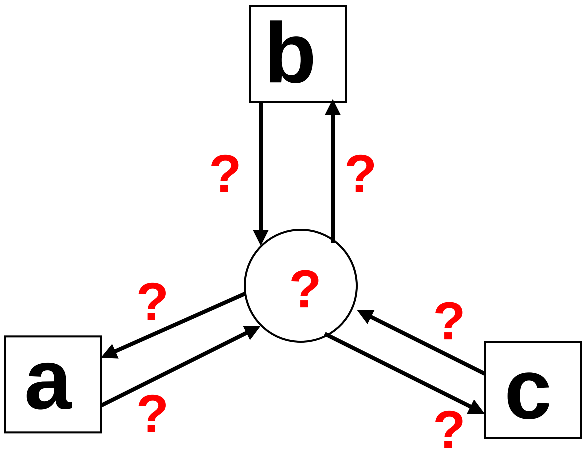


No label-splitting



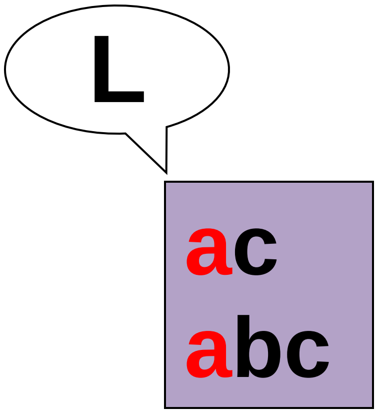
Start with an empty set of places

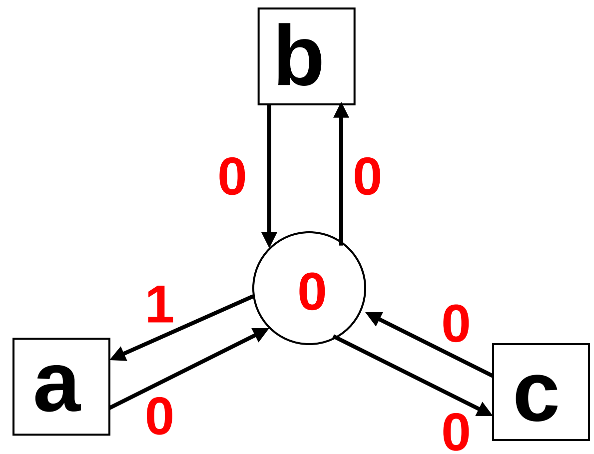




Add places

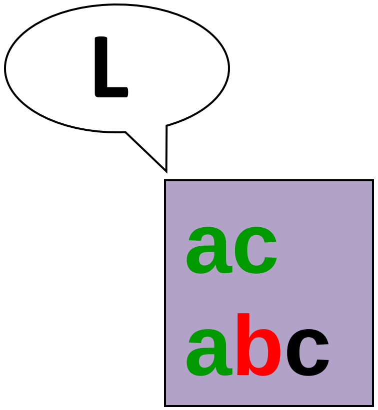
p non-feasible

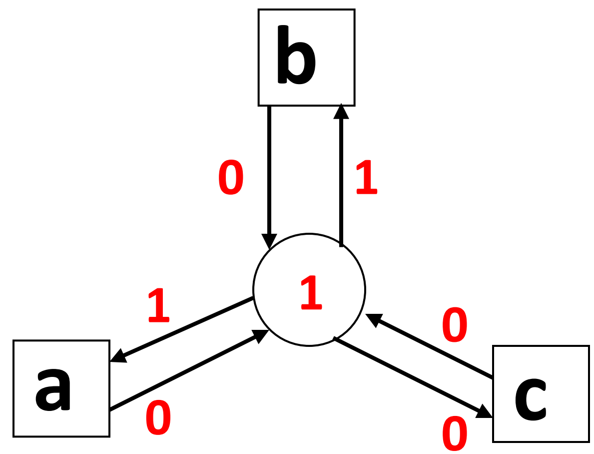




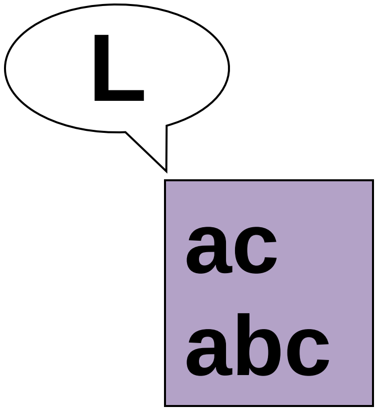
Add places

p non-feasible

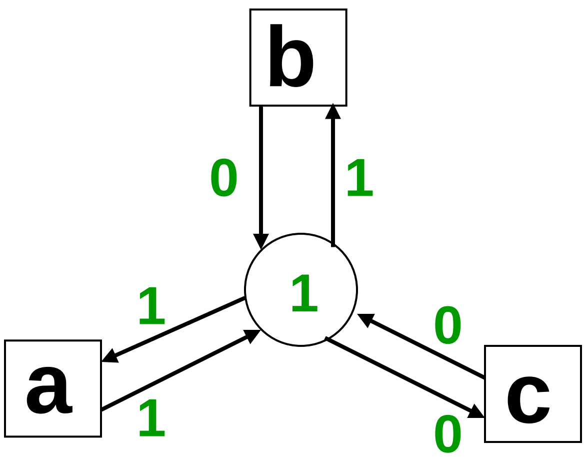




Add places

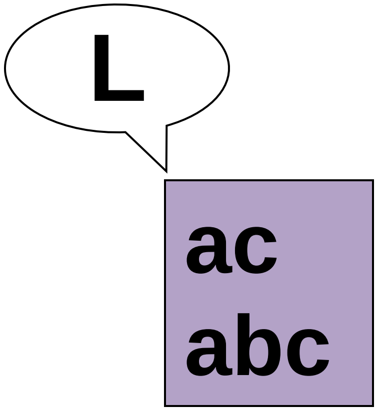


p feasible

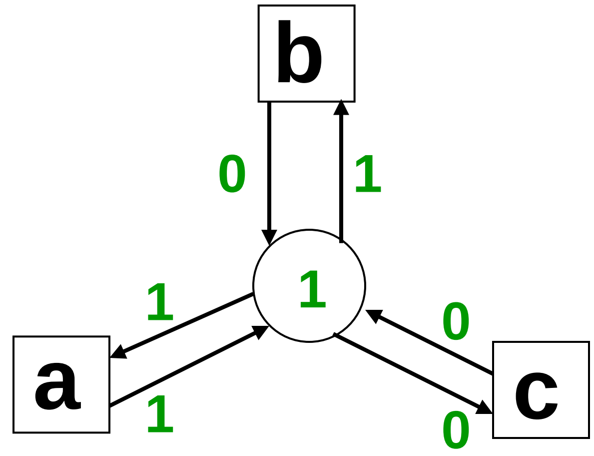


Add places

such that the net still generates L



p feasible



Net with all feasible places:

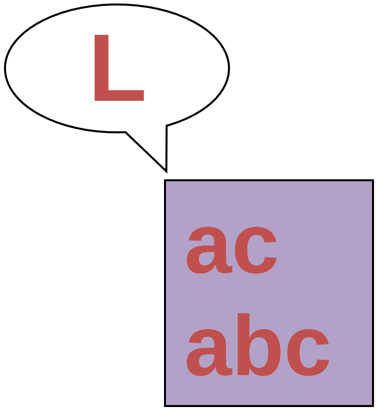
saturated feasible net N_{sat}

p feasible

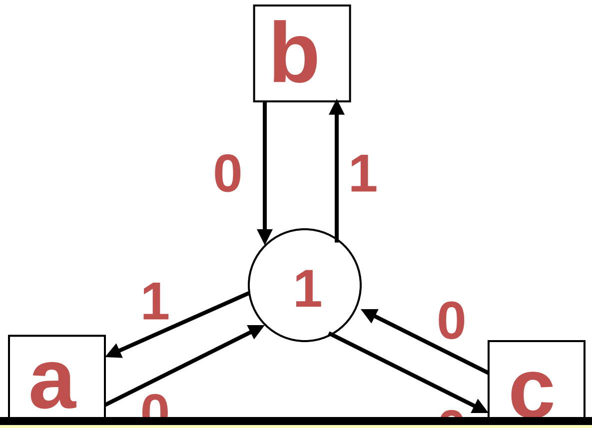
<u>Theorem</u>

$L(N_{sat})$ is the smallest net language with $L \subseteq L(N_{sat})$

Net with all feasible places:
saturated feasible net N_{sat}



p feasible

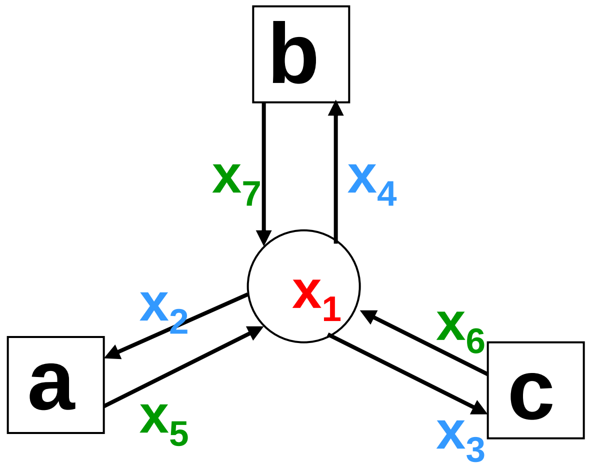


How to compute feasible places?

Net with all feasible places: saturated feasible net N_{sat}

ac
abc

$p=(x_1,x_2,x_3,x_4,x_5,x_6,x_7)$
tuple of non-negative integers



---

p feasible

$\Leftrightarrow$

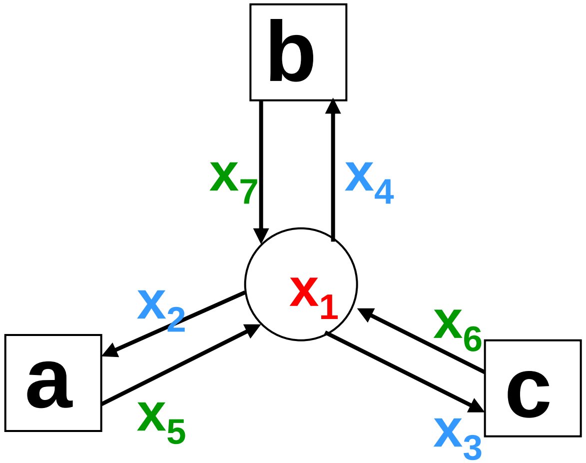
Each proper prefix w enables the subsequent transition t

ac
abc

→

ε enables a
a enables b
a enables c
ab enables c

p=(x_1,x_2,x_3,x_4,x_5,x_6,x_7)
tuple of non-negative integers



---

p feasible

⇔

Each proper prefix w enables the subsequent transition t

ac

abc

ε enables a

a enables b

a enables c

ab enables c

$x_1 \geq x_2$

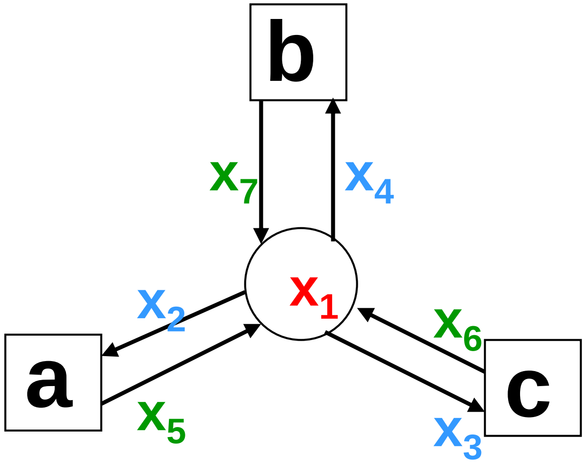
$x_1 - x_2 + x_5 \geq x_4$

$x_1 - x_2 + x_5 \geq x_3$

$x_1 - x_2 + x_5 - x_4 + x_7 \geq x_3$

$p=(x_1,x_2,x_3,x_4,x_5,x_6,x_7)$

tuple of non-negative integers



p feasible

$\Leftrightarrow$

Each proper prefix w enables the subsequent transition t

ac
abc

ε enables a
a enables b
a enables c
ab enables c

$x_1 \geq x_2$	$x_1 - x_2 \geq 0$
$x_1 - x_2 + x_5 \geq x_4$	$x_1 - x_2 - x_4 + x_5 \geq 0$
$x_1 - x_2 + x_5 \geq x_3$	$x_1 - x_2 - x_3 + x_5 \geq 0$
$x_1 - x_2 + x_5 - x_4 + x_7 \geq x_3$	$x_1 - x_2 - x_3 - x_4 + x_5 + x_7 \geq 0$

p feasible

$\Leftrightarrow$

Each proper prefix w enables the subsequent transition t

$\Leftrightarrow$

$$\mathbf{A_L p \geq 0}$$

ac
abc

→

ε enables a
a enables b
a enables c
ab enables c

→

$x_1 \geq x_2$	$x_1 - x_2$	≥ 0
$x_1 - x_2 + x_5 \geq x_4$	$x_1 - x_2 - x_4 + x_5$	≥ 0
$x_1 - x_2 + x_5 \geq x_3$	$x_1 - x_2 - x_3 + x_5$	≥ 0
$x_1 - x_2 + x_5 - x_4 + x_7 \geq x_3$	$x_1 - x_2 - x_3 - x_4 + x_5 + x_7$	≥ 0

Non-negative integer solution of $A_L p \geq 0$:
transition-region

[A_L may have infinite many rows]

ac
abc

ε enables a
a enables b
a enables c
ab enables c

$x_1 \geq x_2$
$x_1 - x_2 + x_5 \geq x_4$
$x_1 - x_2 + x_5 \geq x_3$
$x_1 - x_2 + x_5 - x_4 + x_7 \geq x_3$

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & -1 & 1 & 0 & 1 \end{pmatrix} \bullet \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} \geq 0$$

Non-negative integer solution of $A_L p \geq 0$:
transition-region

[A_L may have infinite many rows]

ac
abc

→

ε enables a
a enables b
a enables c
ab enables c

<u>Theorem</u>

each transition region generates a feasible place and vice versa

Non-negative integer solution of $A_L p \geq 0$: transition-region

[A_L may have infinite many rows]

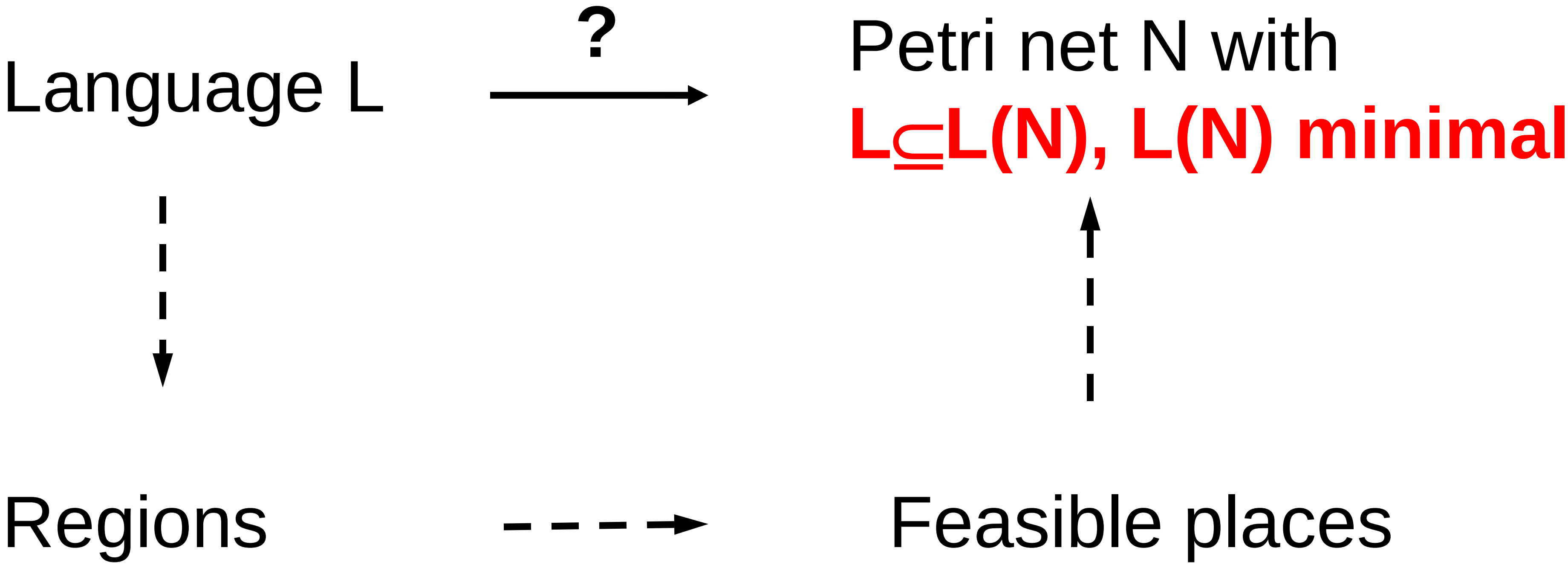
Language L $\xrightarrow{?}$ Petri net N with **L=L(N)**

Language L $\xrightarrow{?}$ Petri net N with **L=L(N)**

What if no such net N exists?

Language L $\xrightarrow{?}$ Petri net N with

$L \subseteq L(N)$, $L(N)$ minimal


?
Language L
Petri net N with
L⊆L(N), L(N) minimal
Regions
Feasible places

Language L $\xrightarrow{?}$ Petri net N with

L⊆L(N), L(N) minimal

Non-negative integer solutions of linear inequation system

transition regions

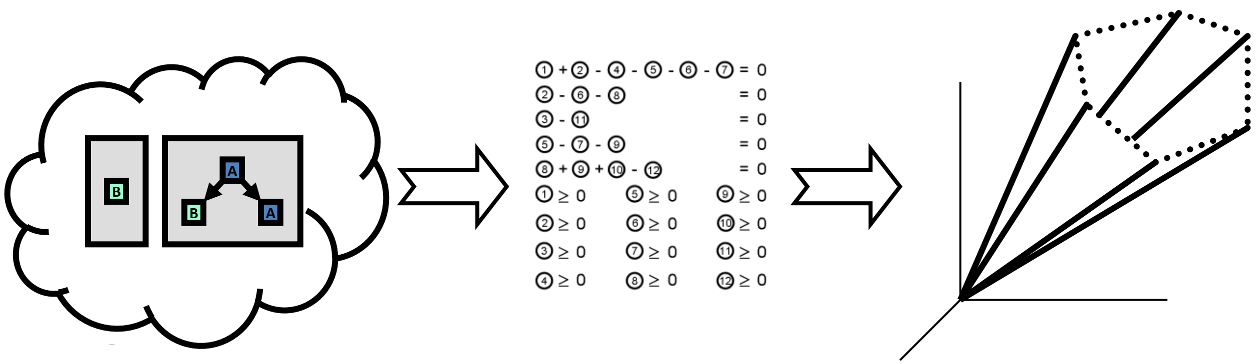
feasible places

Non-negative integer solution of $A_L p \geq 0$: transition-region

- A_L may have infinitely many rows,
  if the language L is infinite

- If the language L is finite then
  the solution space is a pointed polyhedral cone,
  i.e., is generated by a finite set of rays

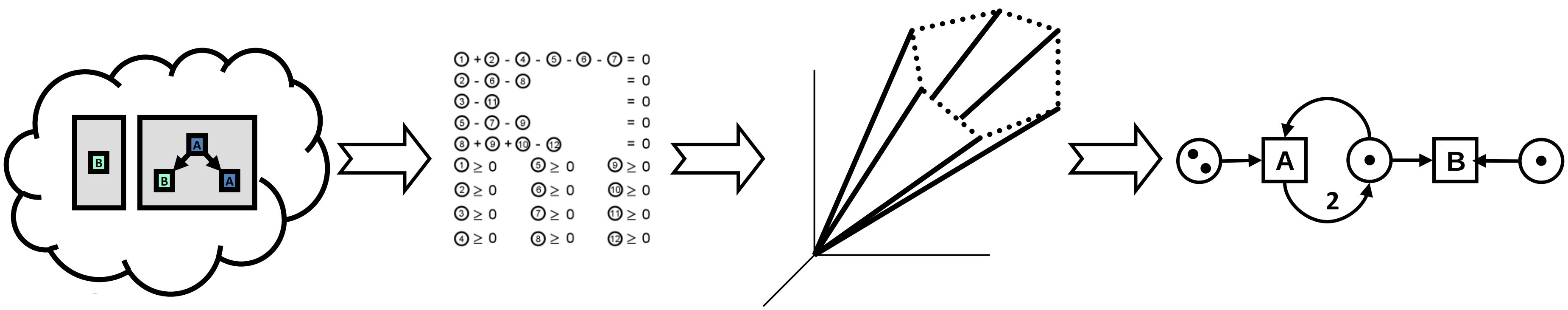
What feasible places should we add?



solution space of this inequality system:

- **pointed polyhedral cone**
- **generated by a finite set of rays.**

What feasible places should we add?



Add places corresponding to the rays of the cone

(… and then find and delete implicit ones)

ac
abc

Another approach

Prefix w extended by … ition t:

wrong continua…

ac
abc

Prefix w extended by new transition t:

wrong continuation $wt \notin L$

ac
abc

→

b	c
aa	aba
abb	aca
acb	acc
abca	abcb
abcc	

Prefix w extended by new transition t:

wrong continuation $wt \notin L$

ac
abc

→

b	c
aa	aba
abb	aca
acb	acc
abca	abcb
abcc	

Net with set of feasible places prohibiting all wrong continuations which can be prohibited:

separating-representation N_{sep}

ac
abc

→

b	c
aa	aba
abb	aca
acb	acc

<u>Theorem</u>

$L(N_{sep})=L(N_{sat})=L$

if and only if

each wrong continuation is prohibited

by some feasible place

Set of places prohibiting all

wrong continuations which can be prohibited:

separating-representation

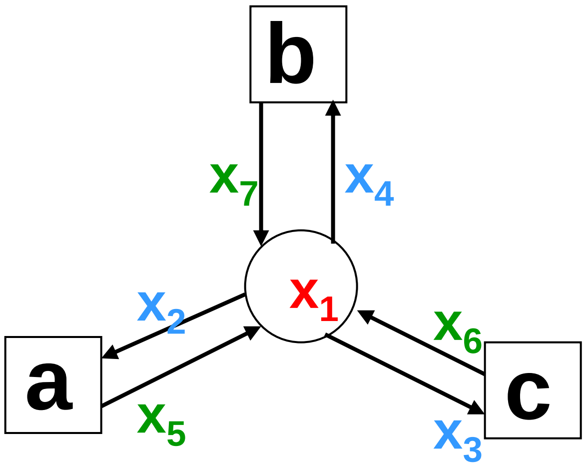
ac
abc

→

b	c
aa	aba
abb	aca
acb	acc
abca	abcb
abcc	

$p=(x_1,x_2,x_3,x_4,x_5,x_6,x_7)$

tuple of non-negative integers



→ $x_1 < x_4$ $\quad$ $x_1 < x_3$

$x_1 - x_2 + x_5 < x_2$ $\quad$ …

p is feasible and prohibits wt

$$\Leftrightarrow$$

$$A_L p \geq 0 \wedge b_{wt} p < 0$$

[there may be infinitely many wrong continuations]

ac

b

aa

c

aba

$p=(x_1,x_2,x_3,x_4,x_5,x_6,x_7)$

ers

<u>Theorem</u>

L fulfills conditions on semi-linearity

(L regular, det. context-free, …):

The matrix A_L and the set of wrong continuations can be finitely represented

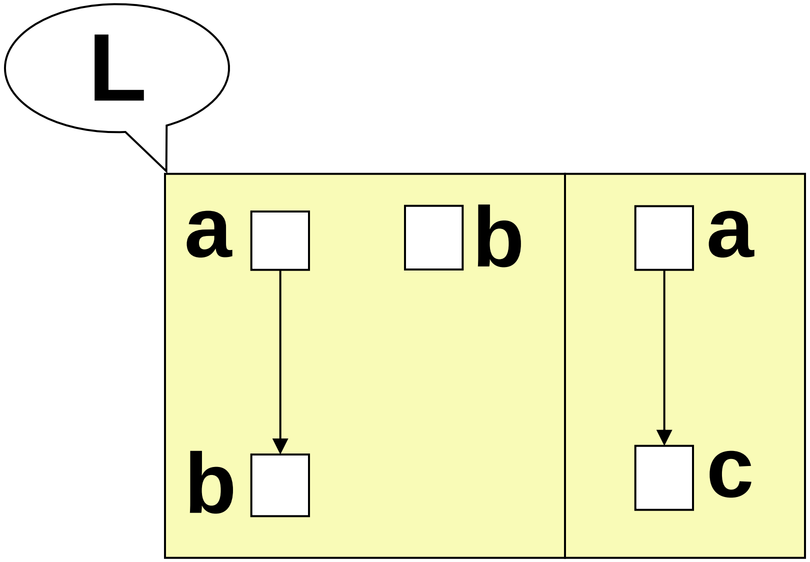
L finite:

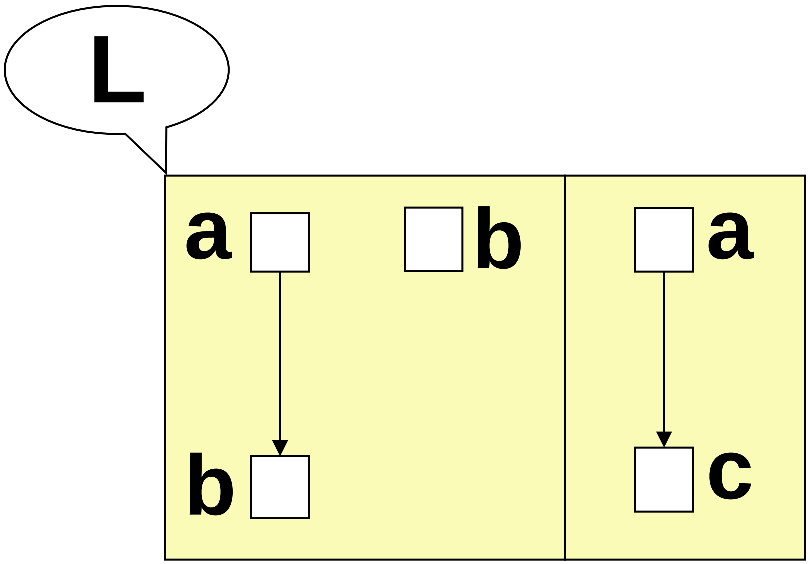
Computation is polynomial

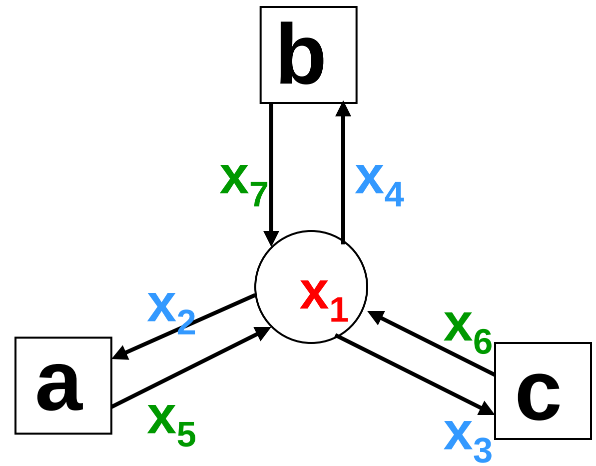
$A_L p \geq 0 \wedge b_{wt} p < 0$

[there may be infinitely many wrong continuations]

General case: language of pomsets



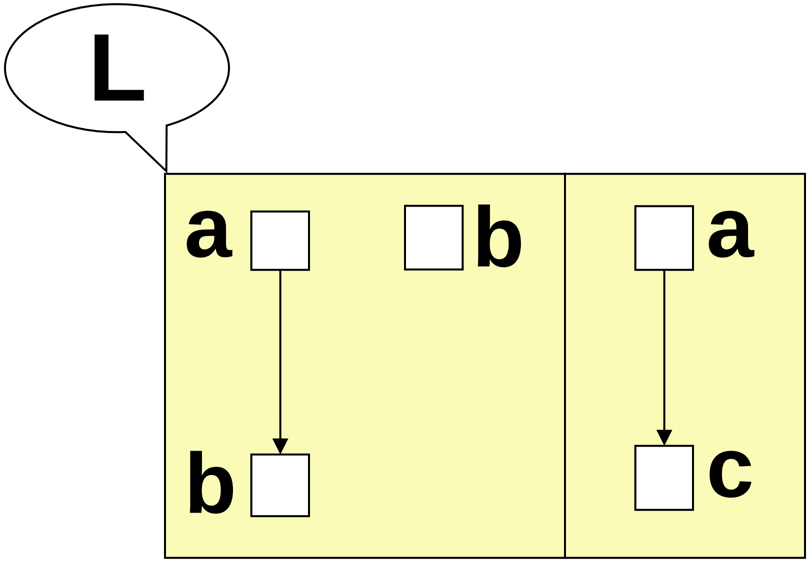


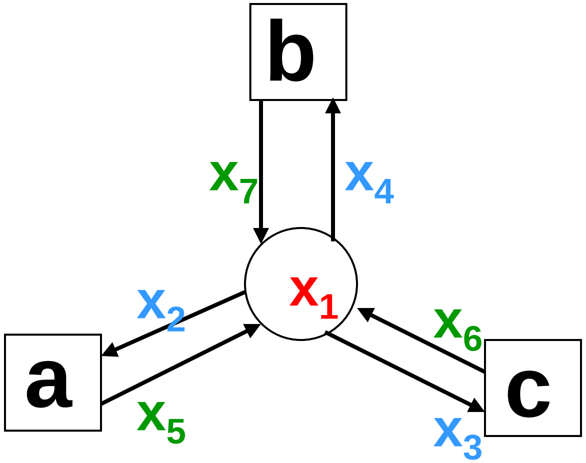


p feasible

$\Leftrightarrow$

- **Each prefix w enables subsequent step s**

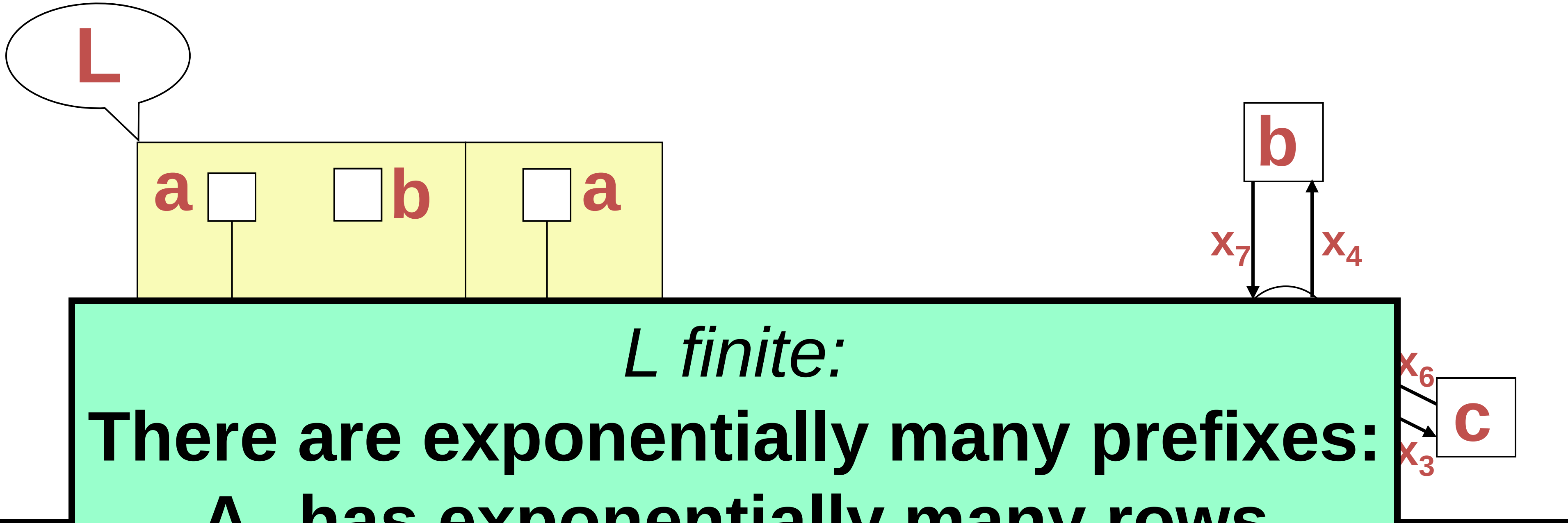




p feasible

$$\Leftrightarrow$$

$$A_L p \geq 0$$



L finite:

There are exponentially many prefixes:

A_L has exponentially many rows

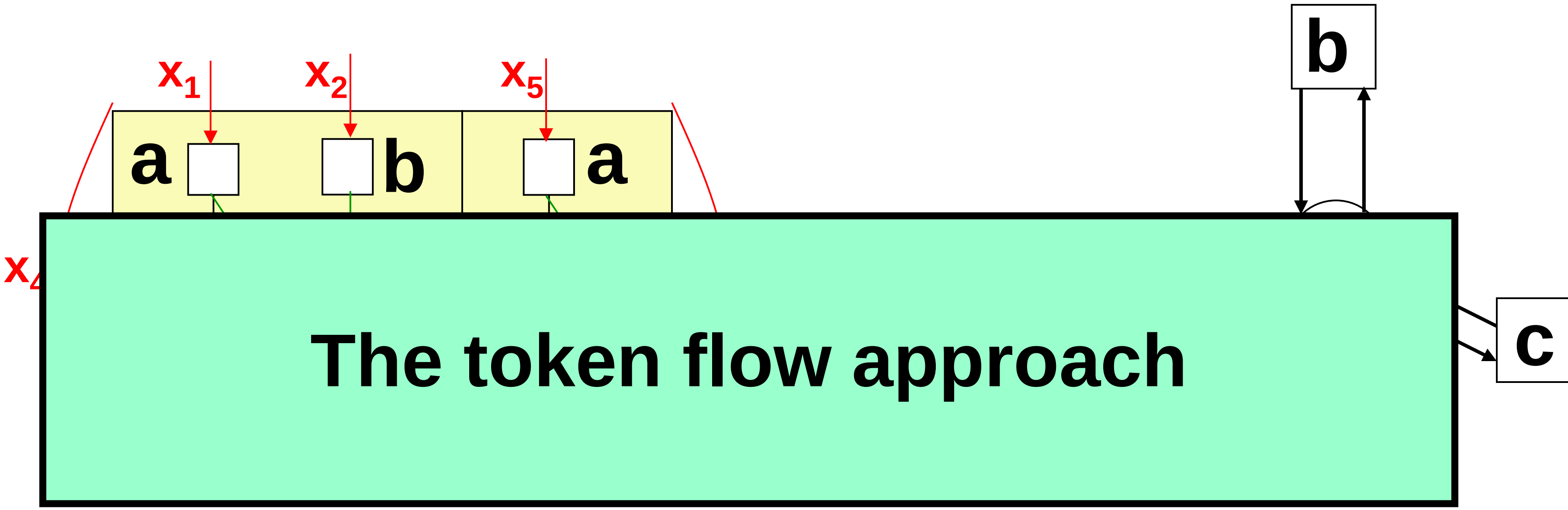
p feasible

$$\Leftrightarrow$$

$$A_L p \geq 0$$

$p=(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10},x_{11},x_{12},x_{13},x_{14})$

tuple of non-negative integers

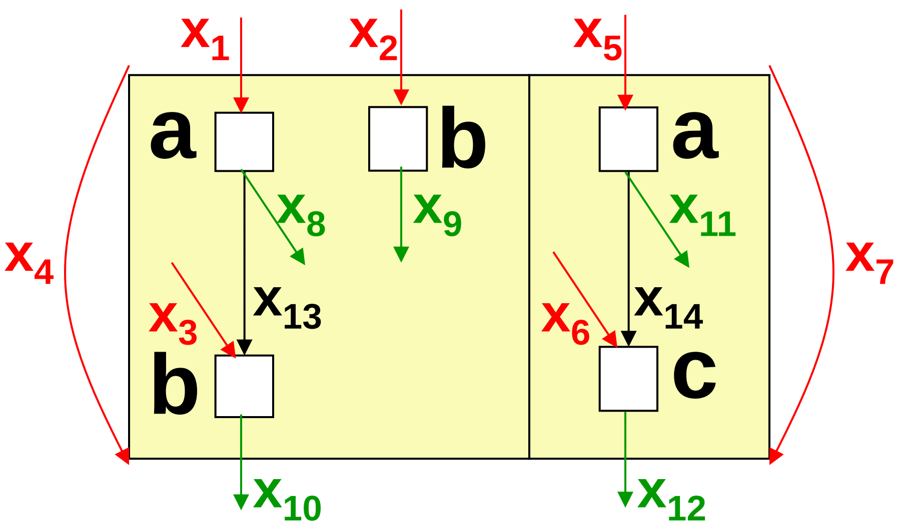


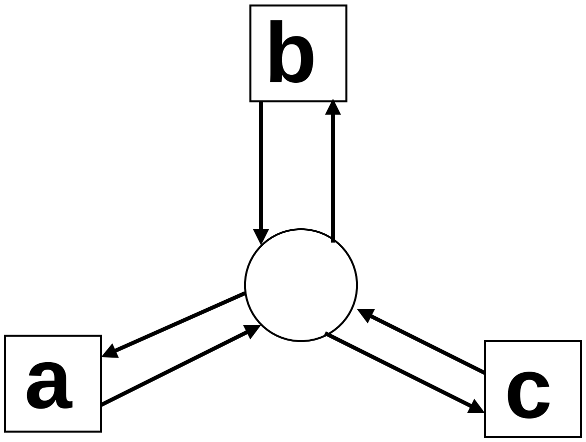
Tokens consumed from the initial marking

Token flow between transition occurrences

Tokens remaining in the final marking

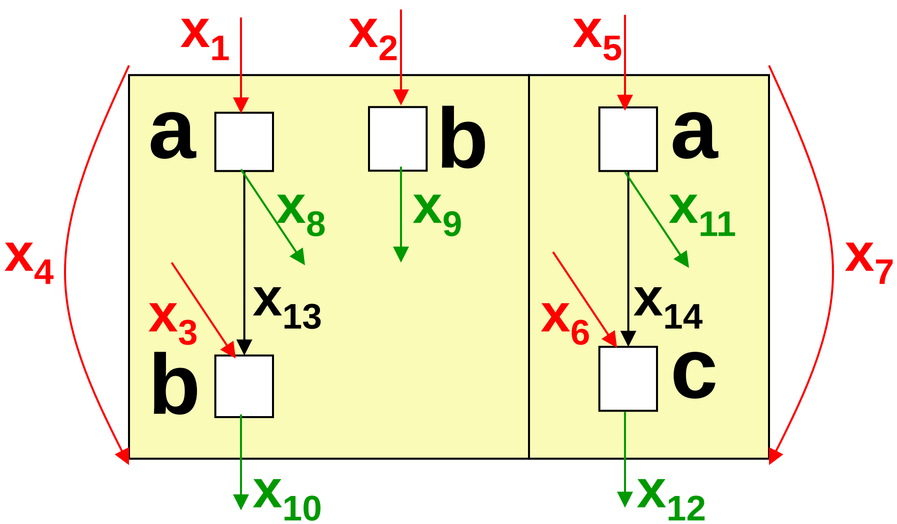
$p=(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10},x_{11},x_{12},x_{13},x_{14})$
tuple of non-negative integers

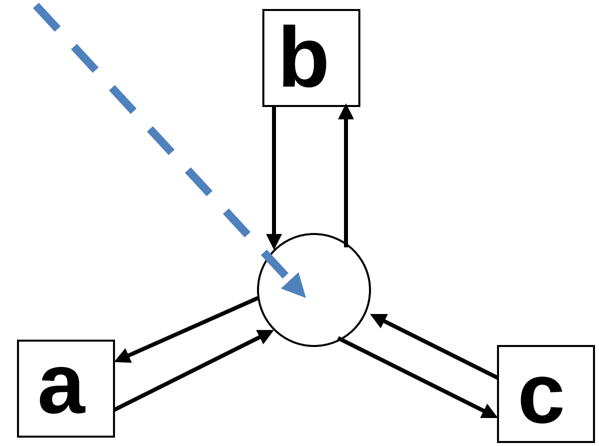




The initial marking of the place
Token flow between transition occurrences on the place
New tokens remaining in the final marking of the place

$$\overbrace{x_1+x_2+x_3+x_4}^{\text{initial flow of left pomset}} = \overbrace{x_5+x_6+x_7}^{\text{initial flow of right pomset}}$$



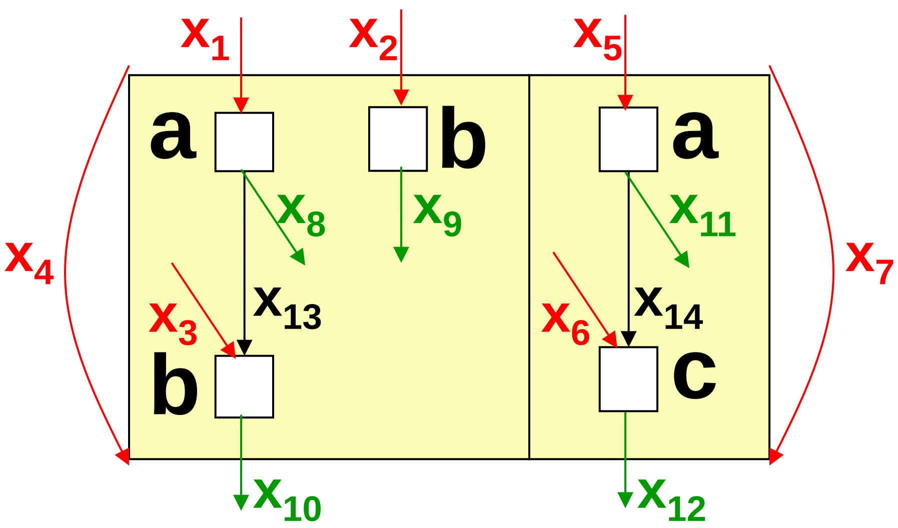


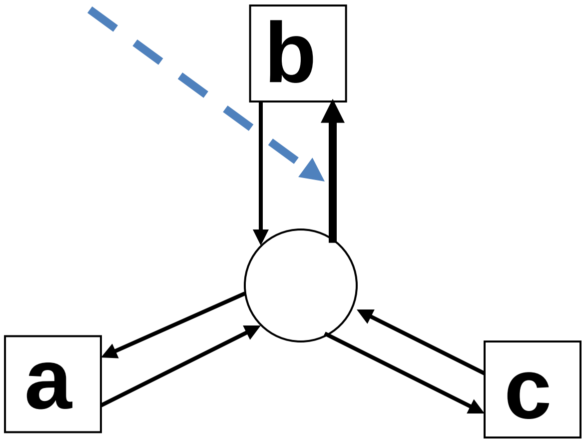
The initial marking of the place

Token flow between transition occurrences on the place

New tokens remaining in the final marking of the place

$$x_2 = \overbrace{x_3 + x_{13}}^{\text{in-flow of } \mathbf{b}}$$



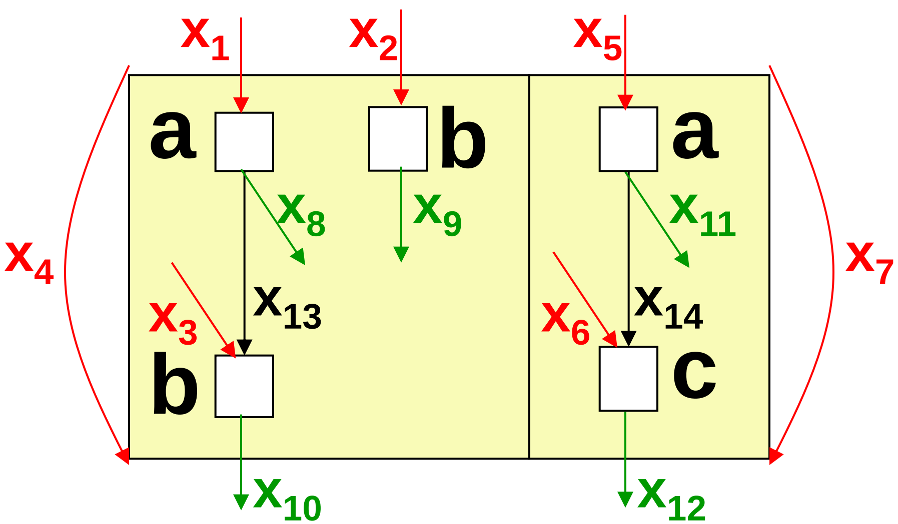


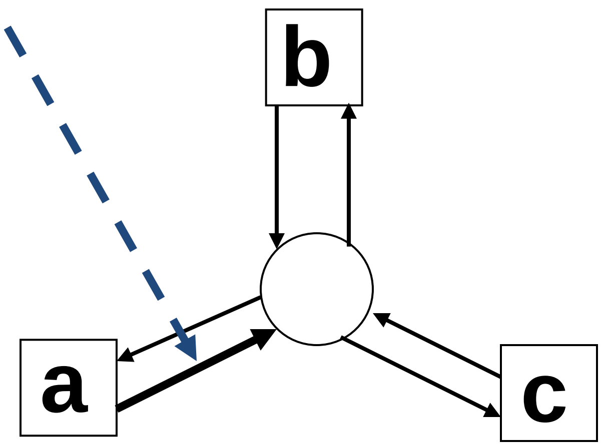
The initial marking of the place

Token flow between transition occurrences on the place

New tokens remaining in the final marking of the place

$$\overbrace{x_{11}+x_{14}}^{\text{out-flow of } \mathbf{a}}=x_8+x_{13}$$

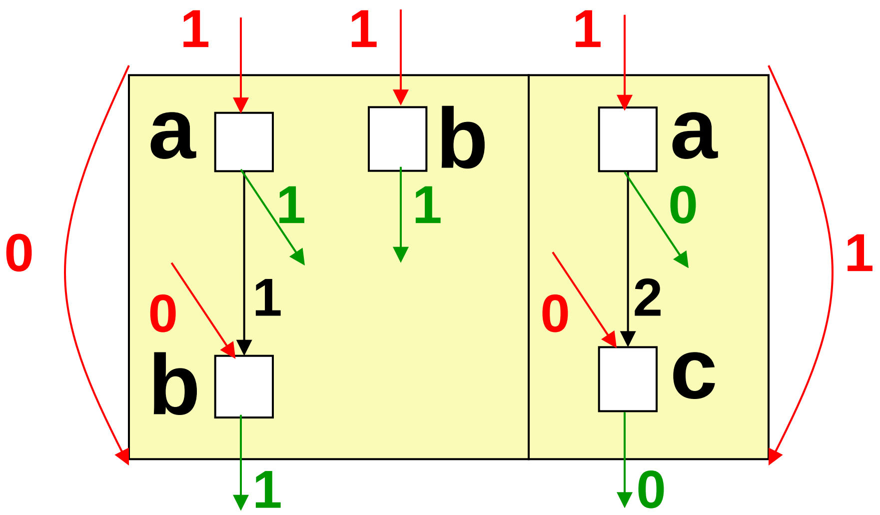


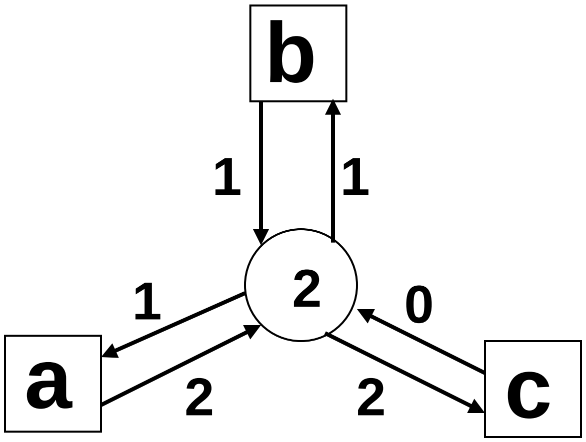


The initial marking of the place

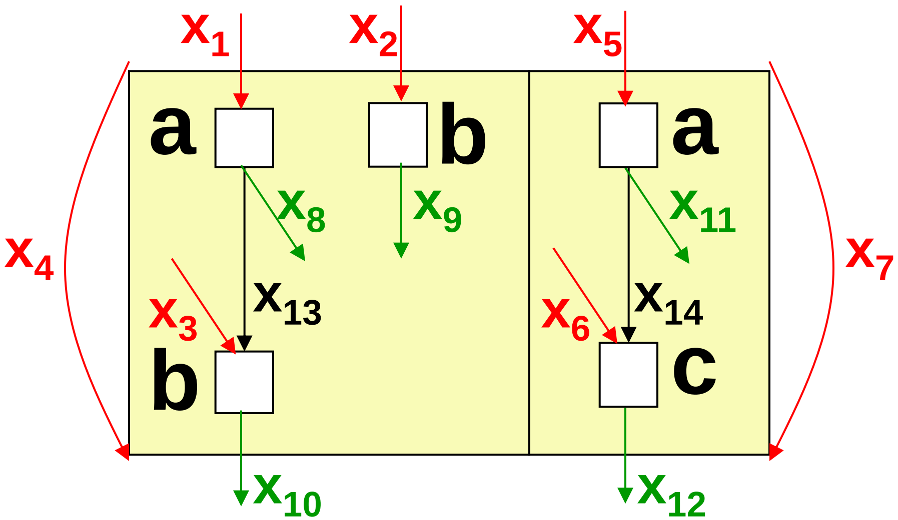
Token flow between transition occurrences on the place

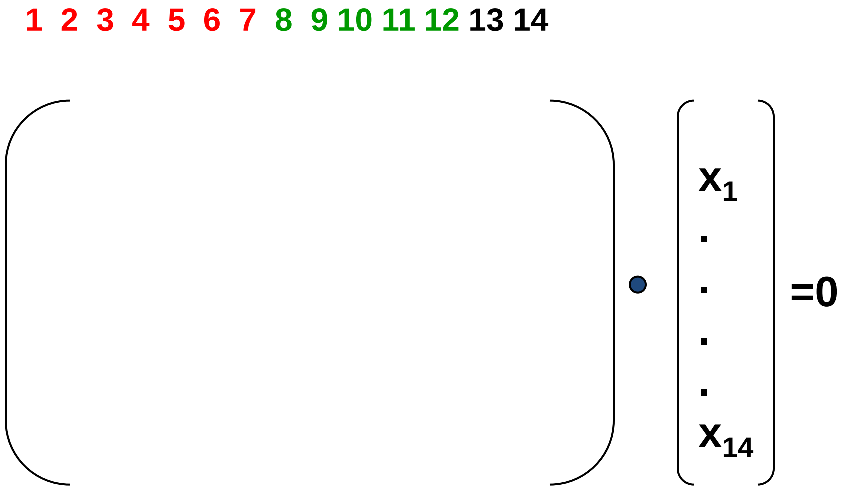
New tokens remaining in the final marking of the place





Equally labeled nodes have equal in-flow and equal out-flow
Each pomset has the same initial flow

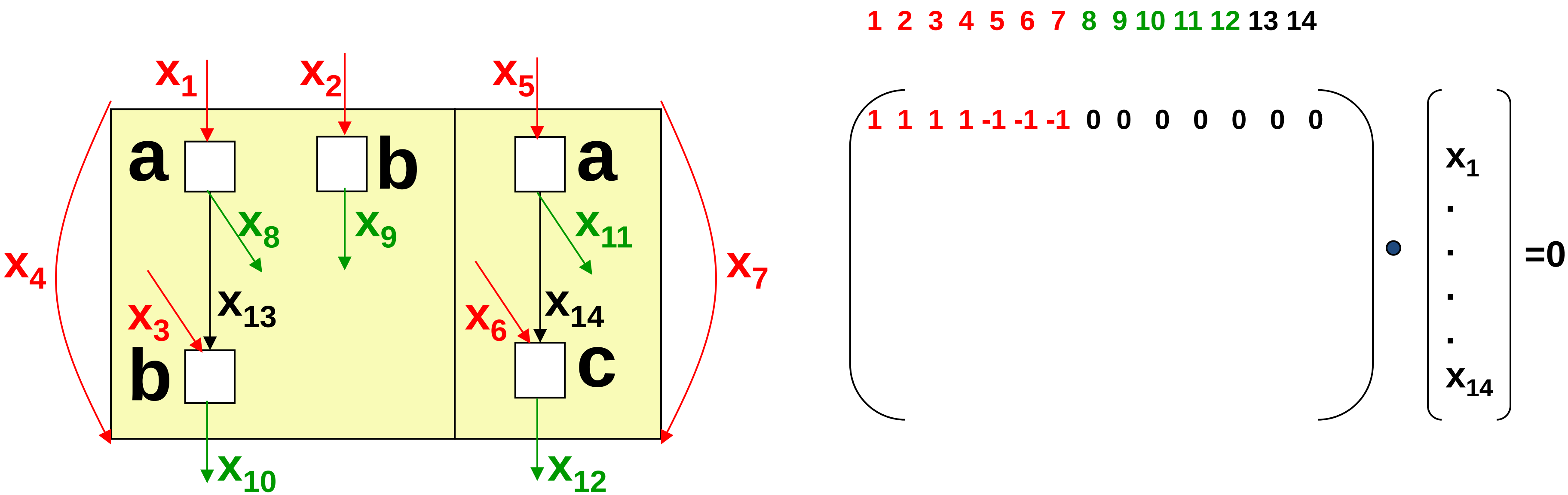




Equally labeled nodes have equal in-flow and equal out-flow

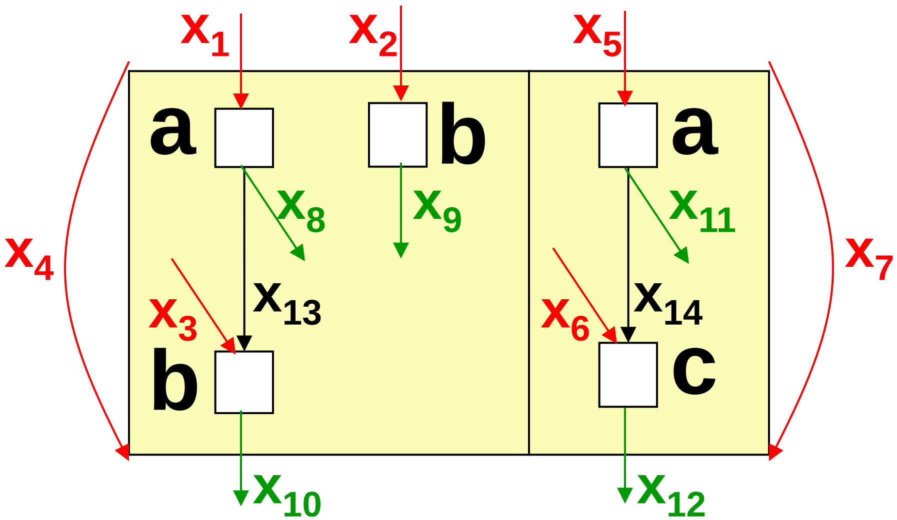
Each pomset has the same initial flow

Each pomset has the same initial flow



Equally labeled nodes have equal in-flow and out-flow
Each pomset has the same initial flow

a-labeled nodes have the same in-flow



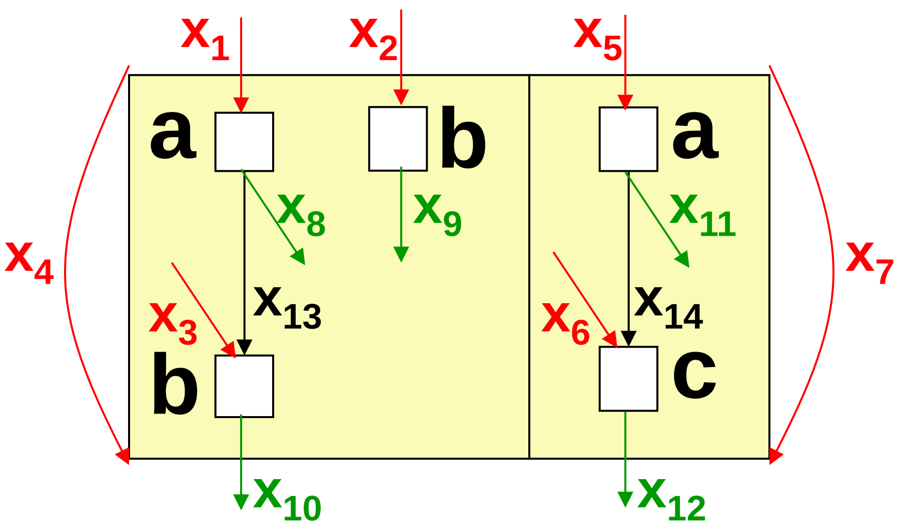
$$
\begin{array}{cccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14
\end{array}
$$

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & &
\end{pmatrix}
\bullet
\begin{pmatrix}
x_1 \\ . \\ . \\ . \\ . \\ x_{14}
\end{pmatrix}
=0
$$

Equally labeled nodes have equal in-flow and out-flow

Each pomset has the same initial flow

a-labeled nodes have the same out-flow

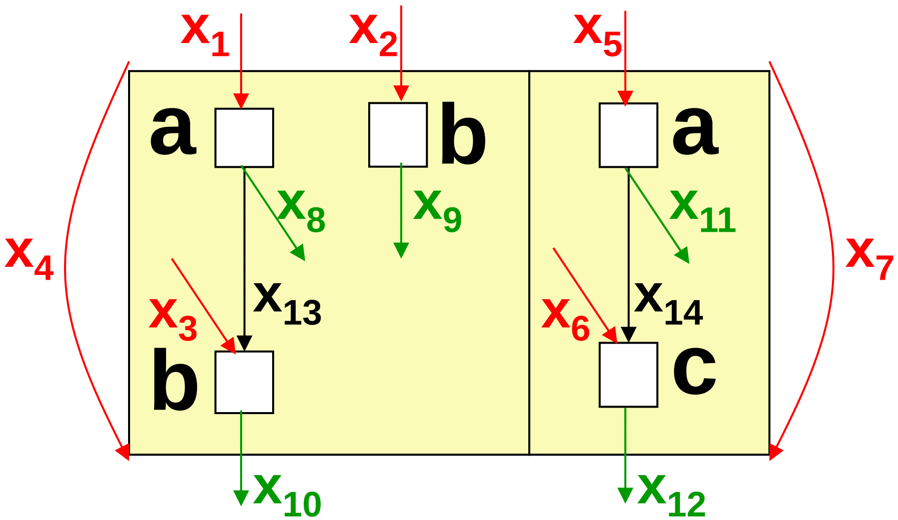


$$\begin{array}{c}
\begin{array}{cccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \end{array} \\
\begin{pmatrix}
1 & 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & -1 \\
 & & & & & & & & & & & & &
\end{pmatrix}
\end{array}
\bullet
\begin{pmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_{14} \end{pmatrix} = 0$$

Equally labeled nodes have equal in-flow and out-flow

Each pomset has the same initial flow

b-labeled nodes have …

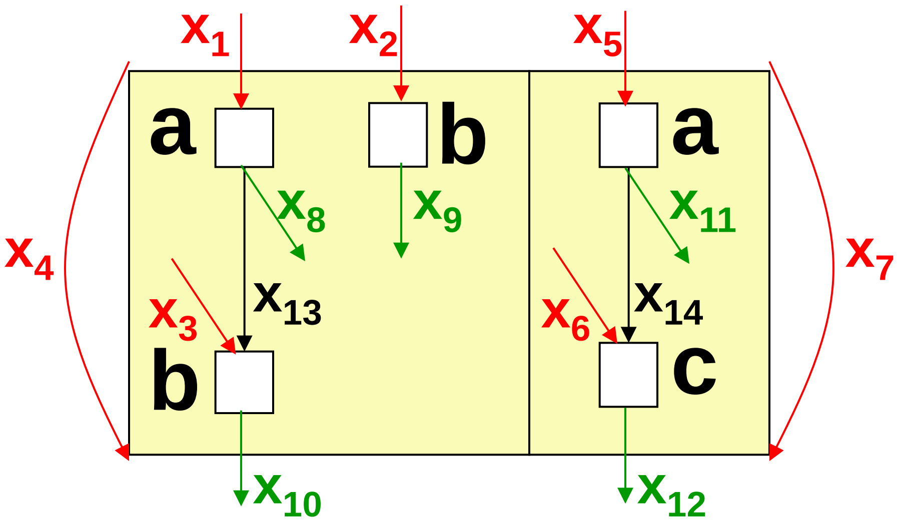


$$
\begin{array}{cccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14
\end{array}
$$

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & -1 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0
\end{pmatrix}
\bullet
\begin{pmatrix}
x_1 \\ . \\ . \\ . \\ . \\ x_{14}
\end{pmatrix}
= 0
$$

Equally labeled nodes have equal in-flow and out-flow

Each pomset has the same initial flow



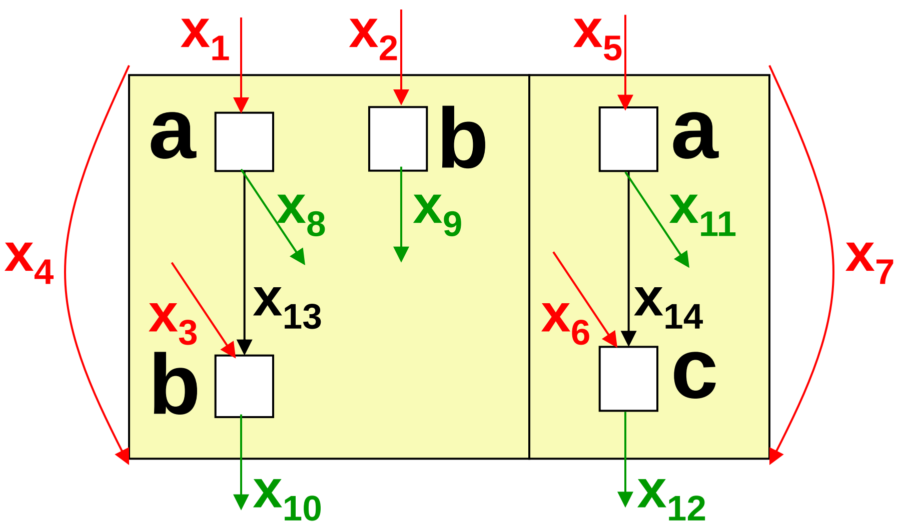
$$
\begin{array}{c}
\begin{array}{cccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \end{array} \\
\begin{pmatrix}
1 & 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & -1 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
\bullet
\begin{pmatrix} x_1 \\ . \\ . \\ . \\ . \\ x_{14} \end{pmatrix} = 0
$$

Equally labeled nodes have equal in-flow and out-flow

Each pomset has the same initial flow

$$\Leftrightarrow$$

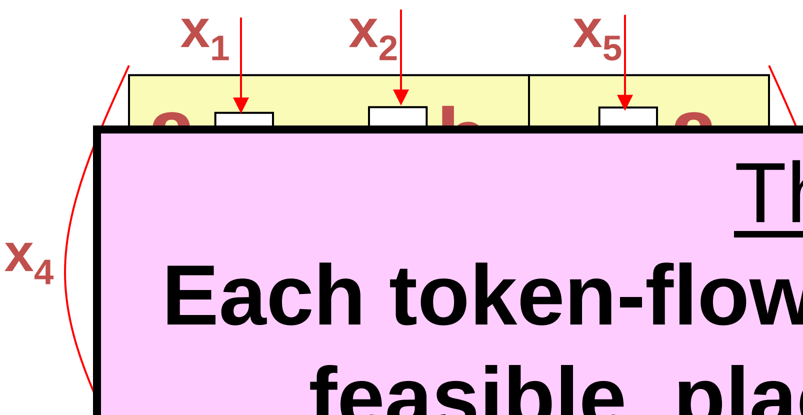
$$B_L p = 0$$



$$
\begin{array}{c}
\begin{array}{cccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \end{array} \\
\begin{pmatrix}
1 & 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & -1 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
\bullet
\begin{pmatrix} x_1 \\ . \\ . \\ . \\ . \\ x_{14} \end{pmatrix} = 0
$$

Non-negative integer solutions of $B_L x=0$:

token-flow-regions



1 2 3 4 5 6 7 8 9 10 11 12 13 14

1 1 1 1 -1 -1 -1 0 0 0 0 0 0 0

=0

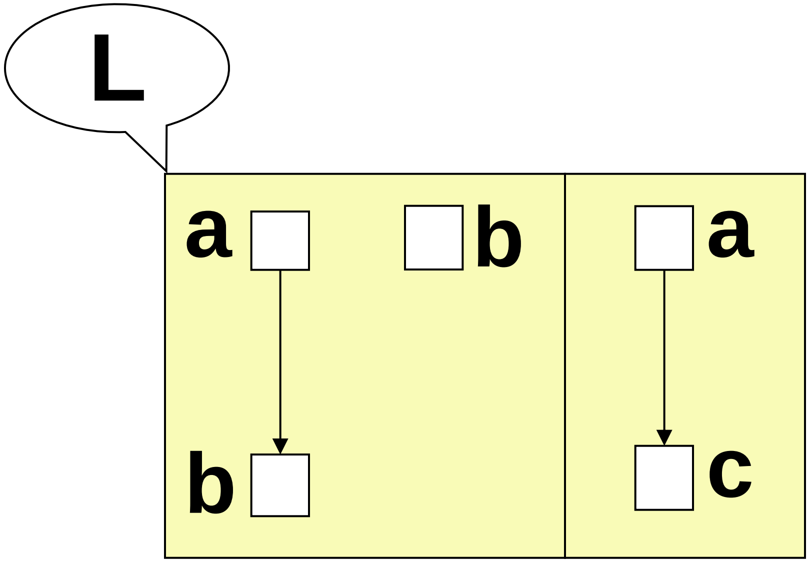
<u>Theorem</u>

Each token-flow-regions generates a feasible place and vice versa

L finite:

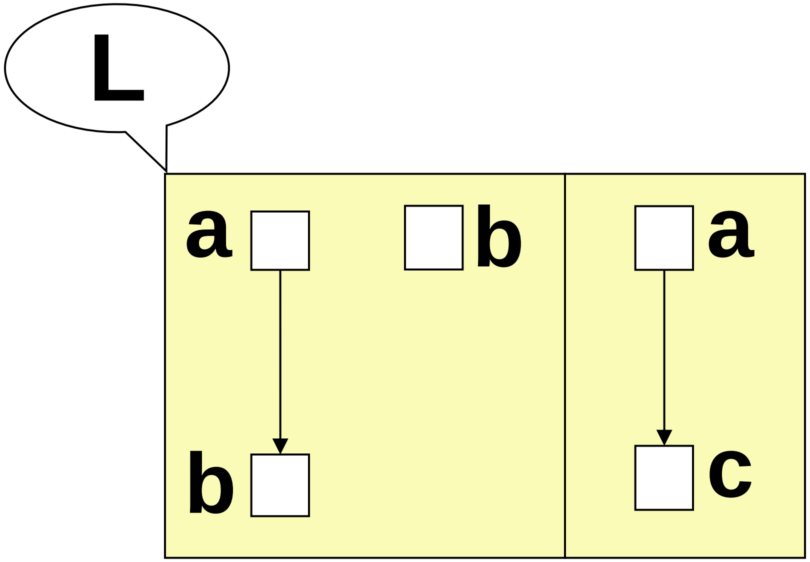
B_L has linear many rows

token-flow-regions



B_L finite:

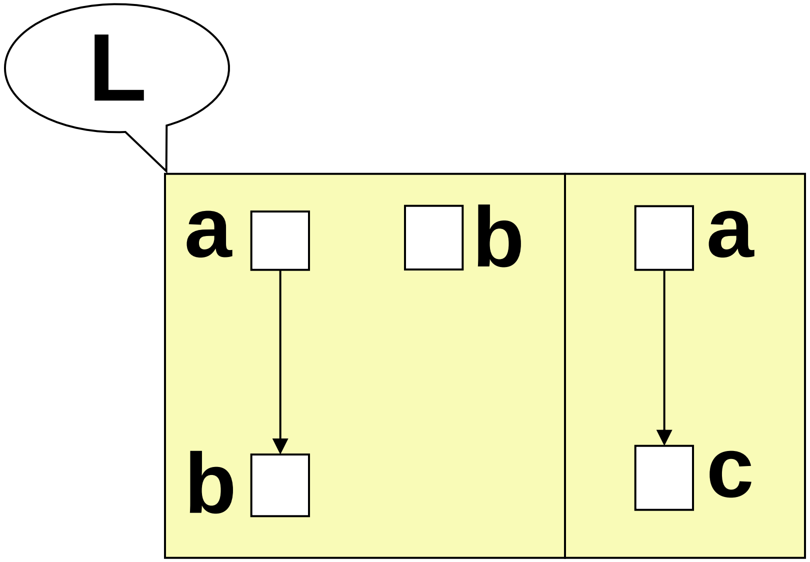
Set of non-negative integer solutions of $B_L p=0$ is finitely generated by a fundamental system of solutions $y_1,\ldots,y_k$



B_L finite:

Net with set of places defined by fundamental system of solutions $y_1,\ldots,y_k$ of $B_Lp=0$:

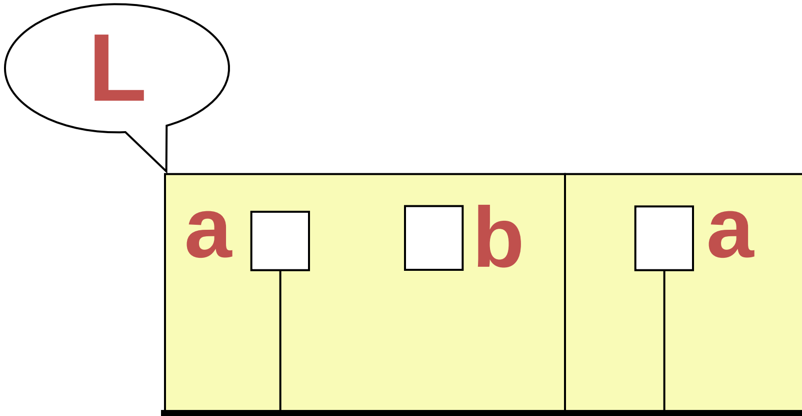
Basis-representation N_{basis}



p is feasible and prohibits ws

$$\Leftrightarrow$$

$$B_L p = 0 \wedge c_{ws} p < 0$$



L finite:

There are exponentially many wrong continuations

$$\Leftrightarrow$$

$$B_L p=0 \wedge c_{ws} p<0$$

L

orem

Term-r... sentation of L
(using oper... ation, union,
parallel & seq... position):
B_L can be finit... ated

What if L is infinite?

B_L fini...

Net with set of places defin... mental system of solutions $y_1, \ldots, y$... $p=0$: Basis-representation N_{basis}

L

Theorem

Term-based representation of L (using operators for iteration, union, parallel & sequential composition):

B_L can be finitely represented

B_L finite:

Net with set of places defined by fundamental system of solutions $y_1,\ldots,y_k$ of $B_L p=0$: Basis-representation N_{basis}

L

Theorem

Term-based representation of L (using operators for iteration, union, parallel & sequential composition):

B_L can be finitely represented

$L(N_{basis})=L(N_{sat})$

B_L finite:

Net with set of places defined by fundamental system of solutions $y_1,\ldots,y_k$ of $B_L p=0$: Basis-representation N_{basis}

L

Theorem

Term-based representation of L (using operators for iteration, union, parallel & sequential composition):

B_L can be finitely represented

$L(N_{basis})=L(N_{sat})$

L finite:

$L=L(N_{basis})$ is decidable

Net with set of places defined by fundamental system of solutions $y_1,\ldots,y_k$ of $B_L p=0$: Basis-representation N_{basis}

L

Theorem

Term-based representation of L
(using operators for iteration, union,
parallel & sequential composition):

Computations are exponential
in the worst case

$L(N_{basis})=L(N_{sat})$

L finite:
$L=L(N_{basis})$ is decidable

Net with set of places defined by fundamental system of solutions $y_1,\ldots,y_k$ of $B_L p=0$:
Basis-representation N_{basis}

Rules of thump

The more concurrency L includes …

…the smaller is B_L (for the definition of token flow-regions)

…the bigger is A_L (for the definition of transition-regions)

…the more wrong continuations have to be considered (for the calculation of the separating-representation)

Adaption to any Petri net class possible …

… through adding rows to A_L resp. B_L

transition-regions
token flow-regions

separating-representation
basis-representation

Non-negative integer solutions of linear inequation systems

p/t-nets
elementary nets
nets with inhibitor arcs
workflow nets
…

classical languages
step languages
partial languages
stratified languages

transition-regions
token flow-regions

separating-representation
basis-representation

solved by the group of
Philippe Darondeau

p/t-nets
elementary nets
nets with inhibitor arcs
workflow nets
…

classical languages
step languages
partial languages
stratified languages

transition-regions
token flow-regions

separating-representation
basis-representation

The „Eichstätt“ group

Non-negative integer solutions of linear inequation systems

p/t-nets
elementary nets
nets with inhibitor arcs
workflow nets
…

classical languages
step languages
partial languages
stratified languages

Application

Process-Mining

Business Process Design

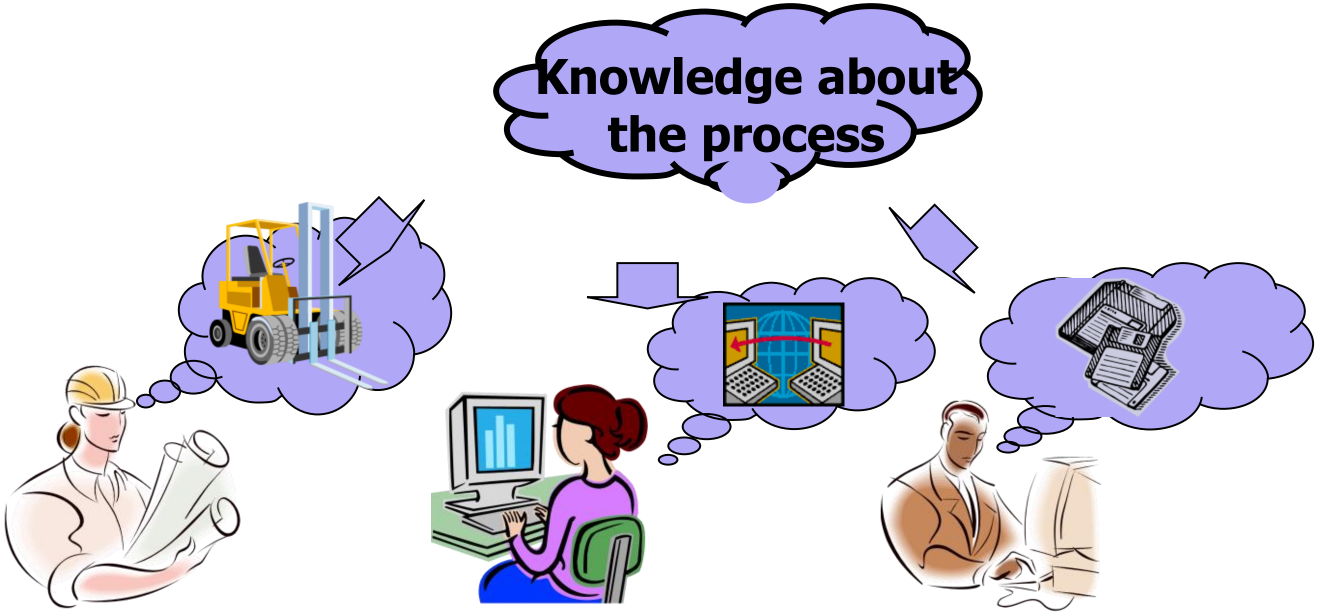
Controller Synthesis

Implementation in VIPtool (*available online*)
(Tool for modeling and partial order based simulation, verification, validation and synthesis of Petri net models)

Case Study

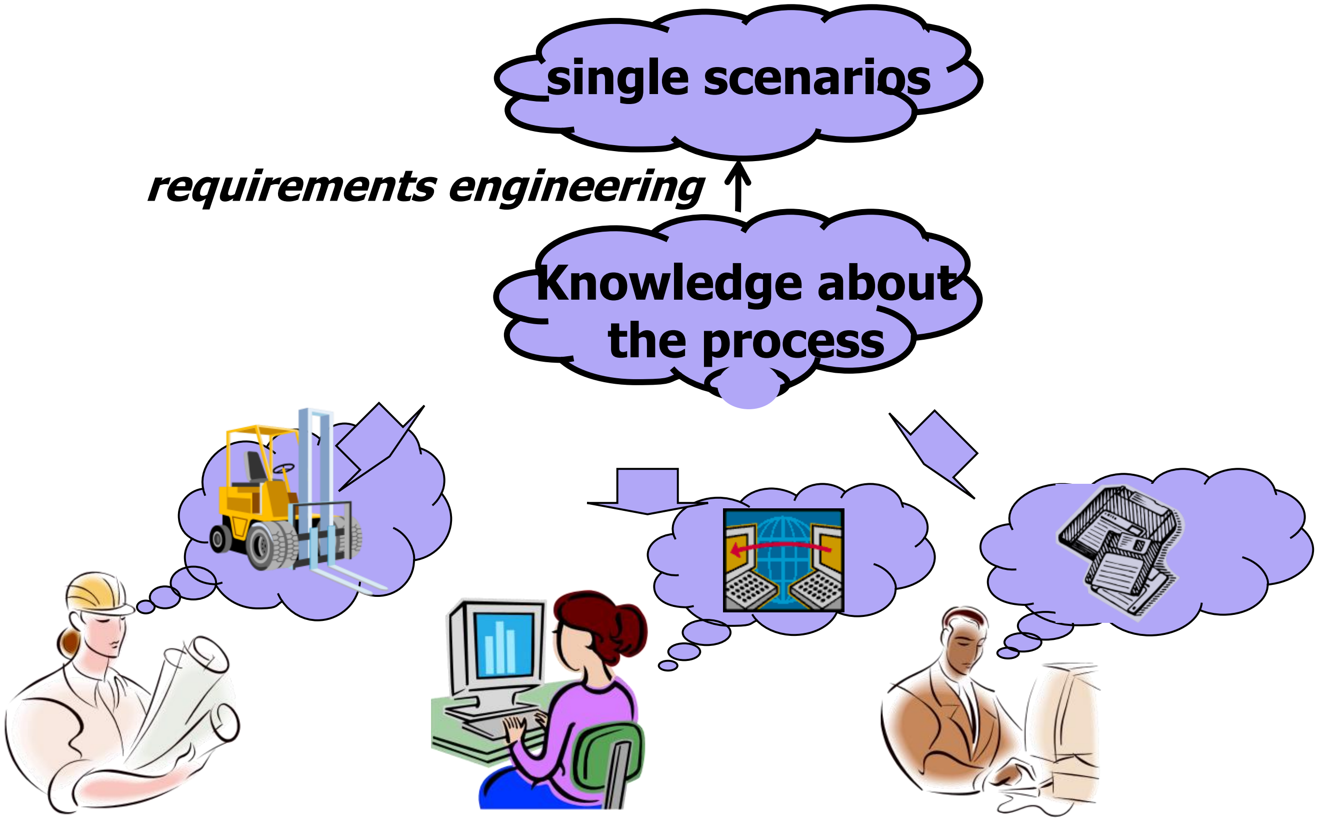
Modeling Business Processes from Scenarios: Initial Situation

Knowledge about a process is distributed in several peoples‘ mind in an informal environment



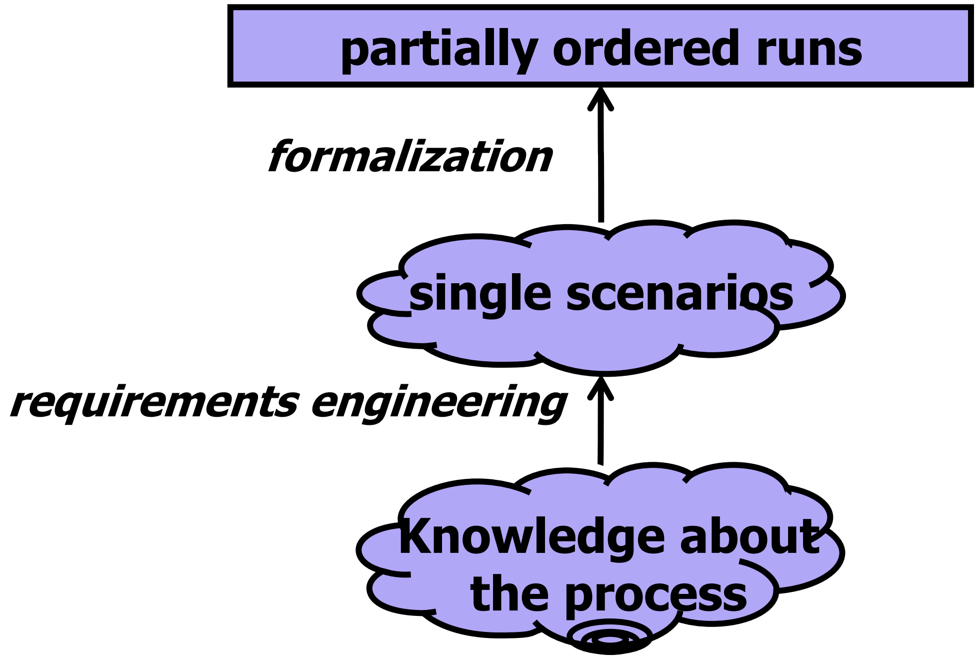
Case Study

Modeling Business Processes from Scenarios: Initial Situation



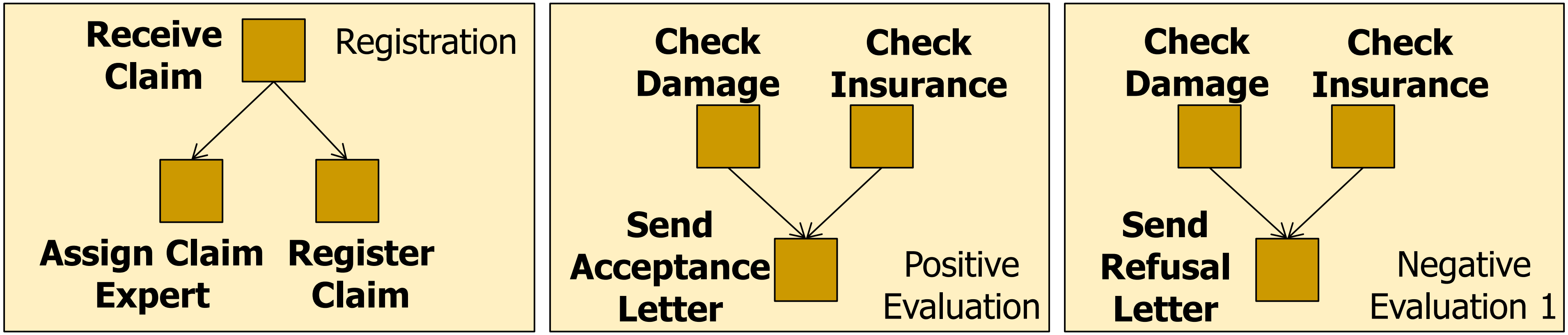
Case Study

Modeling Business Processes from Scenarios: Initial Situation



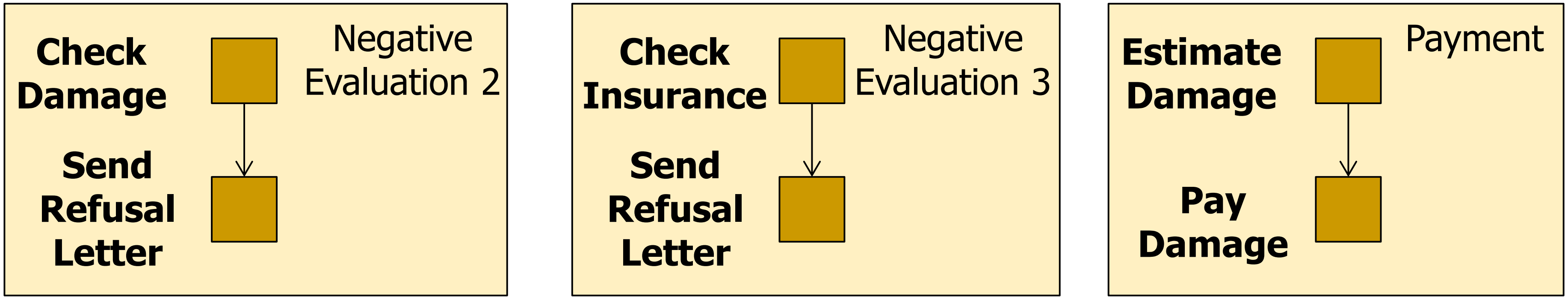
Case Study

Formal runs in an insurance company: Single Scenarios



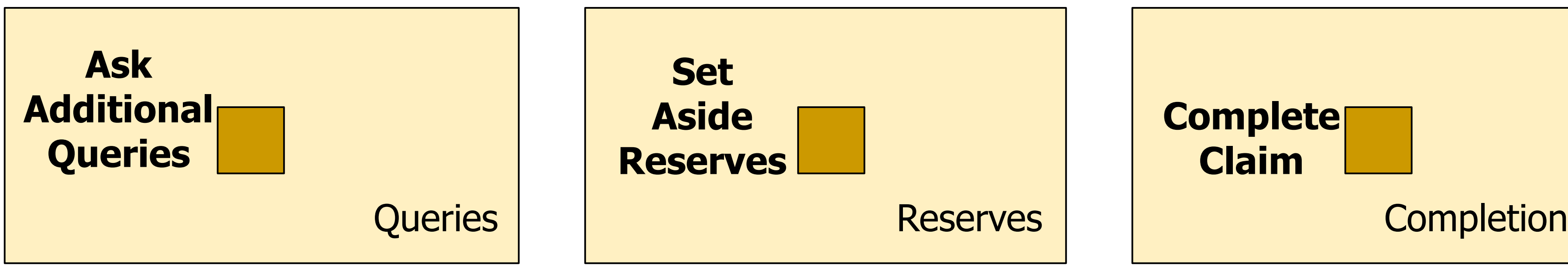
Case Study

Formal runs in an insurance company: Single Scenarios



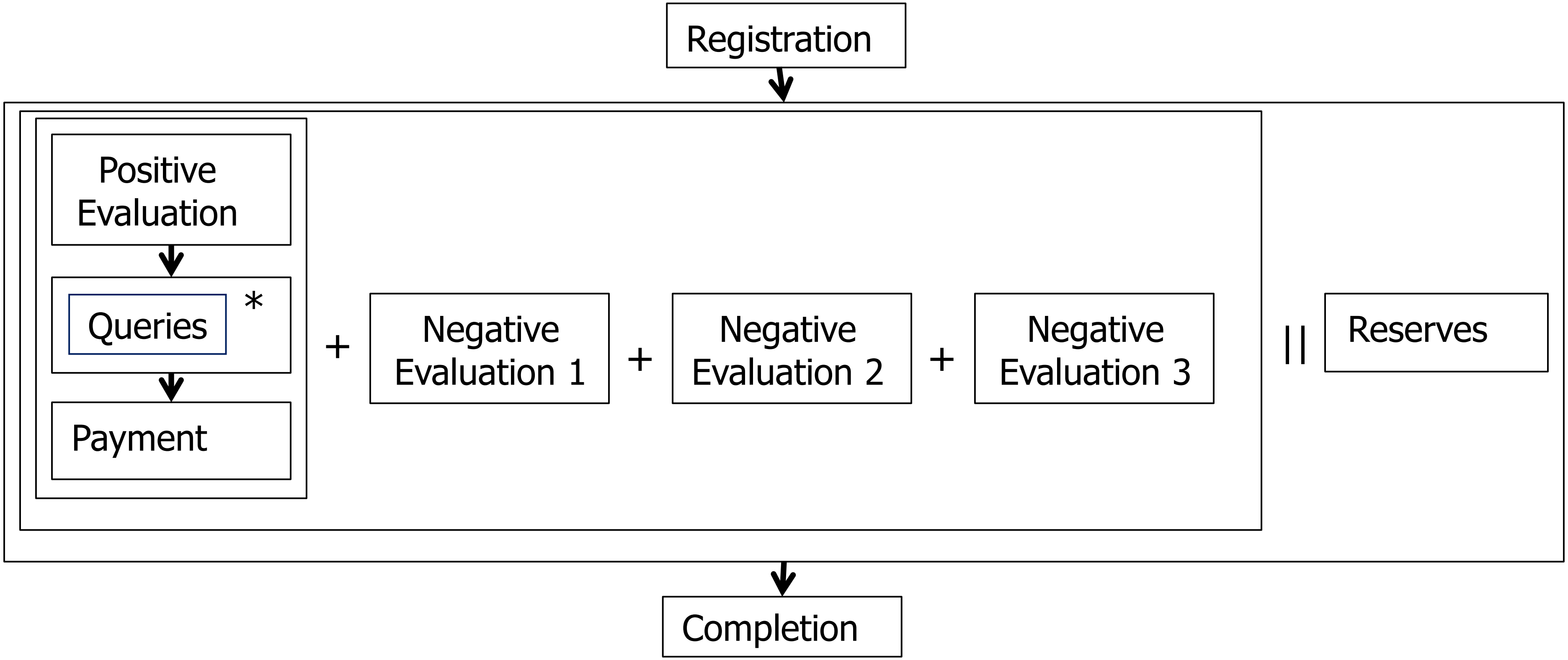
Case Study

Formal runs in an insurance company: Single Scenarios



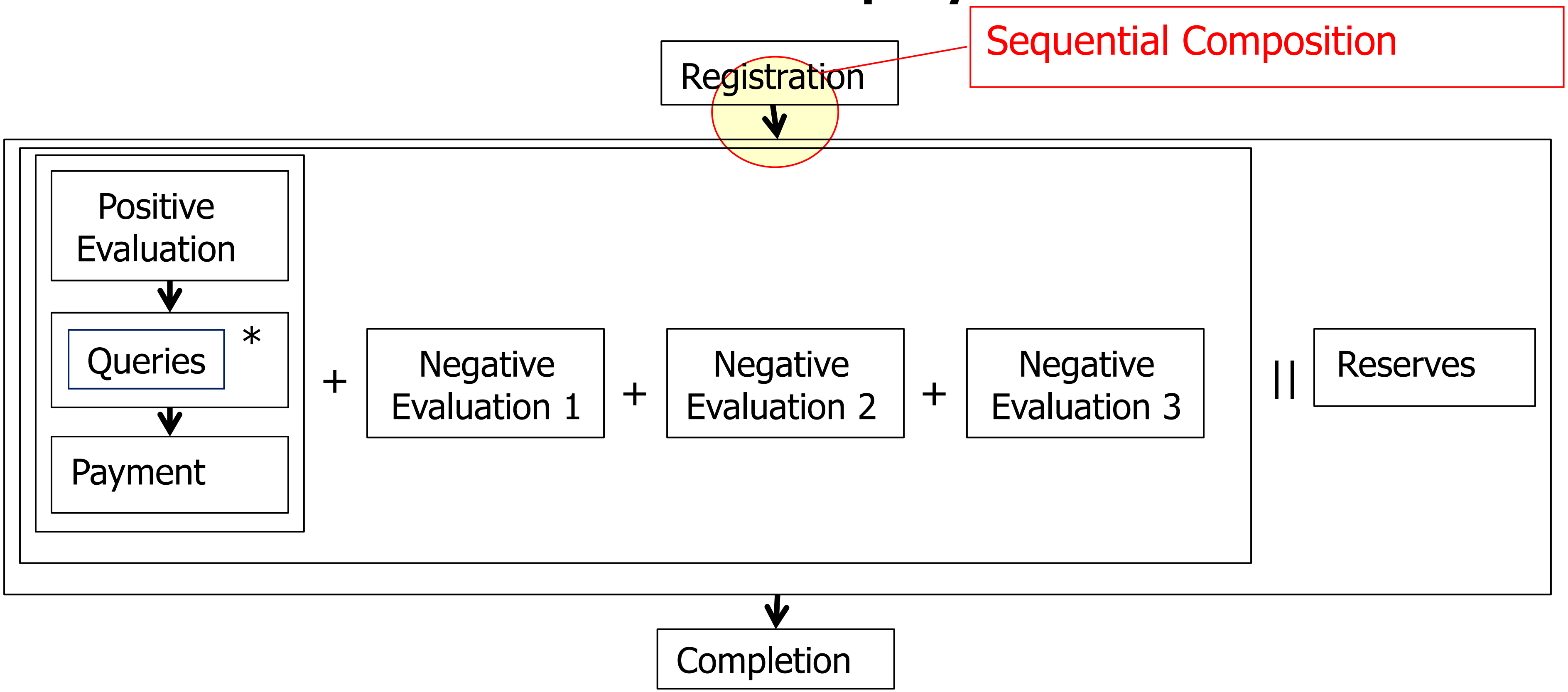
Case Study

Formal runs in an insurance company: Composition



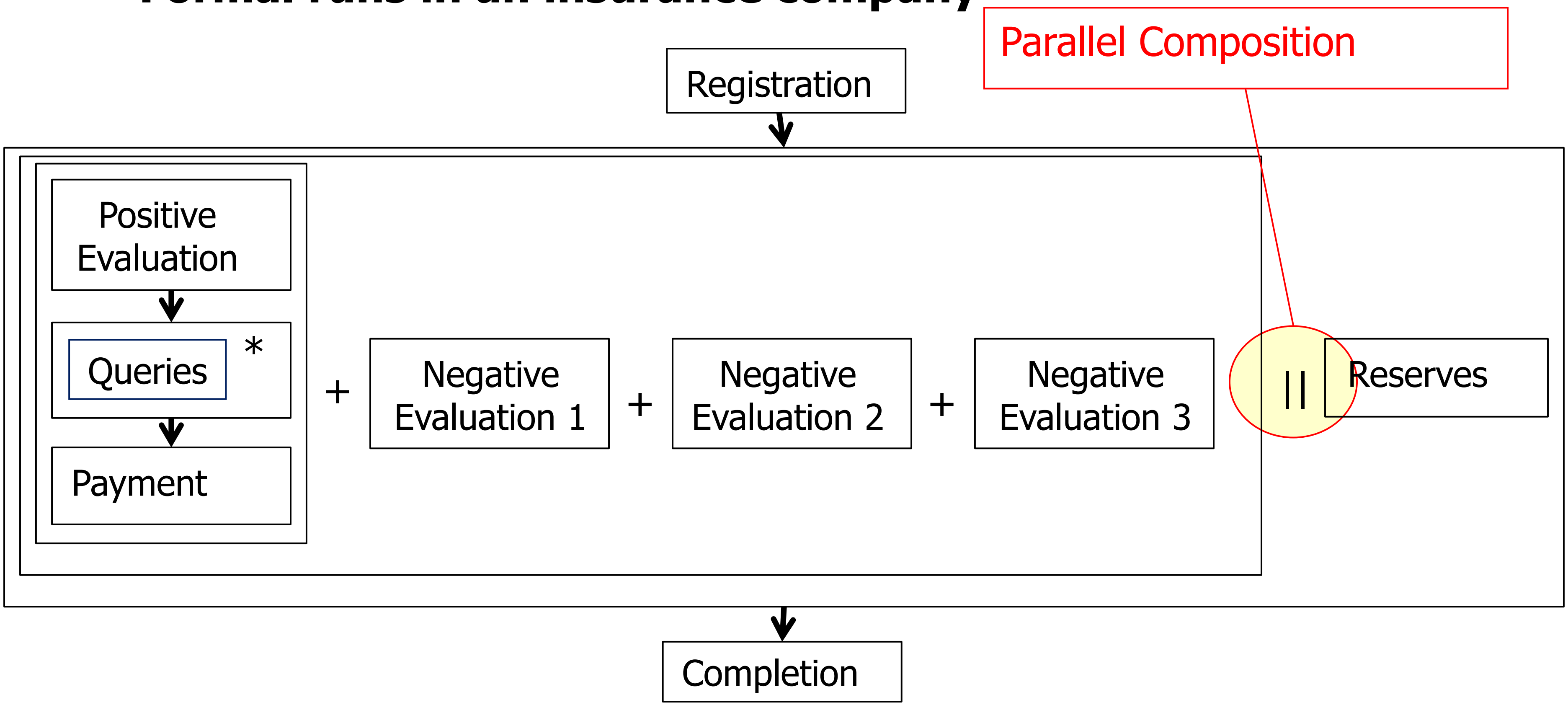
Case Study

Formal runs in an insurance company



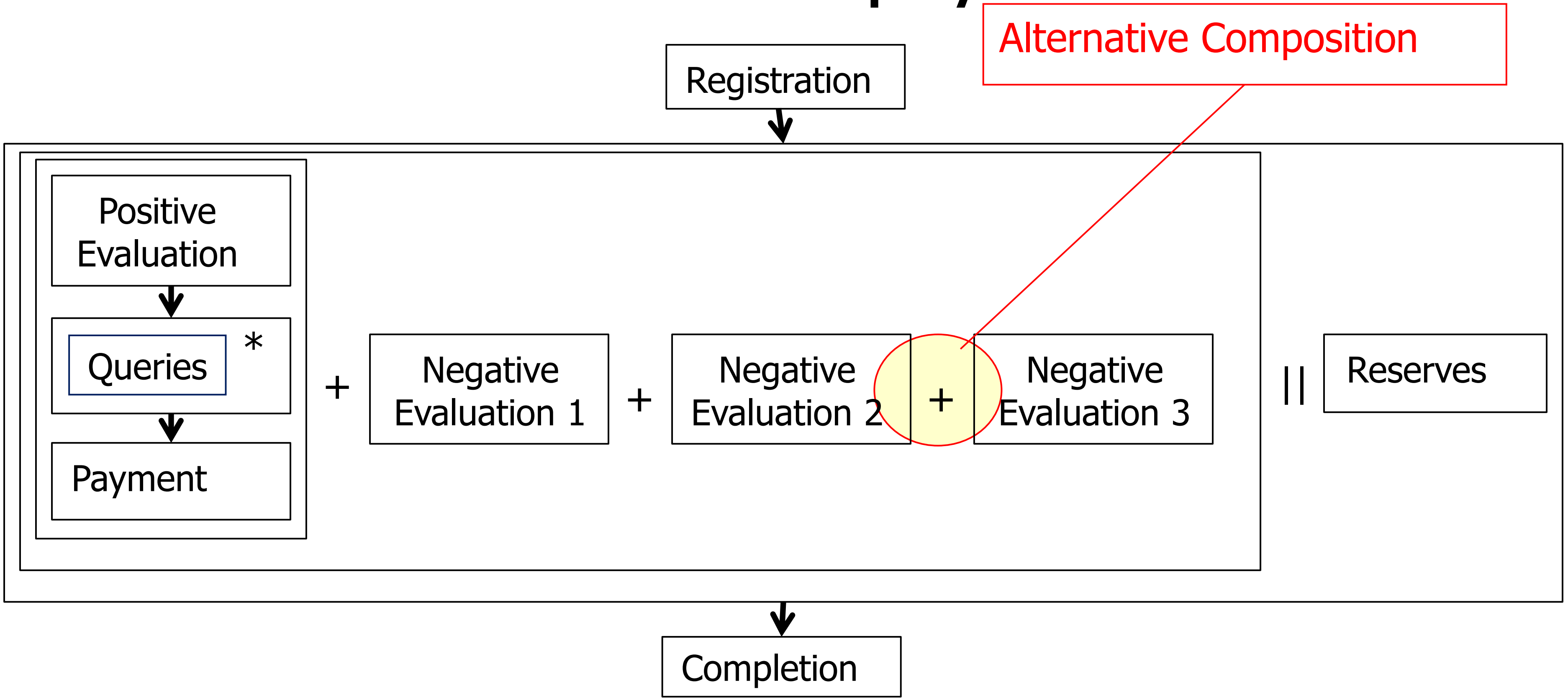
Case Study

Formal runs in an insurance company



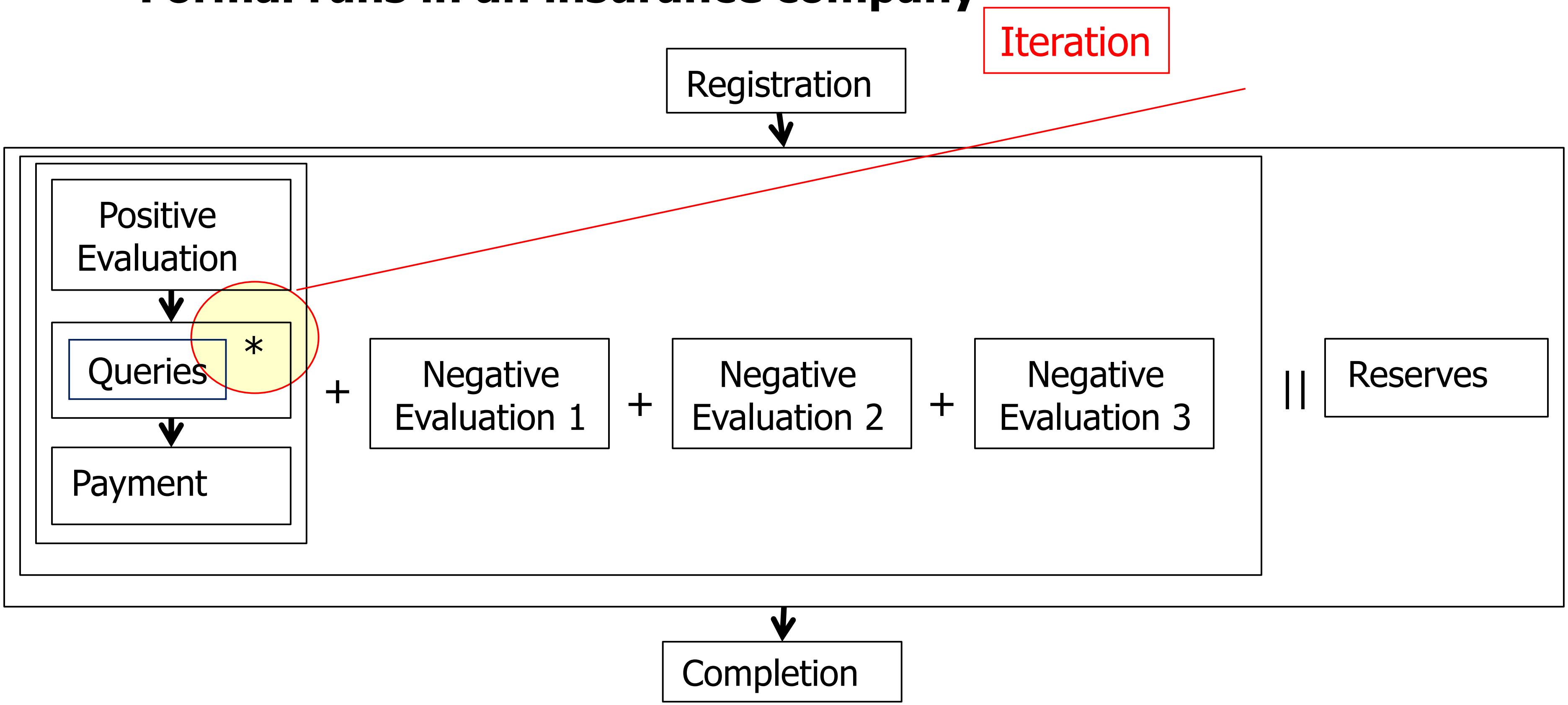
Case Study

Formal runs in an insurance company



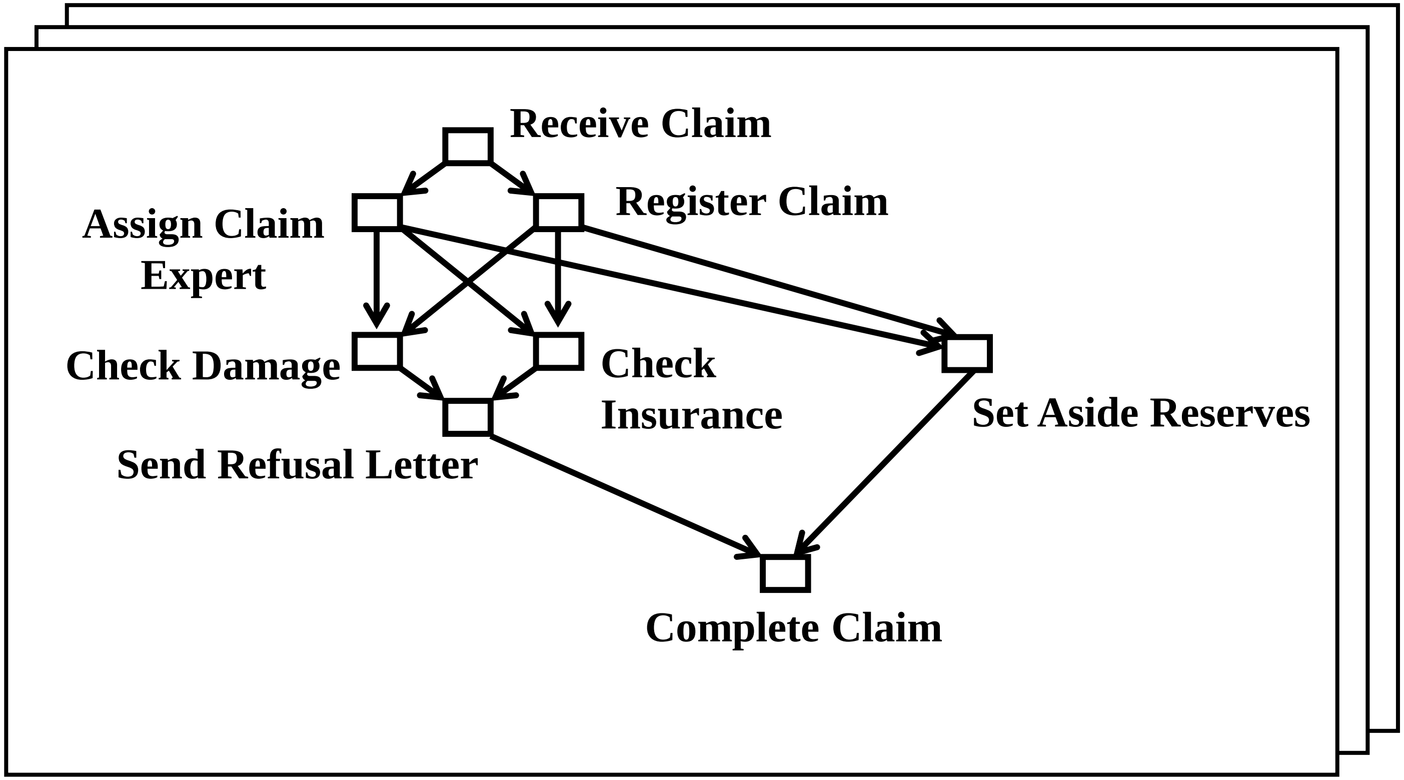
Case Study

Formal runs in an insurance company



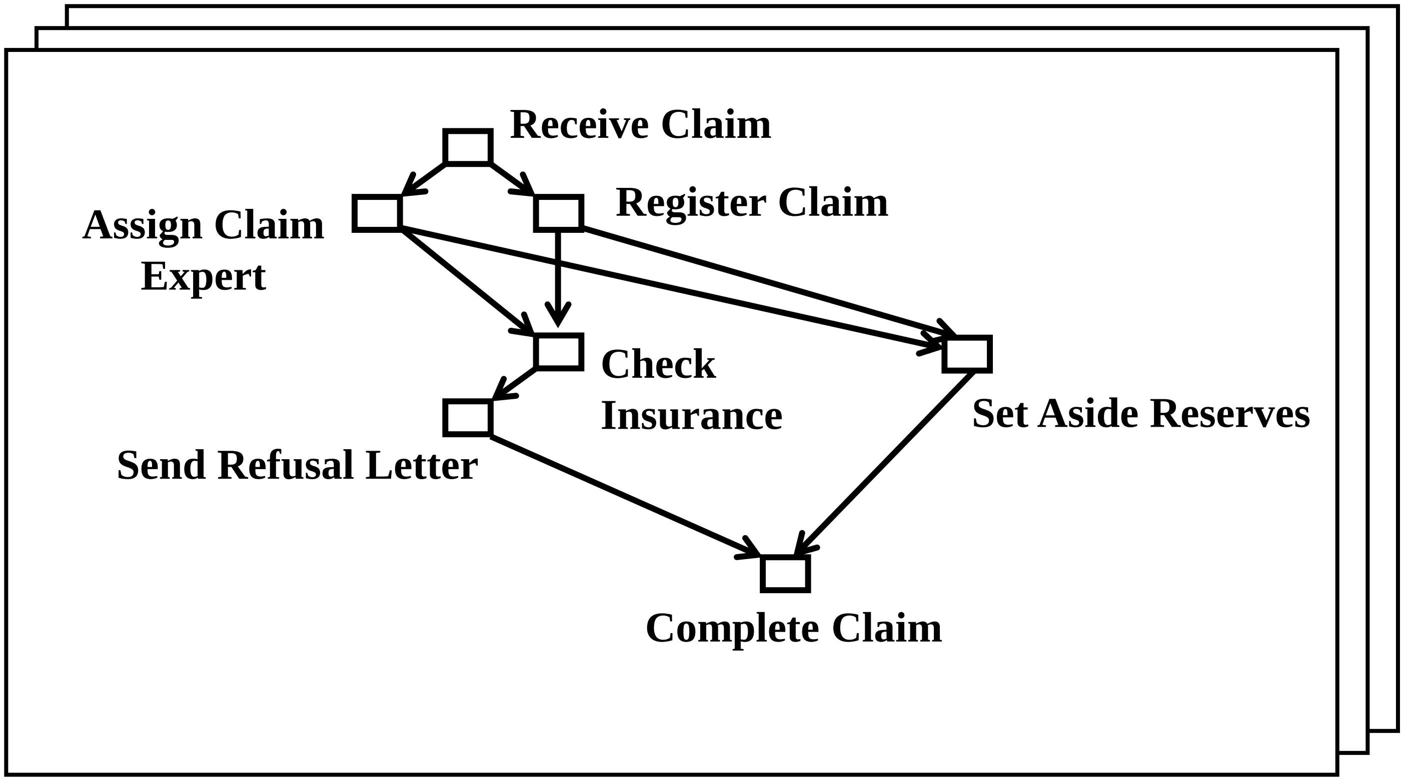
Case Study

Formal runs in an insurance company: Composed Run 1



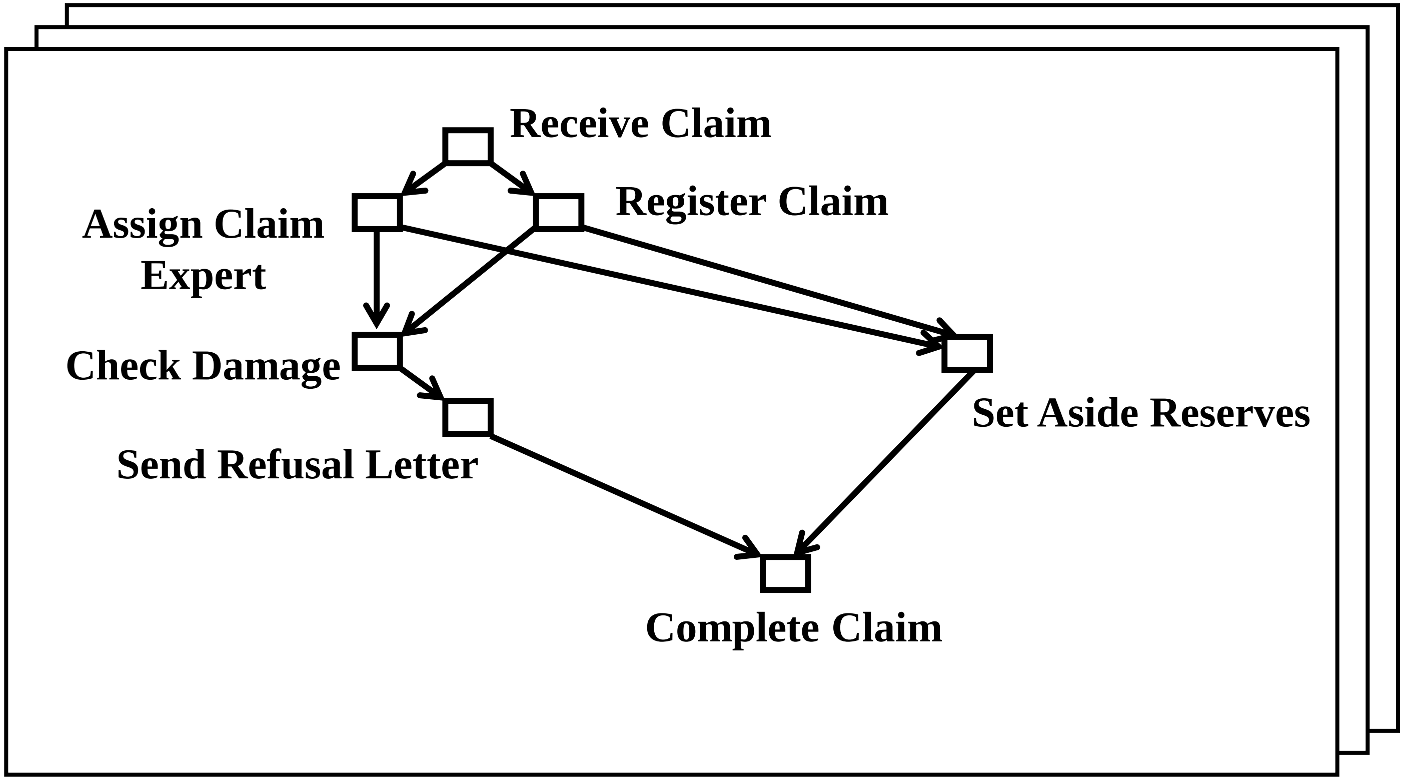
Case Study

Formal runs in an insurance company: Composed Run 2



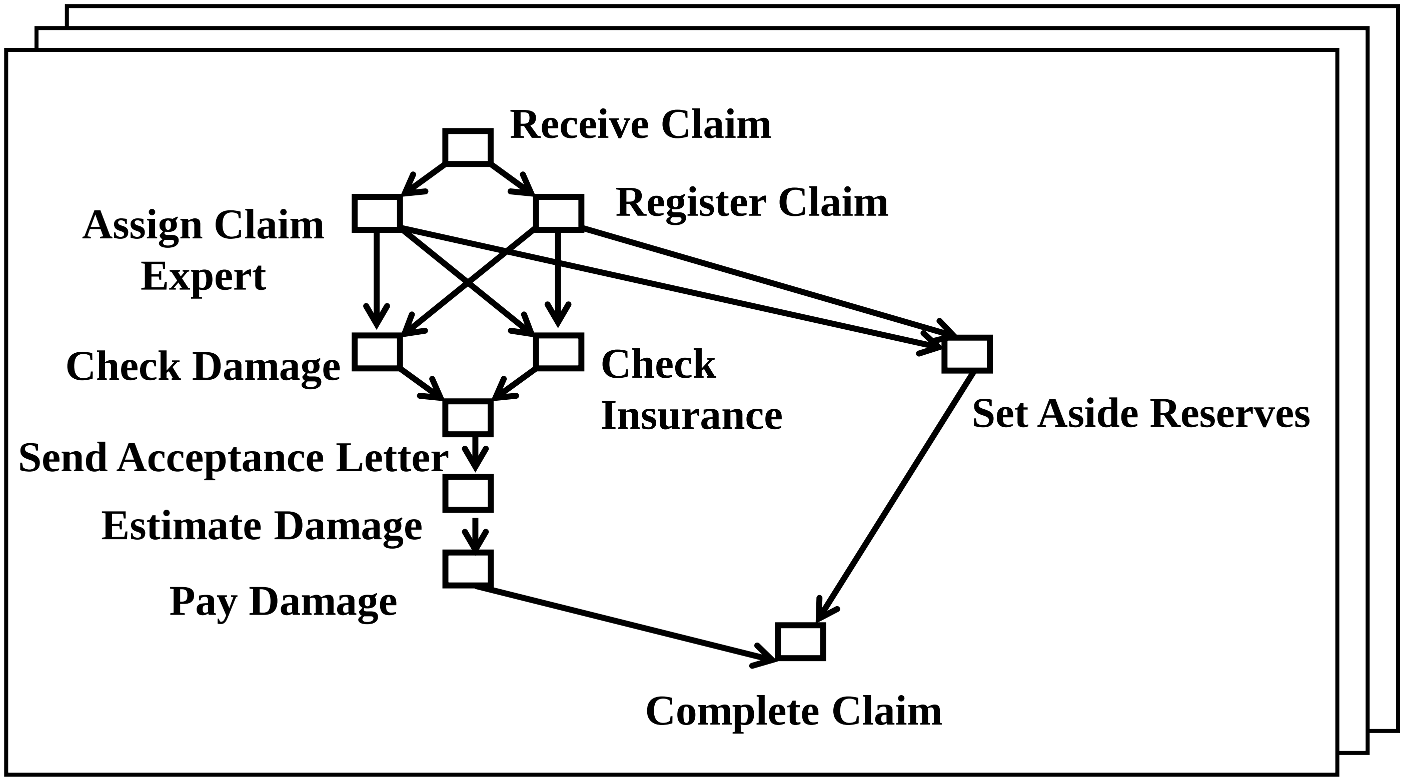
Case Study

Formal runs in an insurance company: Composed Run 3



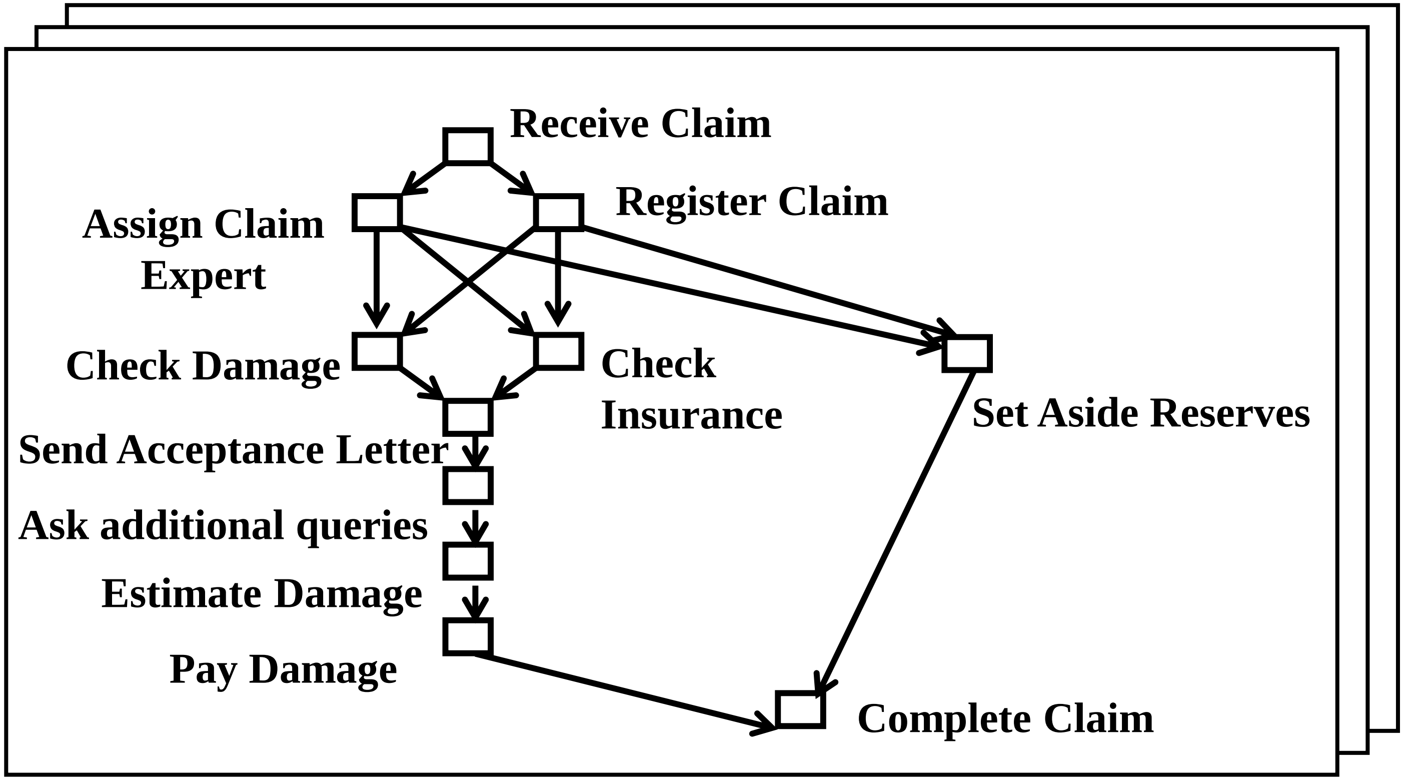
Case Study

Formal runs in an insurance company: Composed Run 4



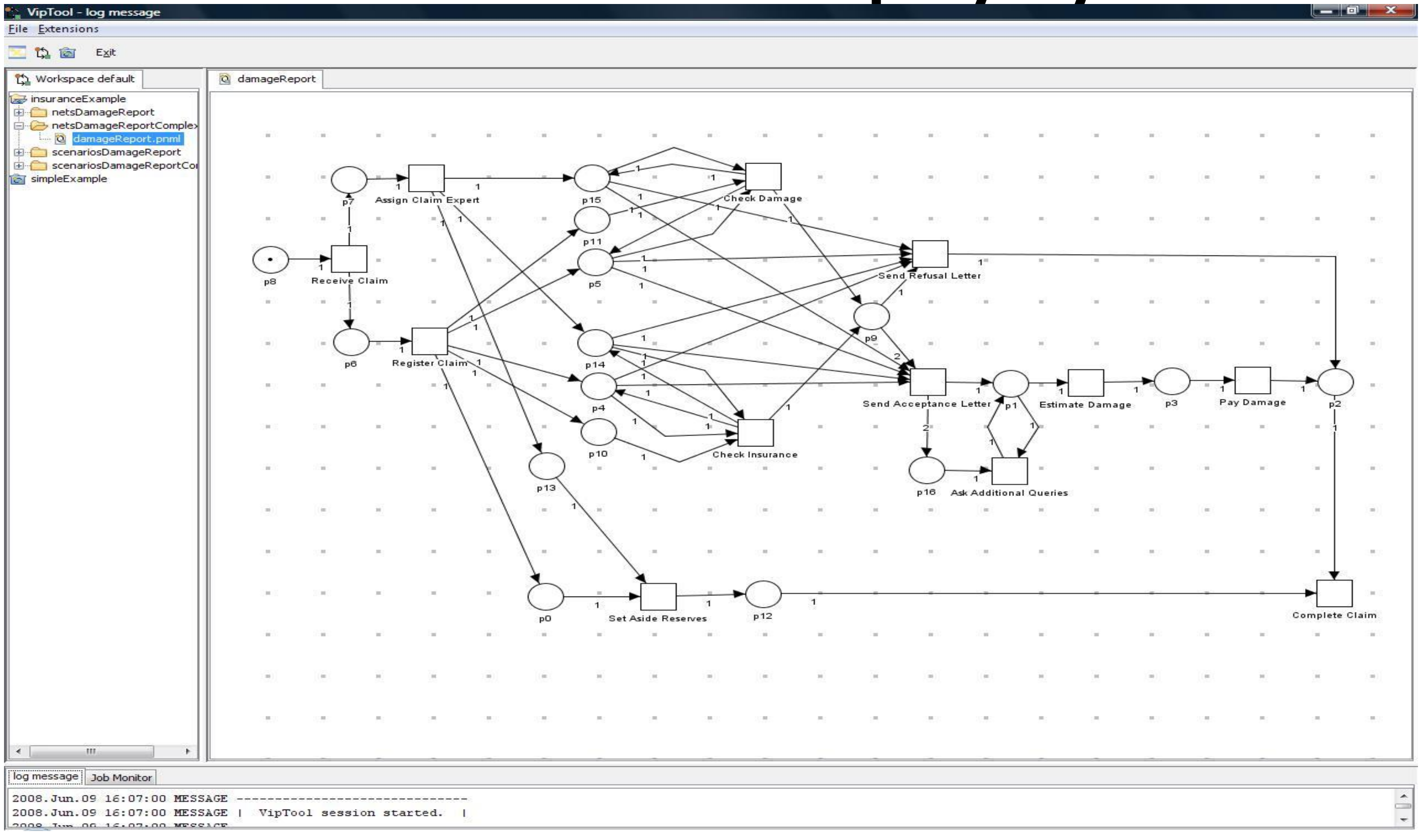
Case Study

Formal runs in an insurance company: Composed Run 5

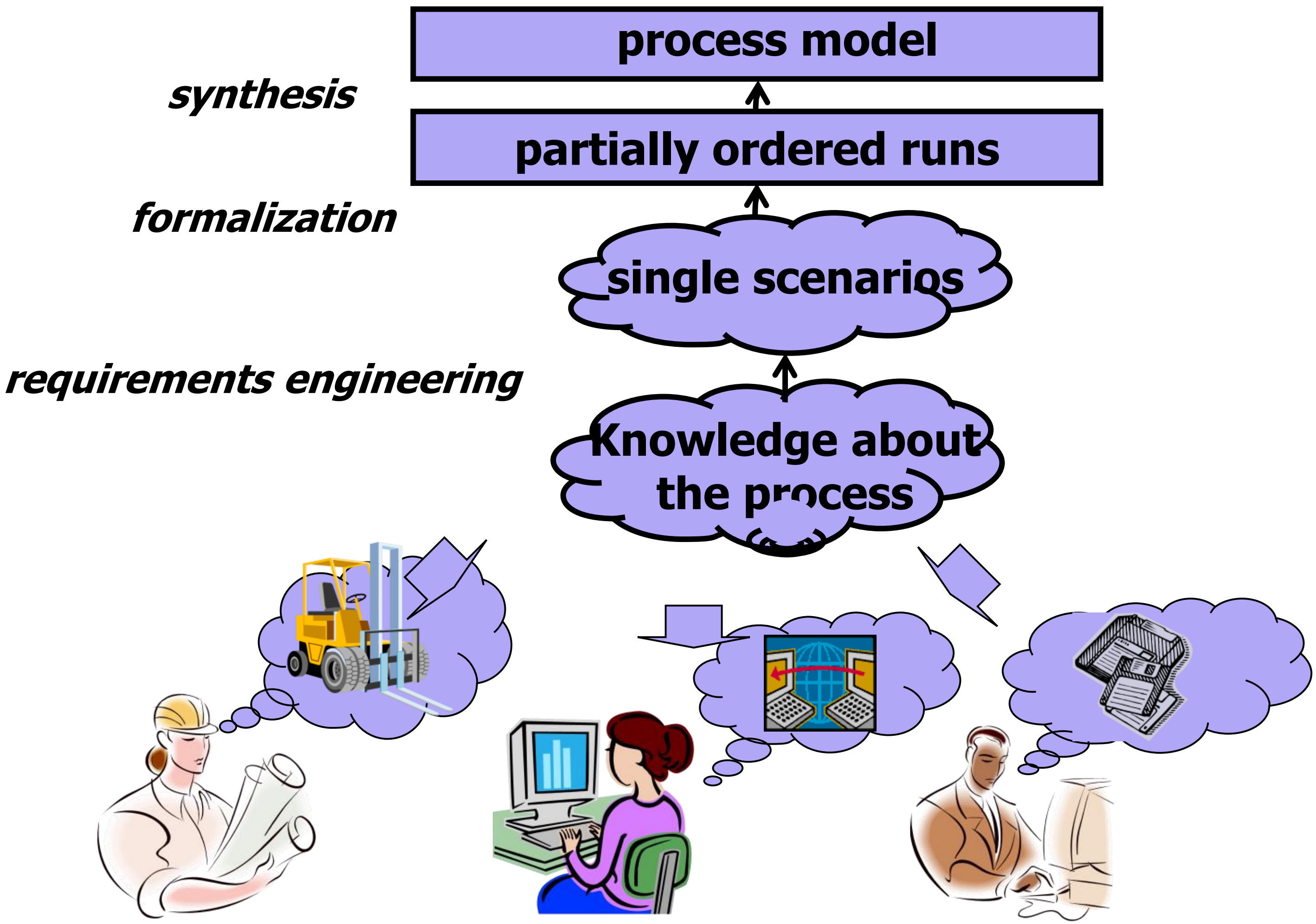


Case Study

Formal runs in an insurance company: Synthesis Result

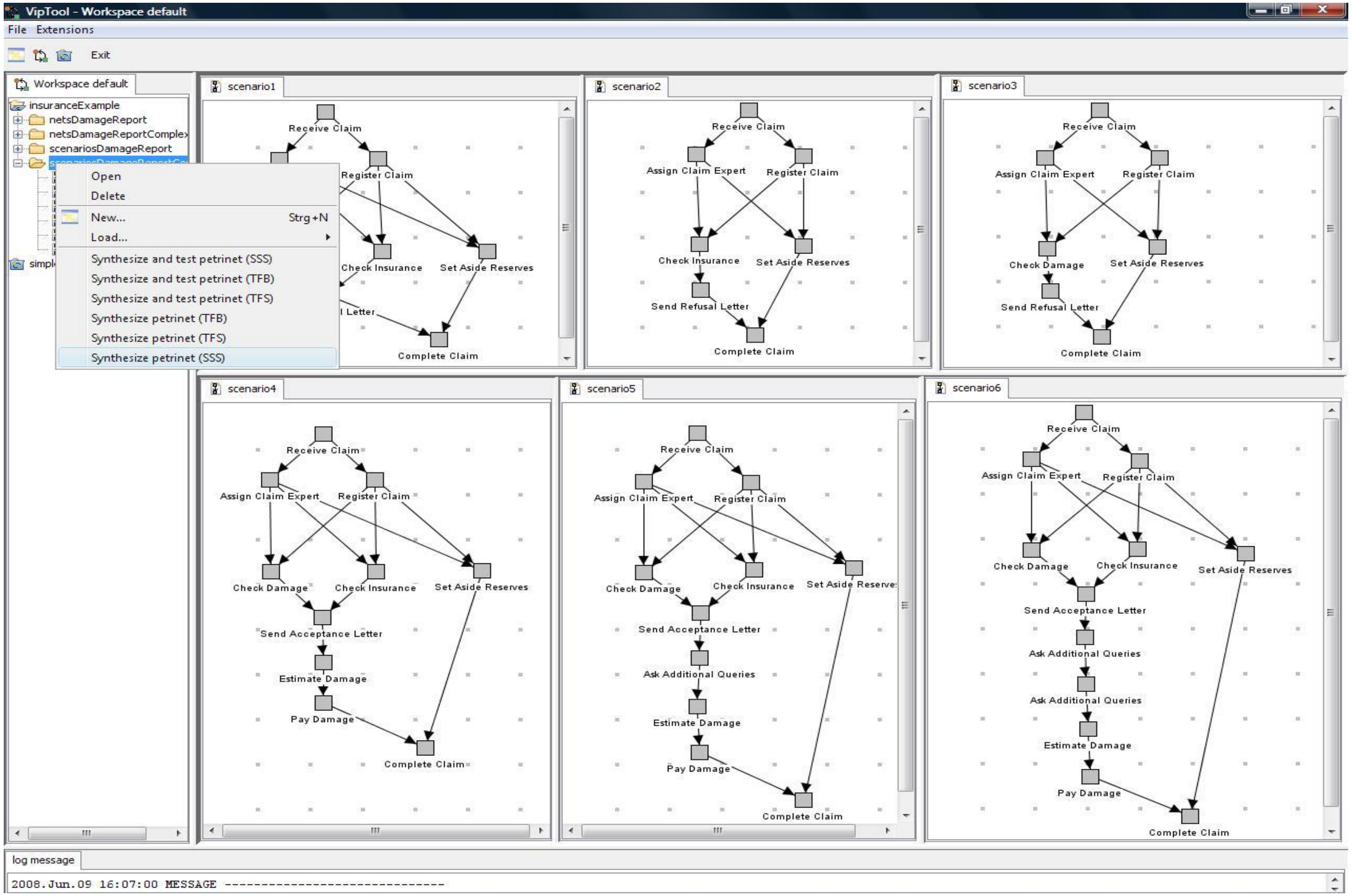


Case Study


process model
synthesis
partially ordered runs
formalization
single scenarios
requirements engineering
Knowledge about the process

Tool Support

Tool Support



Tool Support