Removing Implicit Places Using Regions for Process Discovery

Seminar - Selected Topics in Process Mining

Faizan Zafar

Chair of Process and Data Science

RWTH Aachen University

faizan.zafar@rwth-aachen.de

January 24th, 2021

Outline

- Literature
- Introduction
- Motivation
- Removing Implicit Places
- Evaluation
- Related Work
- Conclusion
- Future Work

Literature

- [L. Mannel, R. Bergenthum, W. Aalst+ 20] Removing Implicit Places Using Regions for Process Discovery. 41st International Conference on Application and Theory of Petri Nets and Concurrency, 2020
- Process discovery aims to discover simple Petri nets
- Petri nets may contain *implicit* places which have adverse outcomes
- A novel technique to identify and remove implicit places
- Integrating the approach with the eST-Miner algorithm

Introduction: Process Discovery

Input: Event Log

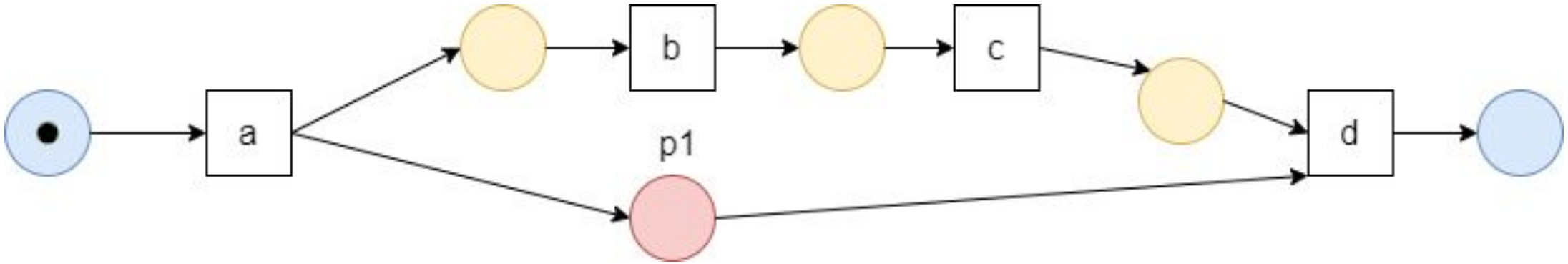
Output: Process model

Introduction: Preliminaries

- Activity -> Trace -> Event Log
- A marked Petri net (N,m_0) is a process model composed of places P, transitions T and a multiset of arcs F -> $\mathbb{N}_0$
- Language L of a Petri net N
- Implicit Place p
- Token Count Function x_p

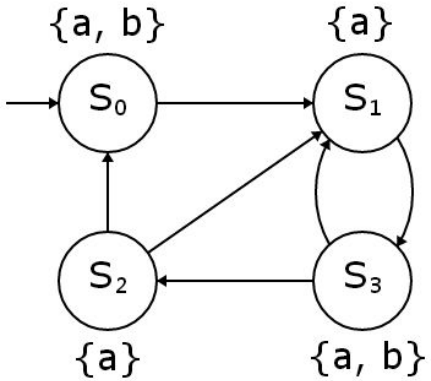
Introduction: Preliminaries

$$L = [\epsilon, \langle a, b, c \rangle^4]$$



Introduction: Region Theory

Transition System



Prefix-closed Language

$\langle a \rangle$
$\langle a, b \rangle$
$\langle a, b, c \rangle$
$\langle a, b, c, d \rangle$

Introduction: Region Theory

- Identifying *regions* in the event log
- A region is a mathematical structure satisfying certain conditions
- State-based region theory makes use of a transition system
- Language-based region theory makes use of a prefix-closed language
- Each region corresponds to a feasible place in the resulting Petri net
- Different approaches exist for finding the subset of regions

Motivation

- Minimal regions method avoids adding implicit places to a model
- *r'* is a sub-region of *r* if *r'* is a subset of *r*. *r* is a minimal region if and only if there is no other region *r'* which is a sub-region of *r*. For each minimal region *r*, a place *p* is produced.
- Transitioning directly from the event log to the resulting model
- The goal is to remove implicit places from a discovered set of places

Removing Implicit Places: Main Idea

➢ **Given an Event Log and the corresponding Petri net N:**

- $p_1, p_2 \in P$ **are two individual places in** N
- **Comparing** p_1 **and** p_2**, we verify the conditions:**

$$\forall \sigma \in L(N) : x_{p_1}(\sigma) \geq x_{p_2}(\sigma) \qquad (1)$$

$$\exists \sigma \in L(N) : x_{p_1}(\sigma) > x_{p_2}(\sigma) \qquad (2)$$

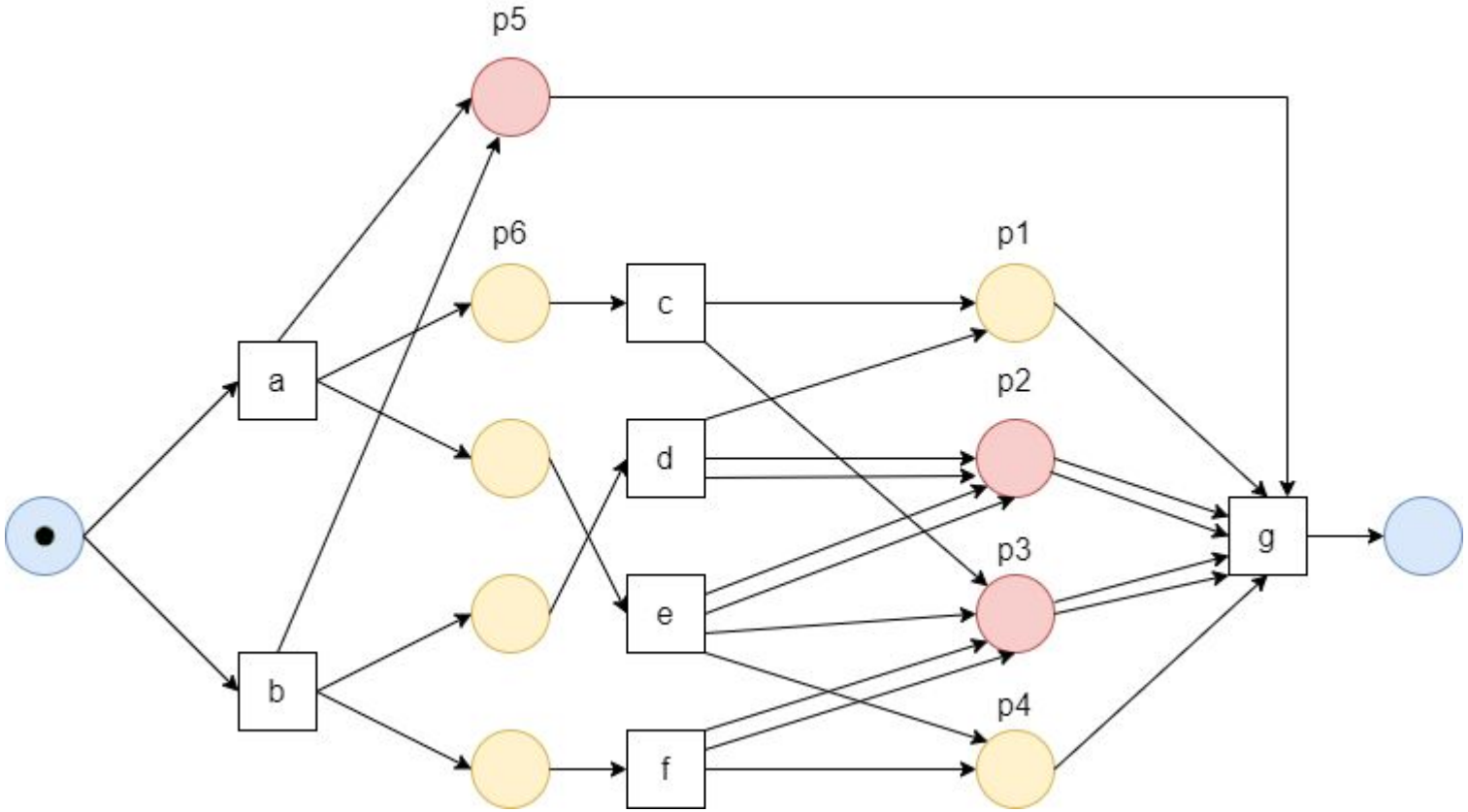
- **If** $x_{p1} > x_{p2}$ **for the prefix-closed language of** N**, then**

$$x_{p_3} = x_{p_1} - x_{p_2} \qquad (3)$$

is a region and p_1 **can be regarded as *implicit***

Removing Implicit Places: Fundamental Case

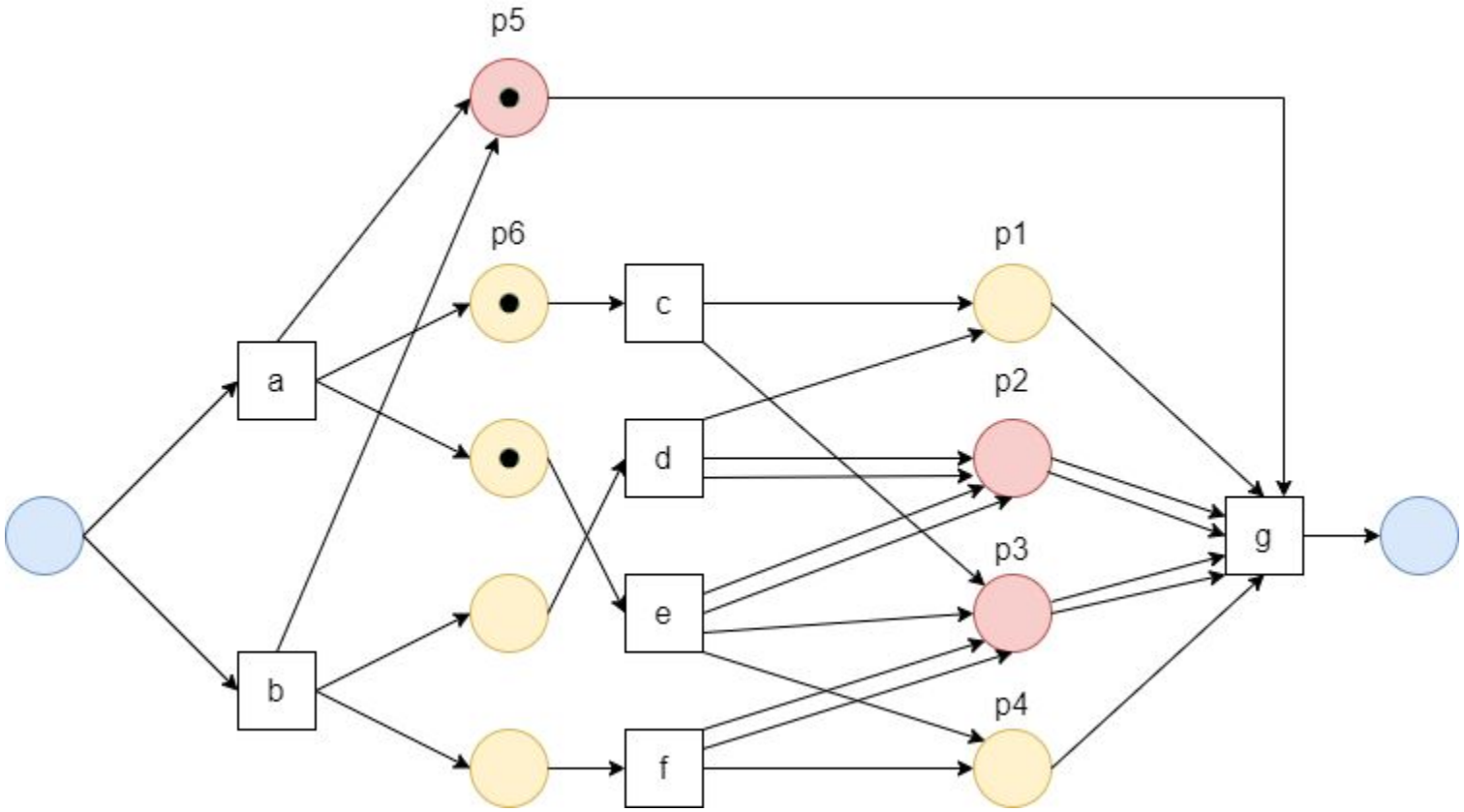
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
p_5	0				
p_6	0				
p_7					

Removing Implicit Places: Fundamental Case

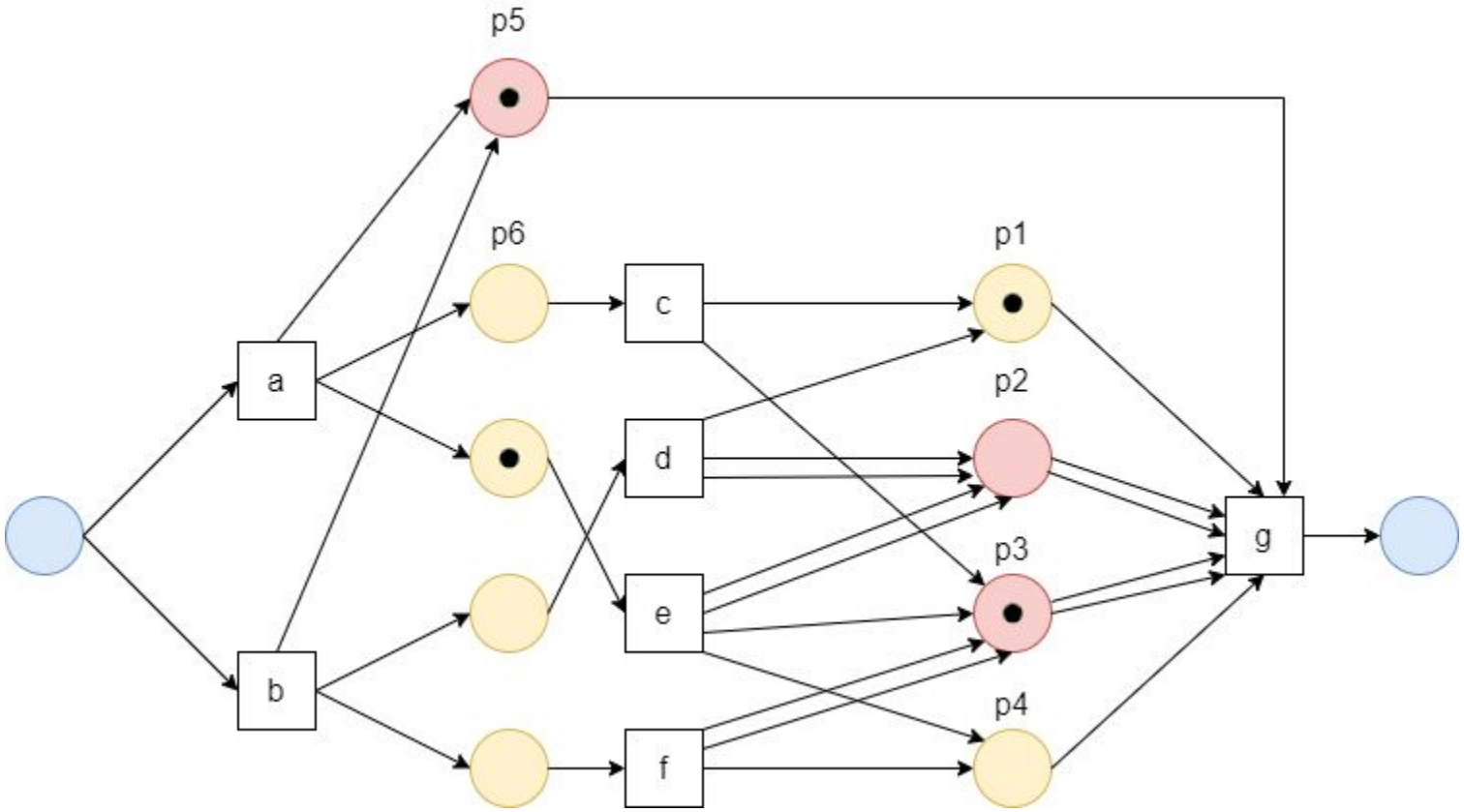
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ϵ	a	c	e	g
p_5	0	1			
p_6	0	1			
p_7					

Removing Implicit Places: Fundamental Case

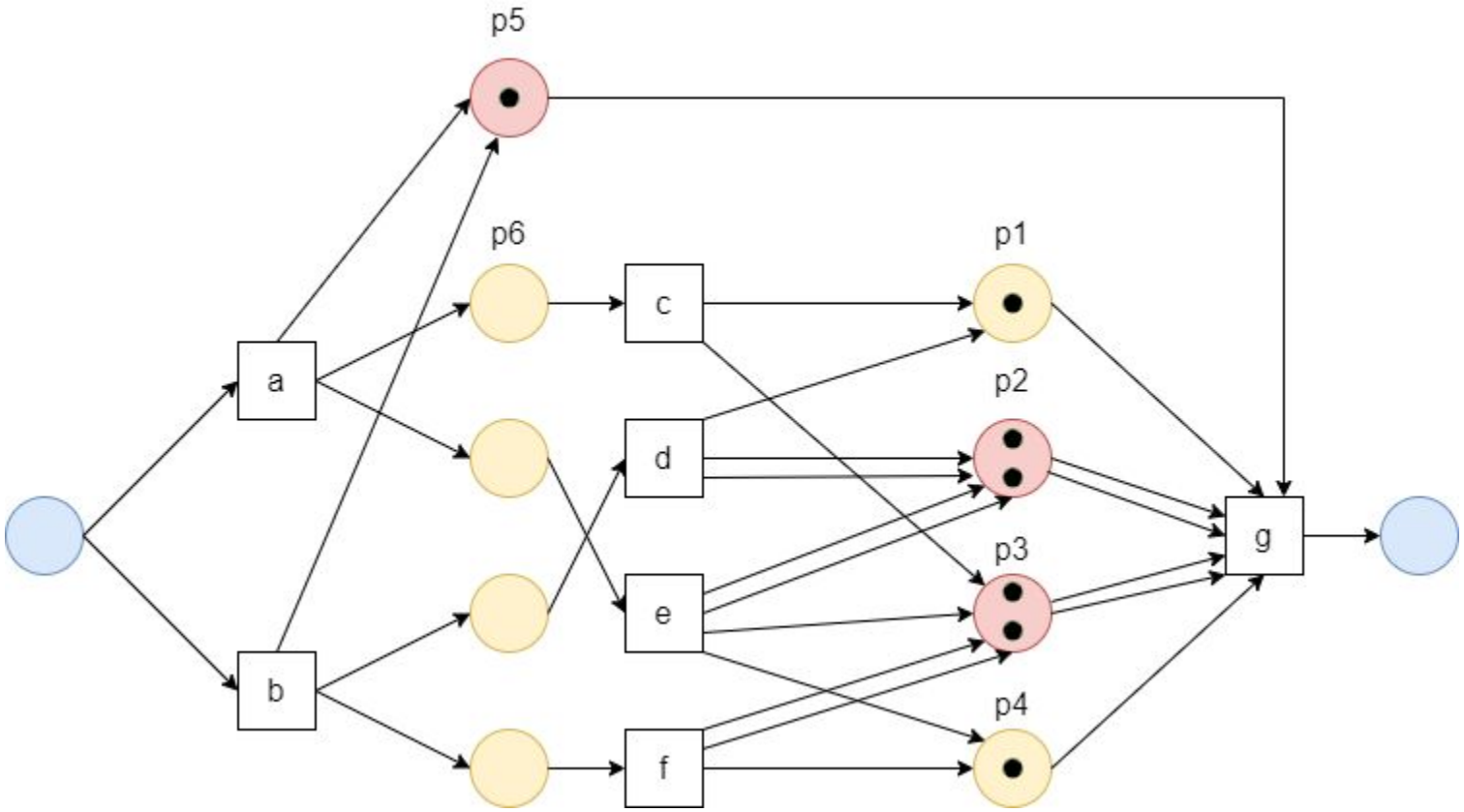
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
p_5	0	1	1		
p_6	0	1	0		
p_7					

Removing Implicit Places: Fundamental Case

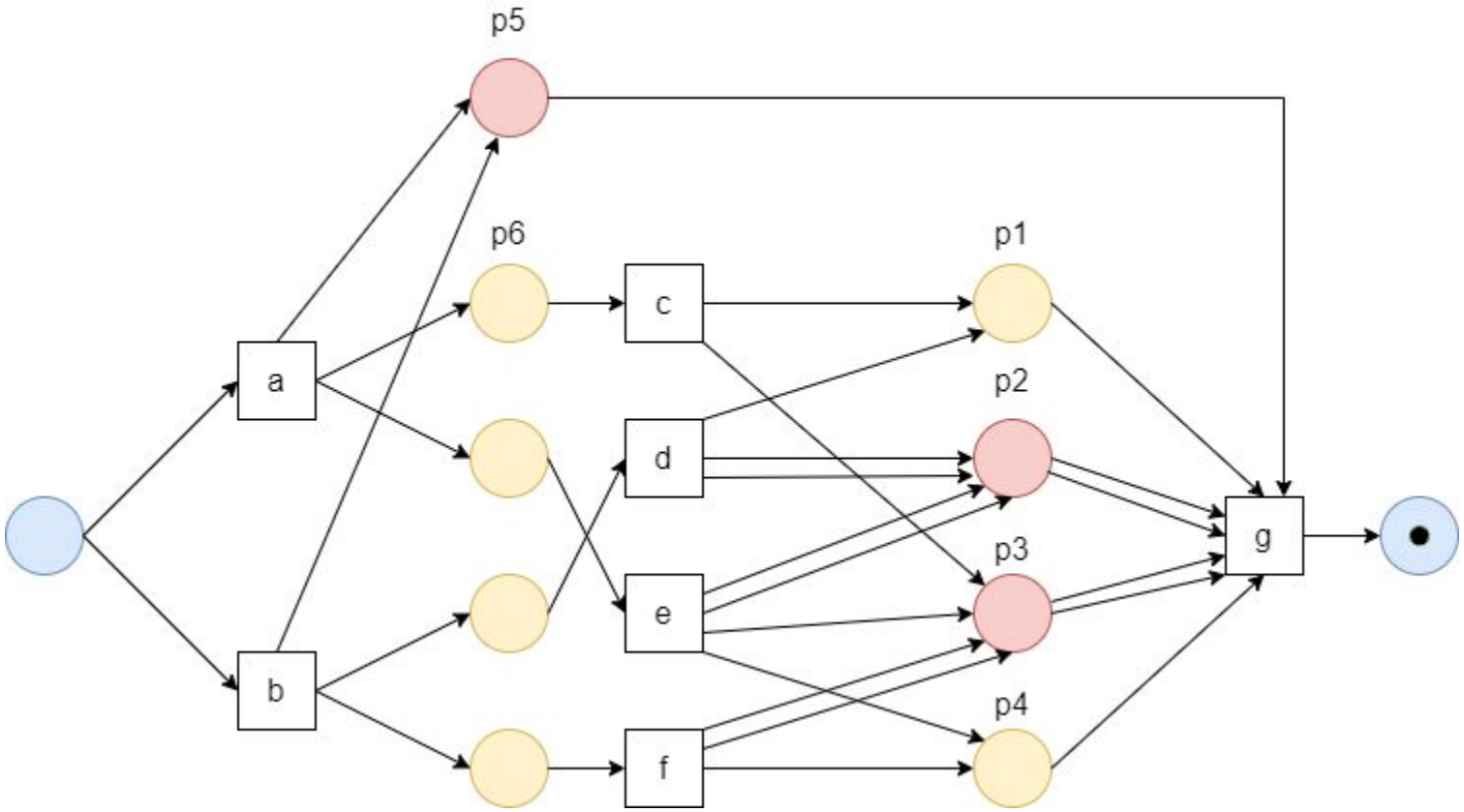
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
p_5	0	1	1	1	
p_6	0	1	0	0	
p_7					

Removing Implicit Places: Fundamental Case

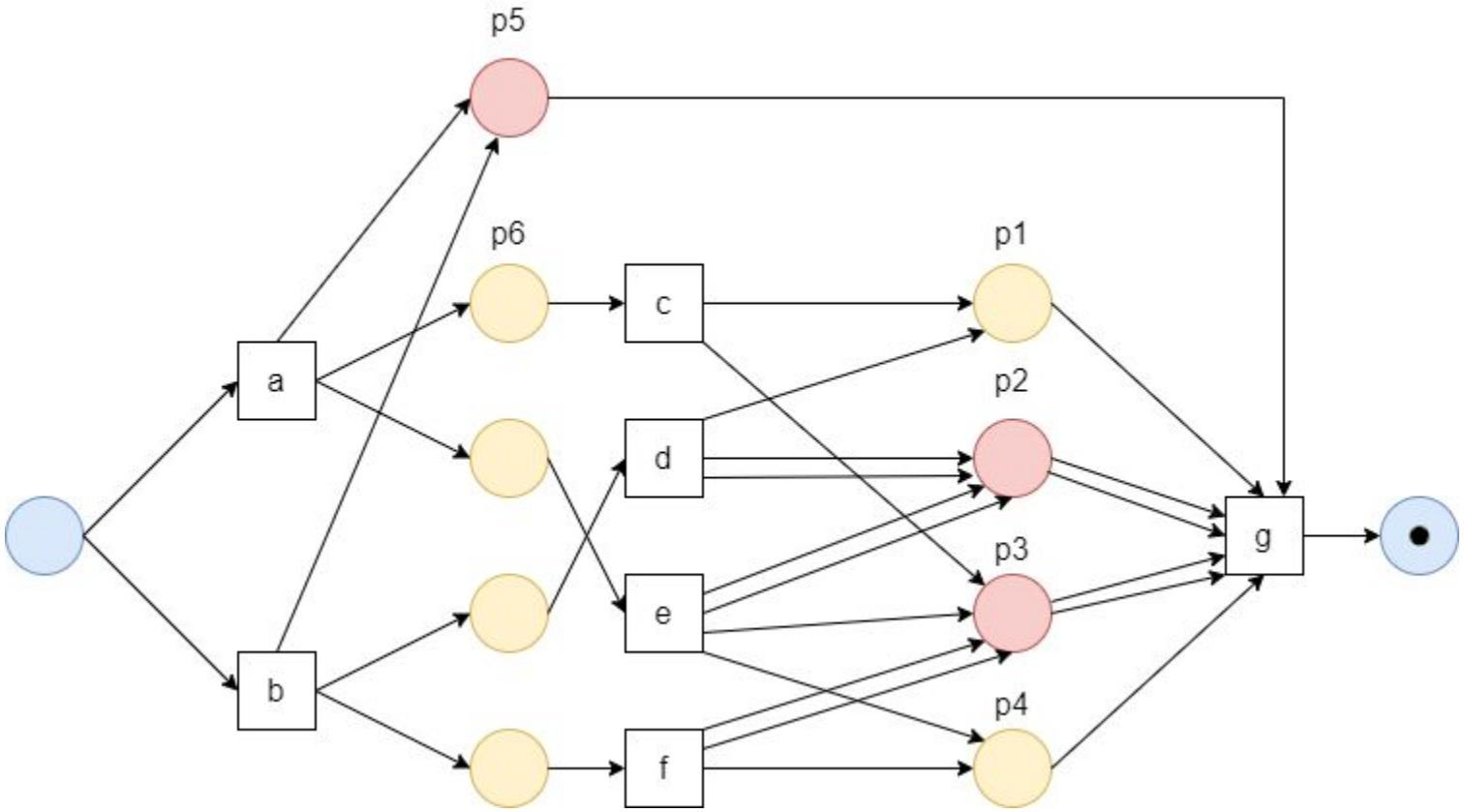
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
p_5	0	1	1	1	0
p_6	0	1	0	0	0
p_7					

Removing Implicit Places: Fundamental Case

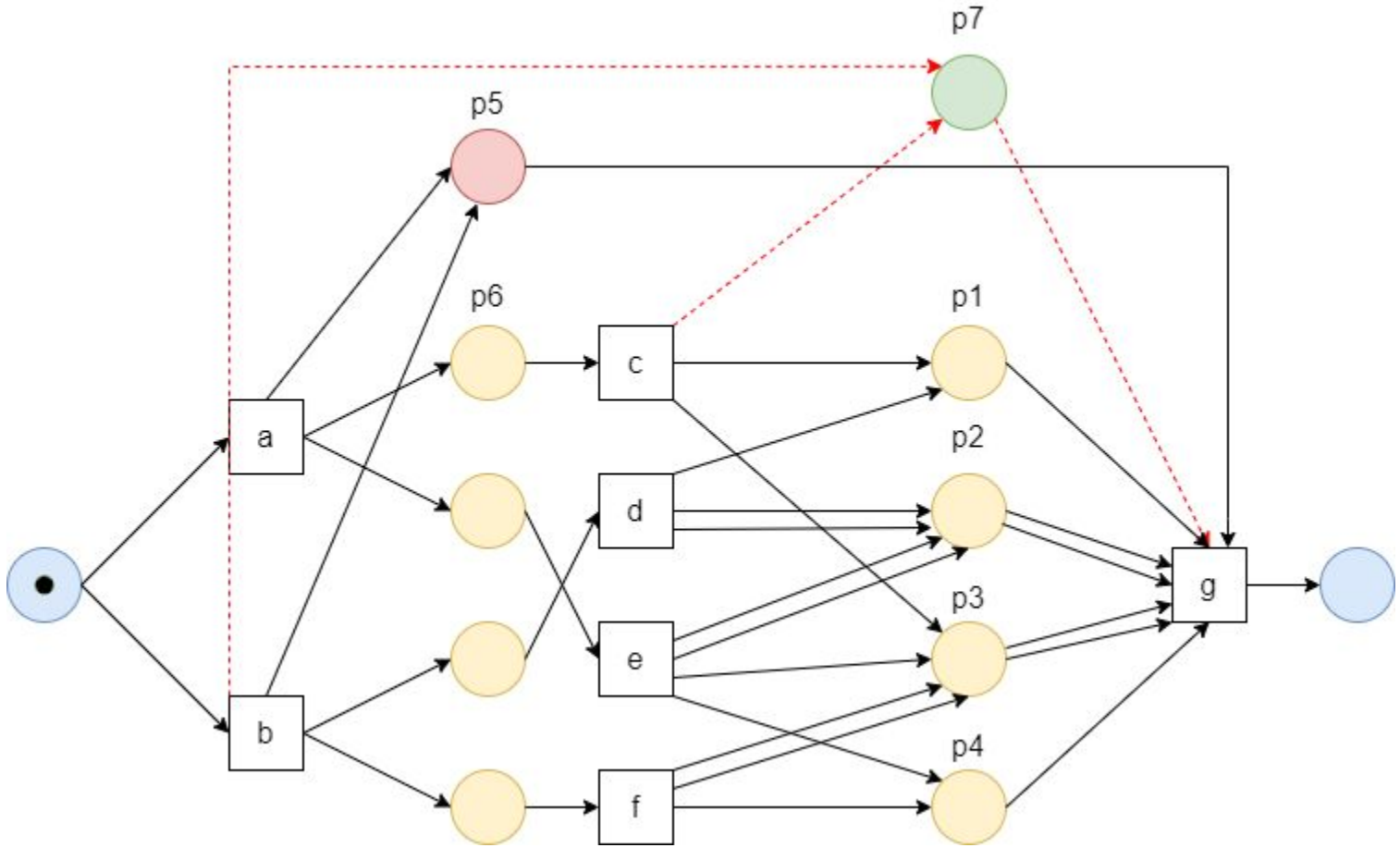
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
p_5	0	1	1	1	0
p_6	0	1	0	0	0
p_7	0	0	1	1	0

Removing Implicit Places: Fundamental Case

$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
p_5	0	1	1	1	0
p_6	0	1	0	0	0
p_7	0	0	1	1	0

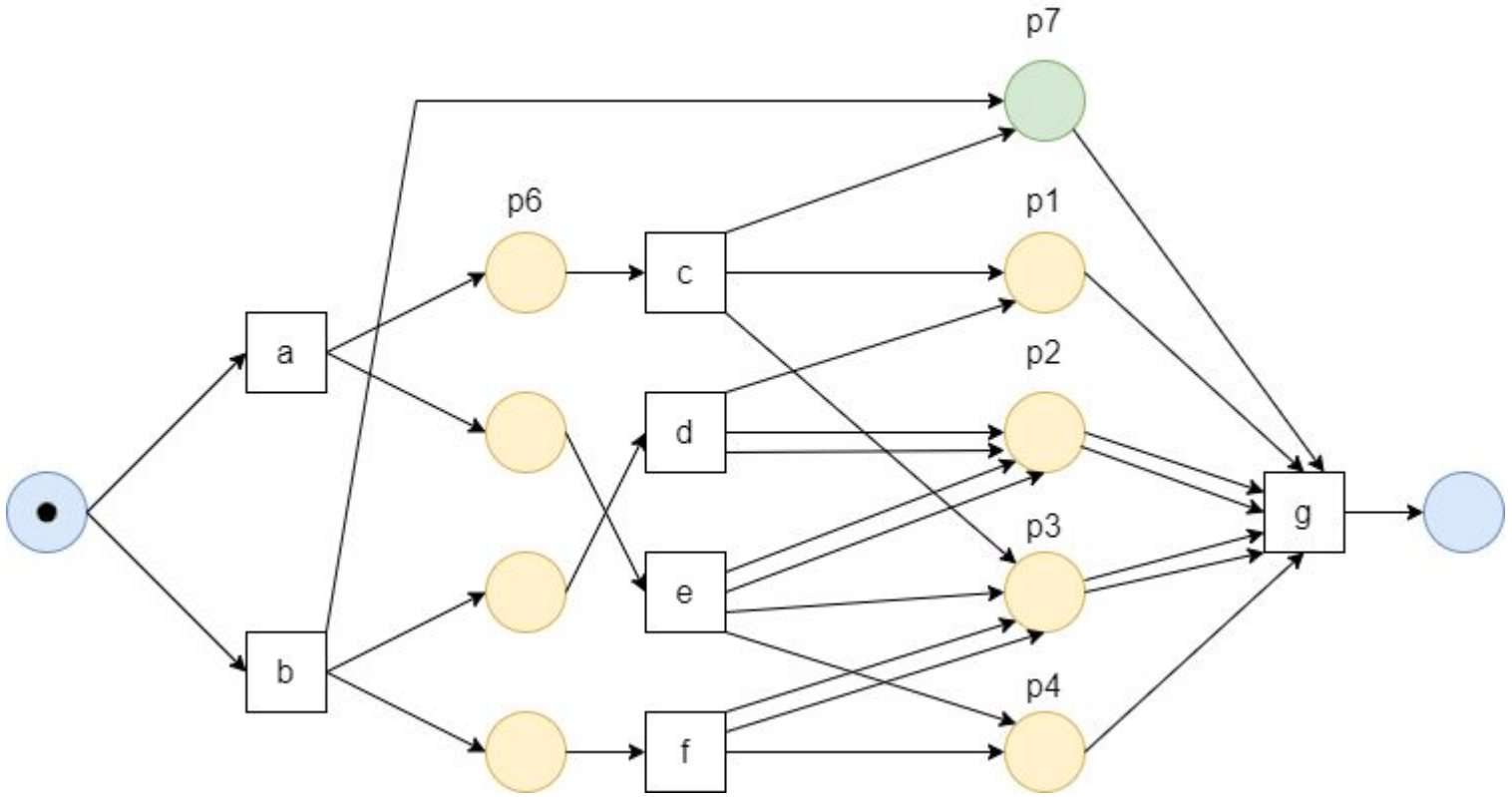
	ε	a	e	c	g
p_5	0	1	1	1	0
p_6	0	1	1	0	0
p_7	0	0	0	1	0

	ε	b	d	f	g
p_5	0	1	1	1	0
p_6	0	0	0	0	0
p_7	0	1	1	1	0

	ε	b	f	d	g
p_5	0	1	1	1	0
p_6	0	0	0	0	0
p_7	0	1	1	1	0

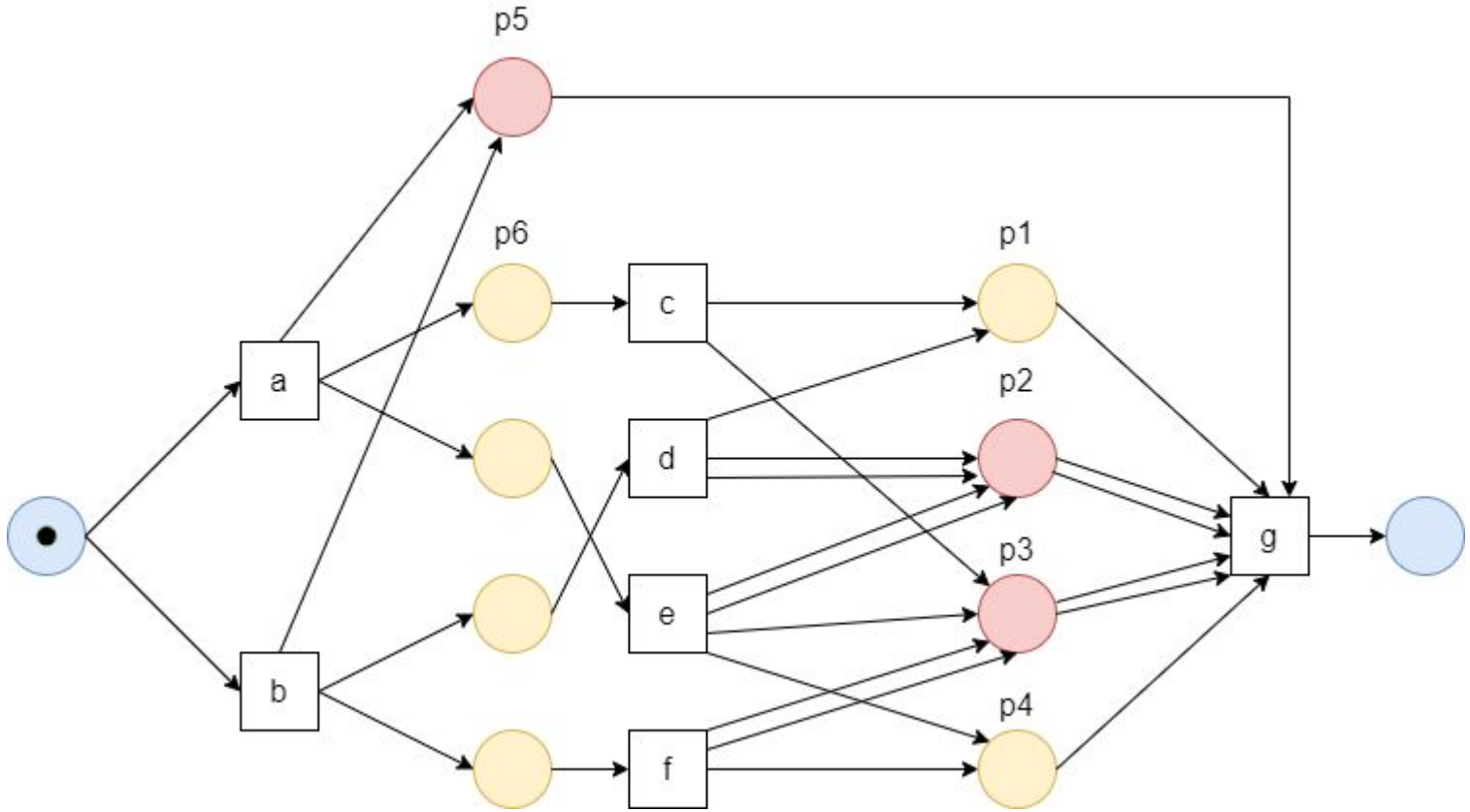
Removing Implicit Places: Fundamental Case

$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



Removing Implicit Places: Extended Case

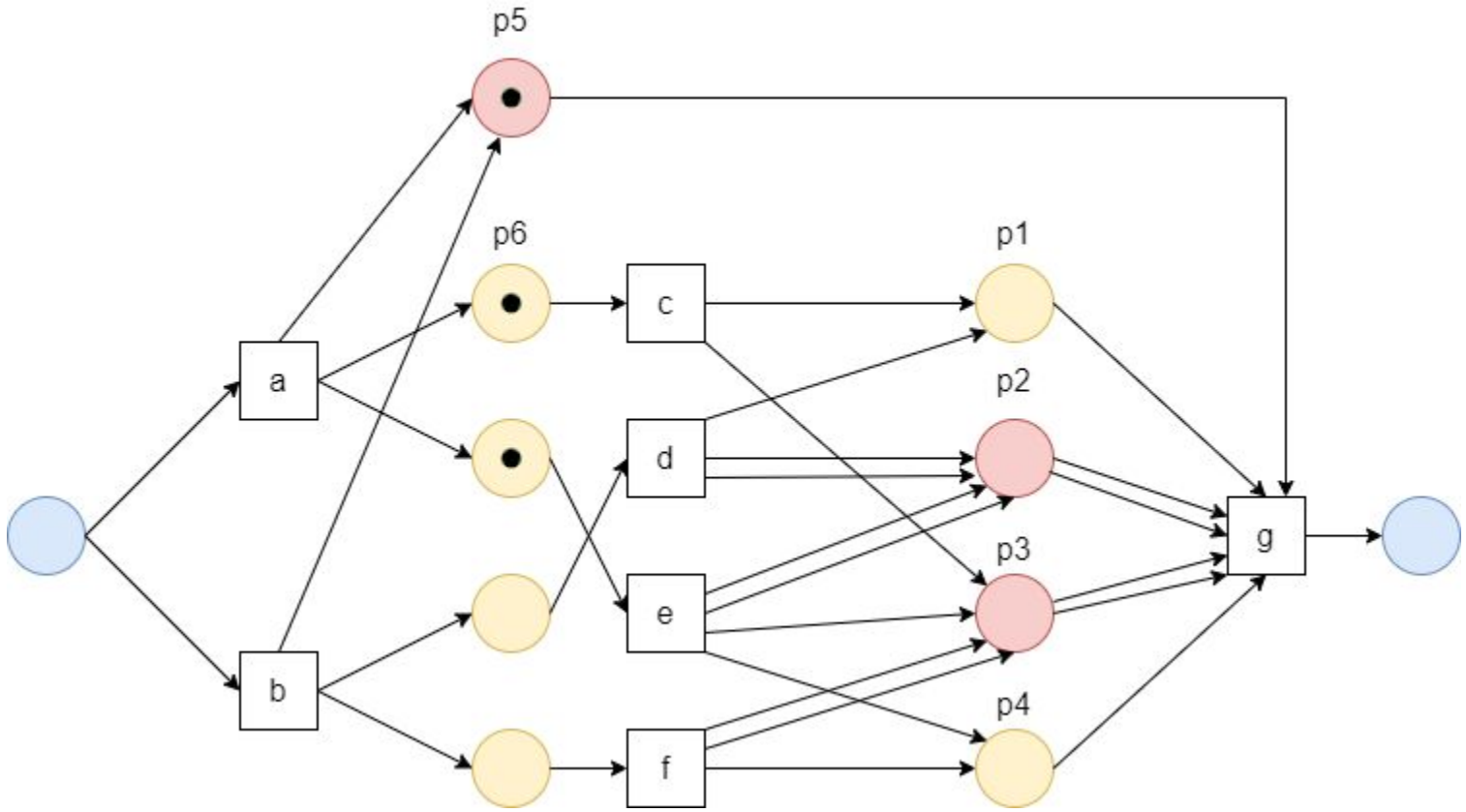
$$L = [\epsilon, \langle a, c, e, g \rangle^2, \langle a, e, c, g \rangle^3, \langle b, d, f, g \rangle^2, \langle b, f, d, g \rangle^4]$$



	ϵ	a	c	e	g
p_3	0				
p_4	0				
p_8					

Removing Implicit Places: Extended Case

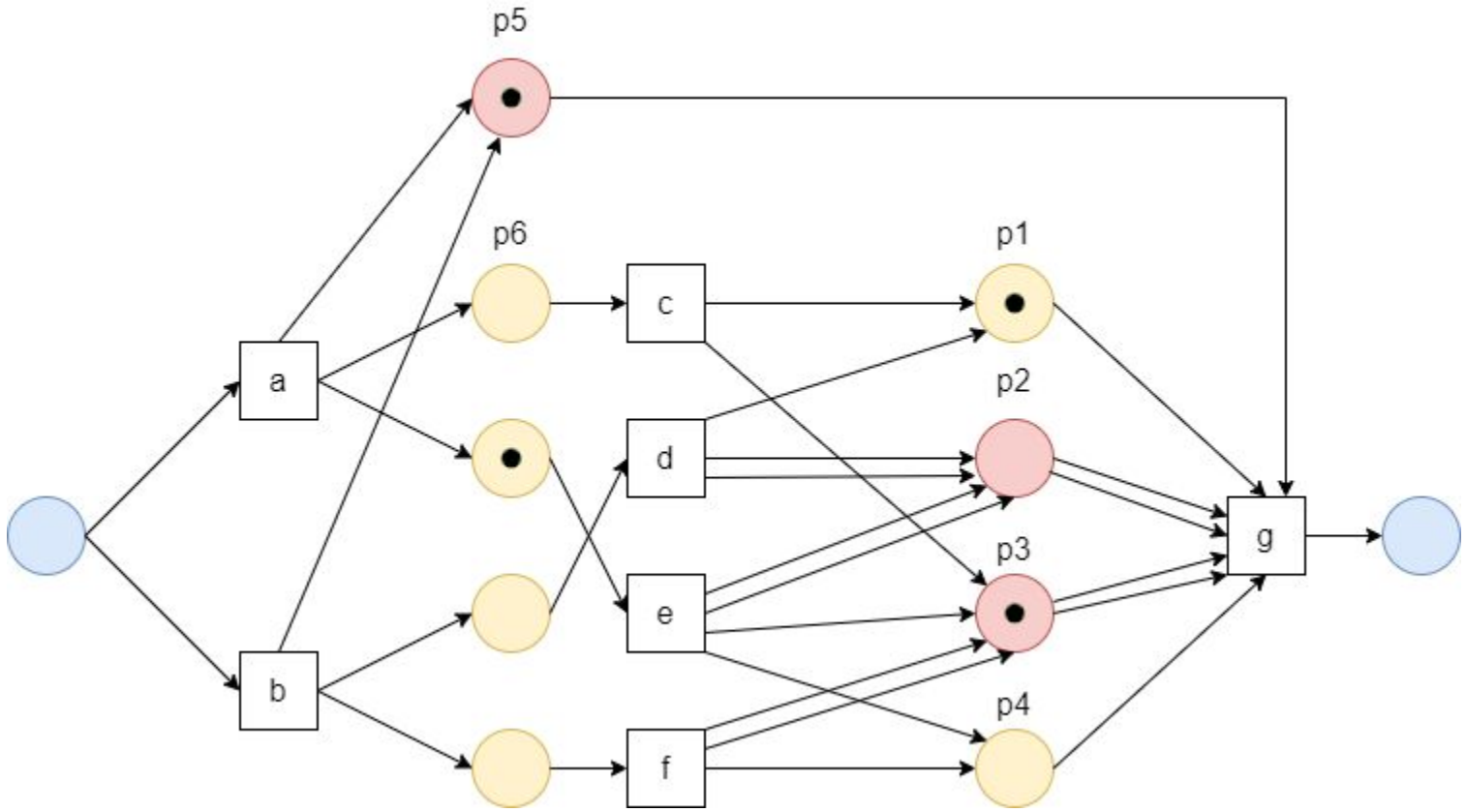
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ϵ	a	c	e	g
p_3	0	0			
p_4	0	0			
p_8					

Removing Implicit Places: Extended Case

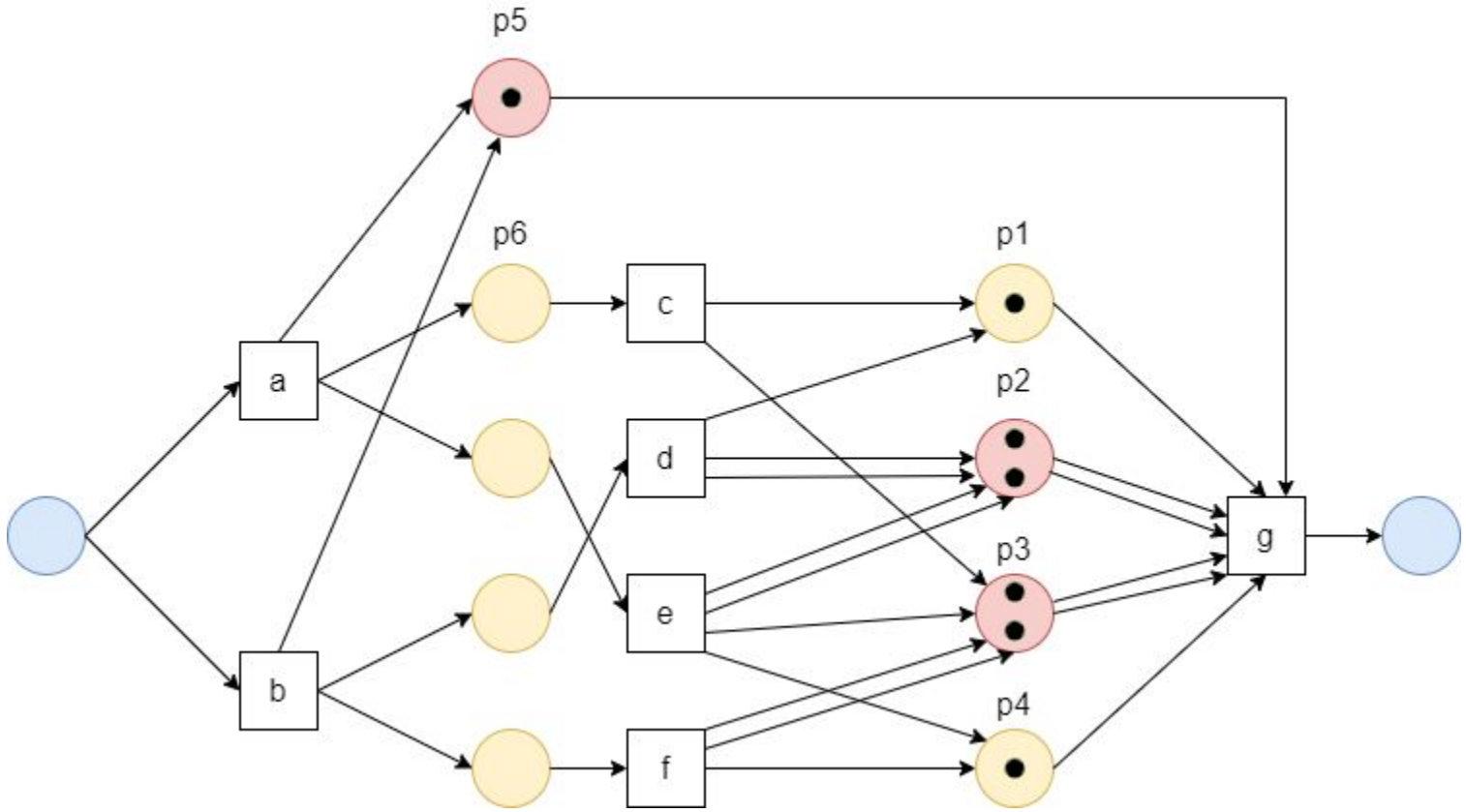
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ϵ	a	c	e	g
p_3	0	0	1		
p_4	0	0	0		
p_8					

Removing Implicit Places: Extended Case

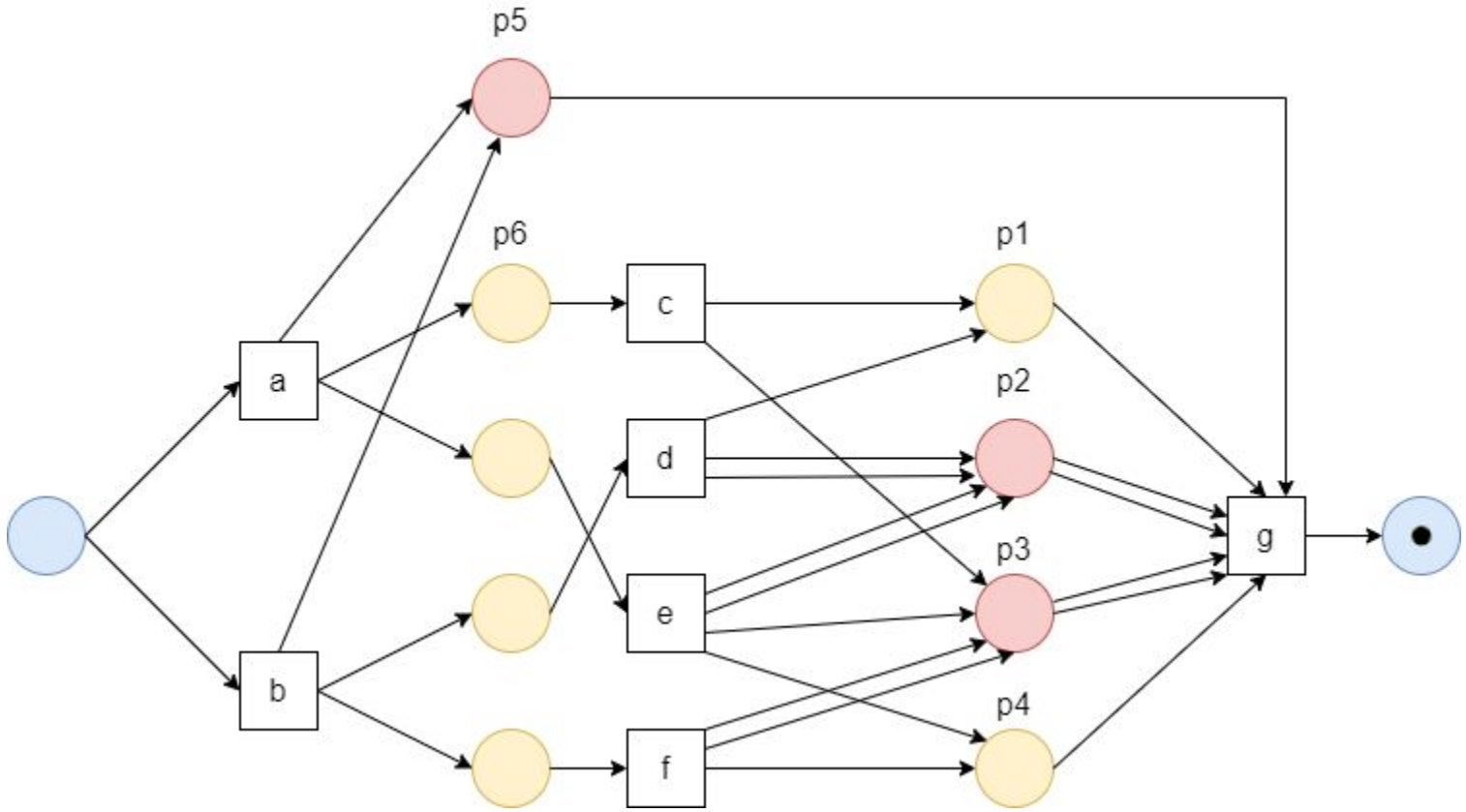
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ϵ	a	c	e	g
p_3	0	0	1	2	
p_4	0	0	0	1	
p_8					

Removing Implicit Places: Extended Case

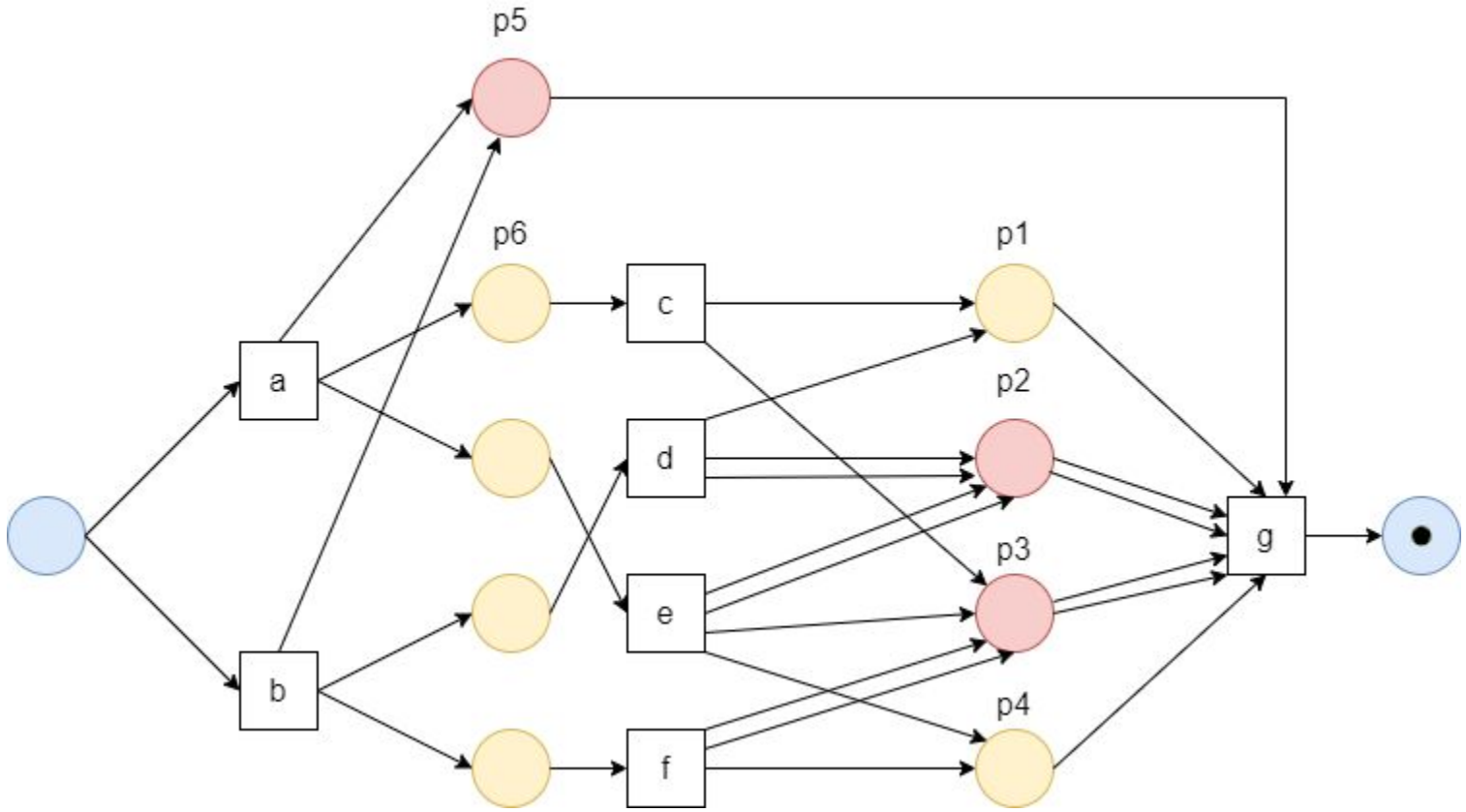
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ϵ	a	c	e	g
p_3	0	0	1	2	0
p_4	0	0	0	1	0
p_8					

Removing Implicit Places: Extended Case

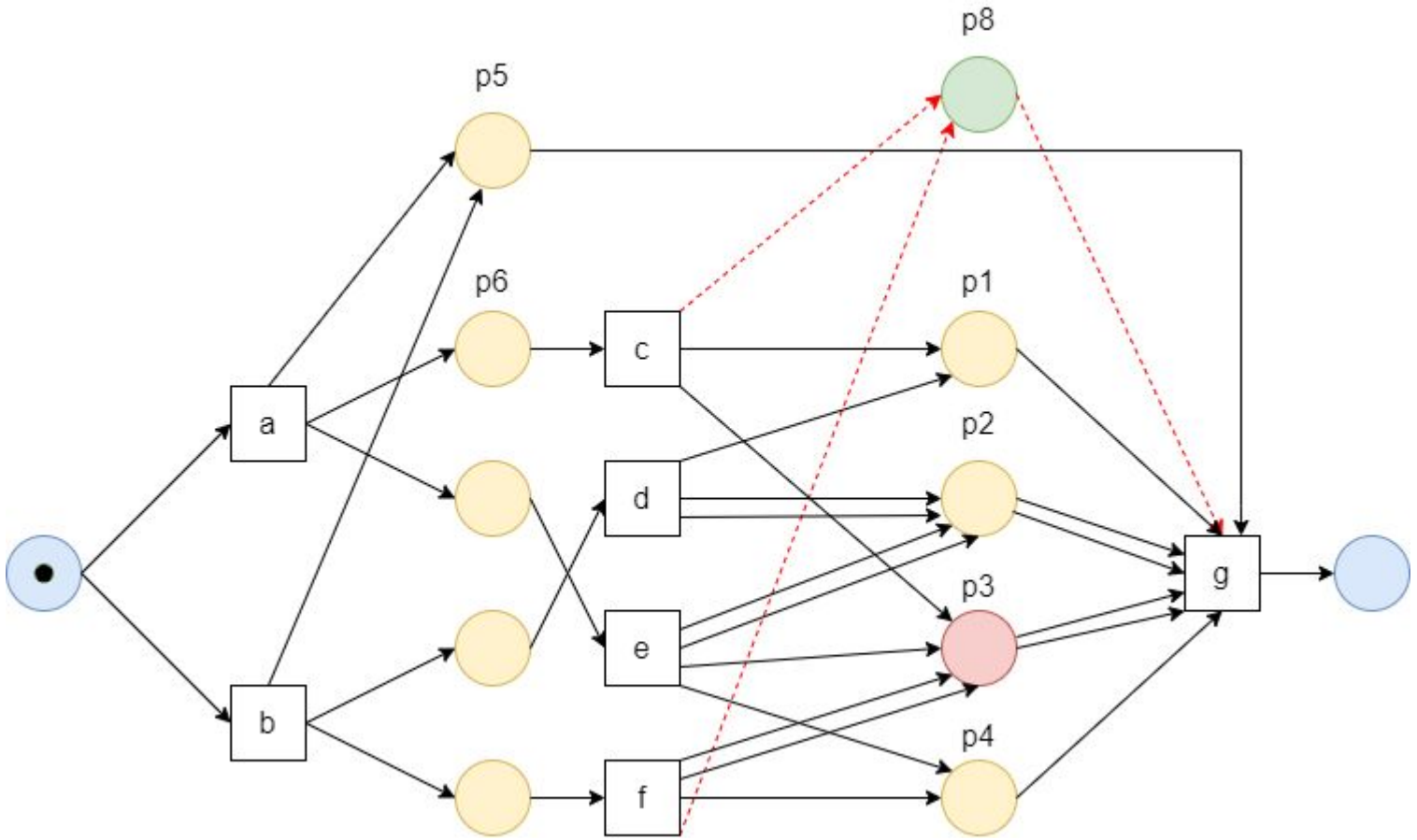
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
p_3	0	0	1	2	0
p_4	0	0	0	1	0
p_8	0	0	1	1	0

Removing Implicit Places: Extended Case

$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
p_3	0	0	1	2	0
p_4	0	0	0	1	0
p_8	0	0	1	1	0

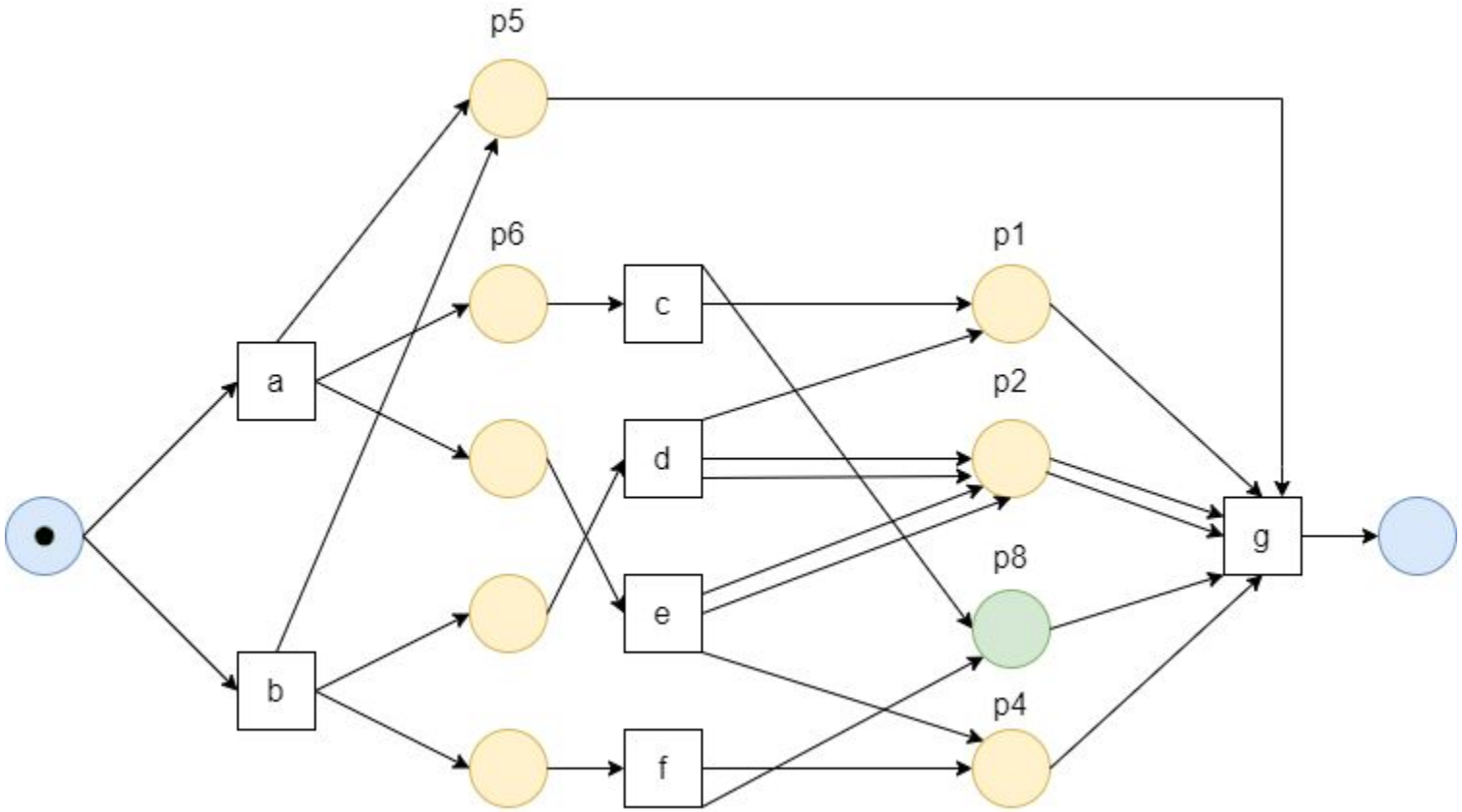
	ε	a	e	c	g
p_3	0	0	1	2	0
p_4	0	0	1	1	0
p_8	0	0	0	1	0

	ε	b	d	f	g
p_3	0	0	0	2	0
p_4	0	0	0	1	0
p_8	0	0	0	1	0

	ε	b	f	d	g
p_3	0	0	2	2	0
p_4	0	0	1	1	0
p_8	0	0	1	1	0

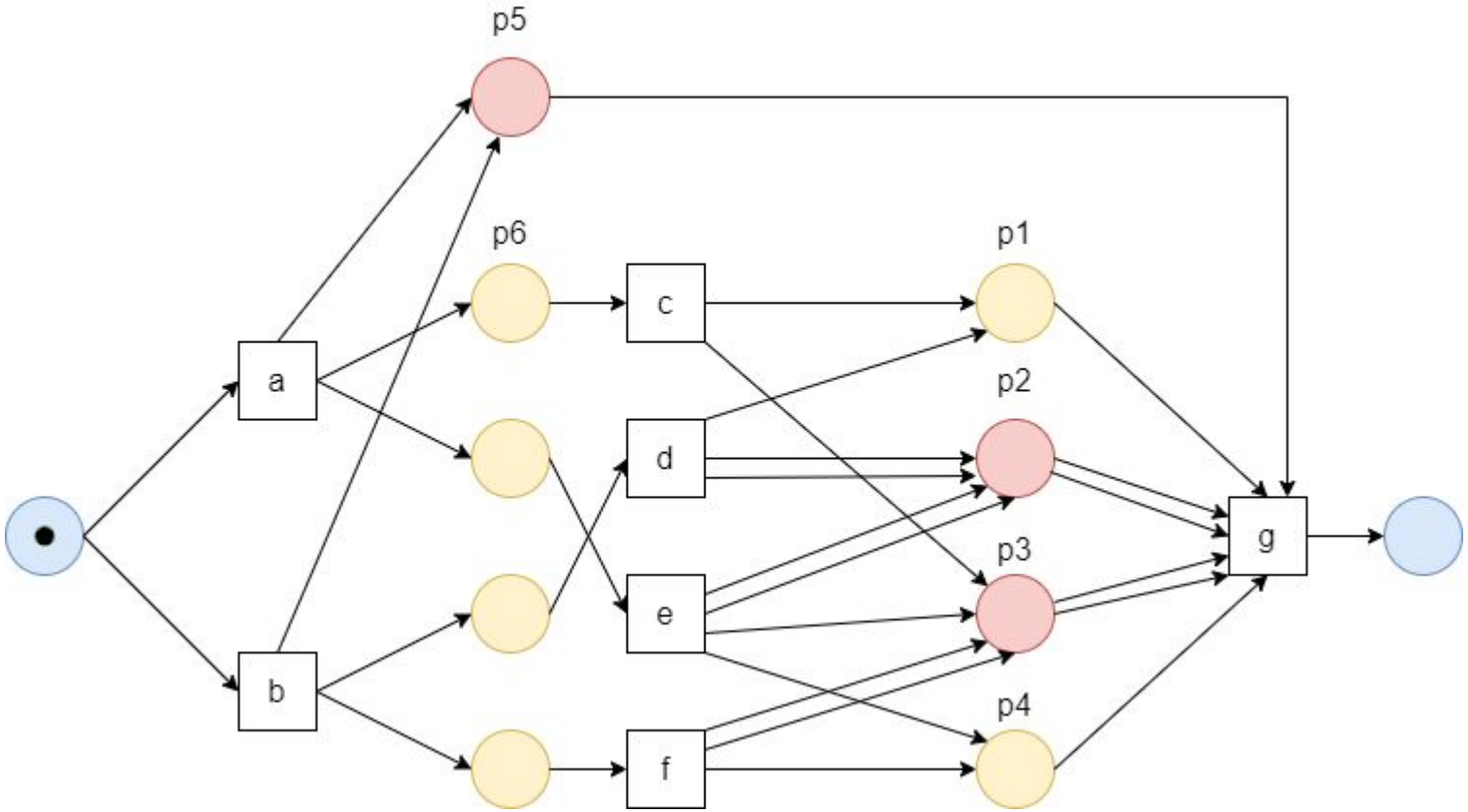
Removing Implicit Places: Extended Case

$$L = [\epsilon, \langle a, c, e, g \rangle^2, \langle a, e, c, g \rangle^3, \langle b, d, f, g \rangle^2, \langle b, f, d, g \rangle^4]$$



Removing Implicit Places: Undiscoverable implicit place

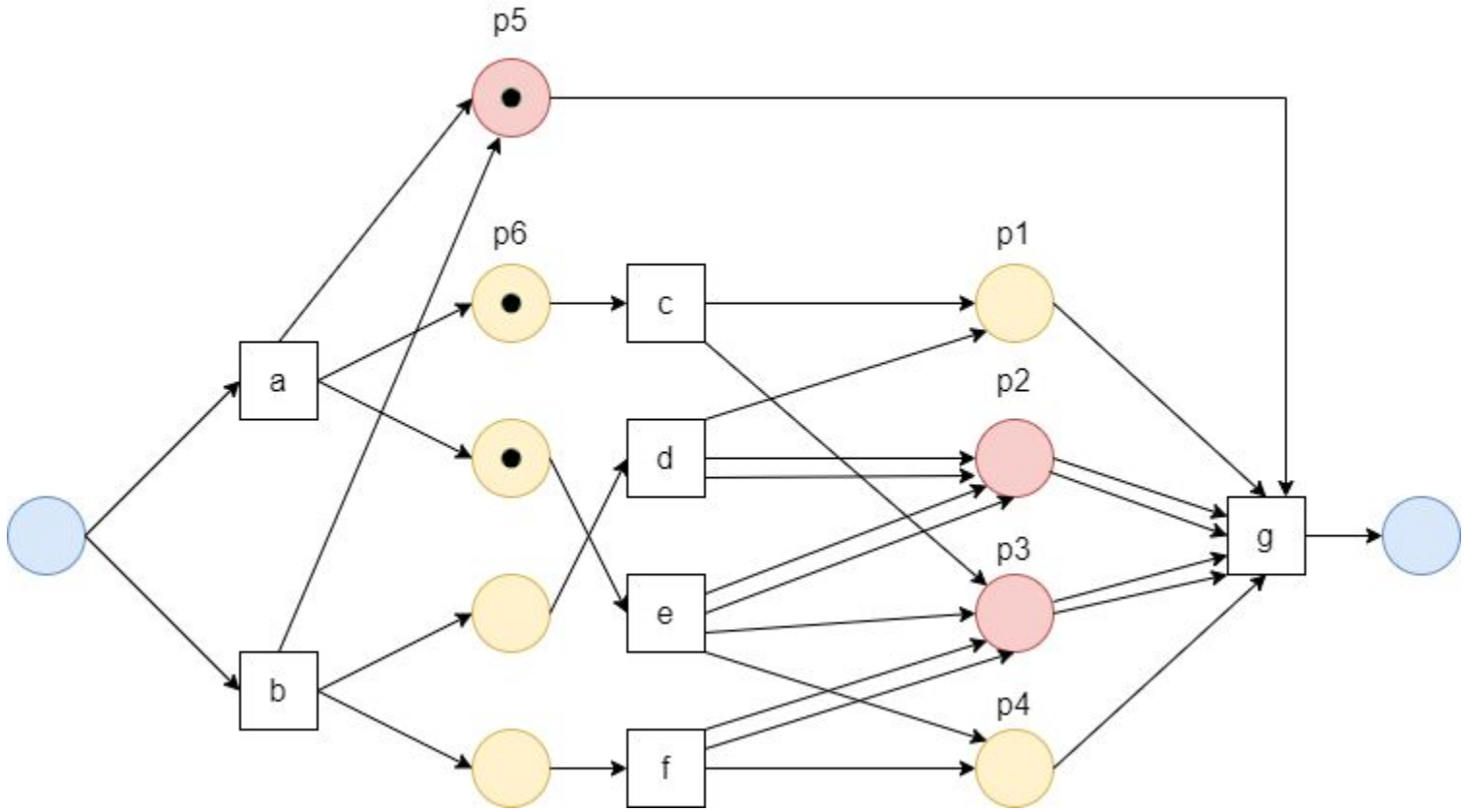
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
p_2	0				
p_1	0				
p_9					

Removing Implicit Places: Undiscoverable implicit place

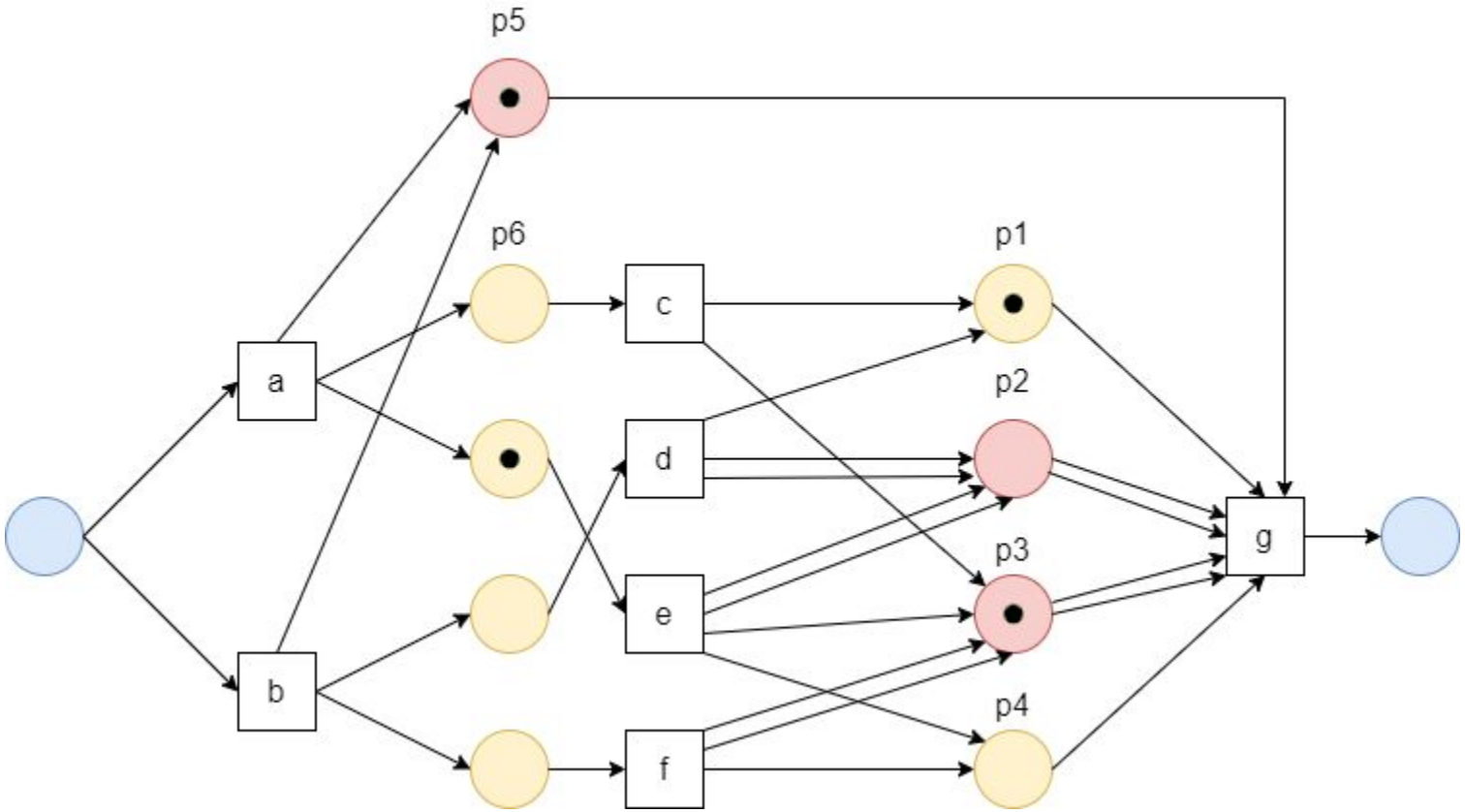
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
p_2	0	0			
p_1	0	0			
p_9					

Removing Implicit Places: Undiscoverable implicit place

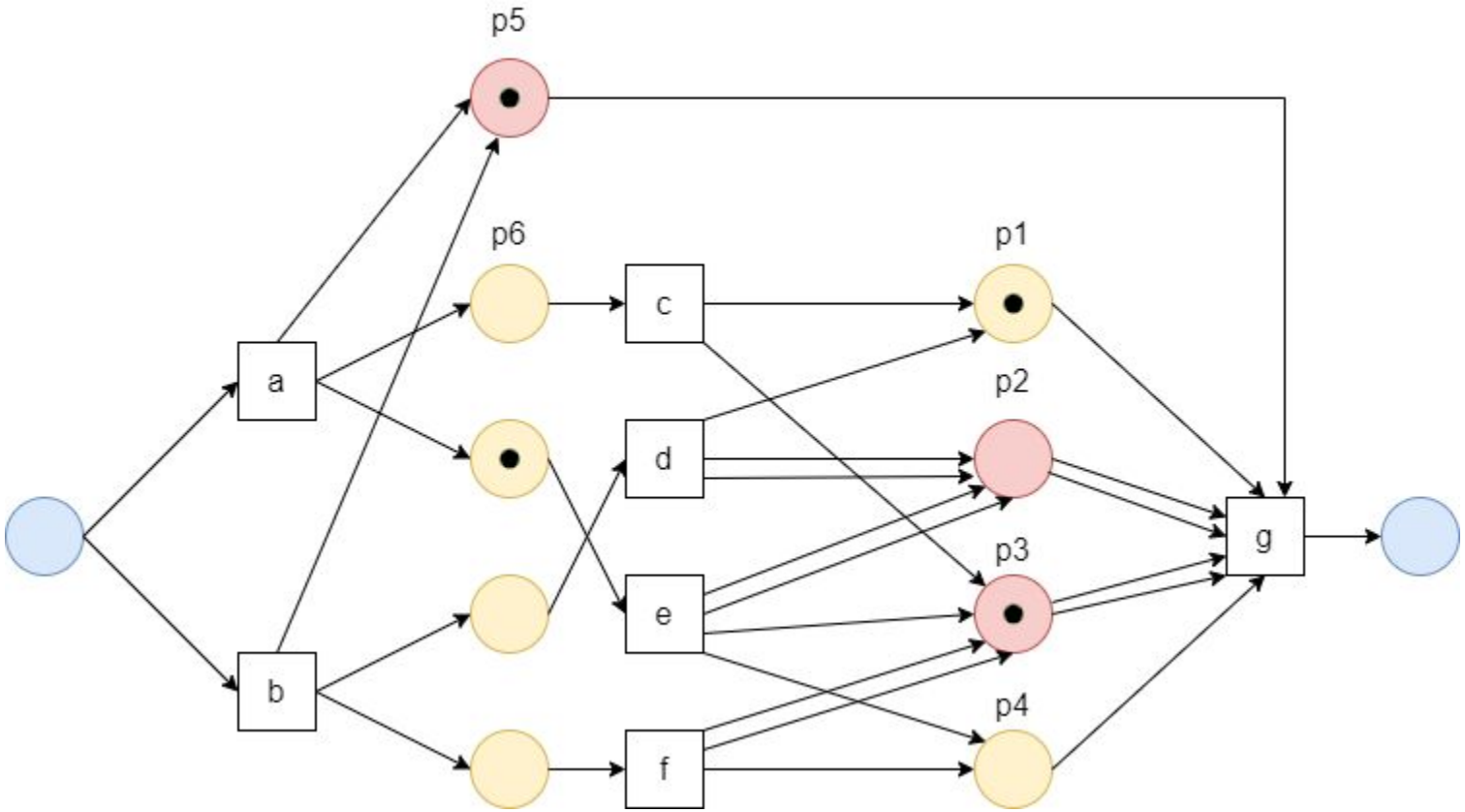
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
p_2	0	0	0		
p_1	0	0	1		
p_9	0	0	-1		

Removing Implicit Places: Undiscoverable implicit place

$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
p_2	0	0	0		
p_1	0	0	1		
p_9	0	0	-1		

$x_{p2} > x_{p1}$ **does not hold!**

Application to the eST-Miner

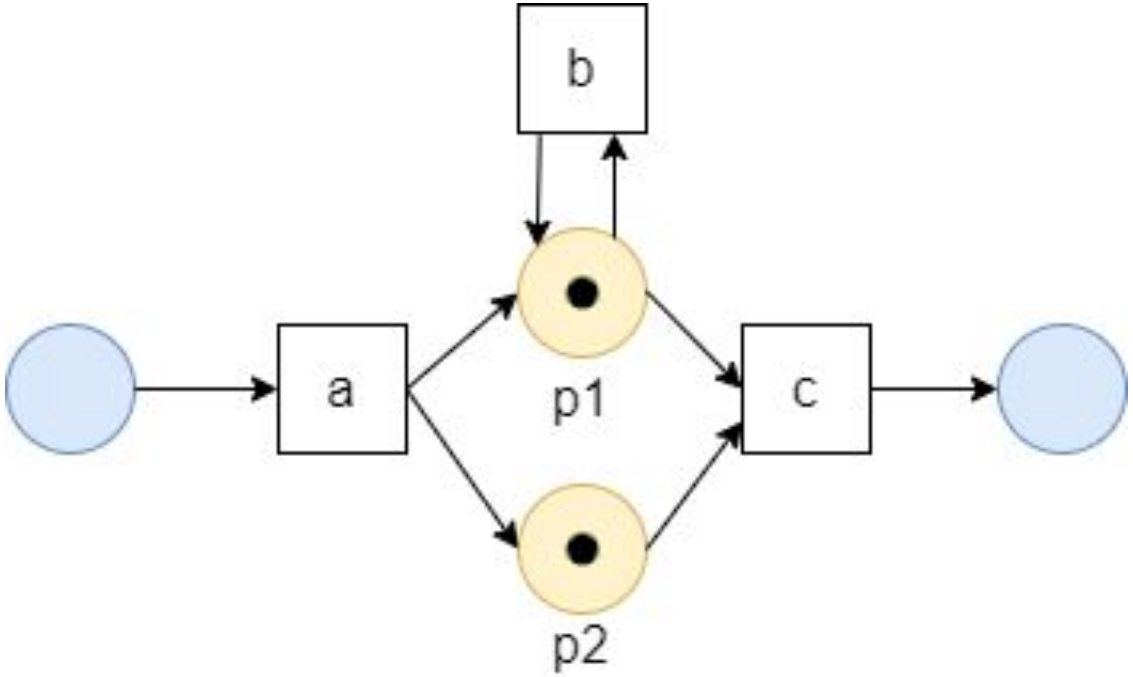
- Using the eST-Miner as a process discovery algorithm
- Candidate places are traversed and evaluated using token replay
- The discovery of a set of all *fitting* places is guaranteed
- A significant number of implicit places are also discovered
- Motivation for applying the implicit place removal technique

Application to the eST-Miner

- Benefitting from the eST-Miner as a space efficient algorithm
- Checking for the existence of places is not necessary
- Place comparisons occur very infrequently
- Final Place Removal (FPR)
- Concurrent Place Removal (CPR)

Challenges

eST-Miner allows for Petri nets with self-loops

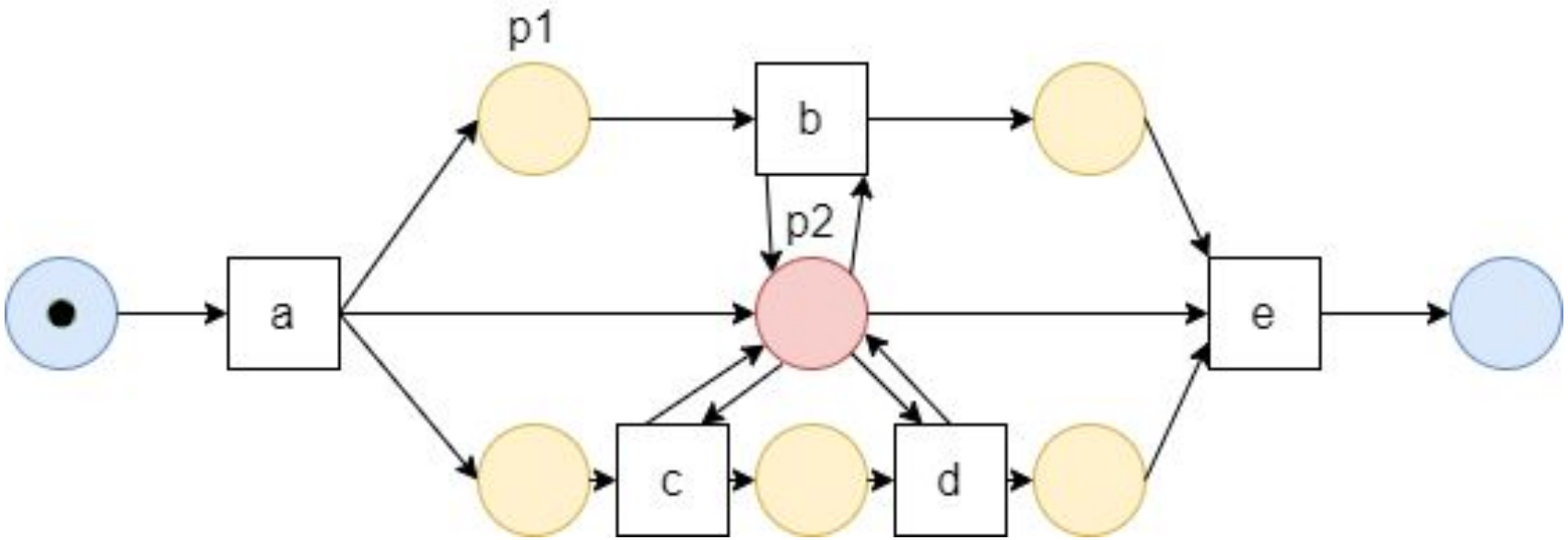


		b
p_1	1	1
p_2	1	1

		b
p_1	1	0/1
p_2	1	1/1

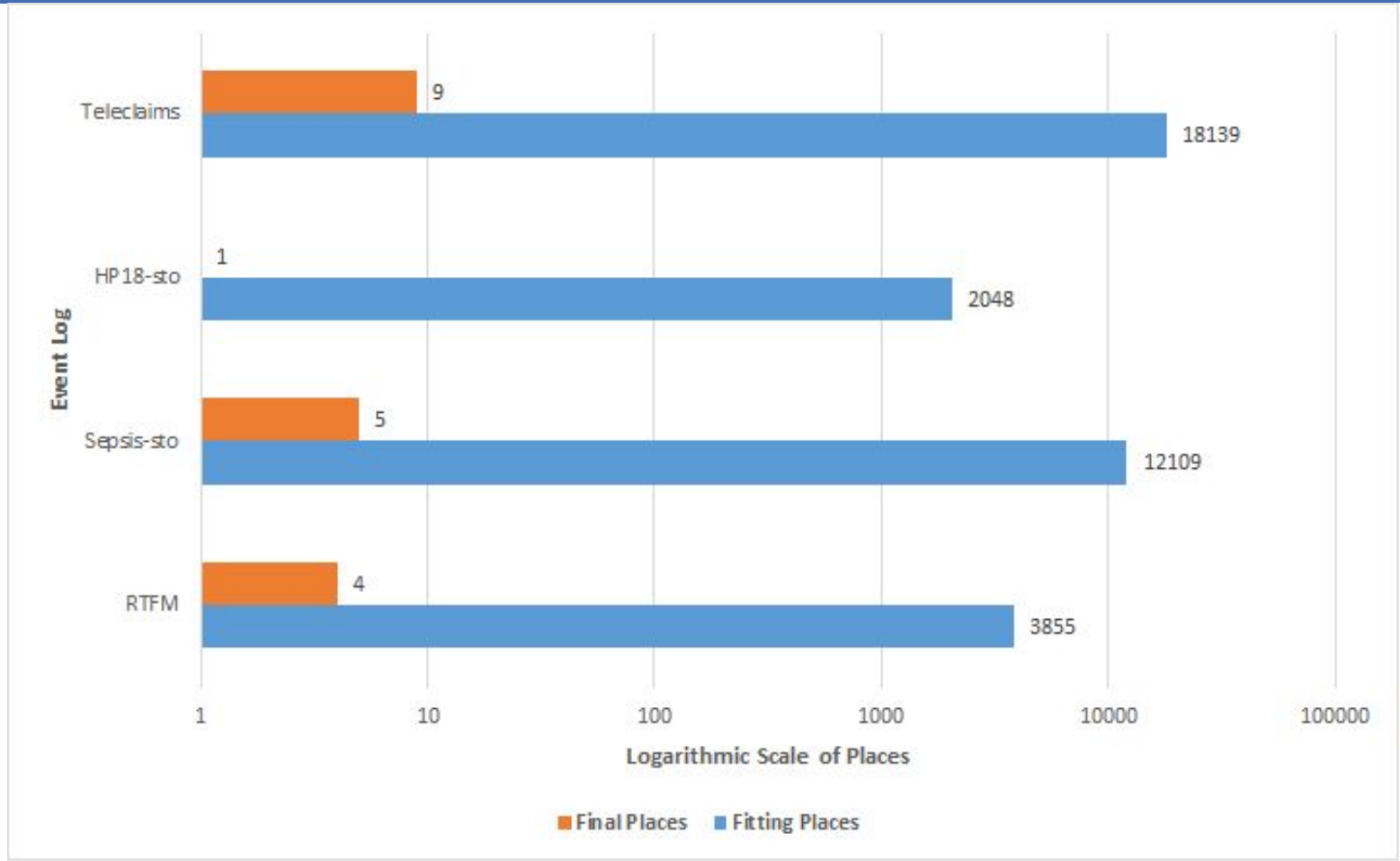
Challenges

Special case of places with self-loops in parallel constructs



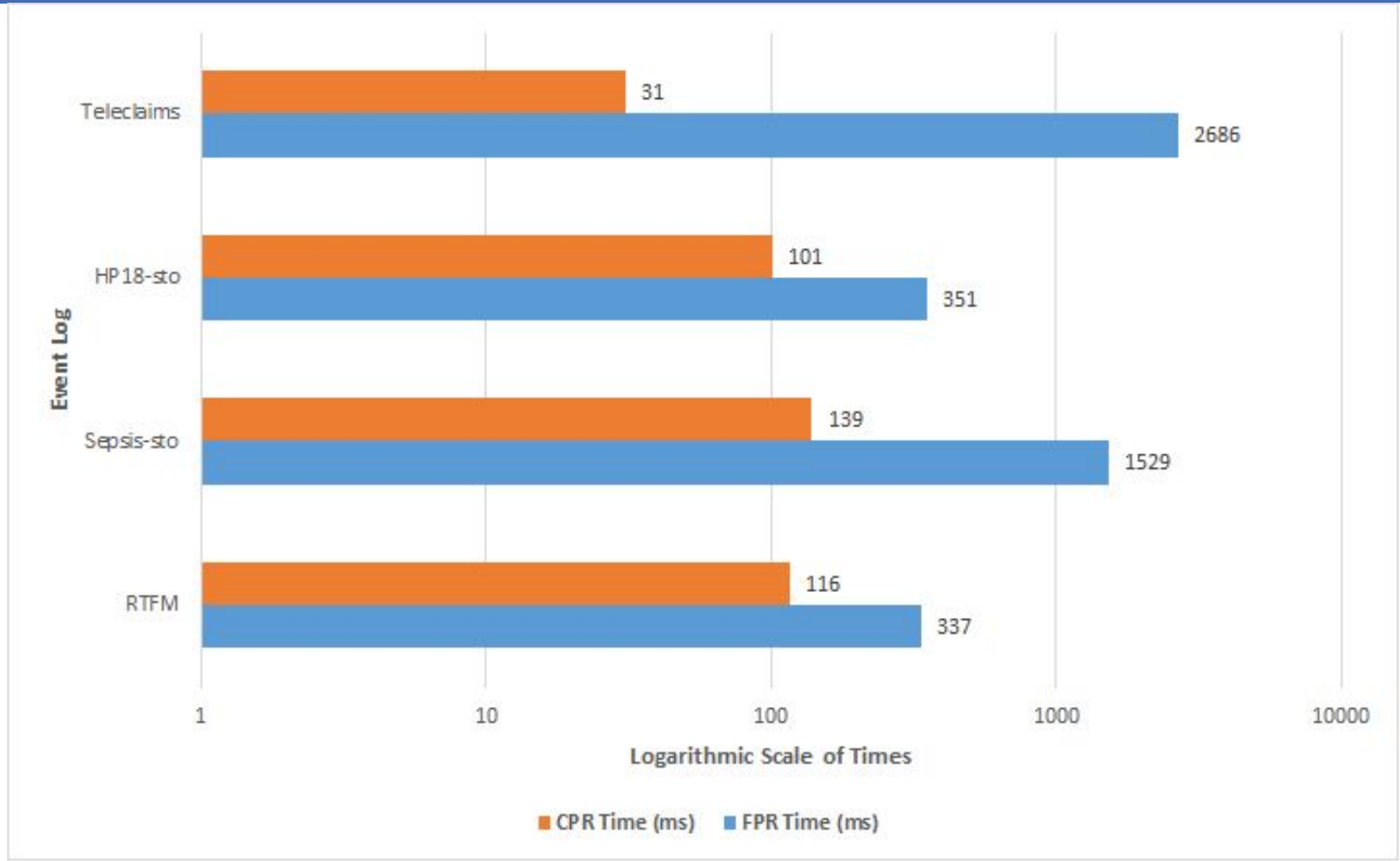
Evaluation: Space Efficiency

Remaining final places are significantly less than total fitting places



Evaluation: Time Efficiency

CPR is always significantly faster than FPR



Related Work

- [B. Berthomieu, D. Botlan, and S. Dal Zilio+ 20] Counting petri net markings from reduction equations. International Journal on Software Tools for Technology Transfer, 2020
- Complex Integer Linear Programming problem
- Developing a reduced Petri net architecture
- Highlighting the state space of a Petri net
- Post-processing step of the eST-Miner

Conclusion

- An approach to identify and remove implicit places from Petri nets
- Sequential and concurrent application schemes
- Combination with the eST-Miner process discovery algorithm
- Robust time and space efficiency of the CPR variant

Future Work

- Further investigations with the eST-Miner
  - Increasing the efficiency of the candidate traversal step
  - Returning results after a certain running time
- Choosing the order of place comparisons
- Solving the problem of self-loop places in parallel constructs
- Application to the Inductive-Miner process discovery algorithm

Thank you!

Questions?

References

[1] W. Aalst and A. Ter. Verification of workflow task structures: A petri-net-based approach. Information Systems, 25:43–69, 03 2000.
[2] W. V. Aalst, V. Rubin, B. V. Dongen, E. Kindler, and C. Gunther. Process mining : A two-step approach using transition systems and regions. 2006.
[3] R. Bergenthum, J. Desel, R. Lorenz, and S. Mauser. Process mining based on regions of languages. In G. Alonso, P. Dadam, and M. Rosemann, editors, Business Process Management, pages 375–383, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
[4] B. Berthomieu, D. Botlan, and S. Dal Zilio. Counting petri net markings from reduction equations. International Journal on Software Tools for Technology Transfer, 22, 04 2020.
[5] J. Desel and W. Reisig. The synthesis problem of petri nets. In P. Enjalbert, A. Finkel, and K. W. Wagner, editors, STACS 93, pages 120–129, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.
[6] B. Dongen, A. Medeiros, H. Verbeek, A. Weijters, and W. Aalst. The prom framework: A new era in process mining tool support. volume 3536, pages 444–454, 06 2005.
[7] F. Garcia-Valles and J. M. Colom. Implicit places in net systems. In Proceedings of the The 8th International Workshop on Petri Nets and Performance Models, PNPM '99, page 104, USA, 1999. IEEE Computer Society