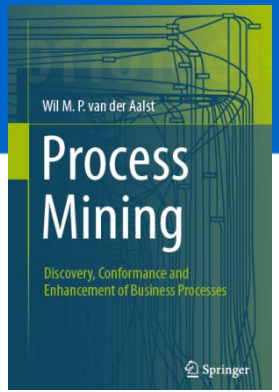
Process Mining: Data Science in Action

Alpha Algorithm: Limitations



prof.dr.ir. Wil van der Aalst
www.processmining.org

Technische Universiteit Eindhoven University of Technology

Where innovation starts

Let L be an event log over T.

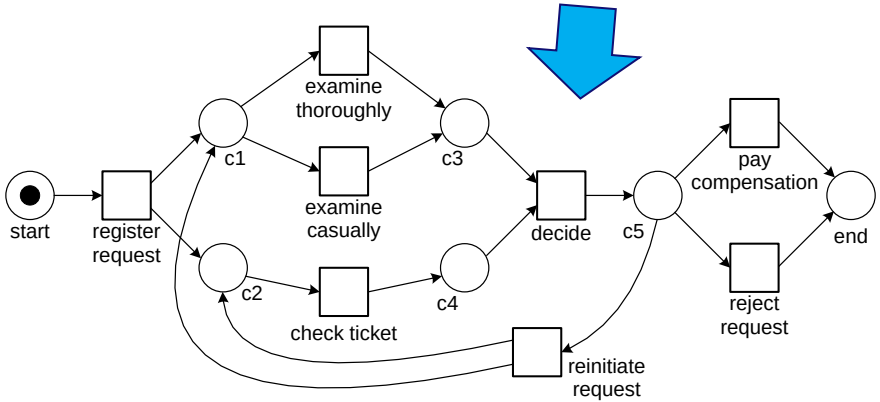
$\alpha(L)$ is defined as follows.

1. $T_L = \{ t \in T \mid \exists_{\sigma \in L}\, t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in L}\, t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in L}\, t = \mathit{last}(\sigma) \}$,
4. $X_L = \{ (A,B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_L b \wedge \forall_{a1,a2 \in A}\, a_1 \#_L a_2 \wedge \forall_{b1,b2 \in B}\, b_1 \#_L b_2 \}$,
5. $Y_L = \{ (A,B) \in X_L \mid \forall_{(A',B') \in X_L}\, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_L = \{ p_{(A,B)} \mid (A,B) \in Y_L \} \cup \{i_L, o_L\}$,
7. $F_L = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_L \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_L \wedge b \in B \} \cup \{ (i_L, t) \mid t \in T_I \} \cup \{ (t, o_L) \mid t \in T_O \}$, and
8. $\alpha(L) = (P_L, T_L, F_L)$.

Alpha algorithm

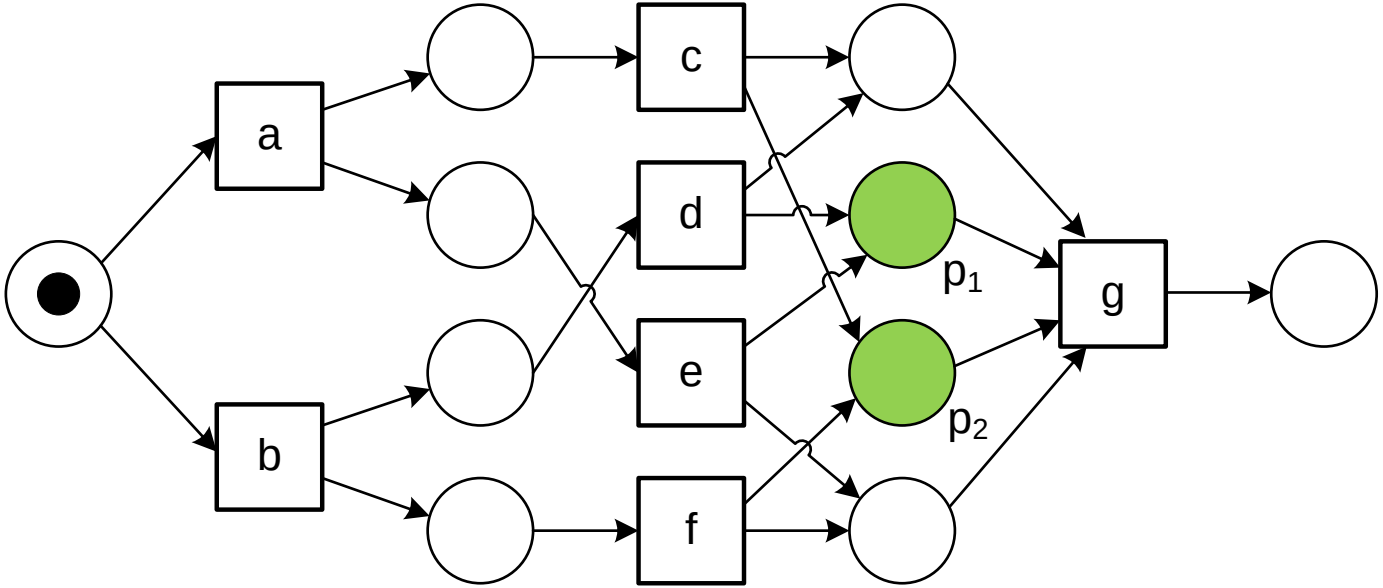
case id	event id	properties				
		timestamp	activity	resource	cost	…
1	35654423	30-12-2010:11.02	register request	Pete	50	…
	35654424	31-12-2010:10.06	examine thoroughly	Sue	400	…
	35654425	05-01-2011:15.12	check ticket	Mike	100	…
	35654426	06-01-2011:11.18	decide	Sara	200	…
	35654427	07-01-2011:14.24	reject request	Pete	200	…
2	35654483	30-12-2010:11.32	register request	Mike	50	…
	35654485	30-12-2010:12.12	check ticket	Mike	100	…
	35654487	30-12-2010:14.16	examine casually	Pete	400	…
	35654488	05-01-2011:11.22	decide	Sara	200	…
	35654489	08-01-2011:12.05	pay compensation	Ellen	200	…
3	35654521	30-12-2010:14.32	register request	Pete	50	…
	35654522	30-12-2010:15.06	examine casually	Mike	400	…
	35654524	30-12-2010:16.34	check ticket	Ellen	100	…
	35654525	06-01-2011:09.18	decide	Sara	200	…
	35654526	06-01-2011:12.18	reinitiate request	Sara	200	…
	35654527	06-01-2011:13.06	examine thoroughly	Sean	400	…
	35654530	08-01-2011:11.43	check ticket	Pete	100	…
	35654531	09-01-2011:09.55	decide	Sara	200	…
	35654533	15-01-2011:10.45	pay compensation	Ellen	200	…
4	35654641	06-01-2011:15.02	register request	Pete	50	…
	35654643	07-01-2011:12.06	check ticket	Mike	100	…
	35654644	08-01-2011:14.43	examine thoroughly	Sean	400	…
	35654645	09-01-2011:12.02	decide	Sara	200	…
	35654647	12-01-2011:15.44	reject request	Ellen	200	…
5	35654711	06-01-2011:09.02	register request	Ellen	50	…
	35654712	07-01-2011:10.16	examine casually	Mike	400	…
	35654714	08-01-2011:11.22	check ticket	Pete	100	…
	35654715	10-01-2011:13.28	decide	Sara	200	…
	35654716	11-01-2011:16.18	reinitiate request	Sara	200	…
	35654718	14-01-2011:14.33	check ticket	Ellen	100	…
	35654719	16-01-2011:15.50	examine casually	Mike	400	…
	35654720	19-01-2011:11.18	decide	Sara	200	…
	35654721	20-01-2011:12.48	reinitiate request	Sara	200	…
	35654722	21-01-2011:09.06	examine casually	Sue	400	…
	35654724	21-01-2011:11.34	check ticket	Pete	100	…
	35654725	23-01-2011:13.12	decide	Sara	200	…
	35654726	24-01-2011:14.56	reject request	Mike	200	…
6	35654871	06-01-2011:15.02	register request	Mike	50	…
	35654873	06-01-2011:16.06	examine casually	Ellen	400	…
	35654874	07-01-2011:16.22	check ticket	Mike	100	…
	35654875	07-01-2011:16.52	decide	Sara	200	…
	35654877	16-01-2011:11.47	pay compensation	Mike	200	…
…	…	…	…	…	…	…

case id	trace
1	$\langle a, b, d, e, h \rangle$
2	$\langle a, d, c, e, g \rangle$
3	$\langle a, c, d, e, f, b, d, e, g \rangle$
4	$\langle a, d, b, e, h \rangle$
5	$\langle a, c, d, e, f, d, c, e, f, c, d, e, h \rangle$
6	$\langle a, c, d, e, g \rangle$
…	…



Limitation of the α algorithm:
Implicit places

$$L_6 = [\langle a,c,e,g\rangle^2, \langle a,e,c,g\rangle^3, \langle b,d,f,g\rangle^2, \langle b,f,d,g\rangle^4]$$



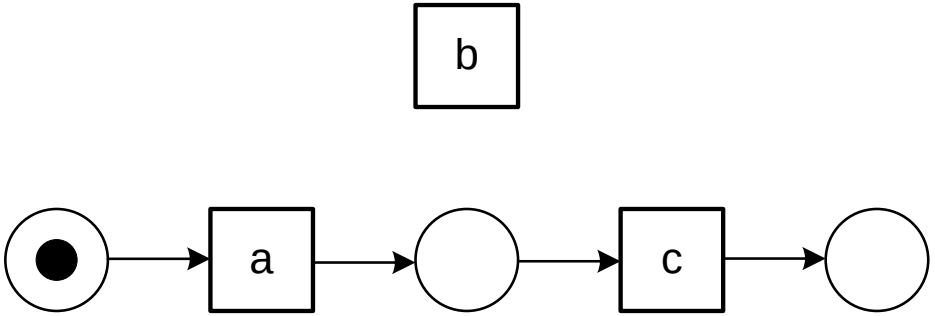
p_1 and p_2 are implicit places!

TU/e

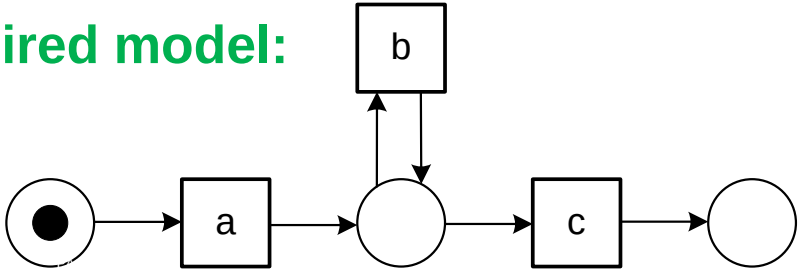
Limitation of the α algorithm:
Loops of length 1

$$L_7 = [\langle a,c\rangle^2, \langle a,b,c\rangle^3, \langle a,b,b,c\rangle^2, \langle a,b,b,b,b,c\rangle^1]$$

a>b a>c b>b b>c	a→b a→c b→c	b‖b	a#a c#c …



desired model:



TU/e

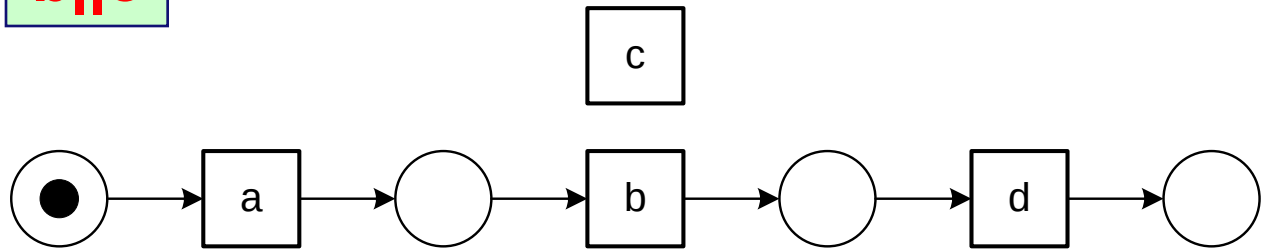
Limitation of the α algorithm:
Loops of length 2

$$L_8 = [\langle a,b,d\rangle^3, \langle a,b,c,b,d\rangle^2, \langle a,b,c,b,c,b,d\rangle]$$

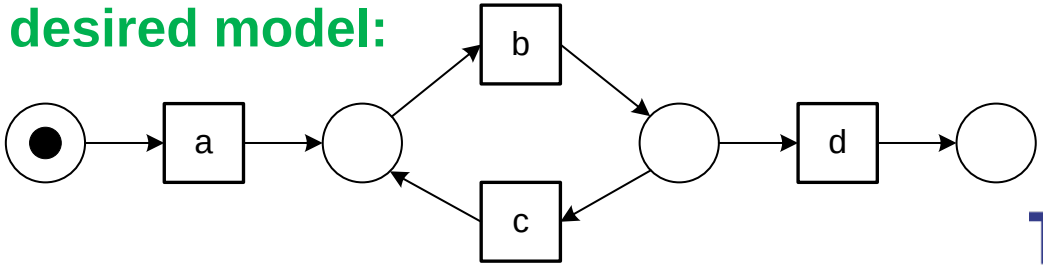
a>b
b>c
b>d
c>b

a→b
b→d

b||c



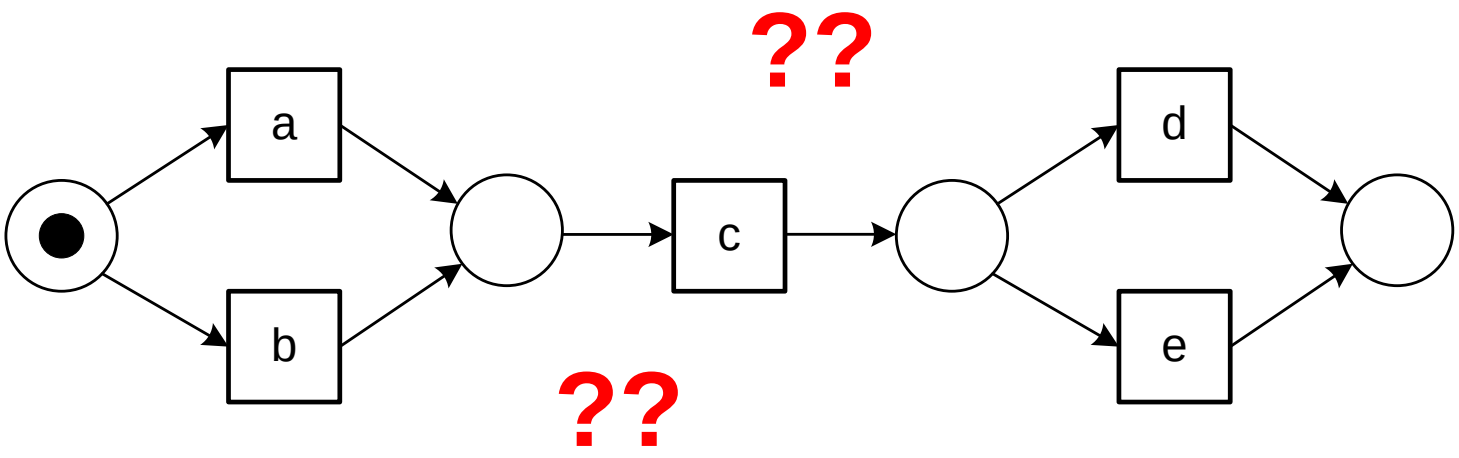
desired model:



TU/e

Limitation of the α algorithm:
Non-local dependencies

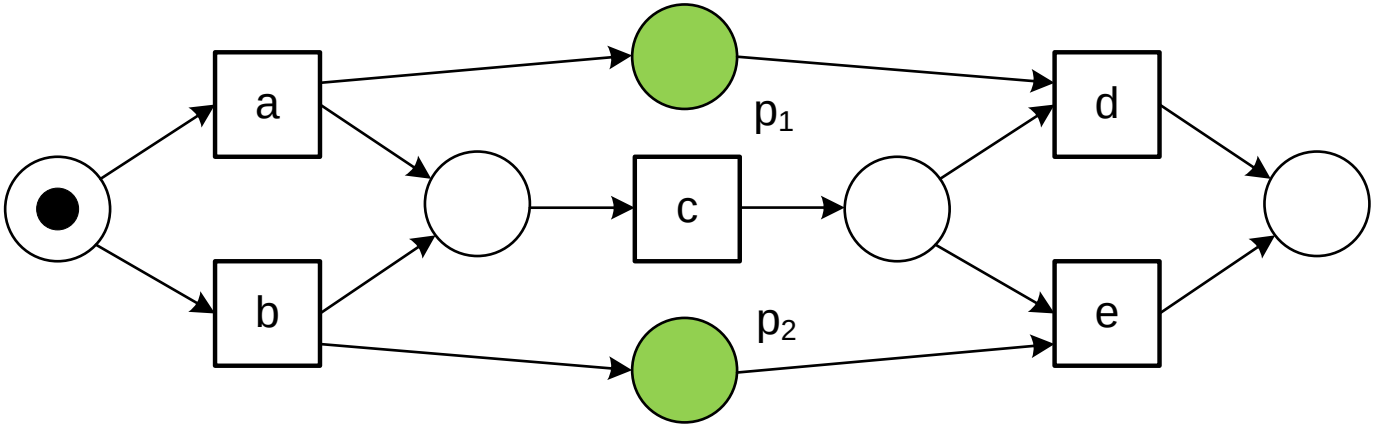
$$L_9 = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}]$$



TU/e

Limitation of the α algorithm:
Non-local dependencies

$$L_9 = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}]$$



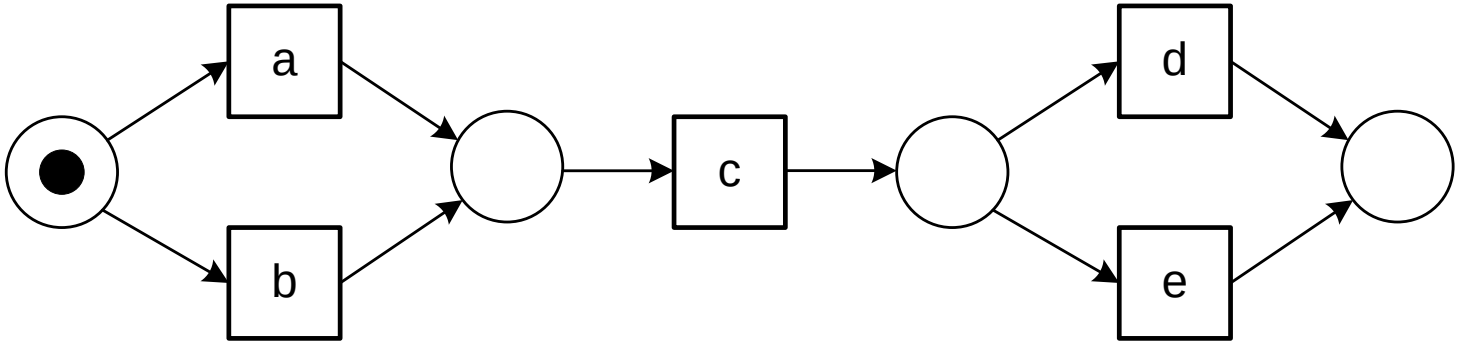
p_1 and p_2 are not discovered!

TU/e

Two event logs: Same discovered model

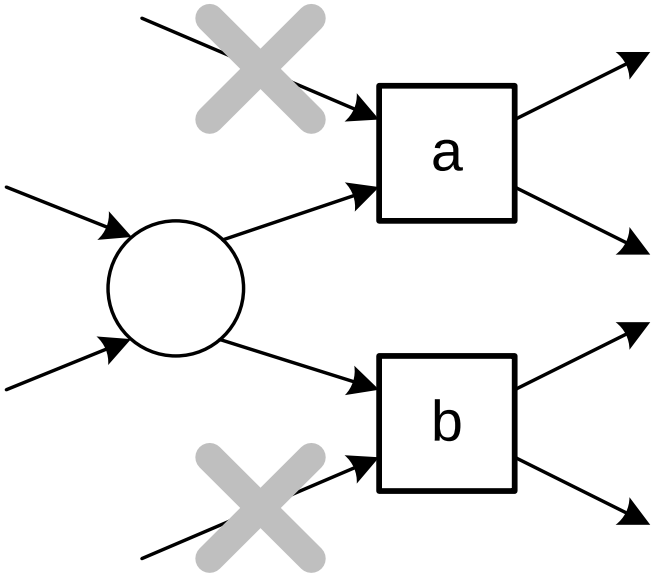
$$L_9 = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}]$$

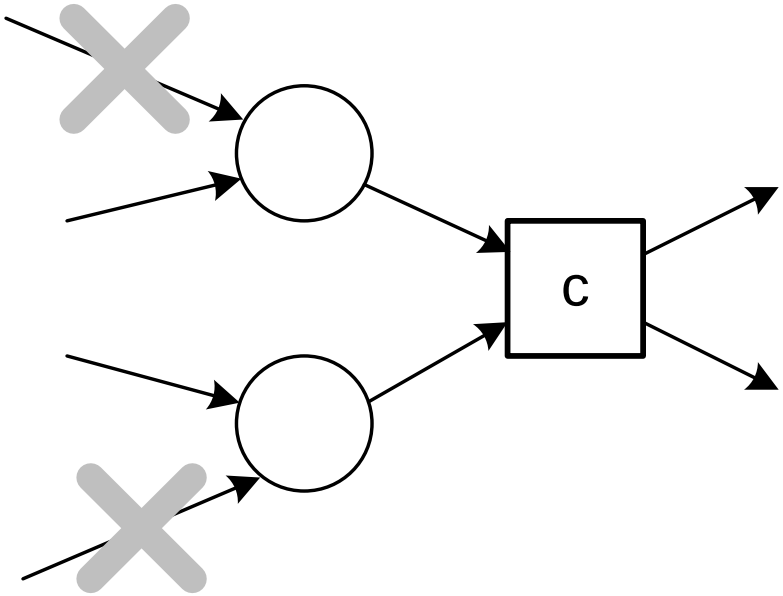
$$L_4 = [\langle a, c, d \rangle^{45}, \langle b, c, d \rangle^{42}, \langle a, c, e \rangle^{38}, \langle b, c, e \rangle^{22}]$$



TU/e

Difficult constructs for the Alpha algorithm





TU/e

Question

Consider the event log:

$$L = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}, \langle a, c, e \rangle^{20}]$$

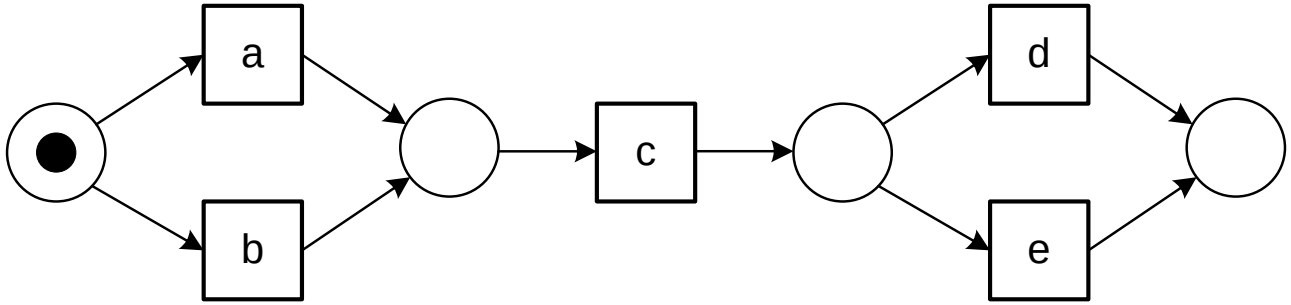
What model will the Alpha algorithm create?

Give a sound WF-net that can produce the observed behavior and nothing more?

TU/e

Answer (1/2):
Model generated by Alpha algorithm

$$L = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}, \langle a, c, e \rangle^{20}]$$



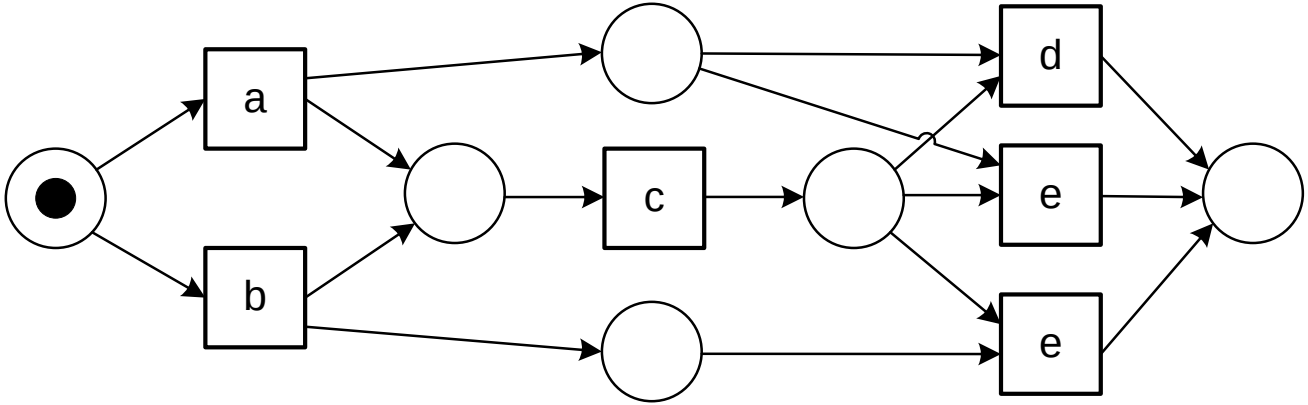
Model generated by Alpha algorithm also allows for trace starting with *b* and ending with *d*!

TU/e

Answer (2/2):

A sound WF-net that can produce the observed behavior and nothing more

$$L = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}, \langle a, c, e \rangle^{20}]$$

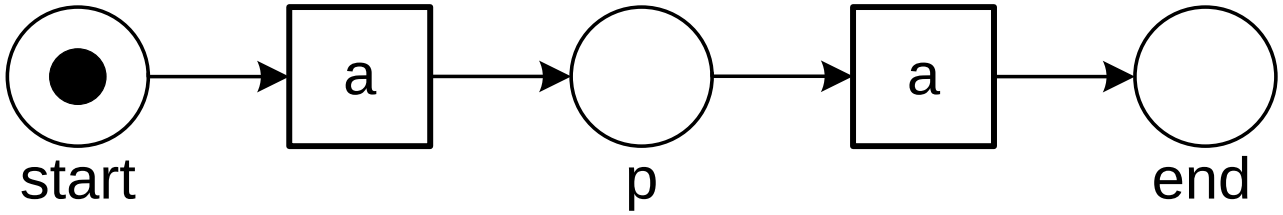


Note the duplicated *e* transition! The Alpha algorithm will never create a WF-net with two transitions having the same label.

TU/e

Limitation of the α algorithm:
representational bias

$$L_{10} = [\langle a, a \rangle^{55}]$$

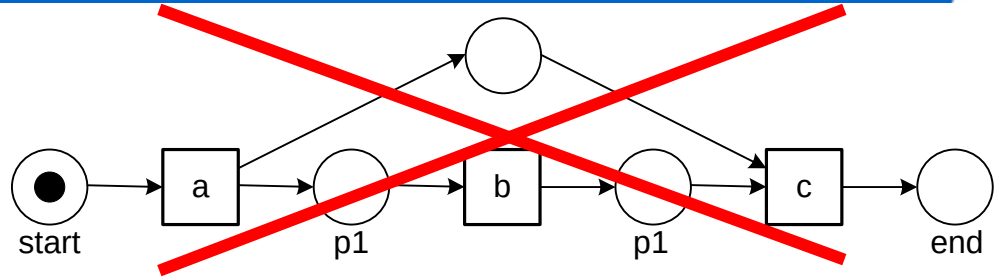


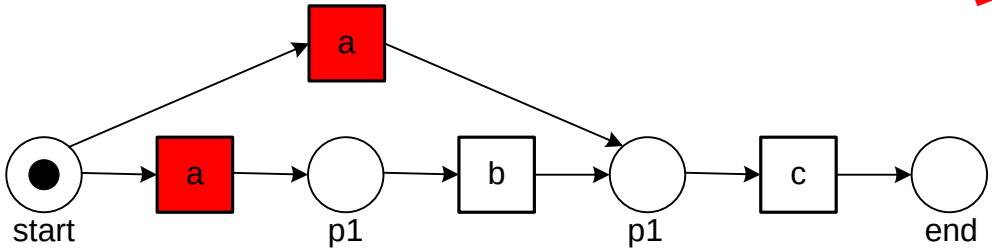
There is no WF-net with unique visible labels that exhibits this behavior.

TU/e

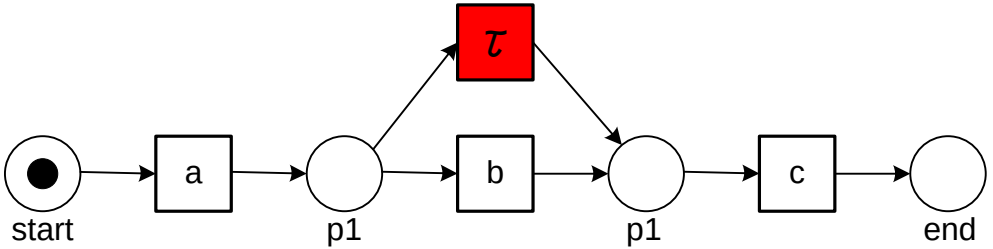
Another example

$$L_{11} = [\langle a, b, c \rangle^{20}, \langle a, c \rangle^{30}]$$



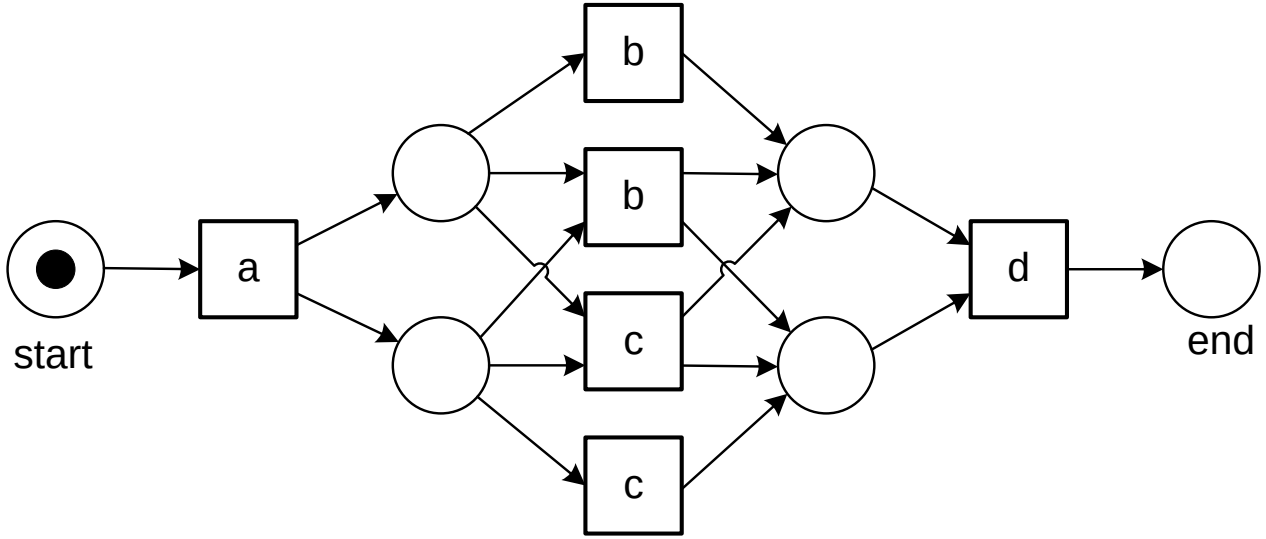


There is no WF-net with unique visible labels that exhibits this behavior.



TU/e

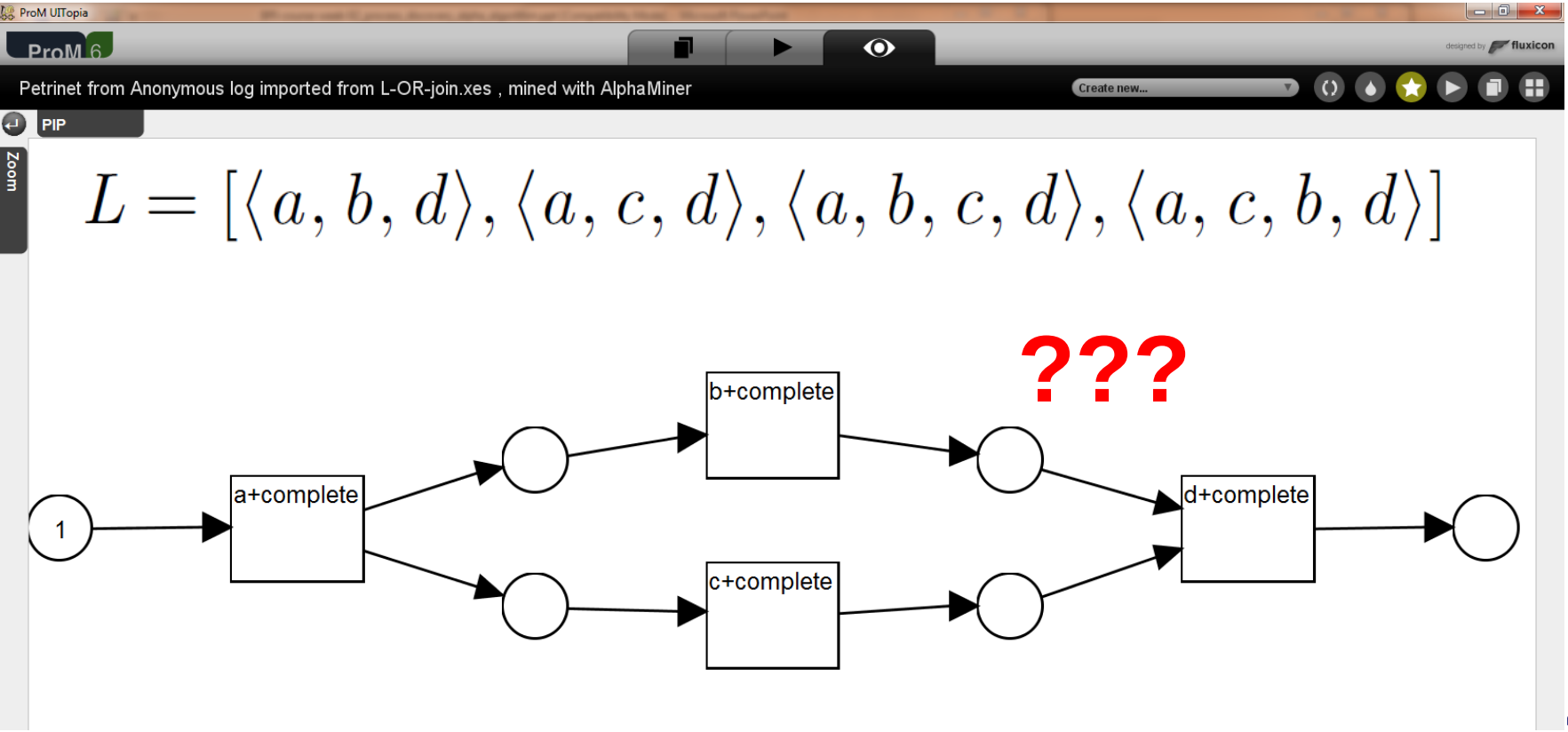
OR-split/join model



- **Let us take an event log containing all possible full firing sequences and apply the Alpha algorithm.**
- **What will happen?**

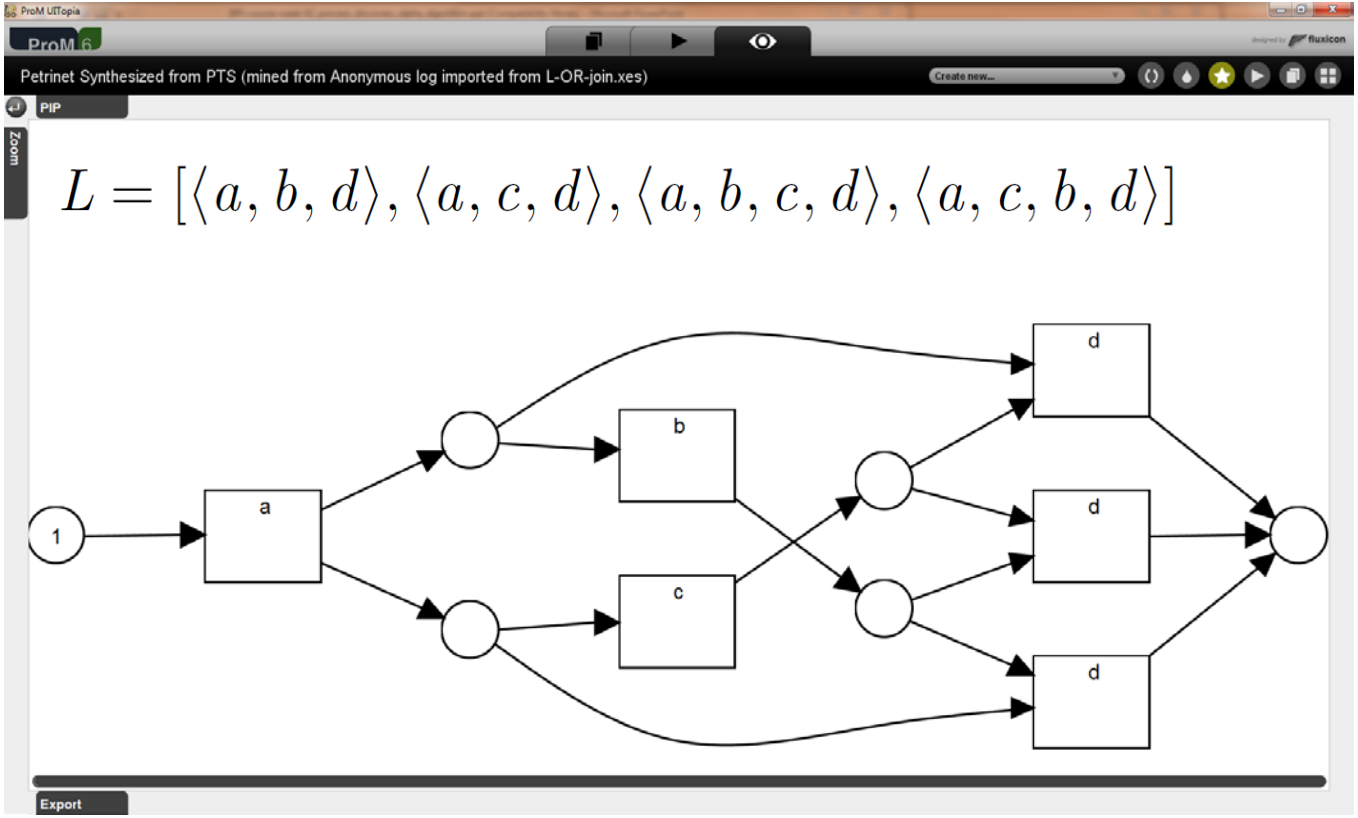
TU/e

Applying the Alpha algorithm using ProM



TU/e

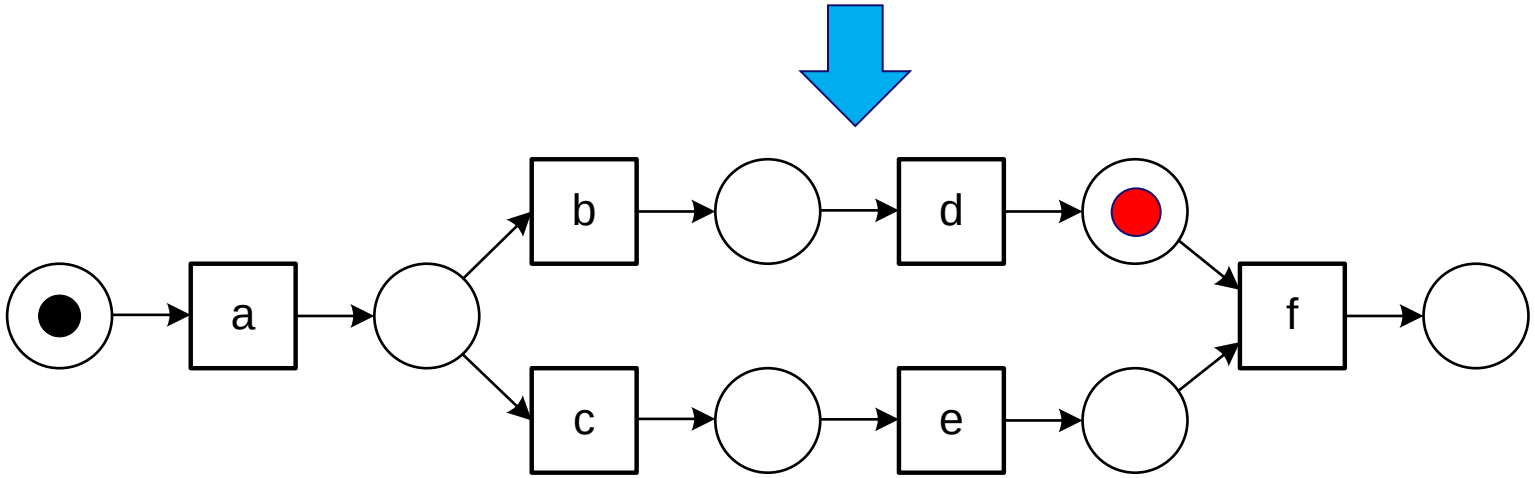
Region-based miner (with label splitting)



TU/e

Limitation of the α algorithm:
resulting model does not need to be a sound WF-net

$$L = [\langle a, b, d, e, f \rangle^{10}, \langle a, c, e, d, f \rangle^{10}]$$



The discovered model is not sound (has deadlock).

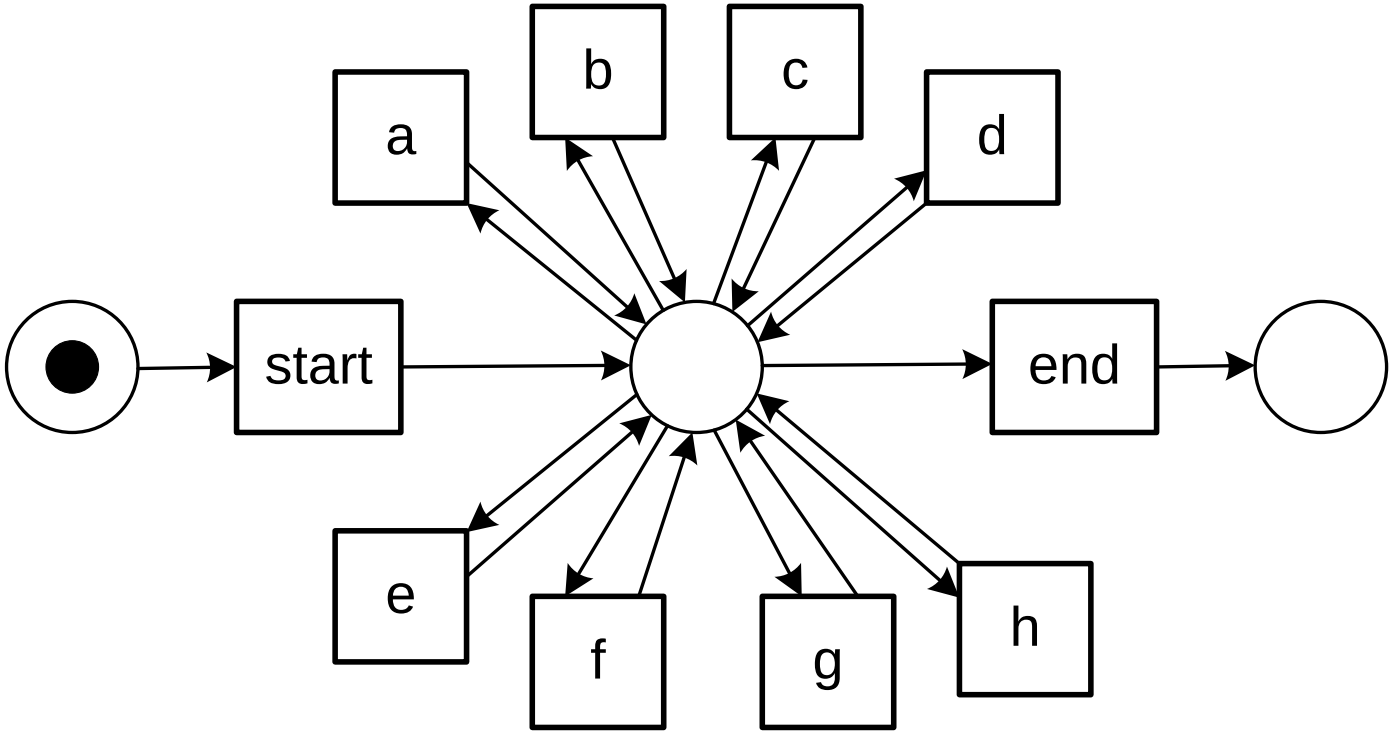
TU/e

Challenge: Noise and Incompleteness

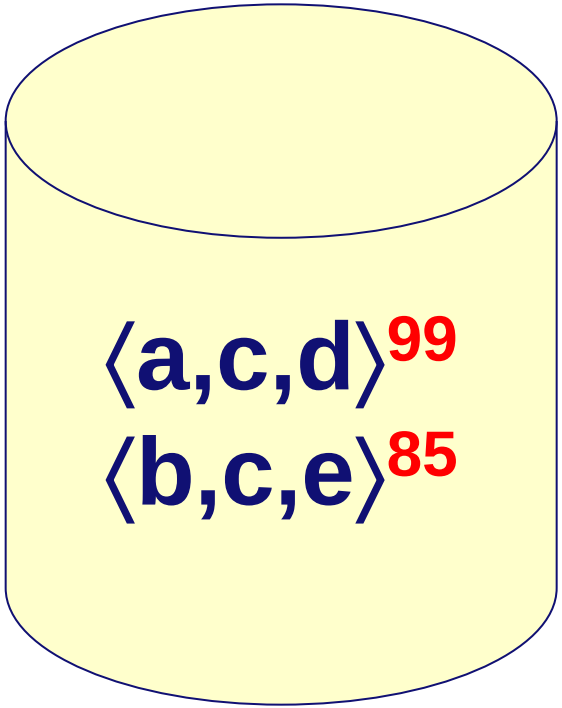
- To discover a suitable process model it is assumed that the event log contains a representative sample of behavior.
- Two related phenomena:
  - Noise: the event log contains rare and infrequent behavior not representative for the typical behavior of the process.
  - Incompleteness: the event log contains too few events to be able to discover some of the underlying control-flow structures.

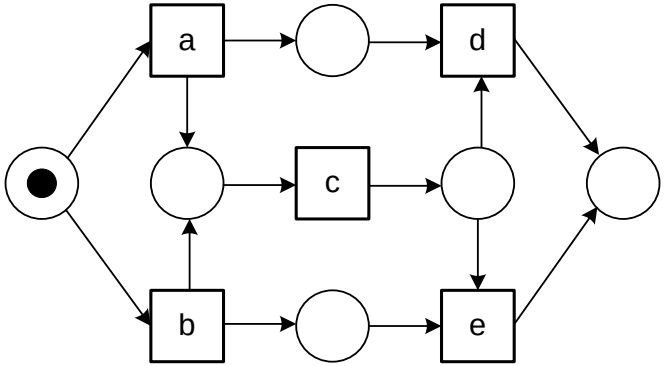
TU/e

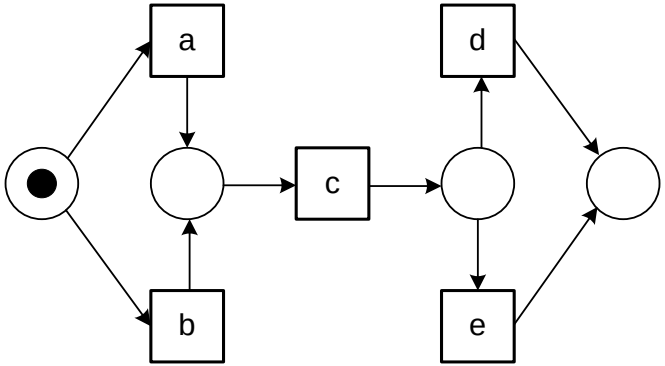
Flower model



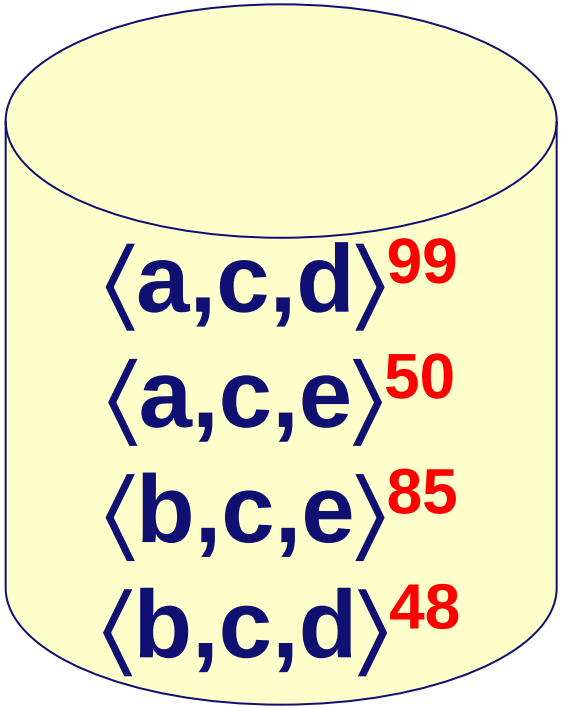
What is the best model?

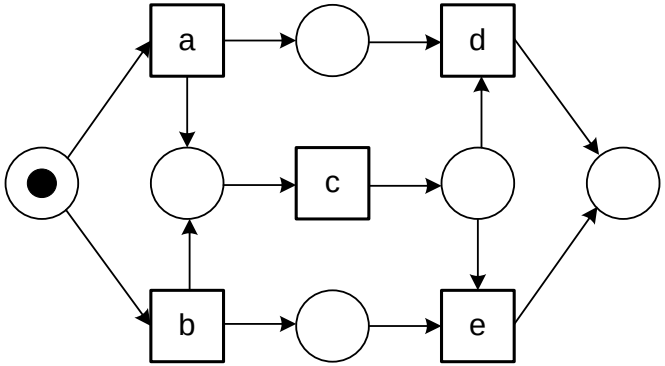


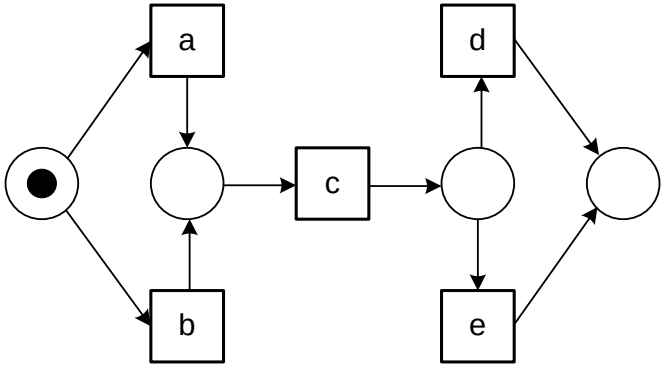




What is the best model?

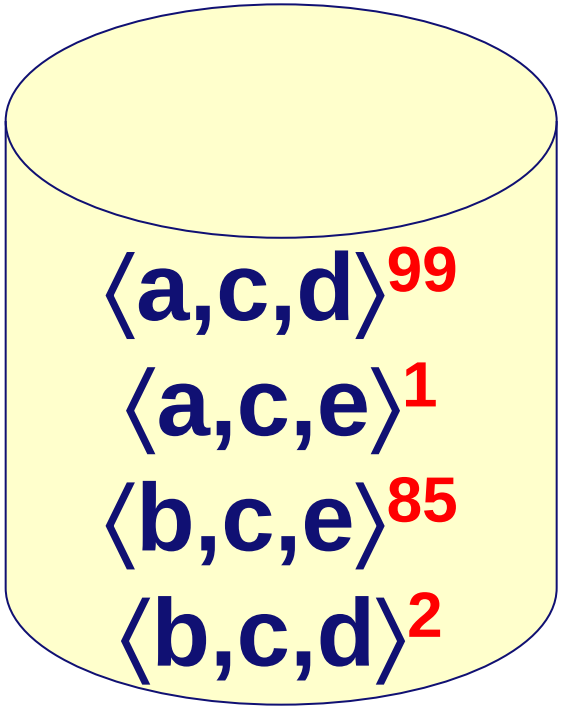


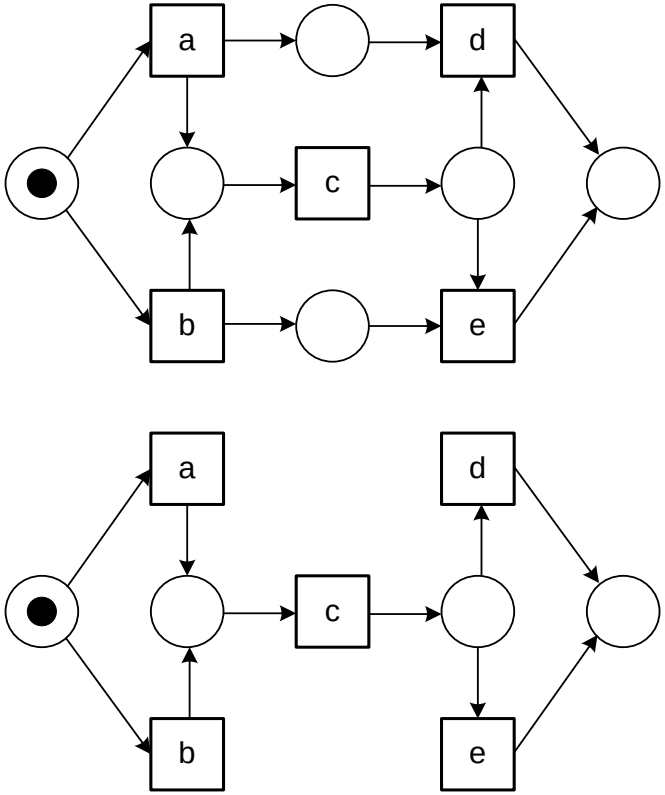




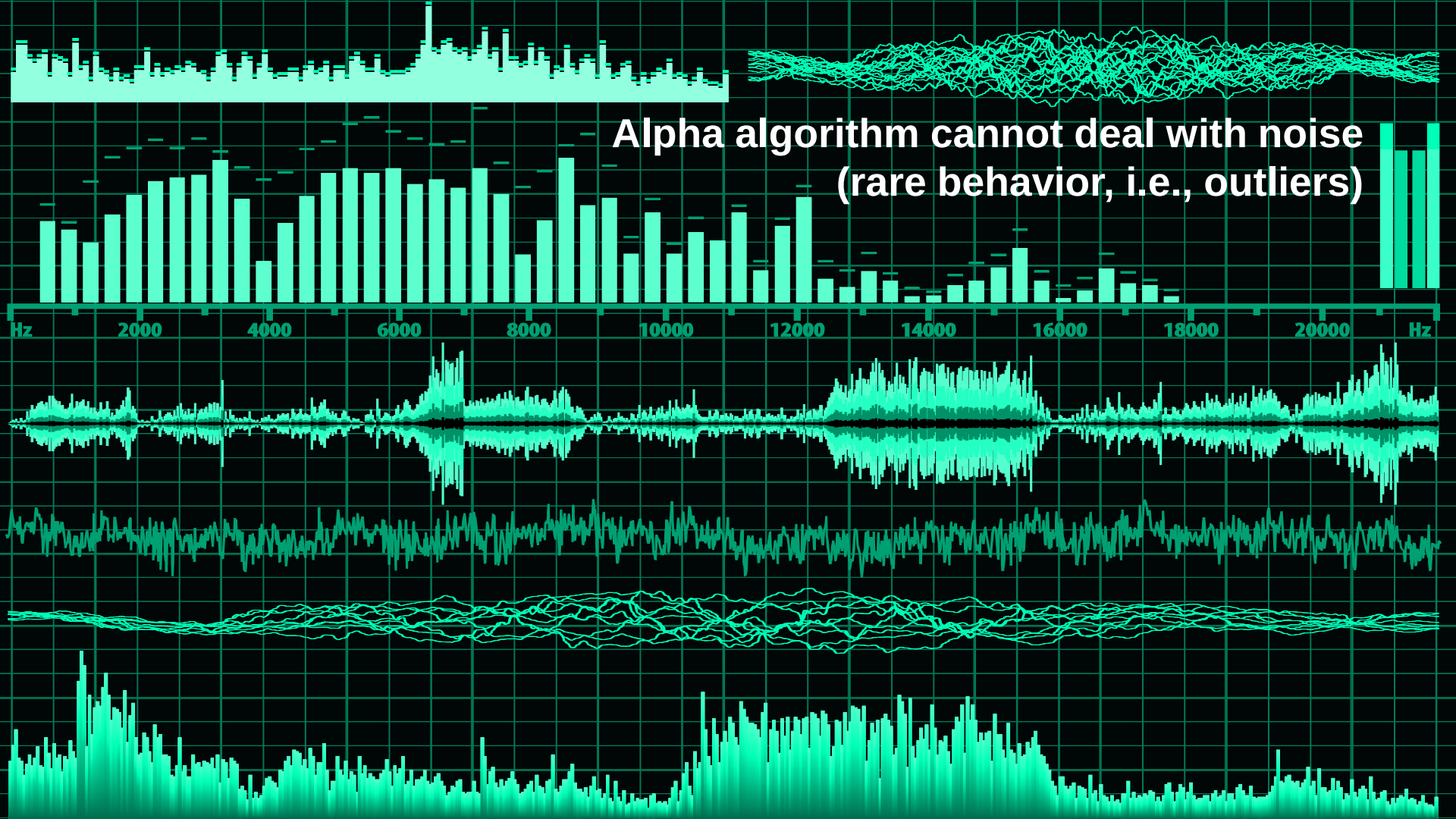
TU/e

What is the best model?



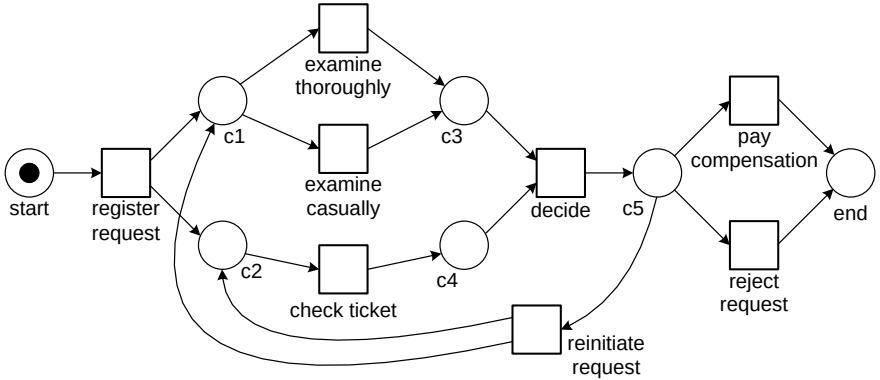


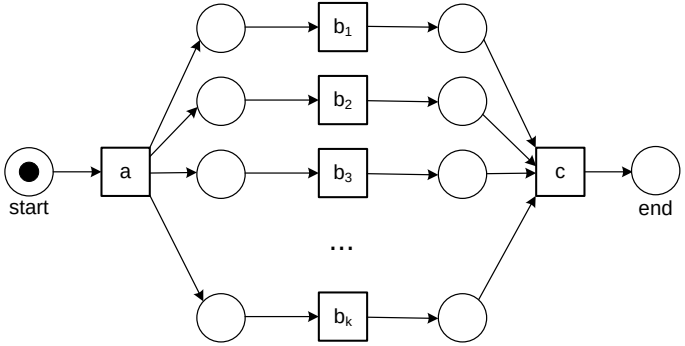
TU/e


Alpha algorithm cannot deal with noise
(rare behavior, i.e., outliers)
Hz
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
Hz

Process models are like maps:
we may not want to see all paths
and only see the highways

Related to noise: Completeness

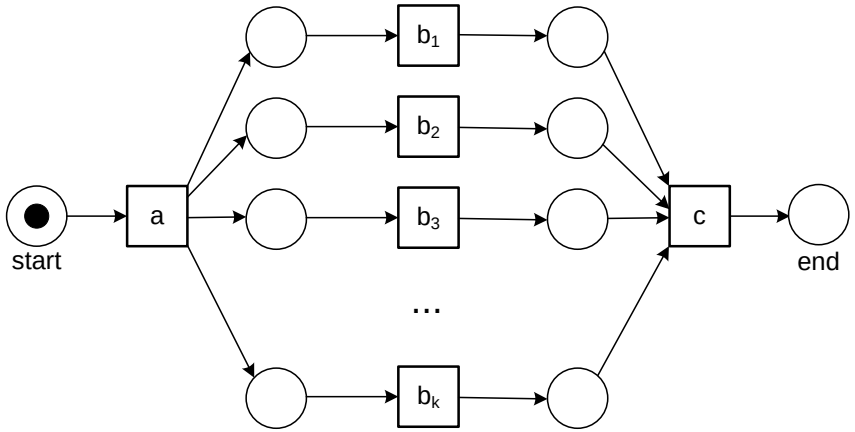




Infinitely many possible traces, 7 possible states

k	number of states: 2^k+2	number of different traces: k!
1	4	1
2	6	2
5	34	120
10	1026	3628800
20	1048578	2.432902e+18

Alpha algorithm depends on the directly follows relation



k	number of states: 2^k+2	number of different traces: k!
1	4	1
2	6	2
5	34	120
10	1026	3628800
20	1048578	2.432902e+18

Only k(k-1) observations are needed to discover the concurrent part. However, if one of these is missing, the result will be incorrect.

TU/e

$$365!/365^{365} \approx 1.454955 \times 10^{-157} \approx 0$$

Limitations (1/2)

- **Implicit places (places that are redundant): harmless and be solved through preprocessing.**
- **Loops of length 1: can be solved in multiple ways (change of algorithm or pre/post-processing).**
- **Loops of length 2: idem.**
- **Non-local dependencies: foundational problem, not specific for Alpha algorithm.**

TU/e

Limitations (2/2)

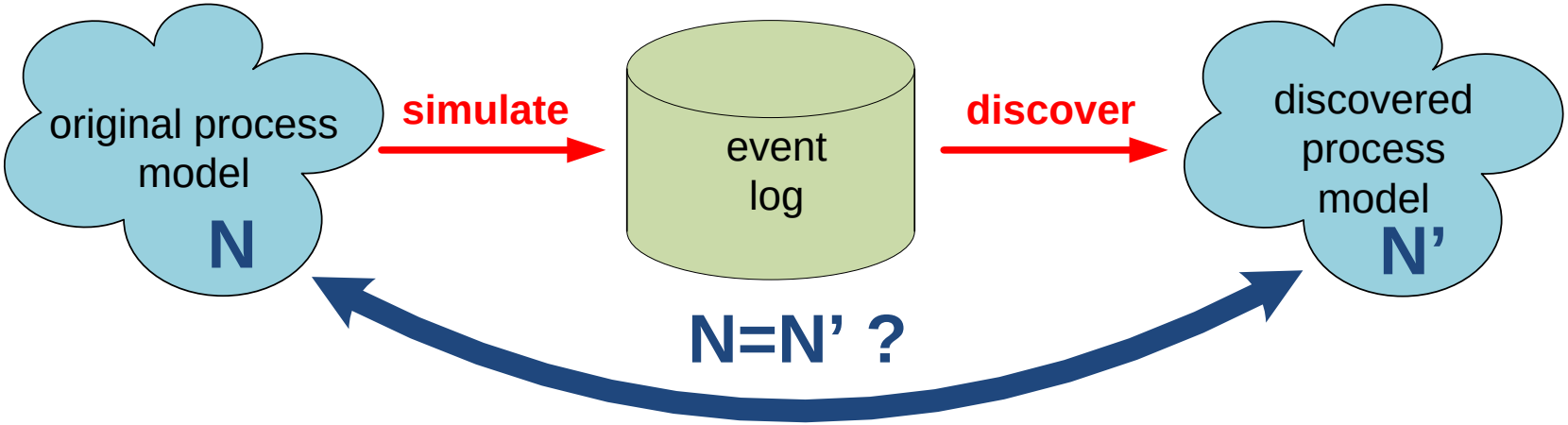
- **Representational bias (cannot discover transitions with duplicate or invisible labels): other algorithms may have a different bias.**
- **Discovered model does not need to be sound: some algorithms ensure this.**
- **Noise: foundational problem, not specific for Alpha algorithm.**
- **Incompleteness: also a foundational problem.**

TU/e

How to measure the quality of a discovered model?

- There may be conflicting requirements (simplicity versus accuracy).
- Confusion matrix and F1-score have the problem that we do not have negative examples.
- Topics will be discussed later.
- For the moment, we only mention the **rediscovery problem** as a quality criterion.

Rediscovering process models



The rediscovery problem: Is the discovered model N' "equivalent" to the original model N?

TU/e

Part I: Preliminaries

- Chapter 1: Introduction
- Chapter 2: Process Modeling and Analysis
- Chapter 3: Data Mining

Part II: From Event Logs to Process Models

- Chapter 4: Getting the Data
- Chapter 5: Process Discovery: An Introduction
- Chapter 6: Advanced Process Discovery Techniques

Part III: Beyond Process Discovery

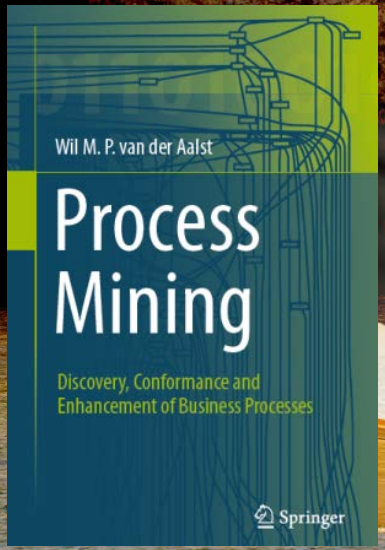
- Chapter 7: Conformance Checking
- Chapter 8: Mining Additional Perspectives
- Chapter 9: Operational Support

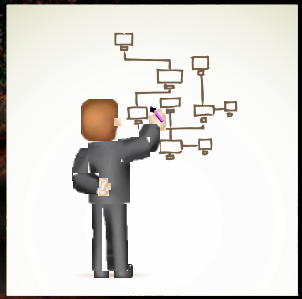
Part IV: Putting Process Mining to Work

- Chapter 10: Tool Support
- Chapter 11: Analyzing “Lasagna Processes”
- Chapter 12: Analyzing “Spaghetti Processes”

Part V: Reflection

- Chapter 13: Cartography and Navigation
- Chapter 14: Epilogue





TU/e