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

February 5, 2021

Outline

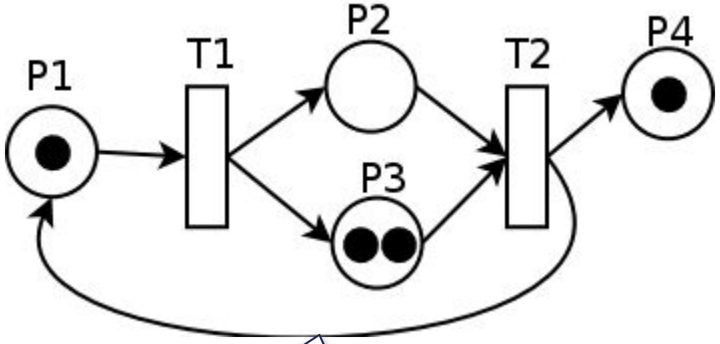
- Introduction
- Motivation
- Preliminaries
- Main Idea
- Running Example
- Limitations
- Application
- Evaluation
- Conclusion
- Future Work
- References

Introduction

Input: Event Log

Output: Process model

Process Discovery



Motivation

- Process discovery aims to discover simple Petri nets
- Petri nets may contain several *implicit* places
- Implicit places increase model complexity
- A novel technique to identify and remove implicit places
- Integrating the approach with the eST-Miner

Preliminaries

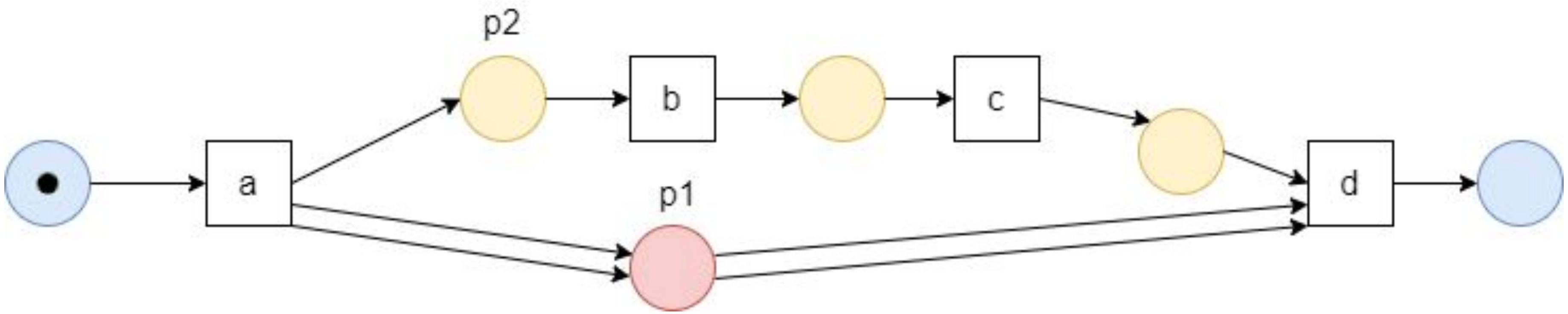
- An activity is simply *work* performed in a business process
- A trace is a sequence of activities
- An event log is a multiset of traces

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

Chair of Process and Data Science | RWTH AACHEN UNIVERSITY

Preliminaries

- A marked Petri net (P, T, F, m_0) is a process model composed of places P, transitions T and a multiset of arcs F -> $\mathbb{N}_0$
- An implicit place p does not contribute to a Petri net's behaviour



Preliminaries

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

The language L of a Petri net is every possible firing sequence in the Petri net

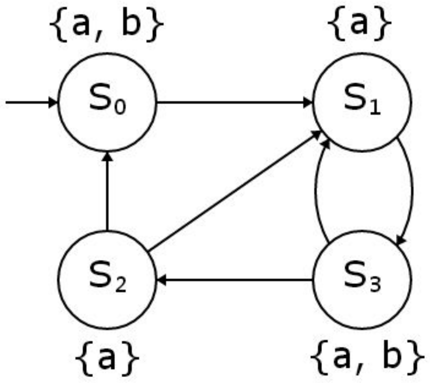
	ε	a	b	c	d
x_{p_2}	0	1	0	0	0

The token count function x_{pi} defines the number of tokens in place p_i

Preliminaries: Region Theory

- Identifying *regions* in the event log
- Each region corresponds to a feasible place in the resulting Petri net

State-based region theory makes use of a *transition system*



Language-based region theory makes use of a *prefix-closed language*

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

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:

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

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

Main Idea

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

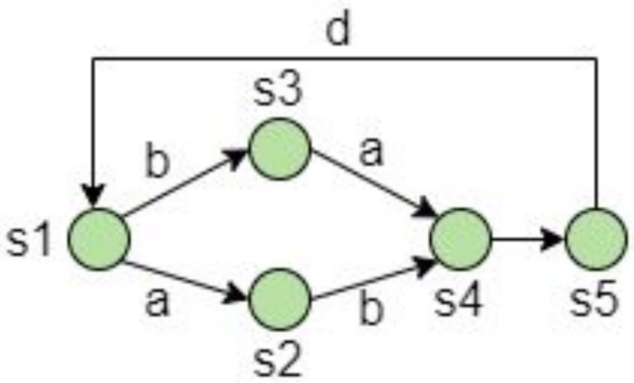
$$x_{p_3} = x_{p_1} - x_{p_2} \quad (3)$$

  is a region and p_1 can be regarded as *implicit* if p_3 exists

- ➢ Incoming arc added for increase in the token count of place p_3
- ➢ Outgoing arc added for decrease in the token count of place p_3

Main Idea

- Minimal regions method avoids adding implicit places to a model



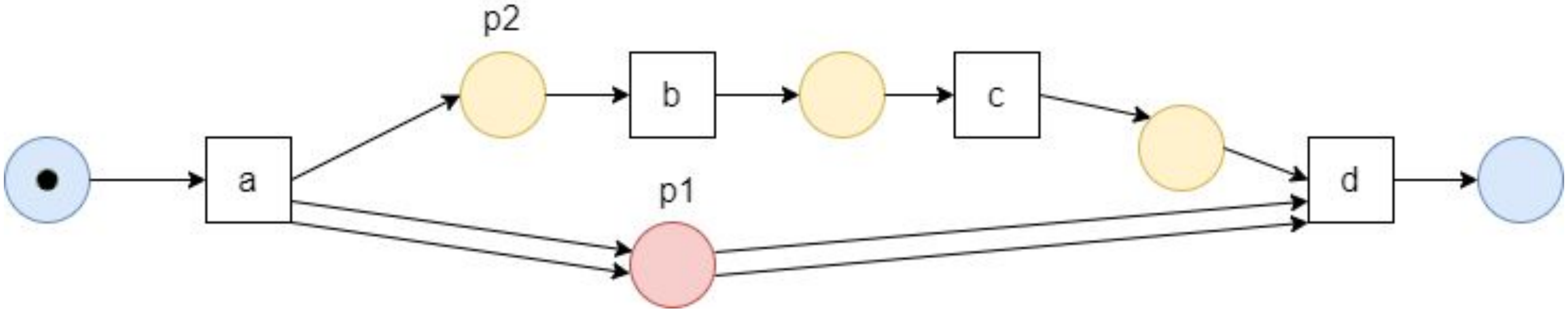
$r_1 = \{s1, s2\}$
$r_2 = \{s1, s3\}$
$r_3 = \{s2, s4\}$
$r_4 = \{s3, s4\}$
$r_5 = \{s5\}$

- Transitioning directly from the event log to the resulting model

- The goal is to remove implicit places from a discovered set of places

Running Example: Simple Case

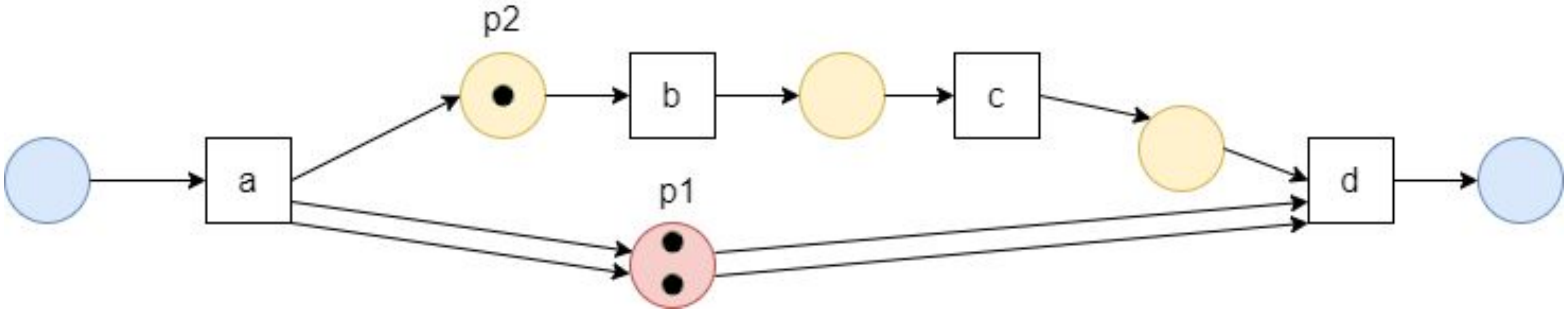
$$L = [\epsilon, \langle a, b, c, d \rangle^4]$$



	ε	a	b	c	d
x_{p_1}	0				
x_{p_2}	0				
x_{p_3}					

Running Example: Simple Case

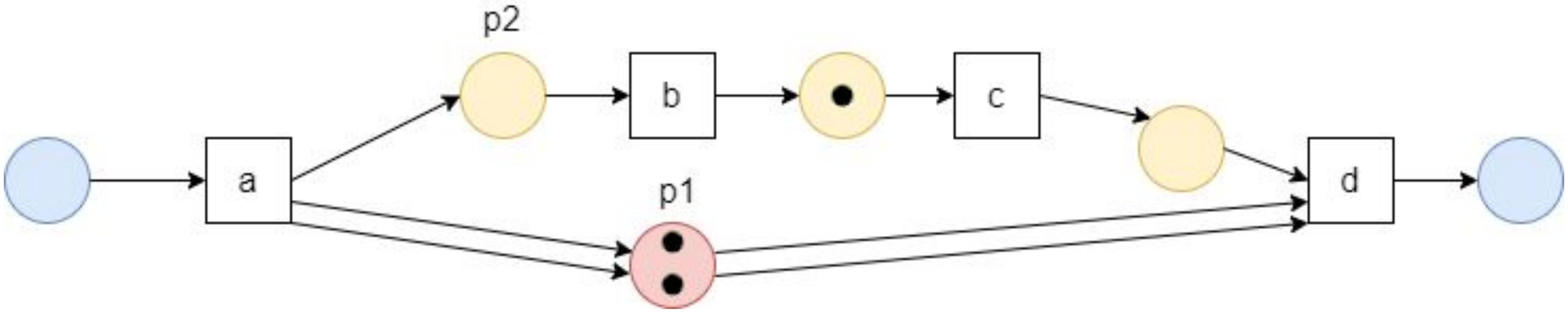
$$L = [\epsilon, \langle a, b, c, d \rangle^4]$$



	ε	a	b	c	d
x_{p_1}	0	2			
x_{p_2}	0	1			
x_{p_3}					

Running Example: Simple Case

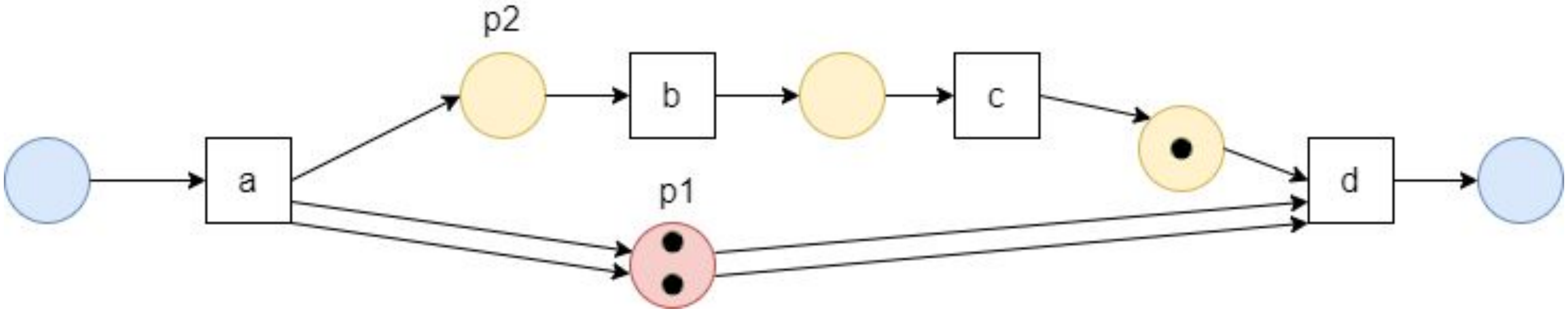
$$L = [\epsilon, \langle a, b, c, d \rangle^4]$$



	ε	a	b	c	d
x_{p_1}	0	2	2		
x_{p_2}	0	1	0		
x_{p_3}					

Running Example: Simple Case

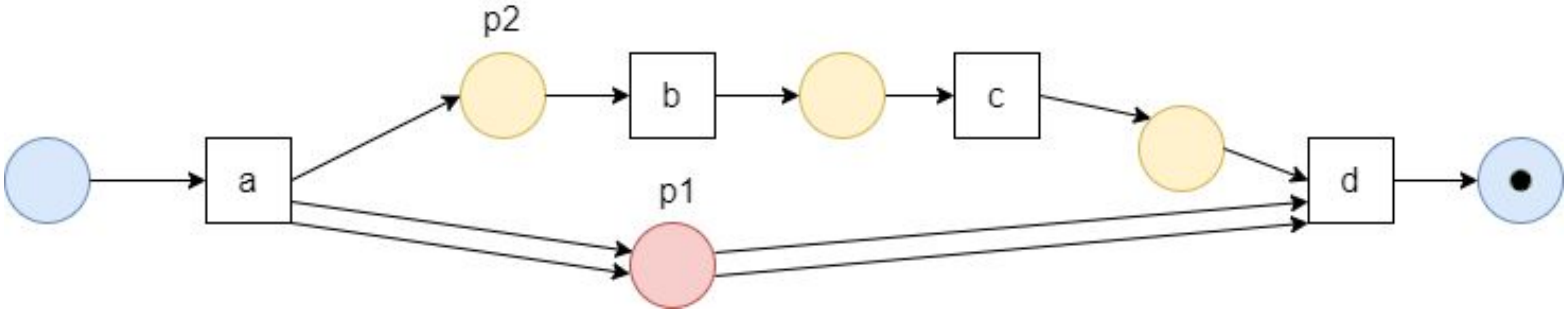
$$L = [\epsilon, \langle a, b, c, d \rangle^4]$$



	ε	a	b	c	d
x_{p_1}	0	2	2	2	
x_{p_2}	0	1	0	0	
x_{p_3}					

Running Example: Simple Case

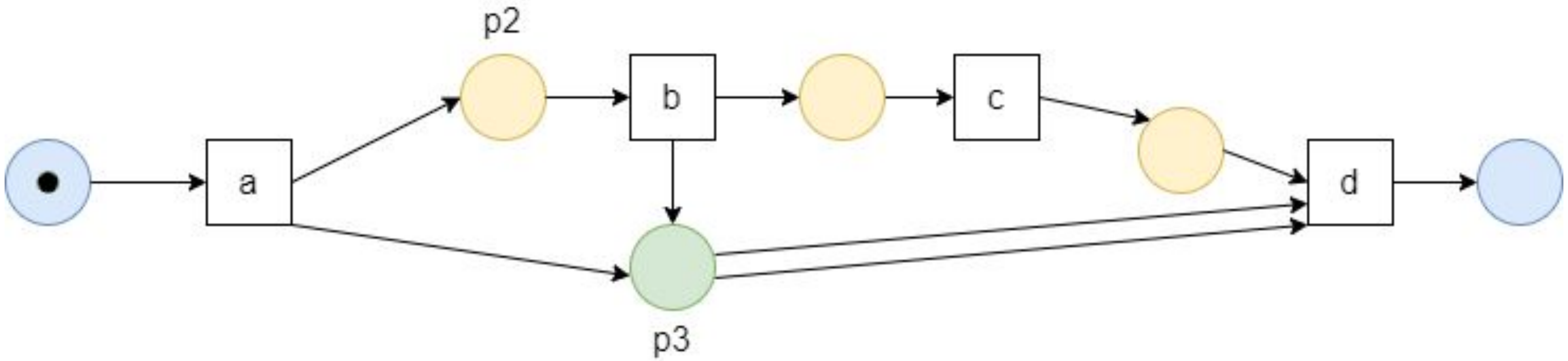
$$L = [\epsilon, \langle a, b, c, d \rangle^4]$$



	ε	a	b	c	d
x_{p_1}	0	2	2	2	0
x_{p_2}	0	1	0	0	0
x_{p_3}					

Running Example: Simple Case

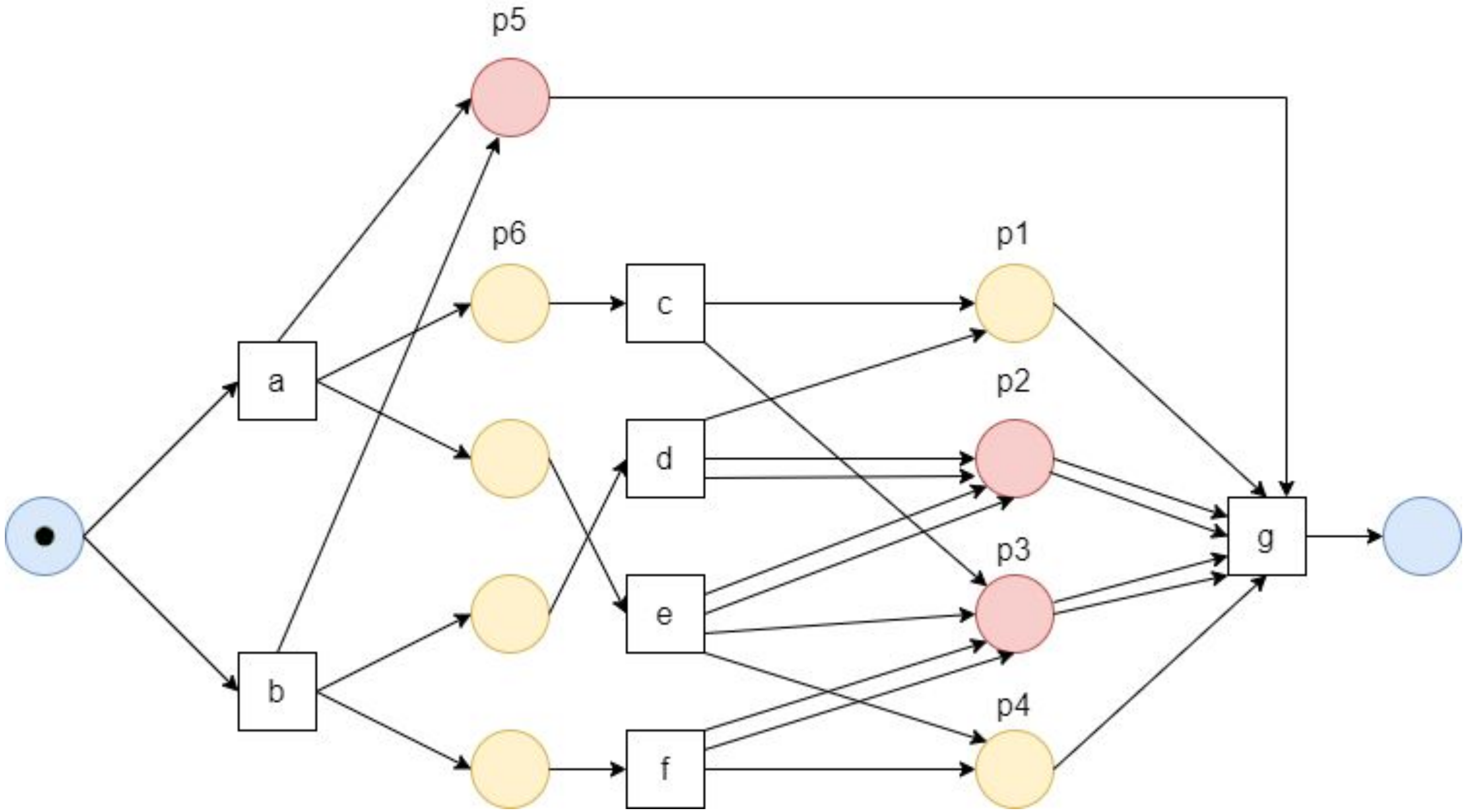
$$L = [\epsilon, \langle a, b, c, d \rangle^4]$$



	ε	a	b	c	d
x_{p_1}	0	2	2	2	0
x_{p_2}	0	1	0	0	0
x_{p_3}	0	1	2	2	0

Running Example: Intermediate Case

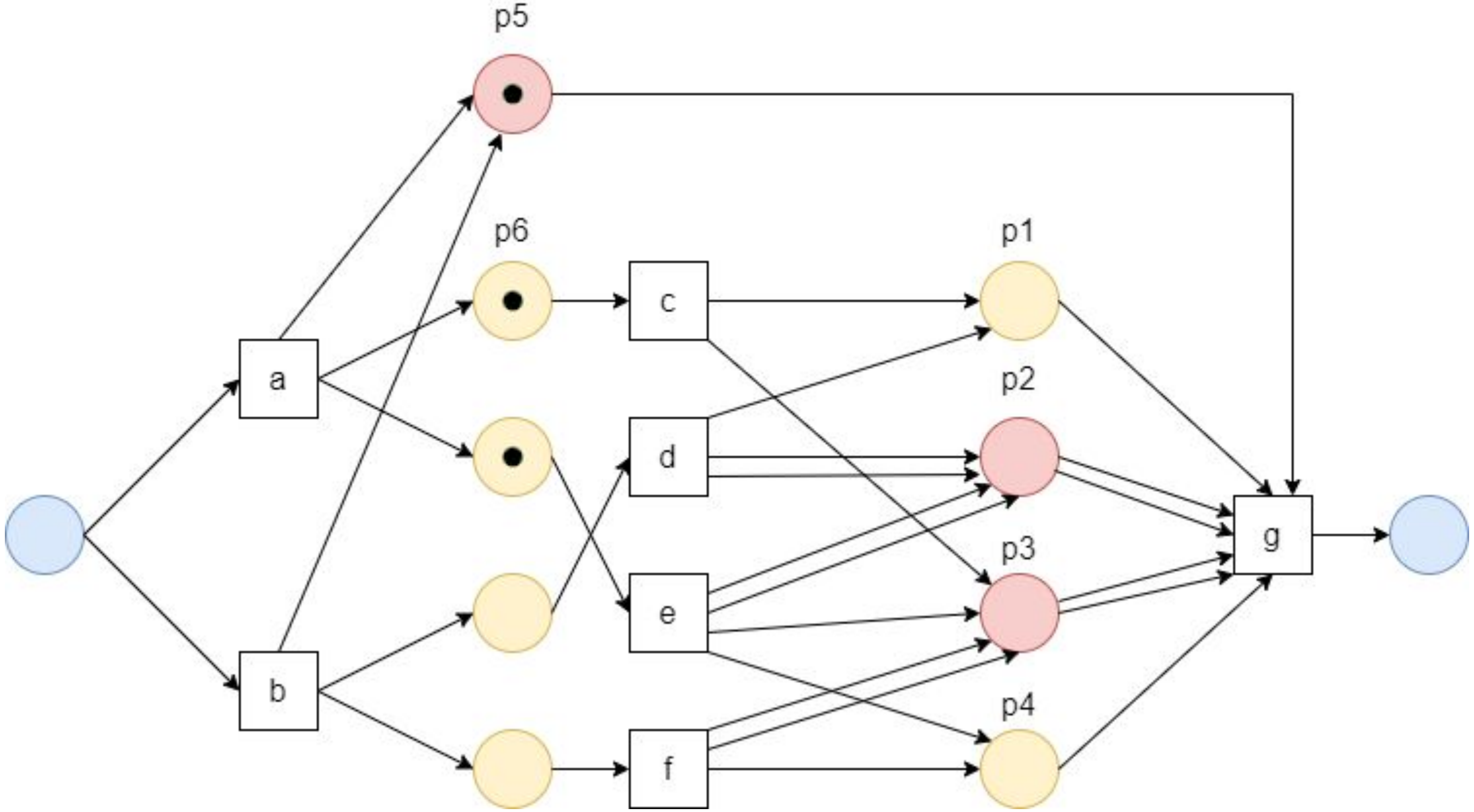
$$L = [\epsilon, \langle a,c,e,g\rangle^2, \langle a,e,c,g\rangle^3, \langle b,d,f,g\rangle^2, \langle b,f,d,g\rangle^4]$$



	ε	a	c	e	g
x_{p5}	0				
x_{p6}	0				
x_{p7}					

Running Example: Intermediate Case

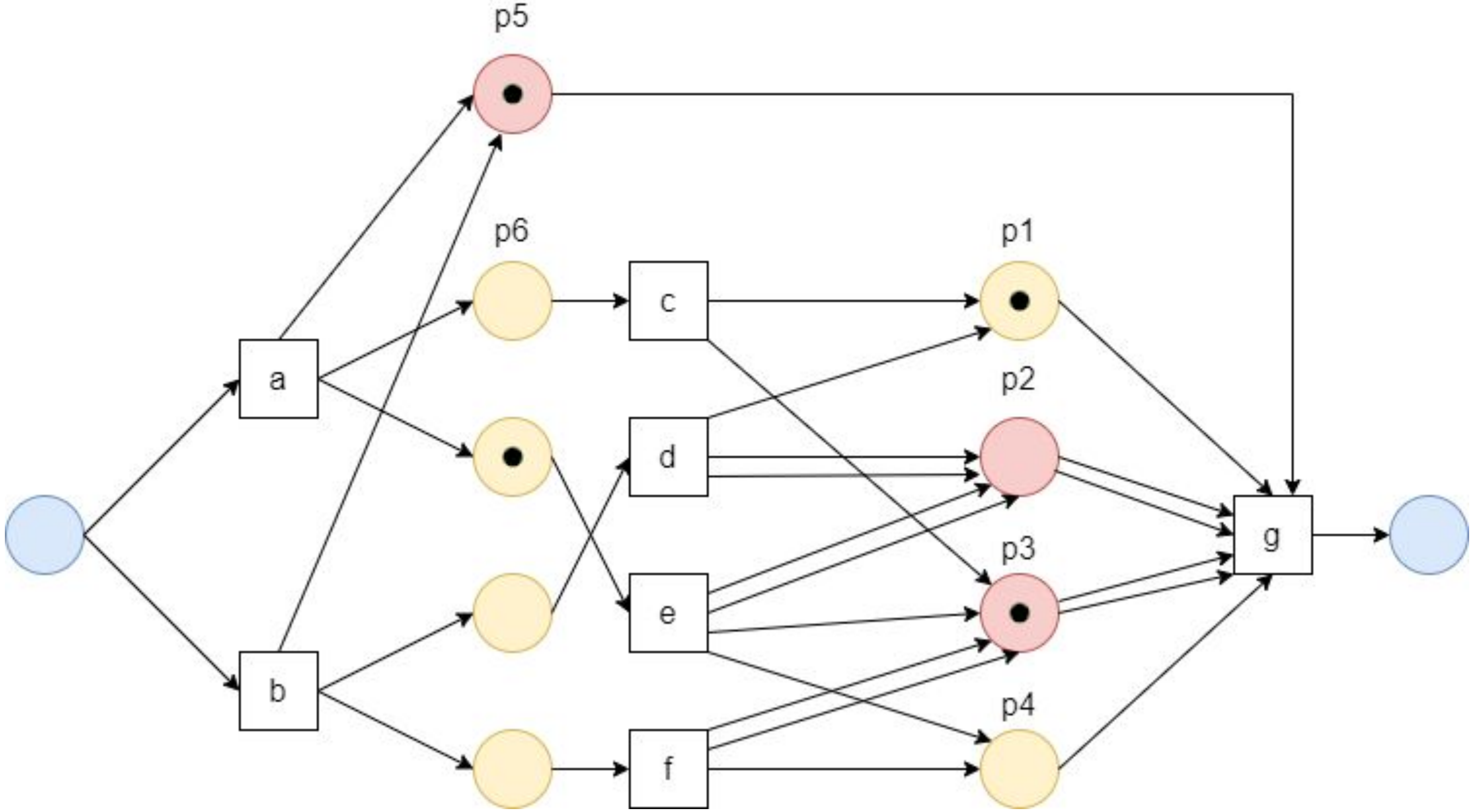
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
x_{p5}	0	1			
x_{p6}	0	1			
x_{p7}					

Running Example: Intermediate Case

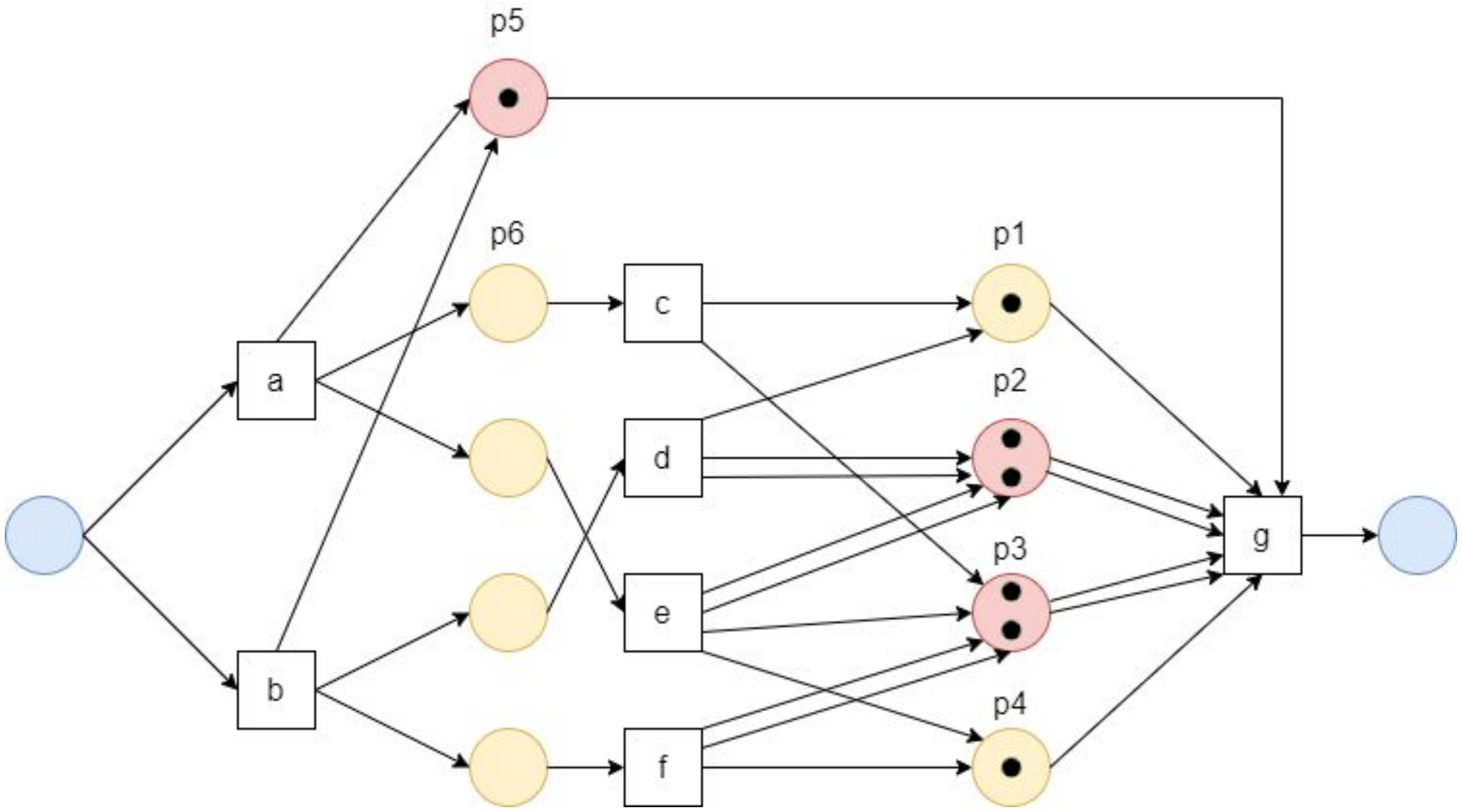
$$L = [\epsilon, \langle a,c,e,g\rangle^2, \langle a,e,c,g\rangle^3, \langle b,d,f,g\rangle^2, \langle b,f,d,g\rangle^4]$$



	ε	a	c	e	g
x_{p_5}	0	1	1		
x_{p_6}	0	1	0		
x_{p_7}					

Running Example: Intermediate Case

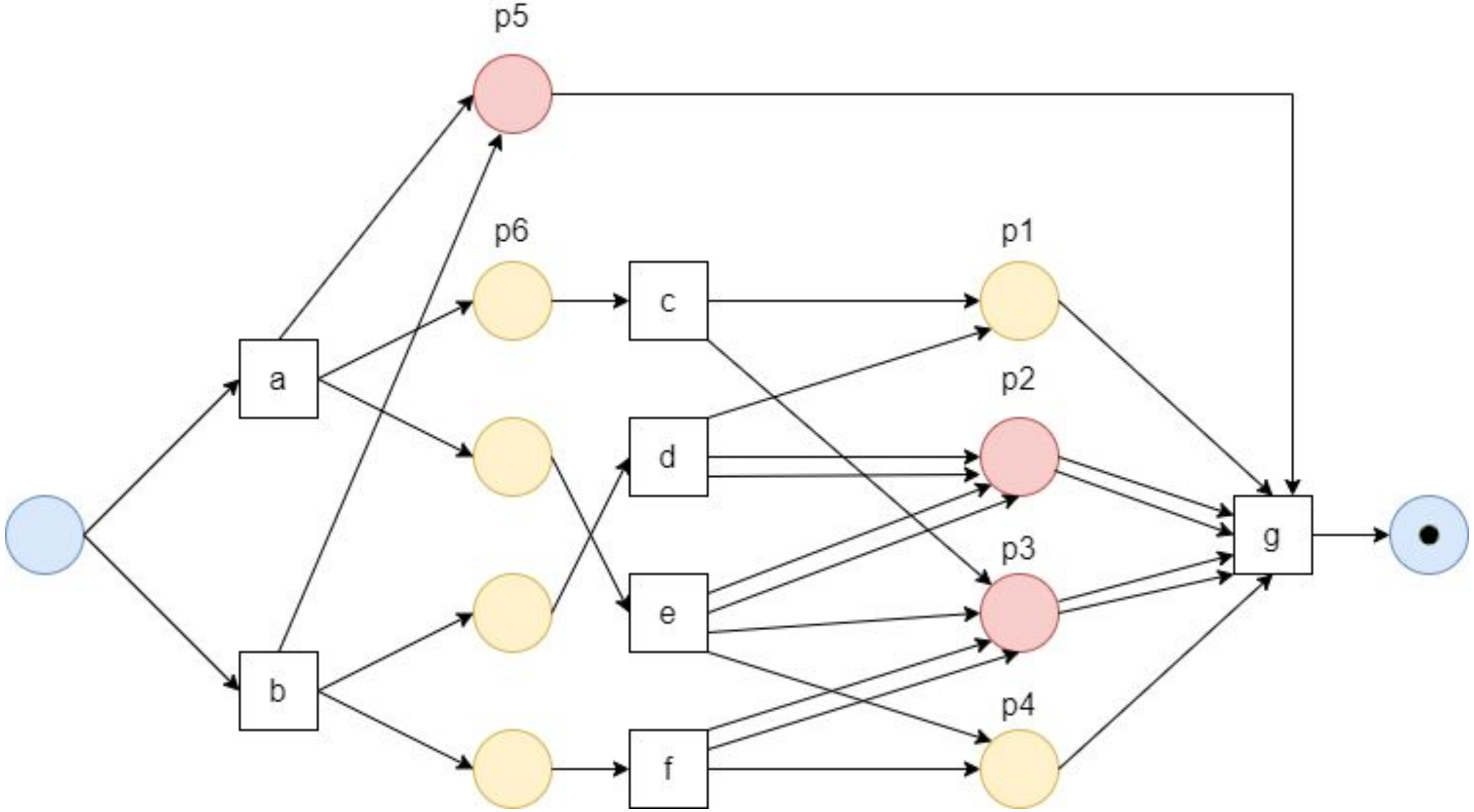
$$L = [\epsilon, \langle a,c,e,g\rangle^2, \langle a,e,c,g\rangle^3, \langle b,d,f,g\rangle^2, \langle b,f,d,g\rangle^4]$$



	ε	**a**	**c**	**e**	**g**
x_{p5}	0	1	1	1	
x_{p6}	0	1	0	0	
x_{p7}					

Running Example: Intermediate Case

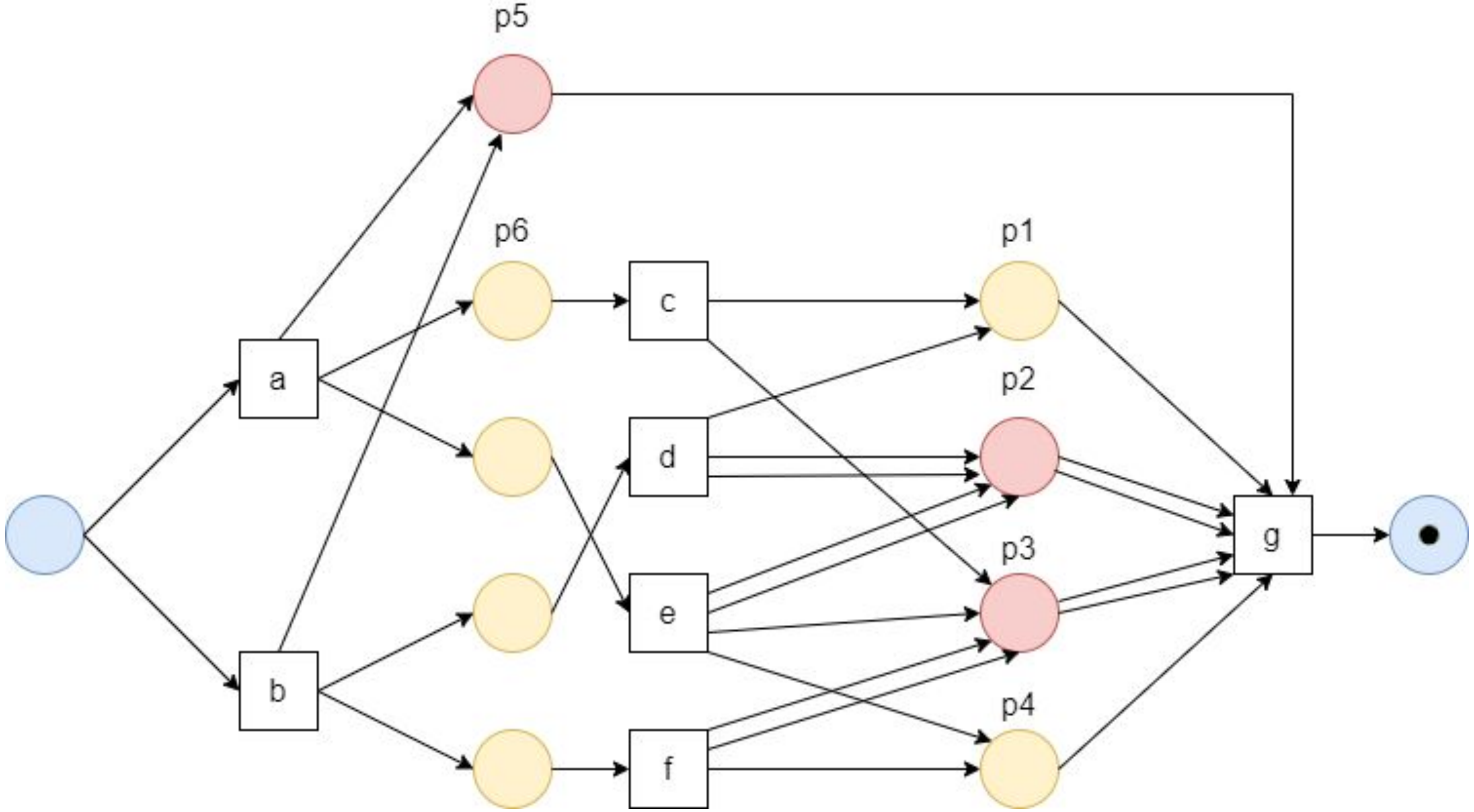
$$L = [\epsilon, \langle a,c,e,g\rangle^2, \langle a,e,c,g\rangle^3, \langle b,d,f,g\rangle^2, \langle b,f,d,g\rangle^4]$$



	ε	**a**	**c**	**e**	**g**
x_{p5}	0	1	1	1	0
x_{p6}	0	1	0	0	0
x_{p7}					

Running Example: Intermediate Case

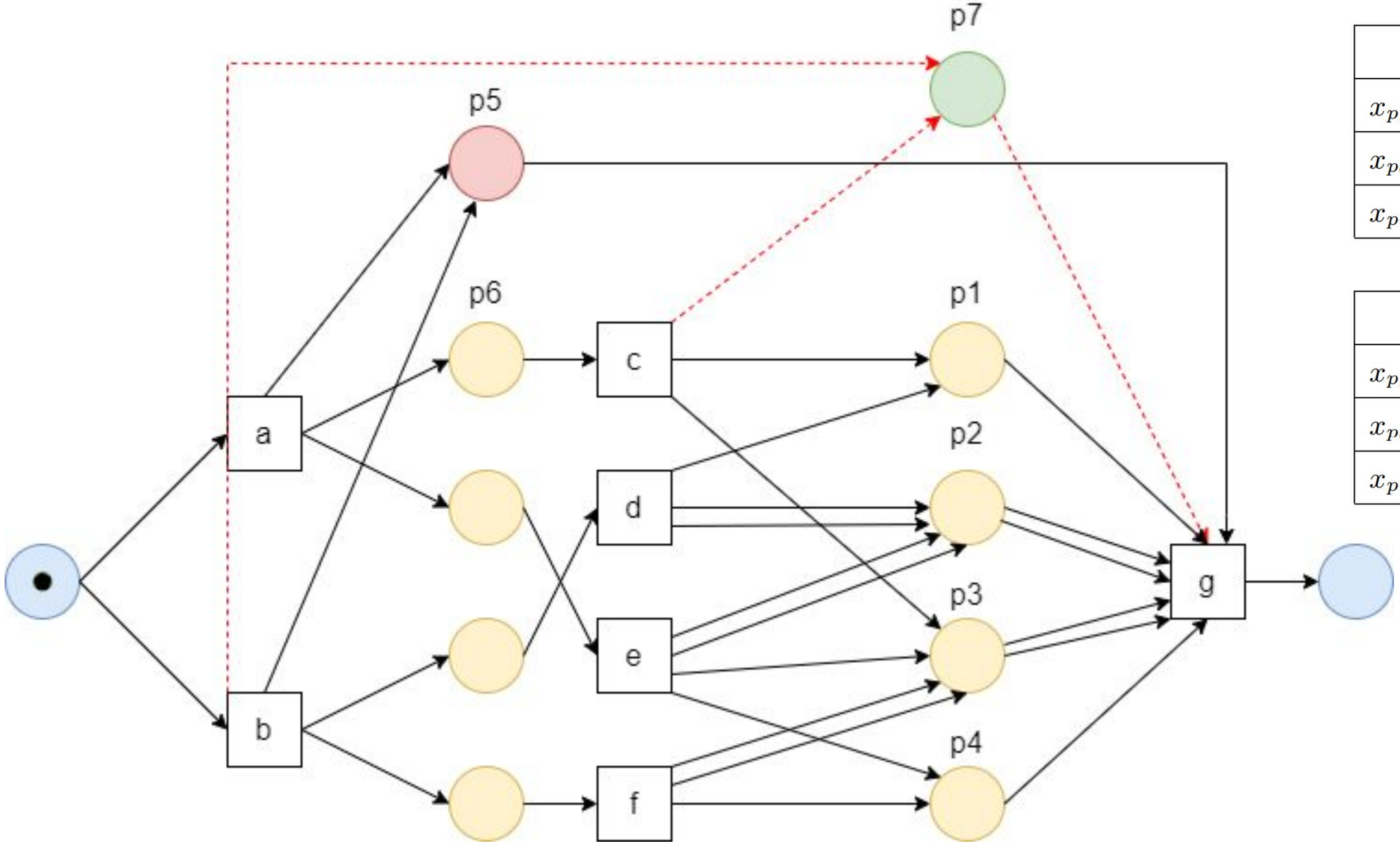
$$L = [\epsilon, \langle a,c,e,g\rangle^2, \langle a,e,c,g\rangle^3, \langle b,d,f,g\rangle^2, \langle b,f,d,g\rangle^4]$$



	ε	a	c	e	g
x_{p5}	0	1	1	1	0
x_{p6}	0	1	0	0	0
x_{p7}	0	0	1	1	0

Running Example: Intermediate 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
x_{p5}	0	1	1	1	0
x_{p6}	0	1	0	0	0
x_{p7}	0	0	1	1	0

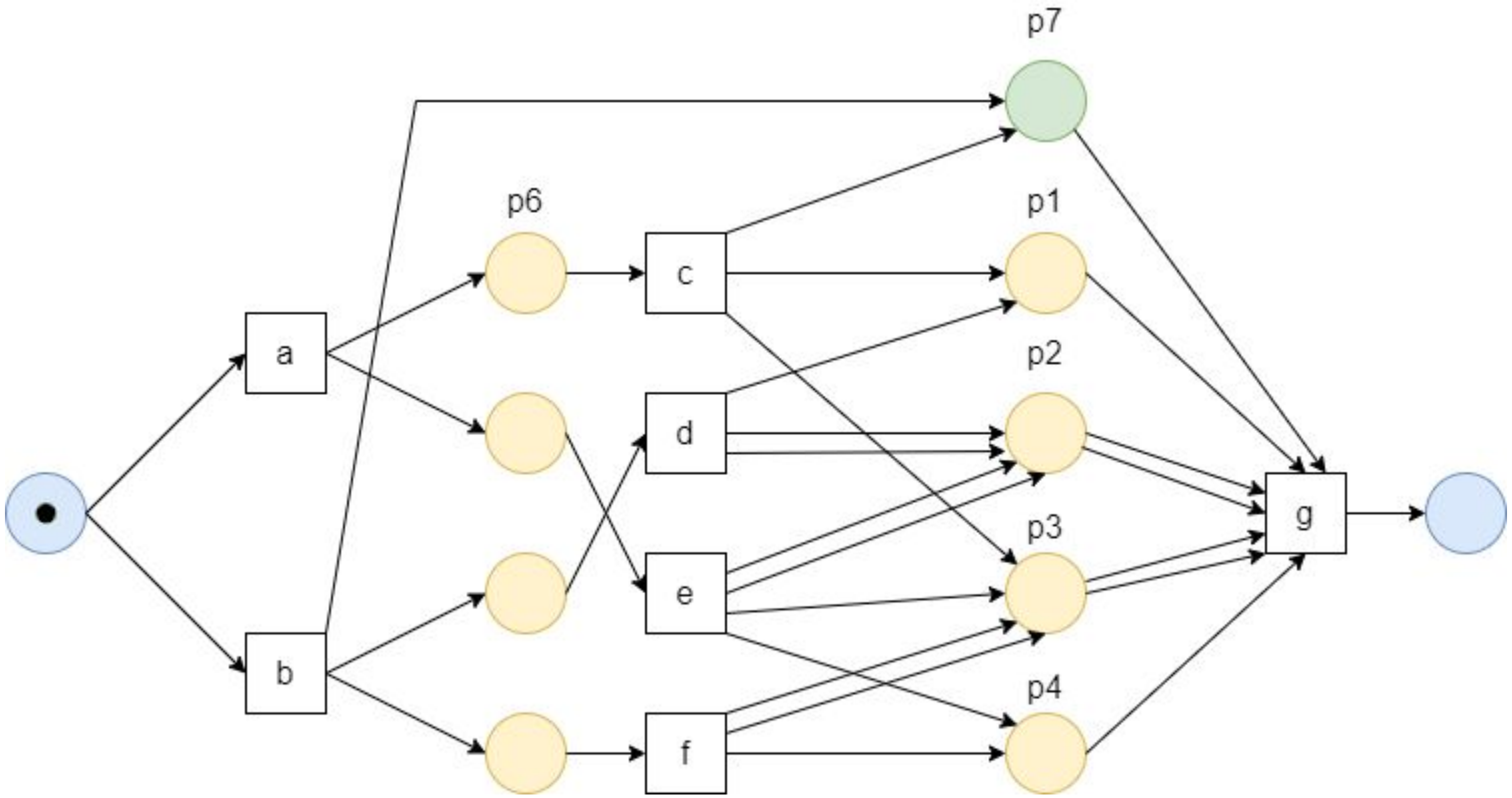
	ε	a	e	c	g
x_{p5}	0	1	1	1	0
x_{p6}	0	1	1	0	0
x_{p7}	0	0	0	1	0

	ε	b	d	f	g
x_{p5}	0	1	1	1	0
x_{p6}	0	0	0	0	0
x_{p7}	0	1	1	1	0

	ε	b	f	d	g
x_{p5}	0	1	1	1	0
x_{p6}	0	0	0	0	0
x_{p7}	0	1	1	1	0

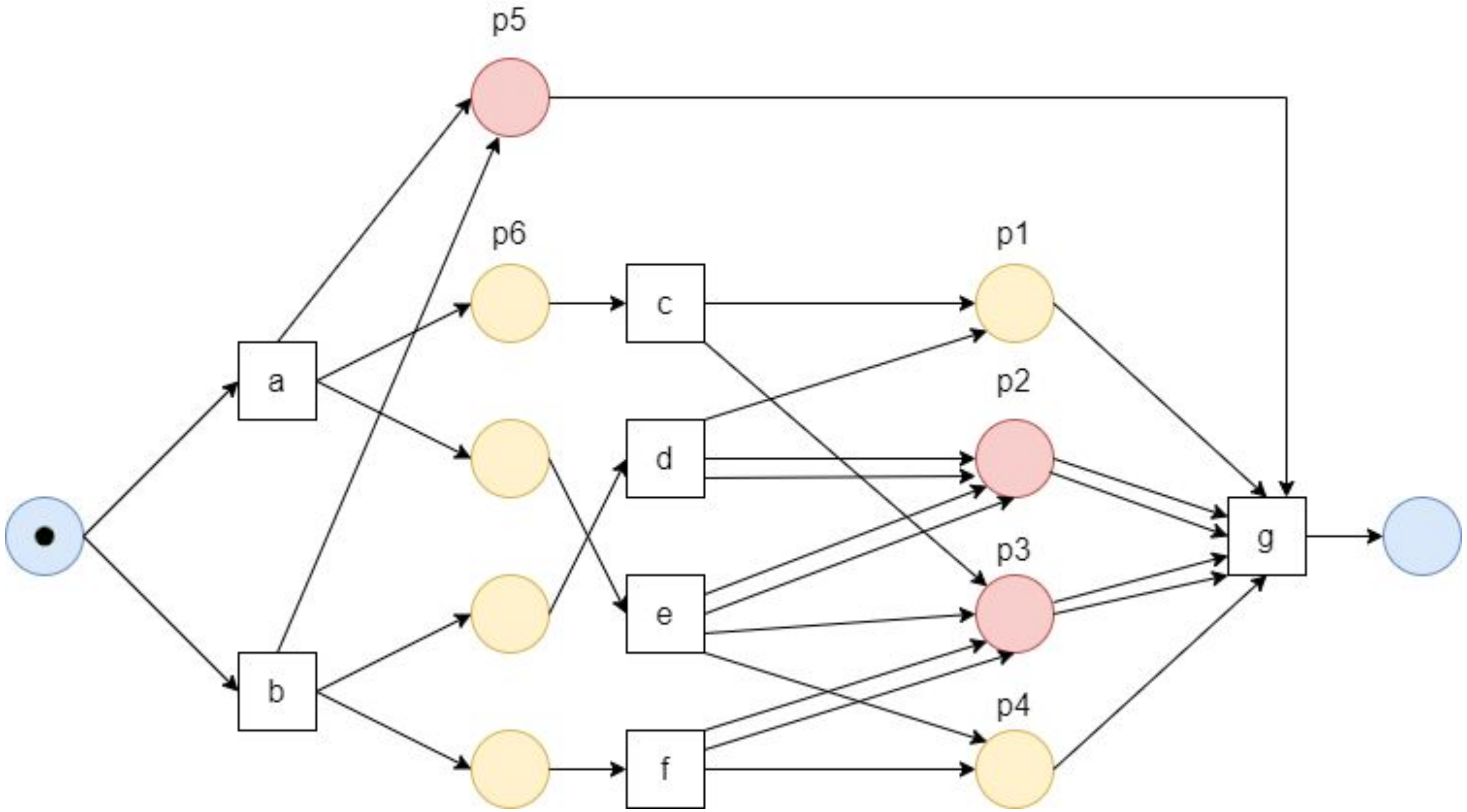
Running Example: Intermediate Case

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



Running Example: Extended Case

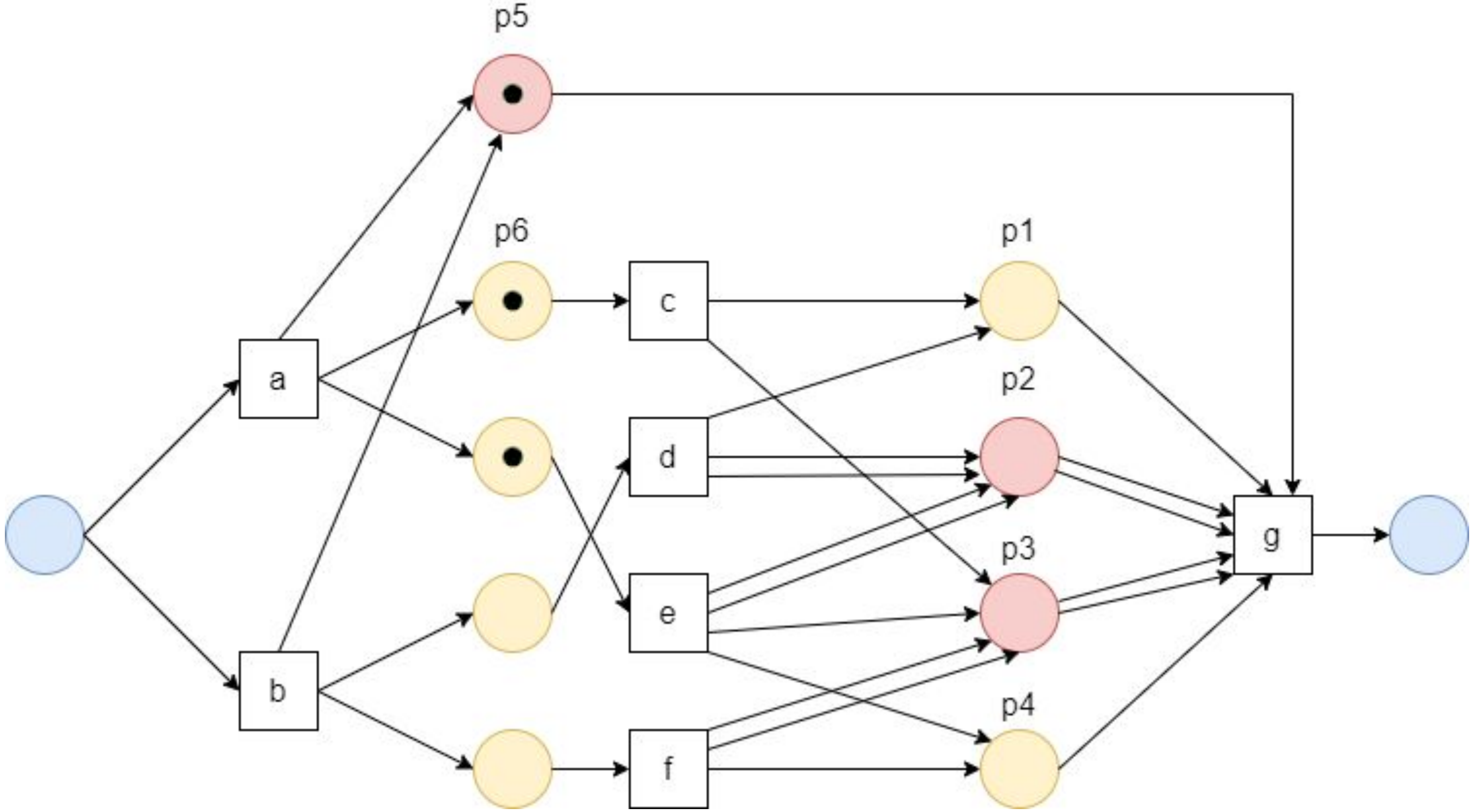
$$L = [\epsilon, \langle a,c,e,g\rangle^2, \langle a,e,c,g\rangle^3, \langle b,d,f,g\rangle^2, \langle b,f,d,g\rangle^4]$$



	ε	a	c	e	g
x_{p_3}	0				
x_{p_4}	0				
x_{p_8}					

Running Example: Extended Case

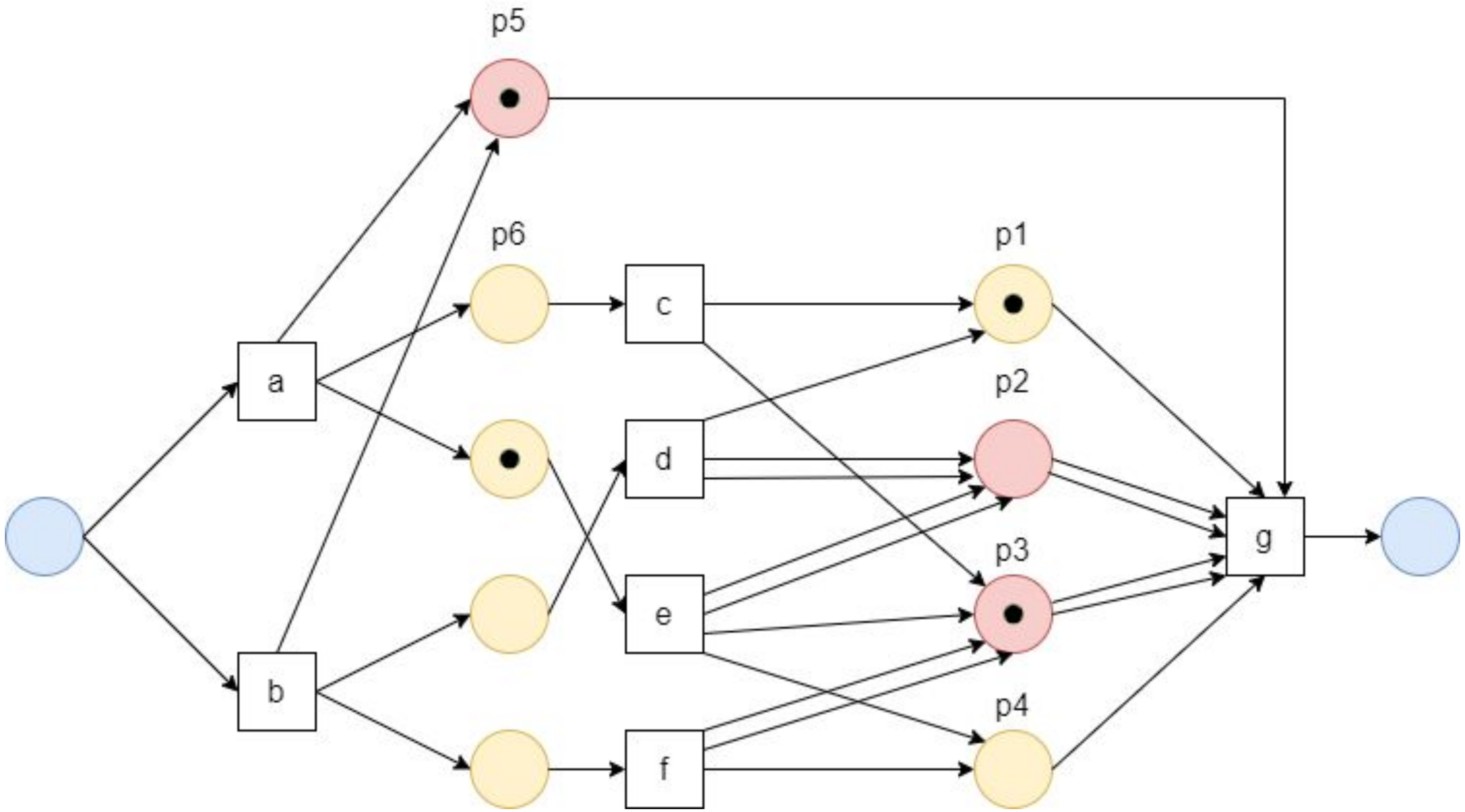
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
x_{p3}	0	0			
x_{p4}	0	0			
x_{p8}					

Running Example: Extended Case

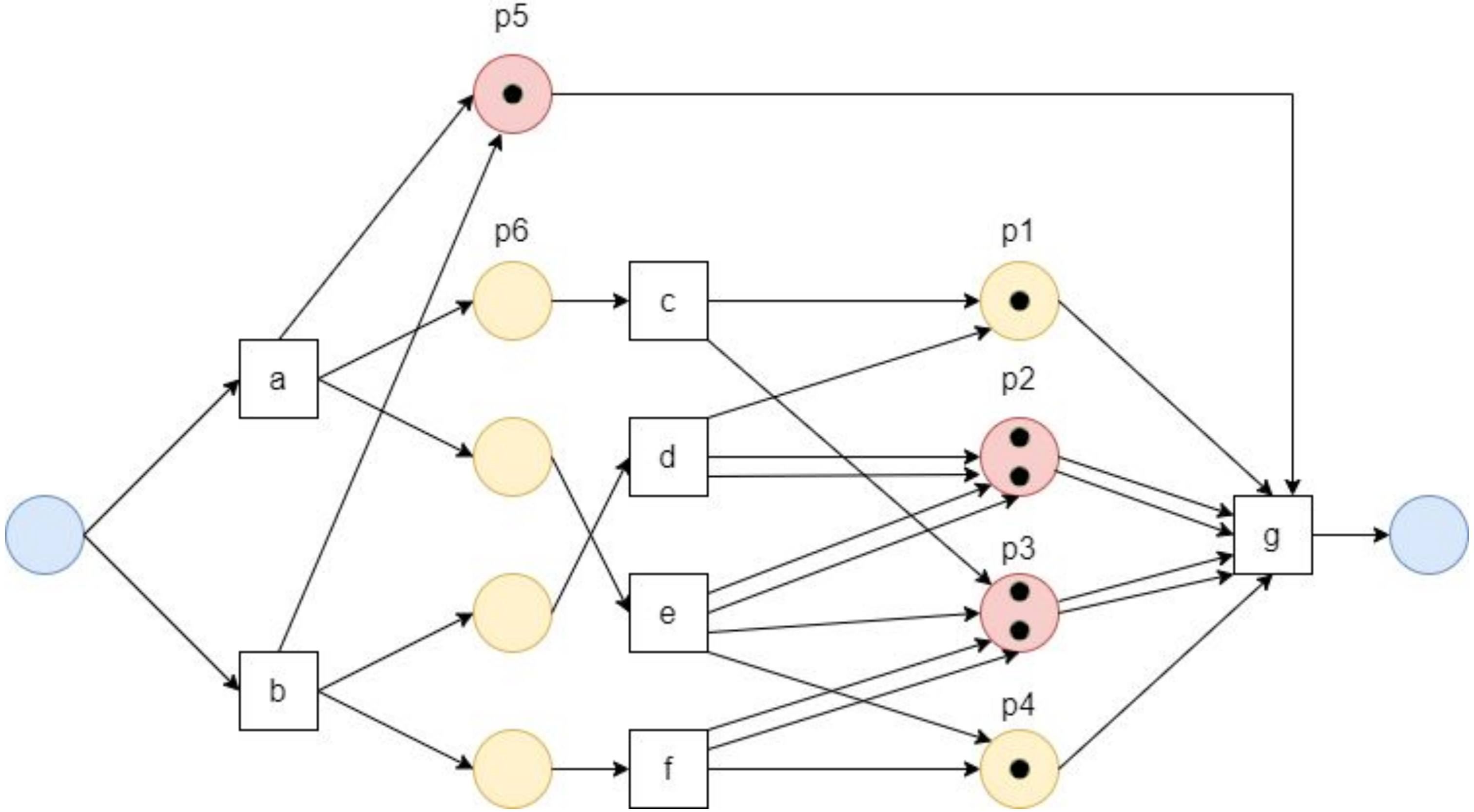
$$L = [\epsilon, \langle a,c,e,g\rangle^2, \langle a,e,c,g\rangle^3, \langle b,d,f,g\rangle^2, \langle b,f,d,g\rangle^4]$$



	ε	a	c	e	g
x_{p3}	0	0	1		
x_{p4}	0	0	0		
x_{p8}					

Running Example: Extended Case

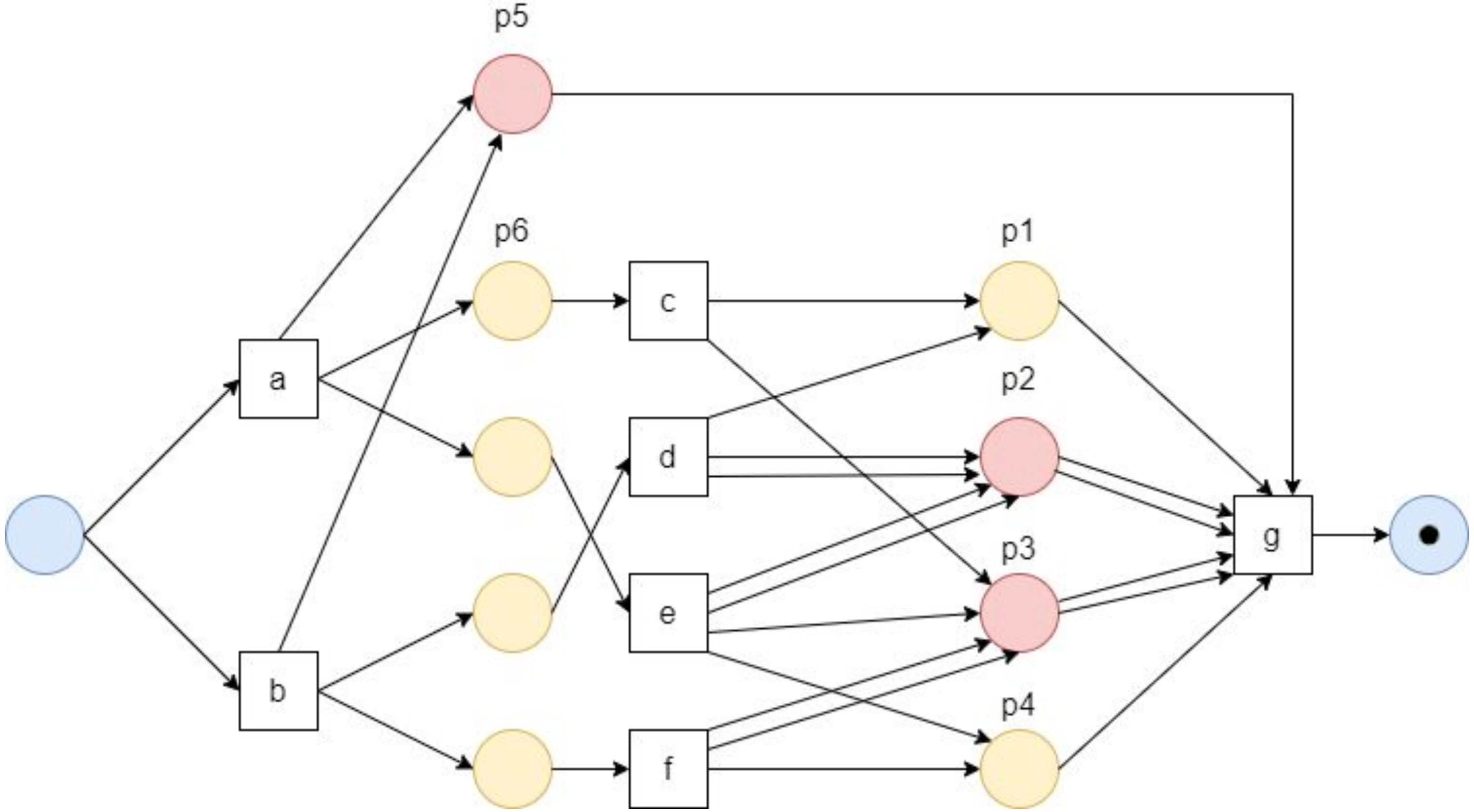
$$L = [\epsilon, \langle a,c,e,g\rangle^2, \langle a,e,c,g\rangle^3, \langle b,d,f,g\rangle^2, \langle b,f,d,g\rangle^4]$$



	ε	a	c	e	g
x_{p3}	0	0	1	2	
x_{p4}	0	0	0	1	
x_{p8}					

Running Example: Extended Case

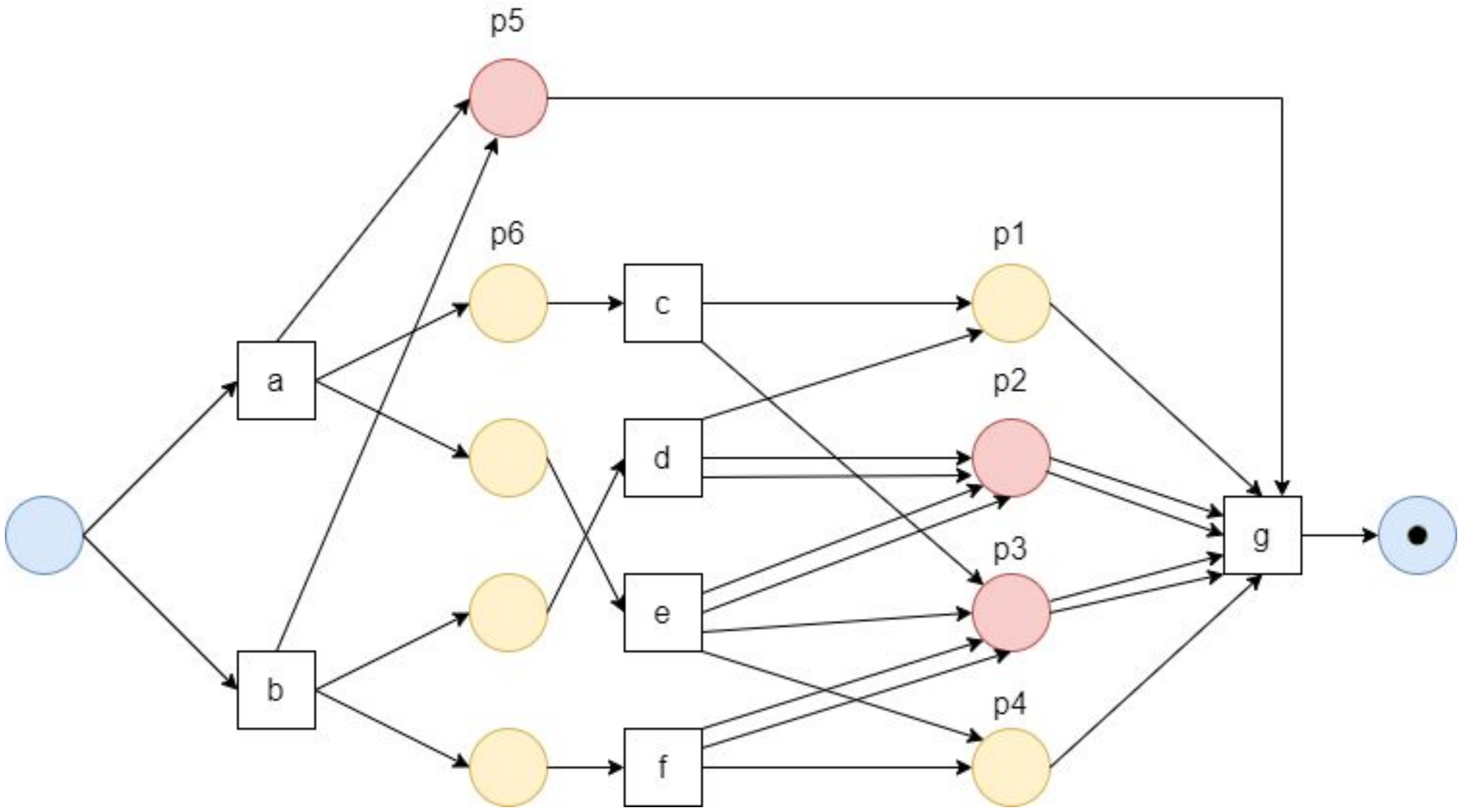
$$L = [\epsilon, \langle a,c,e,g\rangle^2, \langle a,e,c,g\rangle^3, \langle b,d,f,g\rangle^2, \langle b,f,d,g\rangle^4]$$



	ε	a	c	e	g
x_{p3}	0	0	1	2	0
x_{p4}	0	0	0	1	0
x_{p8}					

Running Example: Extended Case

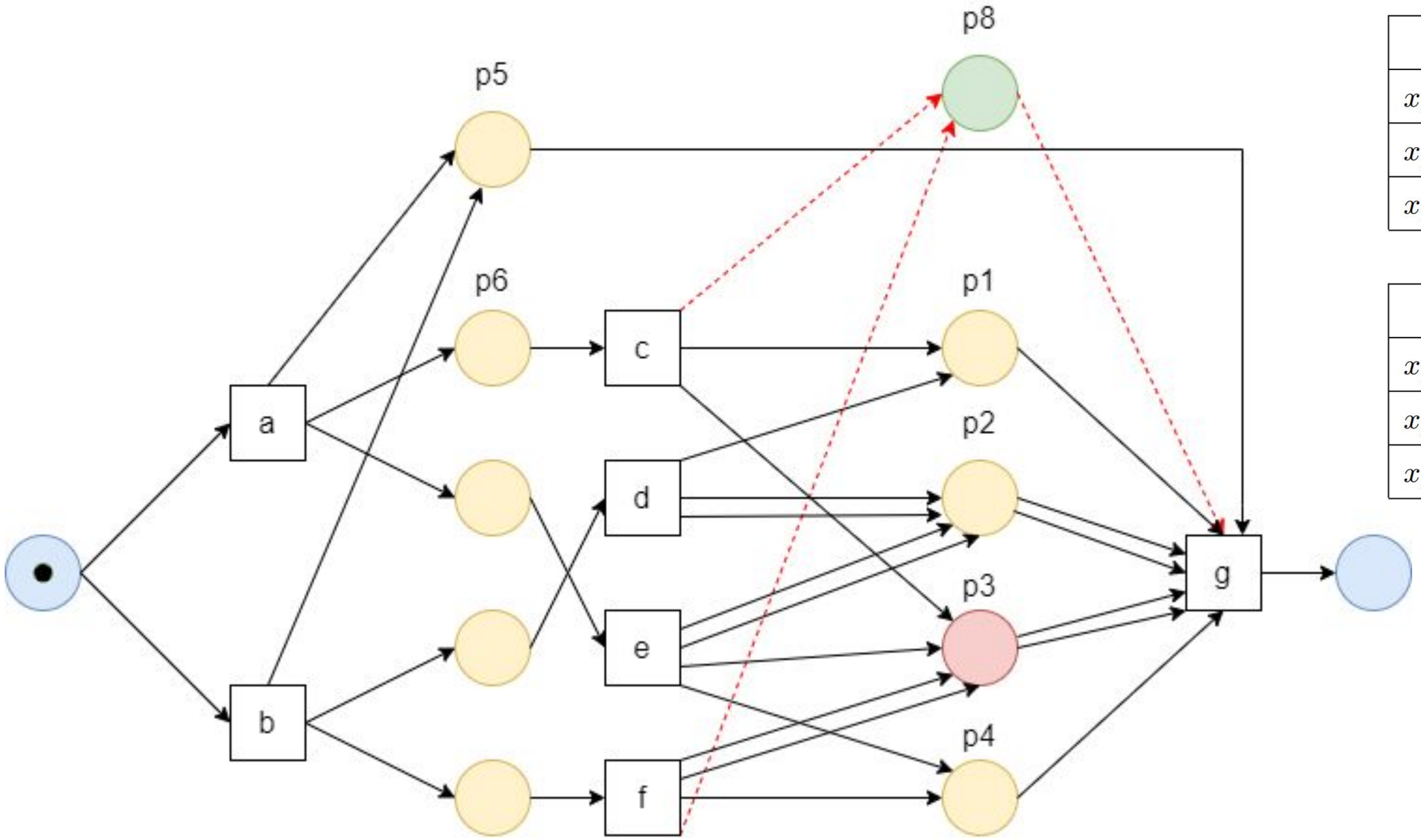
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
x_{p3}	0	0	1	2	0
x_{p4}	0	0	0	1	0
x_{p8}	0	0	1	1	0

Running Example: 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
x_{p3}	0	0	1	2	0
x_{p4}	0	0	0	1	0
x_{p8}	0	0	1	1	0

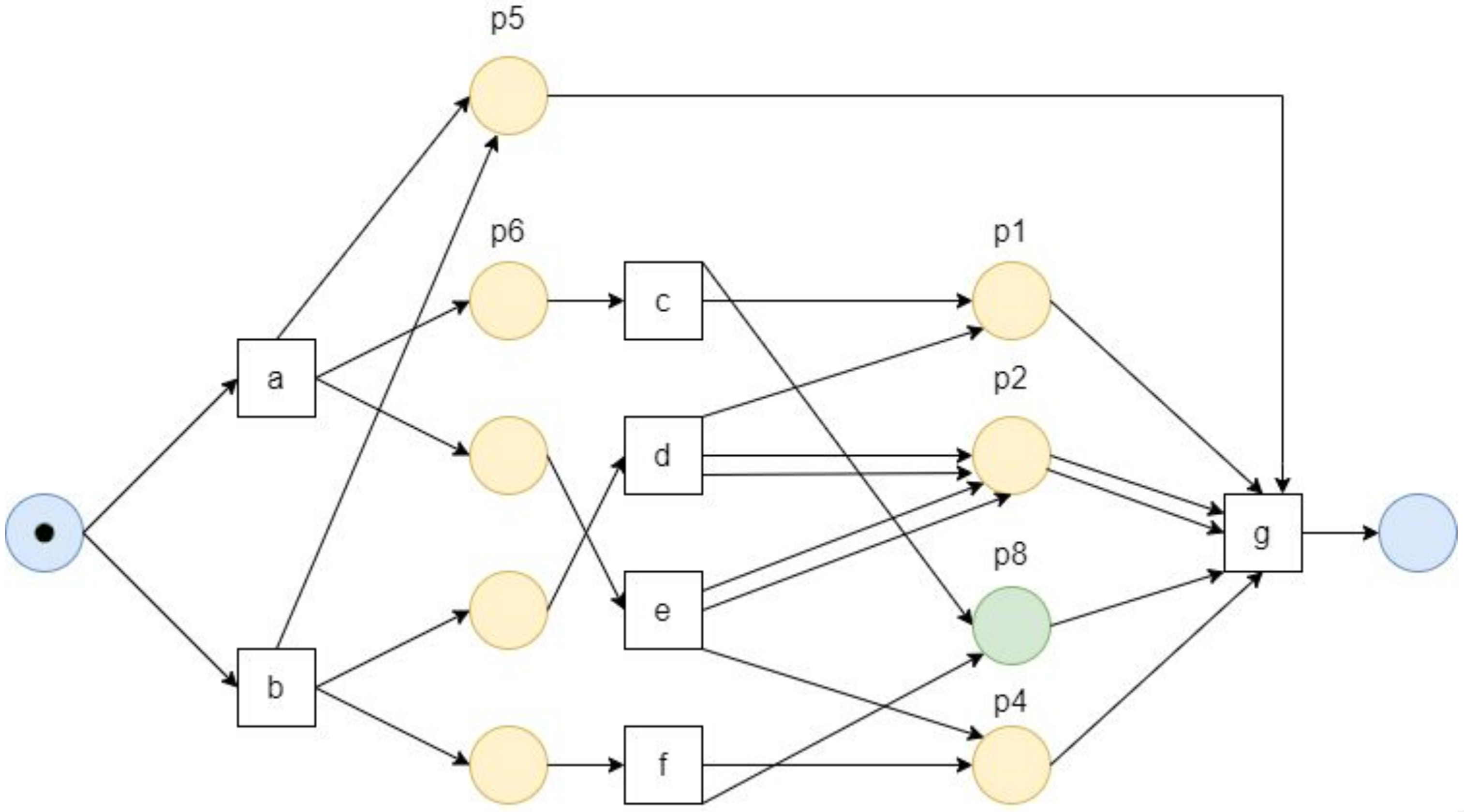
	ε	a	e	c	g
x_{p3}	0	0	1	2	0
x_{p4}	0	0	1	1	0
x_{p8}	0	0	0	1	0

	ε	b	d	f	g
x_{p3}	0	0	0	2	0
x_{p4}	0	0	0	1	0
x_{p8}	0	0	0	1	0

	ε	b	f	d	g
x_{p3}	0	0	2	2	0
x_{p4}	0	0	1	1	0
x_{p8}	0	0	1	1	0

Running Example: Extended Case

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

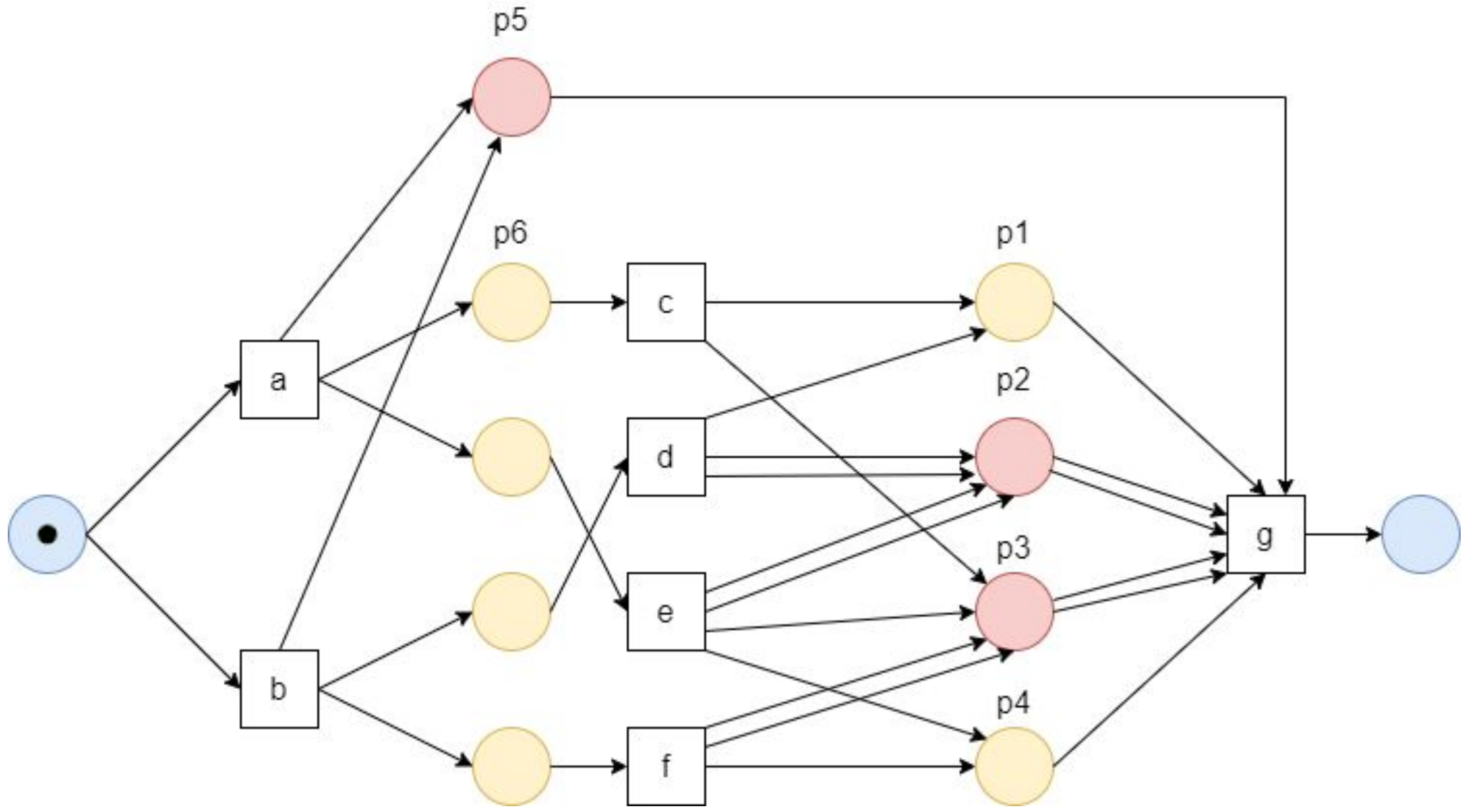


Review

- Realized the algorithm via its simple, intermediate and extended cases
- However, can all implicit places be discovered using this method?
- Uncovering the limitations of our implicit place removal technique

Limitations

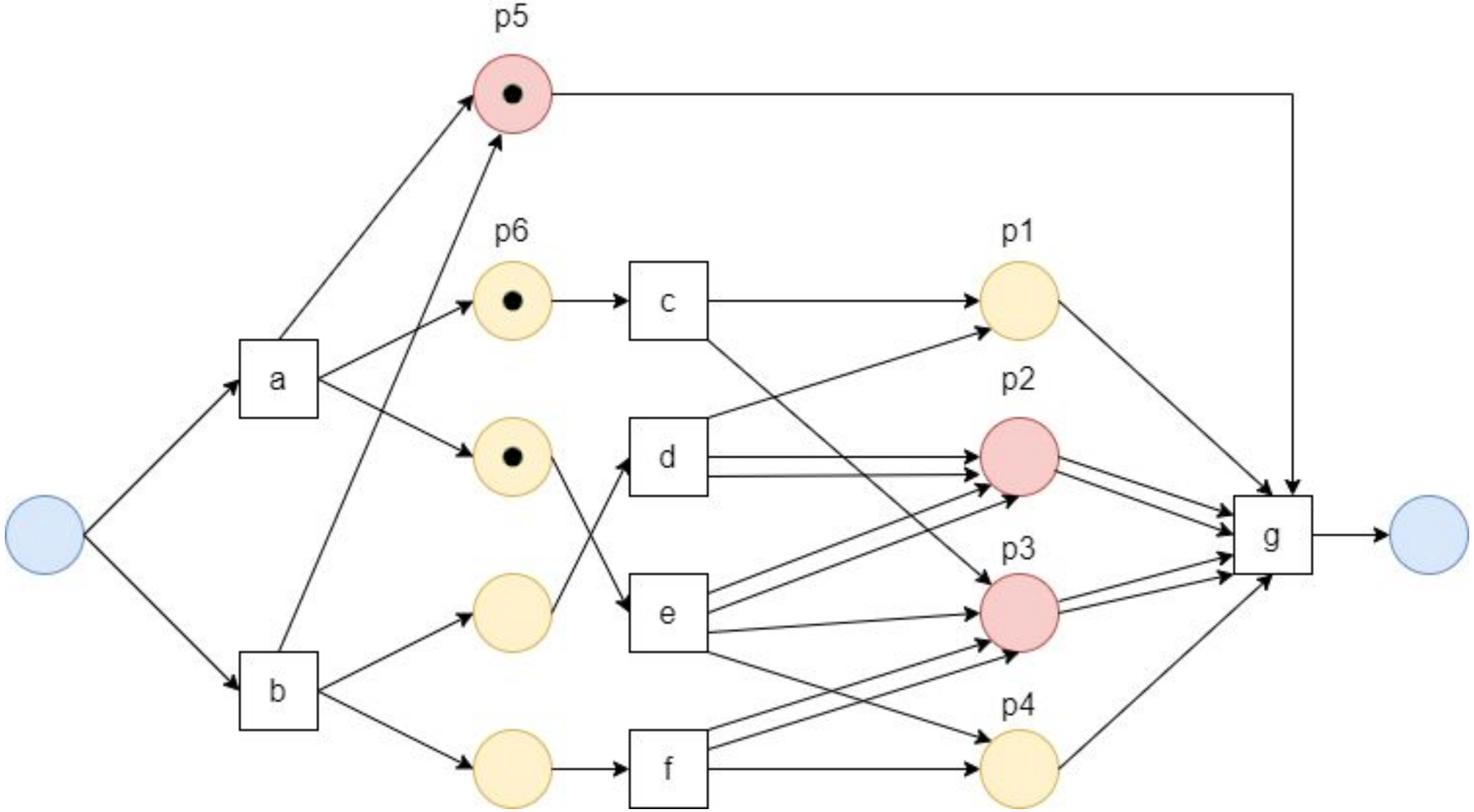
$$L = [\epsilon, \langle a,c,e,g\rangle^2, \langle a,e,c,g\rangle^3, \langle b,d,f,g\rangle^2, \langle b,f,d,g\rangle^4]$$



	ε	a	c	e	g
x_{p_2}	0				
x_{p_1}	0				
x_{p_9}					

Limitations

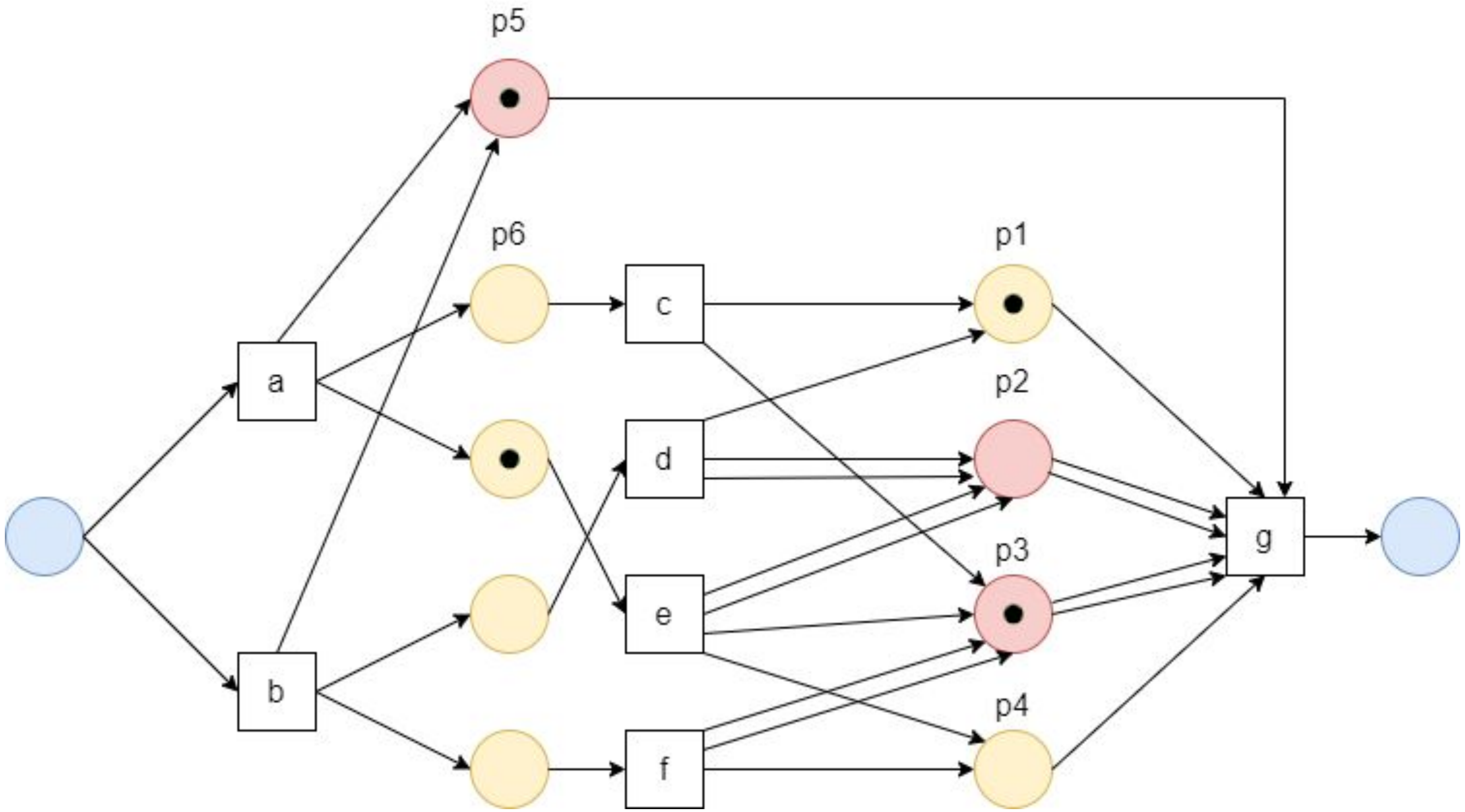
$$L = [\epsilon, \langle a, c, e, g\rangle^2, \langle a, e, c, g\rangle^3, \langle b, d, f, g\rangle^2, \langle b, f, d, g\rangle^4]$$



	ε	a	c	e	g
x_{p_2}	0	0			
x_{p_1}	0	0			
x_{p_9}					

Limitations

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



	ε	a	c	e	g
x_{p_2}	0	0	0		
x_{p_1}	0	0	1		
x_{p_9}	0	0	-1		

$x_{p2} > x_{pi}$ **does not hold for any** i

Review

- Understood the motivation behind removing implicit places
- Examined the main idea of our method
- Witnessed running examples of the algorithm
- Recognized limitations of our technique

Next Steps

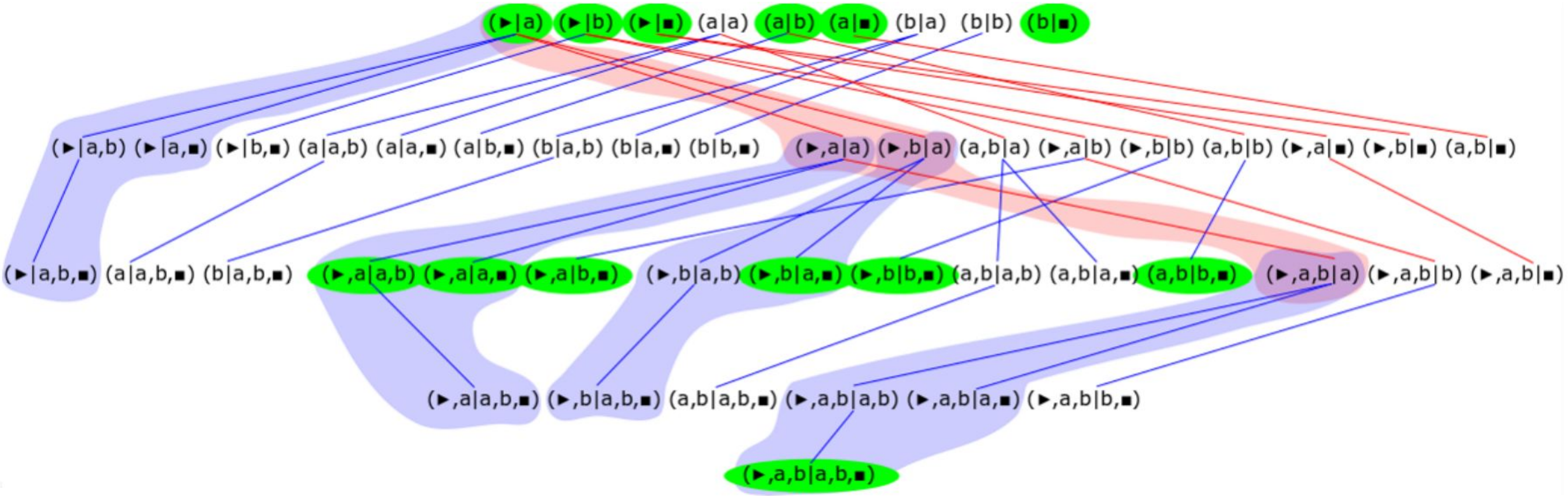
- When and where is this algorithm actually beneficial?
- Diving into the application to the eST-Miner
- Are there any associated implementation challenges?

Application

- Using the eST-Miner as a process discovery algorithm
- Only the event log is used
- Candidate places are traversed and evaluated using token replay
- Replaying the event log is crucial to removing implicit places

Application

➢ **A clever way to enumerate places using a *tree* like structure**



Application

- The discovery of a set of all *fitting* places is guaranteed
- A significant number of implicit places are also discovered
- Removed by solving an Integer Linear Programming Problem
- Immense time and space complexity
- Reason for applying our implicit place removal technique

Application: Variants

Final Place Removal (FPR)

- The final set of places is computed first
- Then, all places are compared and implicit places are removed

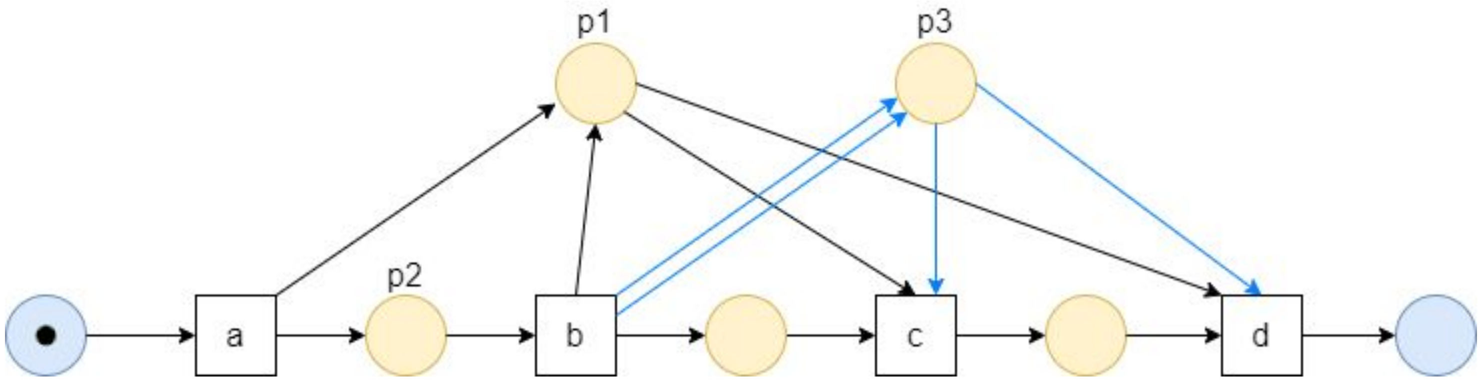
Concurrent Place Removal (CPR)

- Every discovered place is directly compared to existing places
- Removed immediately if found to be implicit

Application: Challenges

➢ The eST-Miner returns a Petri net without arc weights

➢ However, arc weights are required



	ε	a	b	c	d
x_{p_1}	0	1	2	1	0
x_{p_2}	0	1	0	0	0
x_{p_3}	0	0	2	1	0

Application: Challenges

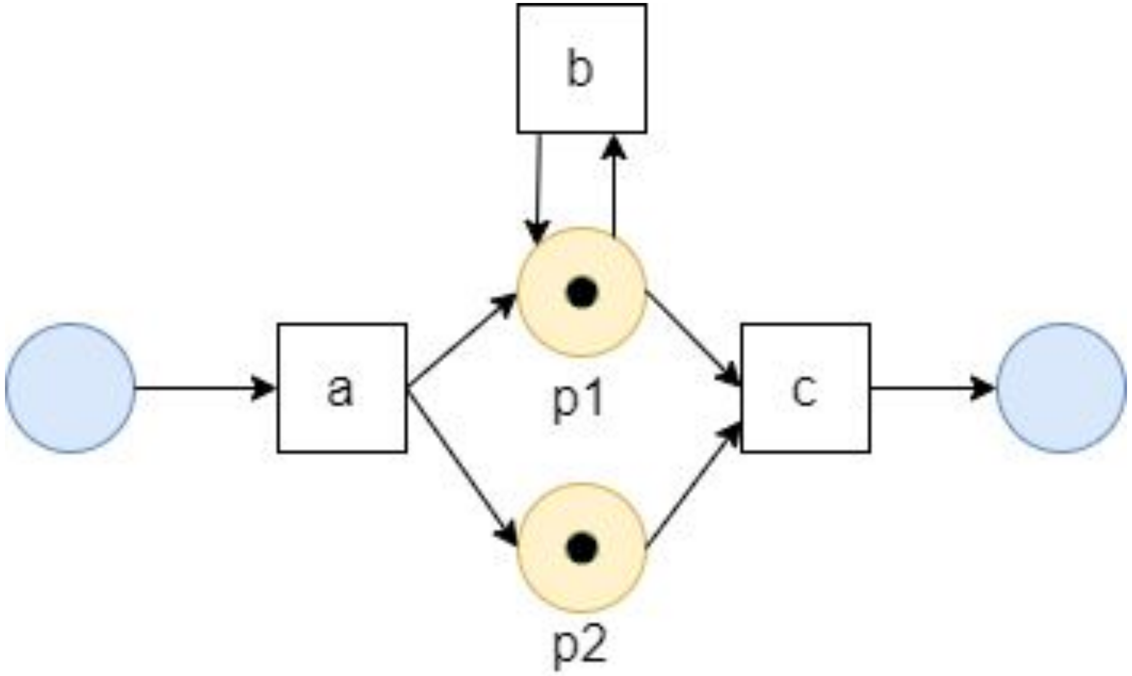
- The eST-Miner discovers only fitting places
- Places empty at the beginning and end of token replay
- Place emptiness is guaranteed by construction

$$x_{p_1} = x_{p_2} = 0 \qquad (4)$$

$$x_{p_3} = 0 \qquad (5)$$

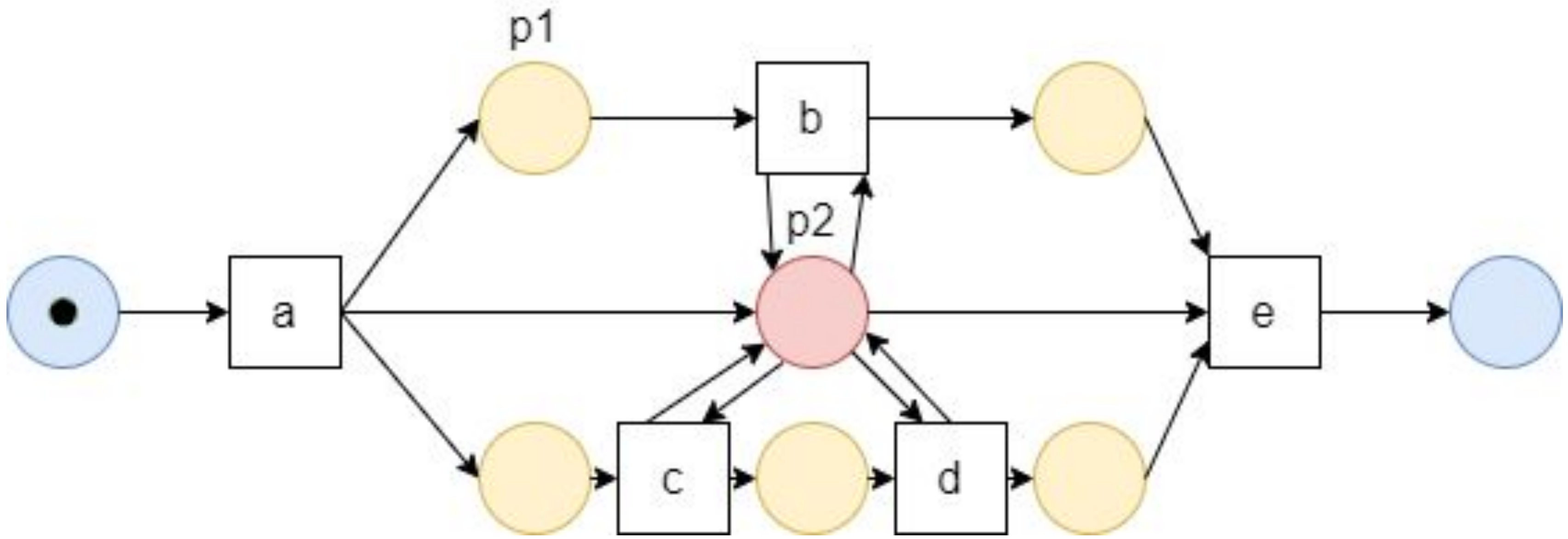
Application: Challenges

eST-Miner allows for Petri nets with self-loops



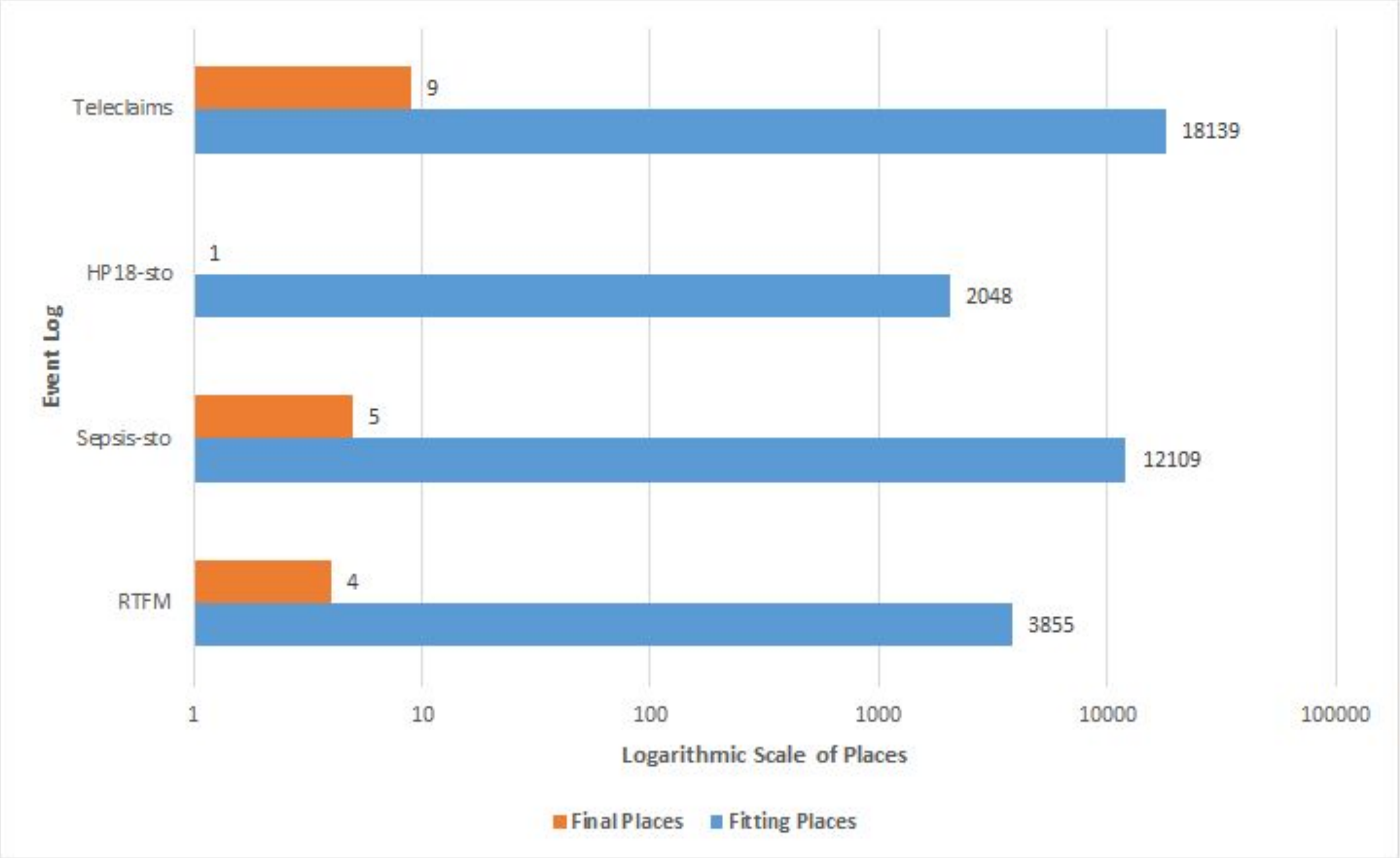
Application: Challenges

Special case of places with self-loops in parallel constructs



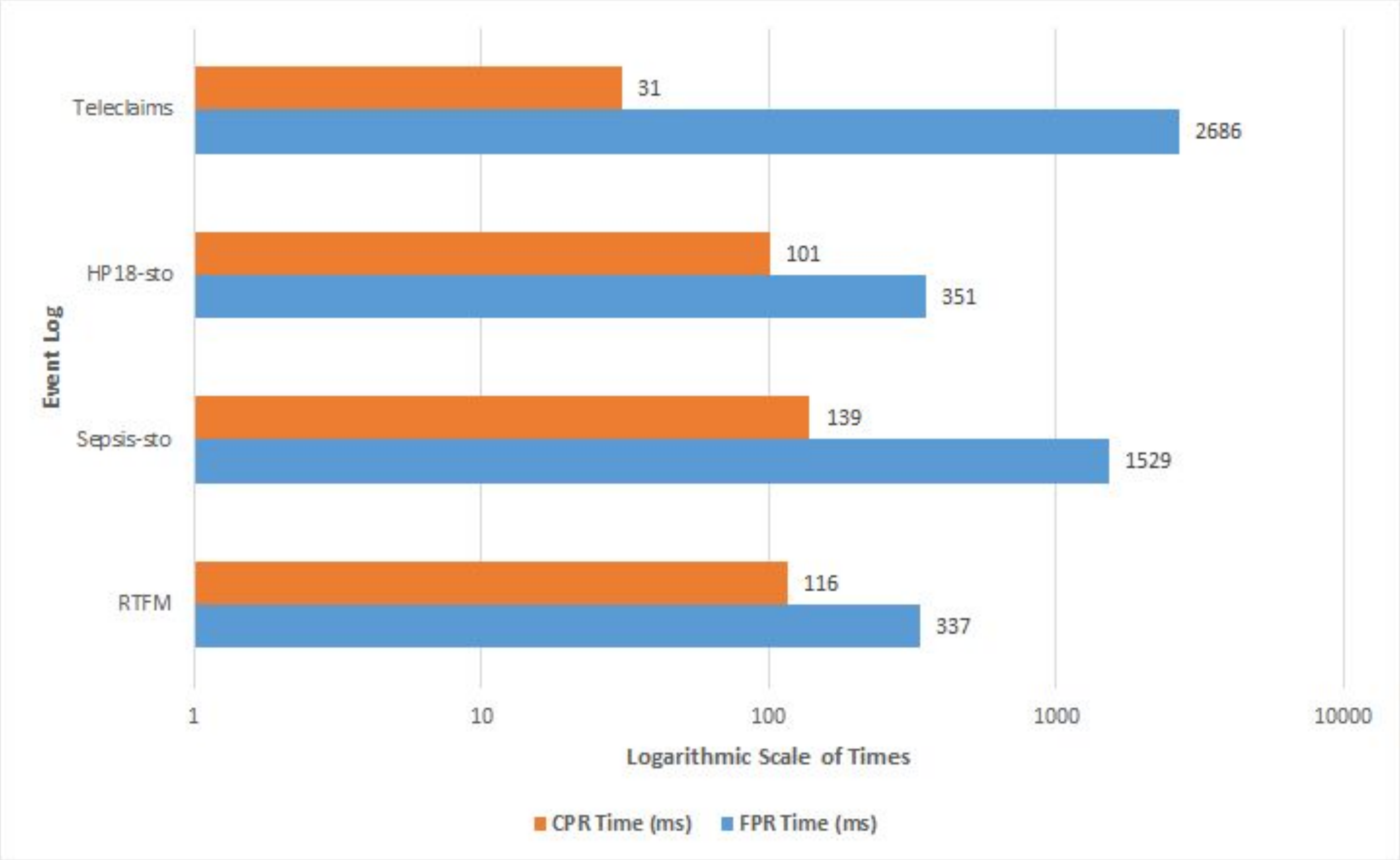
Evaluation: Space Efficiency

Remaining final places are significantly less than total fitting places



Evaluation: Time Efficiency

CPR is always significantly faster than FPR



Conclusion

- An approach to identify and remove implicit places from Petri nets
- Combination with the eST-Miner process discovery algorithm
- Sequential and concurrent application schemes
- 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] L. Mannel. Removing Implicit Places Using Regions for Process Discovery. 06 2020.
[2] W. Aalst and A. Ter. Verification of workflow task structures: A petri-net-based approach. Information Systems, 25:43–69, 03 2000.
[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