Christian-Albrechts-Universität zu Kiel

Advanced Process Mining

Prof. Dr. Agnes Koschmider

Lecture 2: Directly-Follows Graph & Heuristic Miner

Process Analytics

Lecture Overview

Christian-Albrechts-Universität zu Kiel

0	Organization and Introduction
I	Process Discovery
II	Process Conformance
III	Predictive Process Mining
IV	Event Log Preparation
V	Practical Tasks

Simplified Event Log

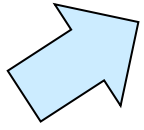
Christian-Albrechts-Universität zu Kiel

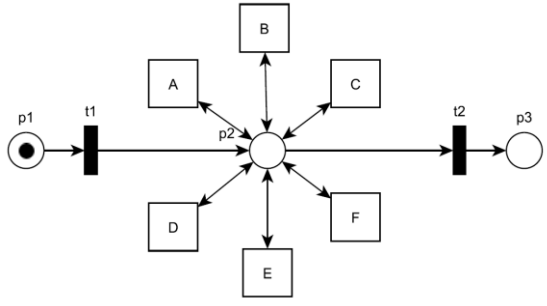
Patient	Activity	Time	Staff	Further Attributes
P1	A	0	S1	…
P1	B	5	S2	…
P1	C	5	S3	…
P2	A	15	S2	…
P2	B	7	S3	…
P2	C	10	S2	…
P3	A	15	S2	…
P3	B	7	S3	…
P4	A	10	S1	…
P4	D	10	S3	…
P5	A	0	S4	…

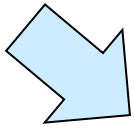
Accuracy of Automated Process Discovery

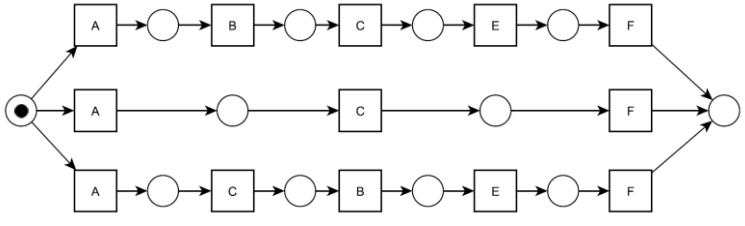
Christian-Albrechts-Universität zu Kiel

Patient	Activity	Time	Staff	Further Attributes
P1	A	0	S1	...
P1	B	5	S2	...
P1	C	5	S3	...
P2	A	15	S2	...
P2	B	7	S3	...
P2	C	10	S2	...
P3	A	15	S2	...
P3	B	7	S3	...
P4	A	10	S1	...
P4	D	10	S3	...
P5	A	0	S4	...









Challenges of Process Discovery

Christian-Albrechts-Universität zu Kiel

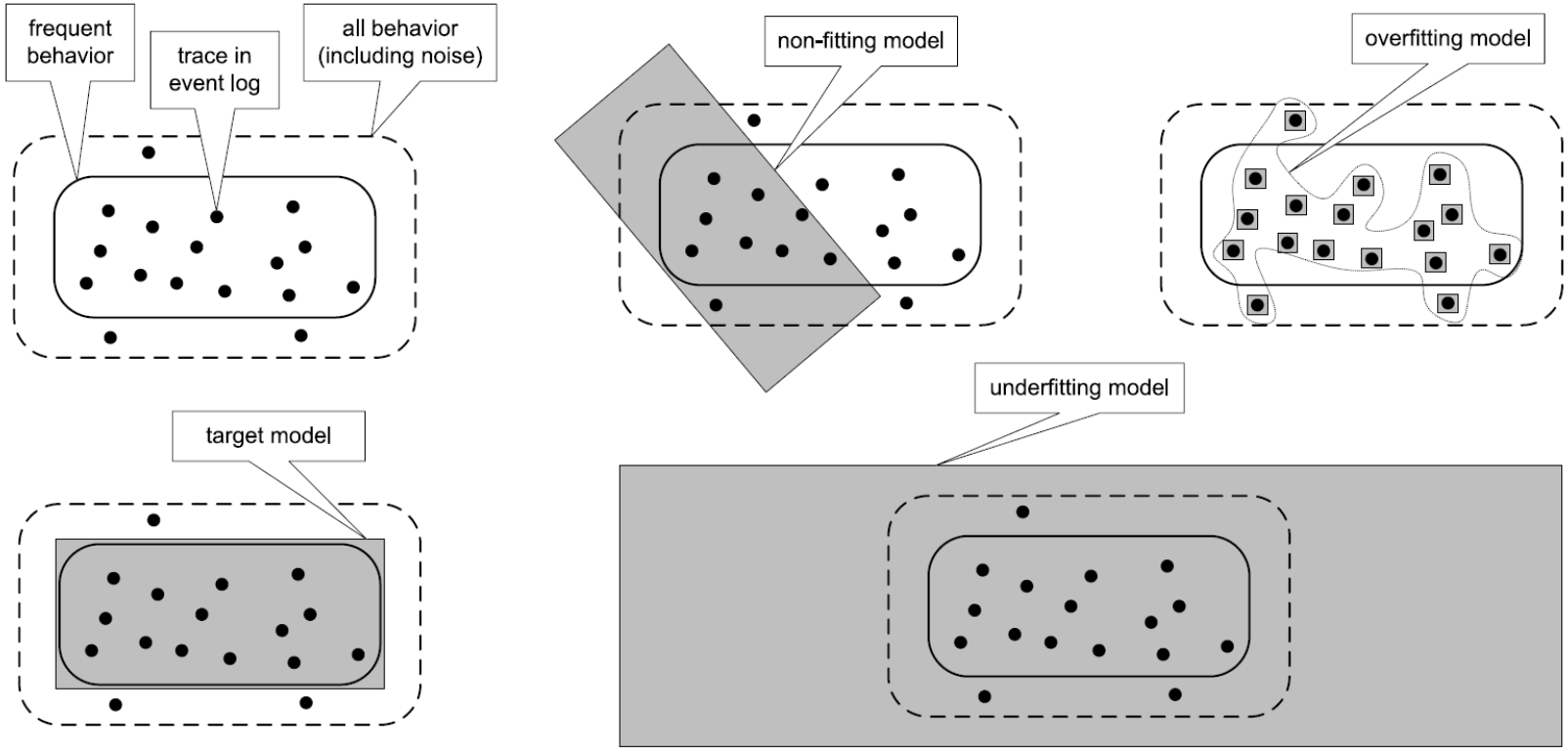
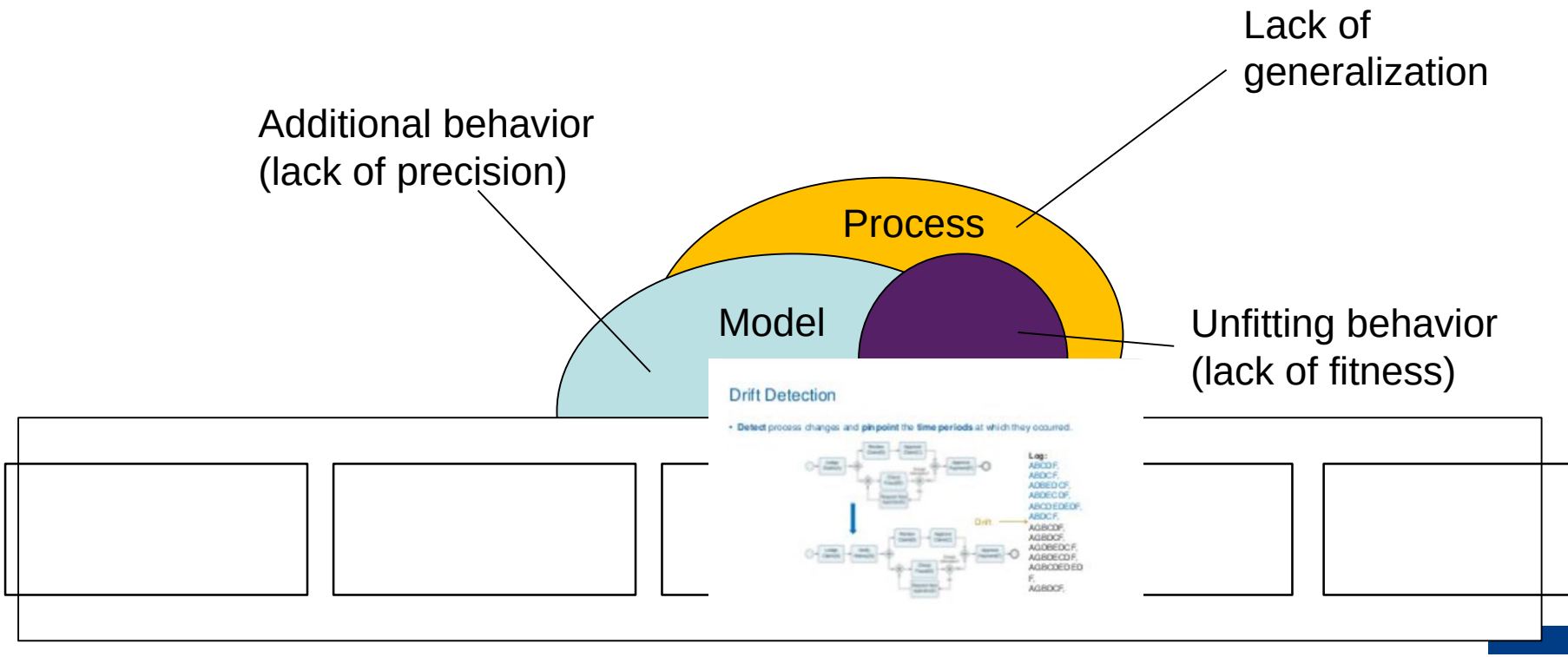


Fig. 6.1 Overview of the challenges that process discovery techniques need to address

Source: Process Mining – Discovery, Conformance and Enhancement of Business Processes, van der Aalst, Wil M. P., Springer 2011, pp. 158

Accuracy of Automated Process Discovery

Christian-Albrechts-Universität zu Kiel



Quality of Discovered Process Models

Christian-Albrechts-Universität zu Kiel

- **Recall** (a.k.a Fitness): behavior recorded in the event log reproducible by the model
- **Precision:** behavior producible by the model observed in the event log
- **F-Score:** (2* fitness * precision) / (fitness + precision)
- **Generalization:** behaviour producible by the model eventually recorded in the event log
- **Complexity:** a set of metrics estimating simplicity and understandability of the process model

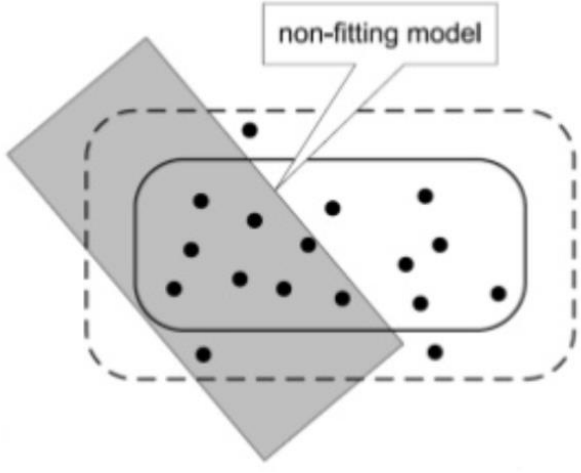
Accuracy of automatically discovered process models

Christian-Albrechts-Universität zu Kiel

- The accuracy of an automatically discovered process model consists of four quality dimensions:

  1. Fitness: the discovered model should allow for the behavior seen in the event log

  - A model has a perfect fitness if all traces in the log can be replayed from the beginning to the end.



Measuring Fitness

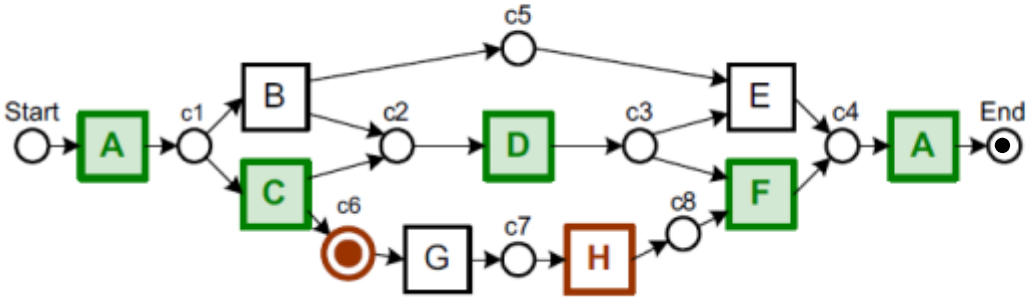
- Replay
  - Replay each trace against the model
  - When a parsing error occurs, repair it locally
  - Keep track of the “parsing error“
  - Does not calculate an exact distance measure

- Optimal Trace Alignment
  - For each trace in the model *t*, find the trace *t‘* of the process model such that the string-edit distance of the *t* and *t‘* is minimal

Measuring Fitness

- Token Replay -

Christian-Albrechts-Universität zu Kiel



$m = 1$
$r = 1$
$c = 8$
$p = 8$

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

- $Fitness_{\mathcal{F}}(\sigma, \mathcal{N}) = 0$
- $Fitness_{\mathcal{E}}(\sigma, \mathcal{N}) = \frac{1}{2}\left(1 - \frac{1}{8}\right) + \frac{1}{2}\left(1 - \frac{1}{8}\right) = 0.875$

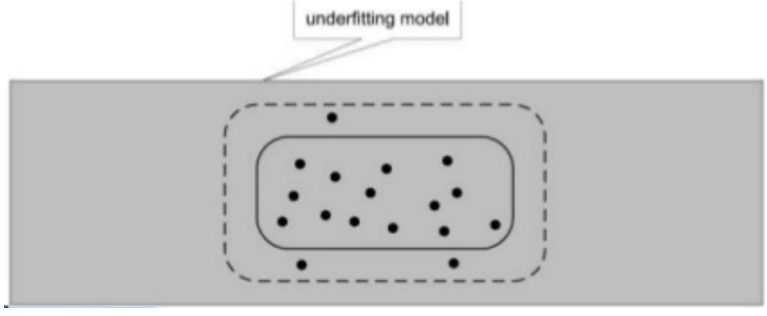
Accuracy of automatically discovered process models

Christian-Albrechts-Universität zu Kiel

- The accuracy of an automatically discovered process model consists of four quality dimensions:

  1. Fitness: the discovered model should allow for the behavior seen in the event log

  2. Precision (avoid underfitting): the discovered model should not allow for behavior completely unrelated to what was seen in the event log.



Accuracy of automatically discovered process models

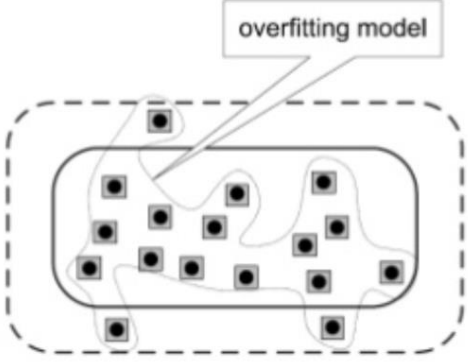
Christian-Albrechts-Universität zu Kiel

- The accuracy of an automatically discovered process model consists of four quality dimensions:

  1. Fitness: the discovered model should allow for the behavior seen in the event log

  2. Precision (avoid underfitting): the discovered model should not allow for behavior completely unrelated to what was seen in the event log.

  3. Generalization (avoid overfitting): the discovered model should generalize the example behavior seen in the event log.



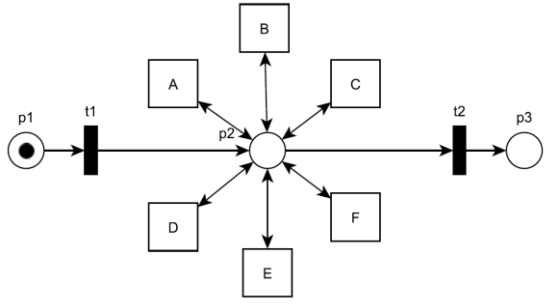
Accuracy of automatically discovered process models

Christian-Albrechts-Universität zu Kiel

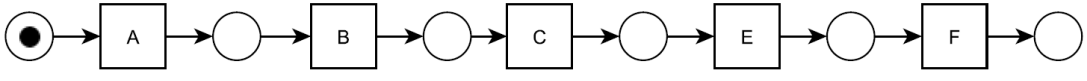
- Even if there are different process models on which all traces of an event log can be played (fitness = 1), not all of these models are good

Let us assume the traces:
$\langle A,B,C,E,F\rangle^{100}$; $\langle A,D,F\rangle^{30}$
$\langle A,C,B,E,F\rangle^{50}$

a)

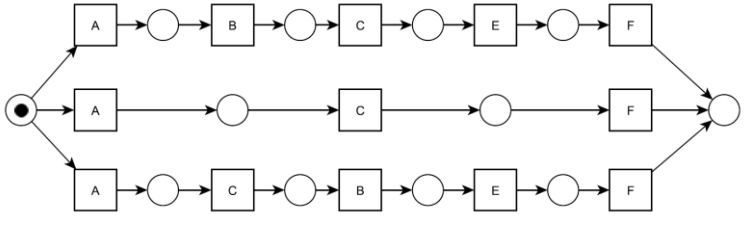


b)



How is the **precision** and **fitness** of a),b) and c) ??

c)



Or something in the middle?

Calculating Precision

Christian-Albrechts-Universität zu Kiel

- Precision = 1 → the behavior allowed by the model is contained or equal to the behavior in the log
- Precision close to 0 → none of the behavior in the model is observed in the log
- Precision can be calculated as a “difference“ between a state space representing the behvior of the model, and a state space representing the behavior of the log

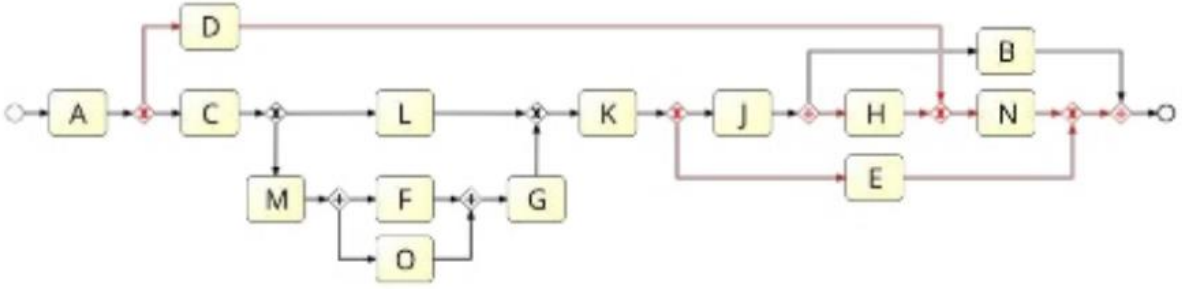
source: A. Augusto et al.: Abstract-and-compare: A Family of Scalable Precision Measures for Automated Process Discovery. In: BPM 2018, Springer

Process Discovery Methods

Christian-Albrechts-Universität zu Kiel

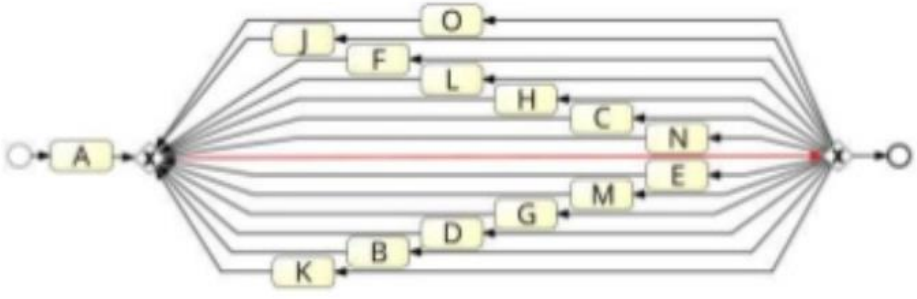
Heuristics Miner

good F-score
complex
semantic errors
tuning complexity



Inductive Miner

high fitness
no semantic errors
simple
low precision



Input requirements for process discovery algorithms

Christian-Albrechts-Universität zu Kiel

Patient	Activity	Time	Staff	Further Attributes
P1	A	0	S1	…
P1	B	5	S2	…
P1	C	5	S3	…
P2	A	15	S2	…
P2	B	7	S3	…
P2	C	10	S2	…
P3	A	15	S2	…
P3	B	7	S3	…
P4	A	10	S1	…
P4	D	10	S3	…
P5	A	0	S4	…

Sequence	Frequency
<A,B,C>	2
<A,B>	1
<A,D>	2
<A>	2

Event/Activity Sequence

Relation	Frequency
(-,A)	4
(A,B)	3
(A,D)	1
(B,C)	2
(A,-)	1
(B,-)	1
(C,-)	2
(D,-)	1

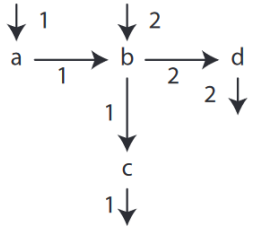
Directly-follows Relations

Directly-Follows-Graph

Christian-Albrechts-Universität zu Kiel

- Recognize "process as a whole“
- It can be used to graphically display all process variants recorded in the event log
- A directly-follows graph can be derived from a log and describes what activities follow one another directly, and with which activities a trace starts or ends.
- In a directly-follows graph, there is an edge from an activity a to an activity b if a is followed directly by b
- The weight of an edge denotes how often that happened

[<a,b,c>,<b,d>^{2}]



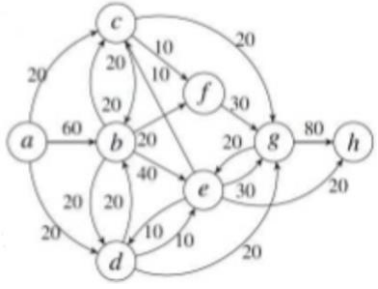
Dependency Graph / Process Maps

A dependency graph of a log is a graph where:

- Each activity is represented by one node
- An arc from activity A to activity B means that B is directly followed by A in at least one trace in the log

Arcs in a dependency graph may be annotated with:

- Absolute frequency: how many times B directly follows A?
- Relative frequency: what percentage of times A is directly followed by B?
- Time: what is the average time between the occurence of A and the occurrence of B?



Directly-Follows-Graph

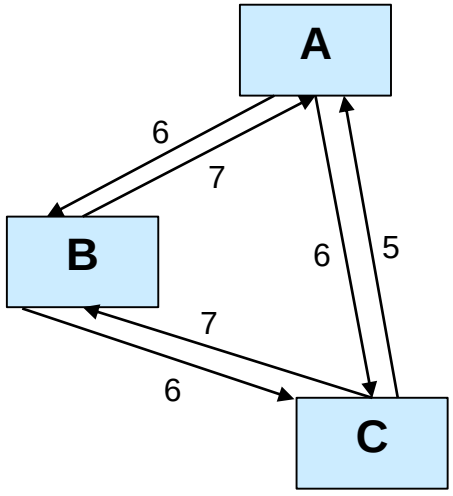
Christian-Albrechts-Universität zu Kiel

ABCABCBACBACABCBACBACABCBACABCABCBACBA

AB BA

BC CB

CA AC

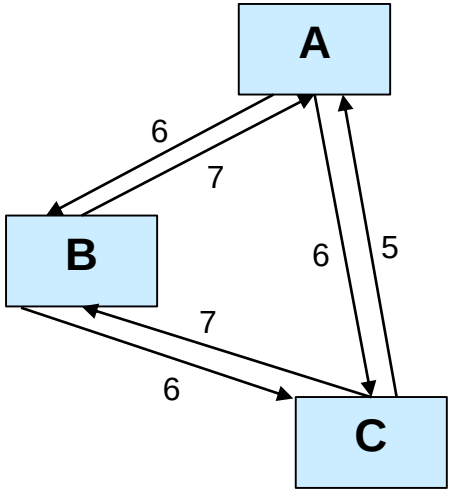


Directly-Follows-Graph

Christian-Albrechts-Universität zu Kiel

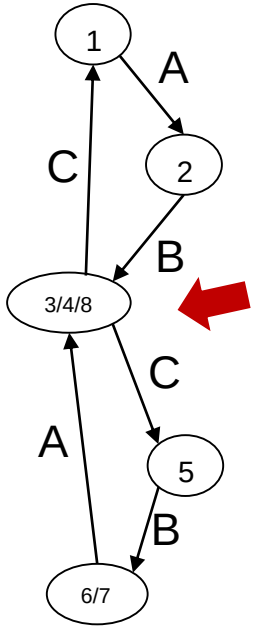
Learning Automata

ABCABCBACBACABCBACBACABCBACABCABCBACBA



DFG

mining =
find structure
in these relations



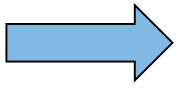
K-Trails

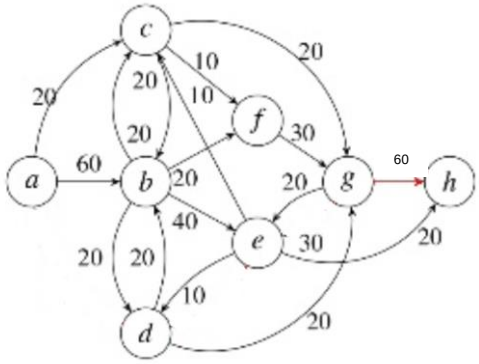
state = "sequences of next k activities"
[Cook, Wolf 1995-1998], [Cohen, Maoz 2014]

Directly-Follows-Graph

Christian-Albrechts-Universität zu Kiel

Trace	#
a b c g e h	10
a b c f g h	10
a b d g e h	10
a b d e g h	10
a b e c g h	10
a b e d g h	10
a c b e g h	10
a c b f g h	10

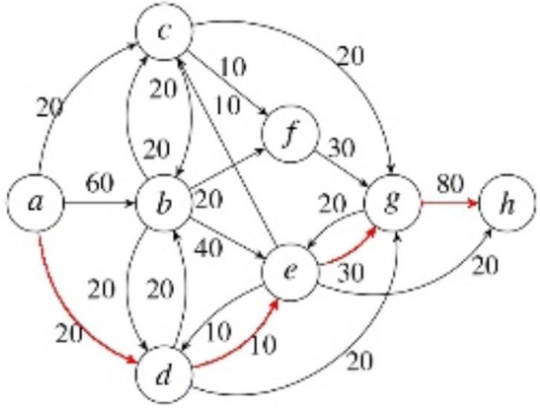




Directly-Follows-Graph

Christian-Albrechts-Universität zu Kiel

Trace	#
a b c g e h	10
a b c f g h	10
a b d g e h	10
a b d e g h	10
a b e c g h	10
a b e d g h	10
a c b e g h	10
a c b f g h	10
a d b e g h	10
a d b f g h	10



Write the dependency graph (process map) on the following event log (include absolute frequency)

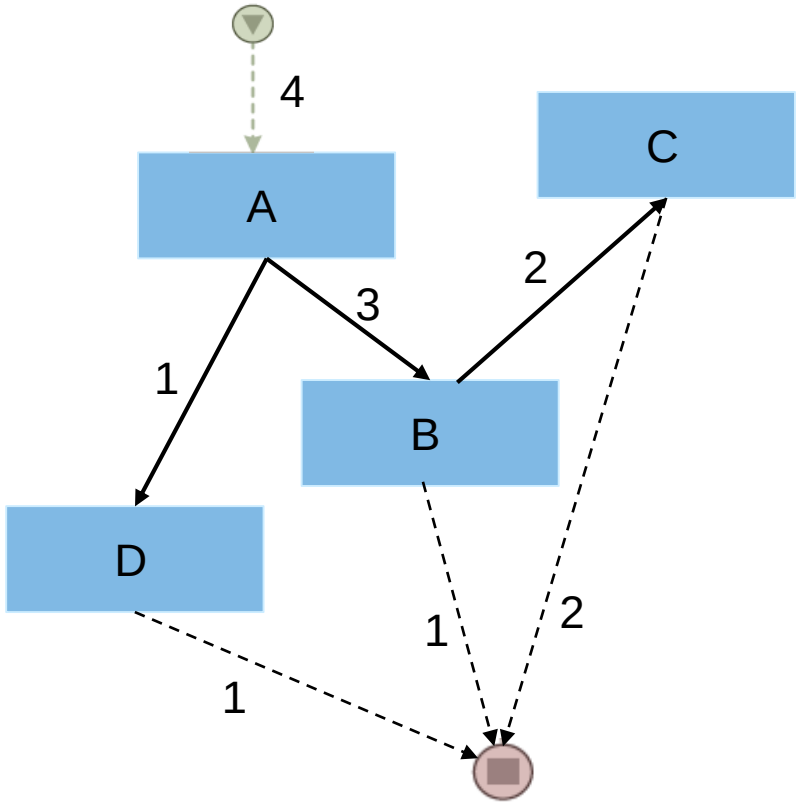
Christian-Albrechts-Universität zu Kiel

CaseID	Task Label	Timestamp
1	Create Fine	19-04-2019
2	Create Fine	19-04-2019
1	Send Bill	19-04-2019
2	Send Bill	19-04-2019
3	Create Fine	20-04-2019
3	Send Bill	20-04-2019
4	Create Fine	21-04-2019
4	Send Bill	21-04-2019
1	Process Payment	24-04-2019
1	Close Case	24-04-2019
2	Send Reminder	19-04-2019
3	Send Reminder	20-05-2019
2	Process Payment	22-05-2019
2	Close Case	22-05-2019
4	Send Reminder	21-05-2019
4	Send Reminder	21-05-2019
4	Process Payment	26-05-2019
4	Close Case	26-05-2019
3	Send Reminder	20-06-2019
3	Send Reminder	20-07-2019
3	Process Payment	25-07-2019
3	Close Case	25-07-2019

Directly-Follows Graph

Christian-Albrechts-Universität zu Kiel

Relation	Frequency
(-,A)	4
(A,B)	3
(A,D)	1
(B,C)	2
(A,-)	1
(B,-)	1
(C,-)	2
(D,-)	1



- describes what activities follow one another directly, and with which activities a trace starts or ends

Abstraction and Filtering

- To cope with the complexity of large real-life logs, process maps are often used in conjunction with two operations:

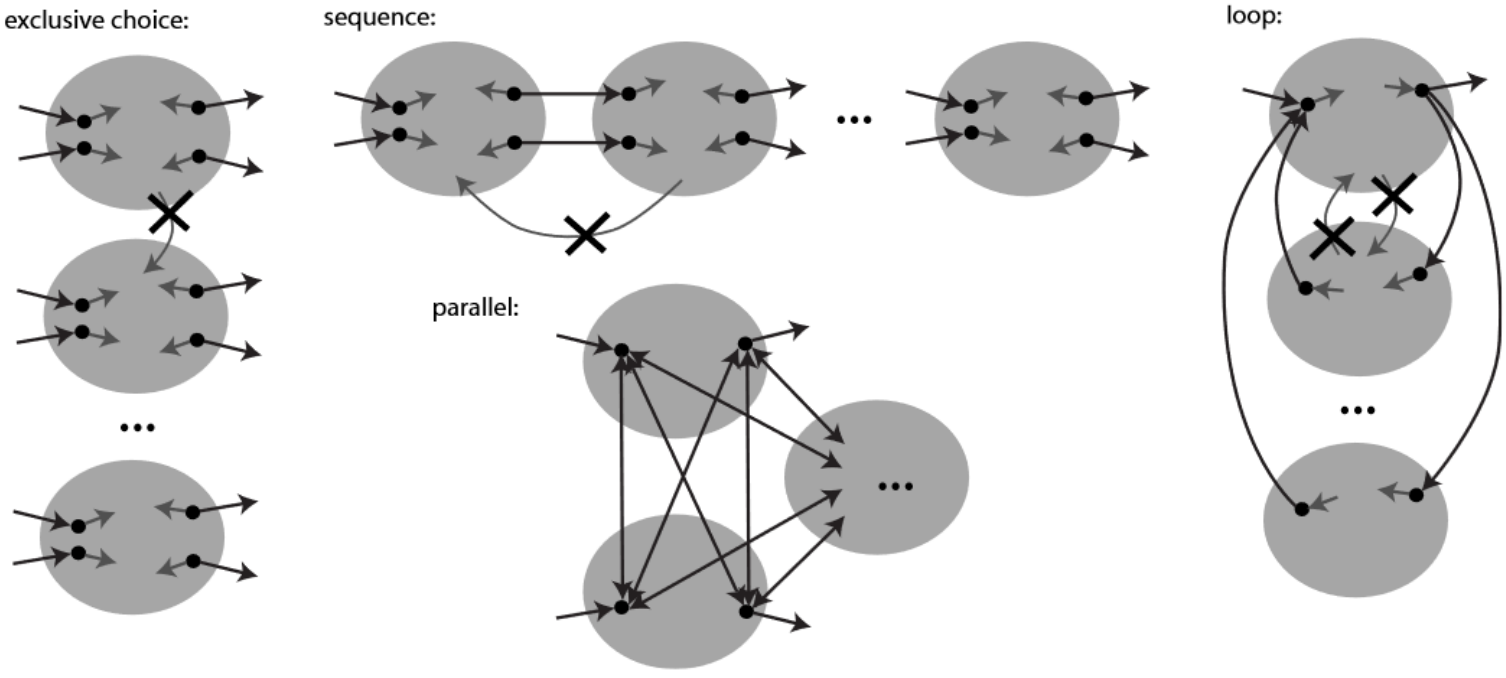
1. Abstract the process map:

- show only most frequent activities
- show only most frequent arcs

2. Filter the traces in the event log

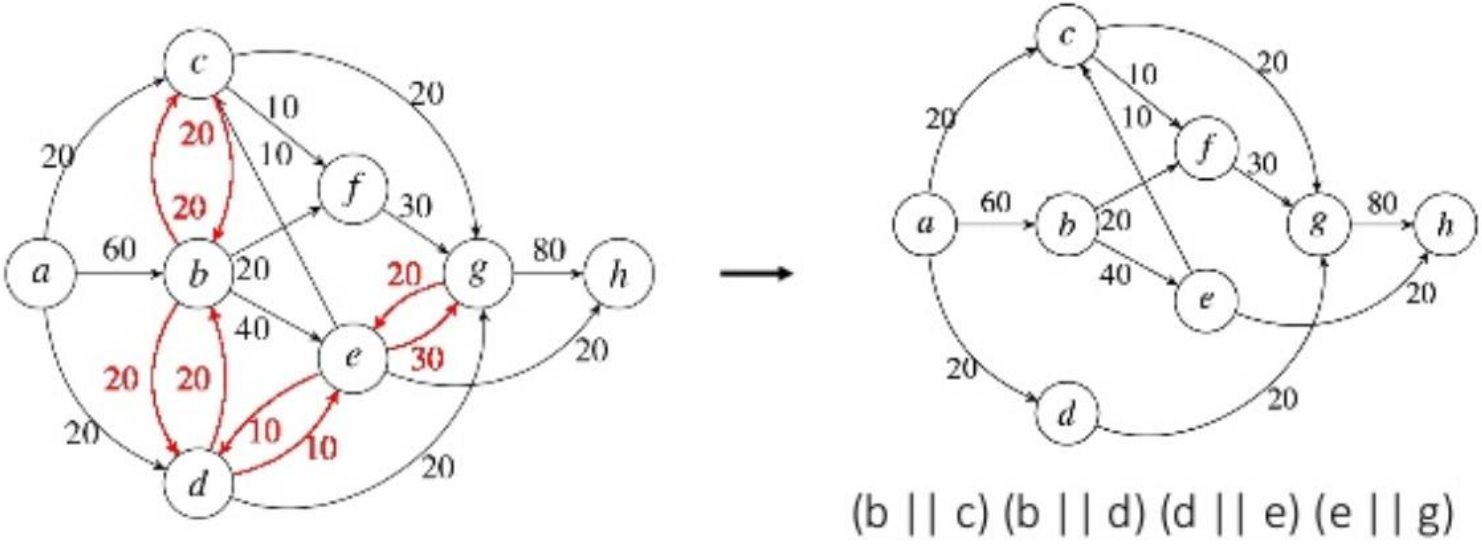
- remove all events that fulfil a condition
- remove traces that fulfil a condition (or traces that do not fulfil a condition)

Cut characteristics



Cut characteristics

Christian-Albrechts-Universität zu Kiel

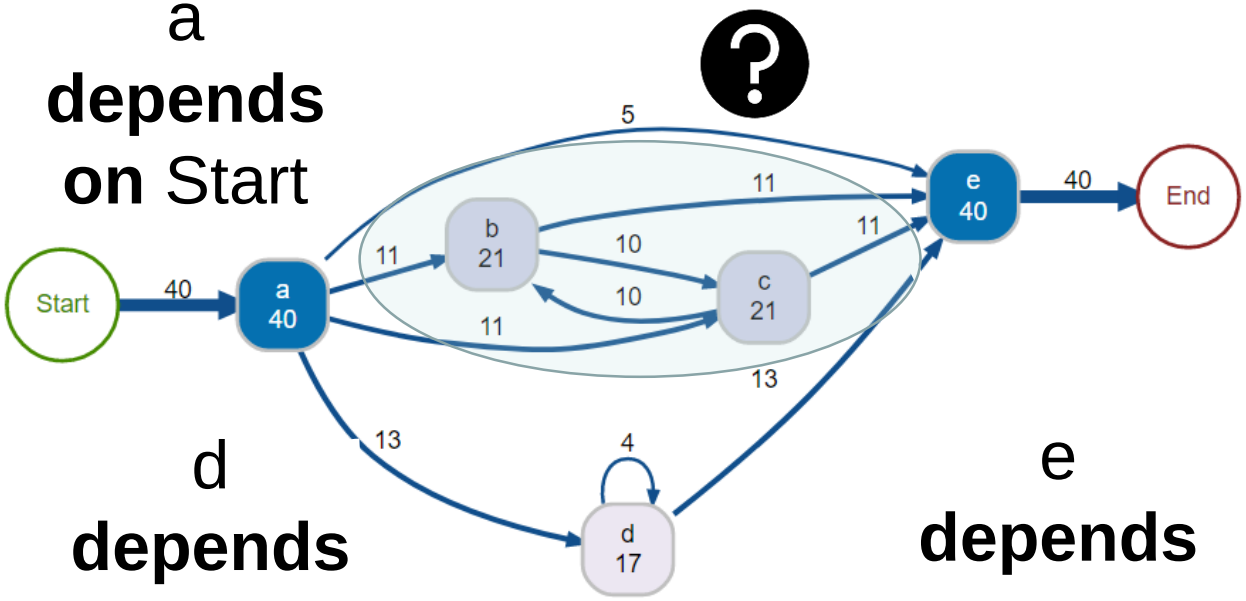


Heuristic Miner

Christian-Albrechts-Universität zu Kiel

- attempts to capture **'causal' dependencies** instead of precedes/follows relations
- focuses on calculating dependency frequency

a **depends on** Start



d **depends on** a

e **depends on** {a,b,c,d}

Heuristic Miner

Christian-Albrechts-Universität zu Kiel

Dependency Measure

$$a \Rightarrow^L b = \begin{cases} \dfrac{|a >^L b| - |b >^L a|}{|a >^L b| + |b >^L a| + 1}, & \text{for } a \neq b \\ \dfrac{|a >^L a|}{|a >^L a| + 1} & \text{otherwise} \end{cases}$$

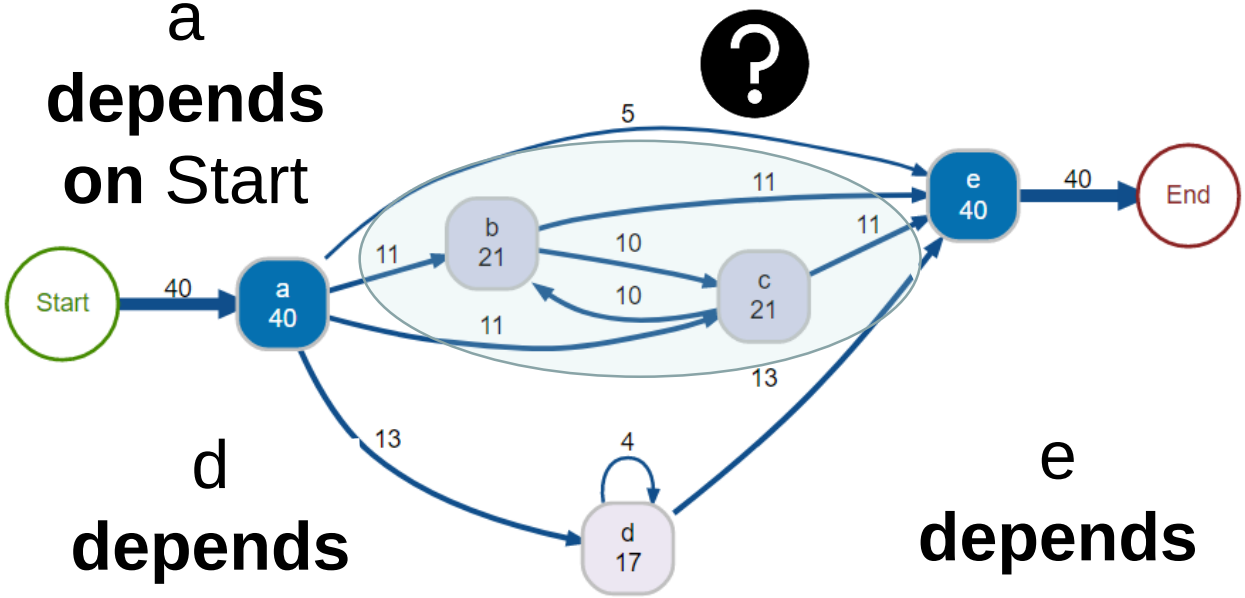
L	Event log
a and **b**	Activities / activity identifiers
$a >^L b$	a directly-follows b
$\lvert \rvert$	Frequency / cardinality

Application of Dependency Measure

Christian-Albrechts-Universität zu Kiel

$$a \Rightarrow^L b = \begin{cases} \dfrac{|a >^L b| - |b >^L a|}{|a >^L b| + |b >^L a| + 1}, & \text{for } a \neq b \\ \dfrac{|a >^L a|}{|a >^L a| + 1} & \text{otherwise} \end{cases}$$

a **depends on** Start



d **depends on** a

e **depends on** {a,b,c,d}

Heuristic Miner

Christian-Albrechts-Universität zu Kiel

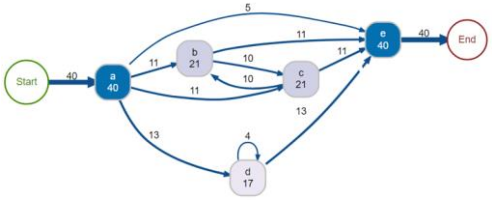
$$a \Rightarrow^L b = \begin{cases} \dfrac{|a >^L b| - |b >^L a|}{|a >^L b| + |b >^L a| + 1}, & \text{for } a \neq b \\ \dfrac{|a >^L a|}{|a >^L a| + 1} & \text{otherwise} \end{cases}$$

Intuition

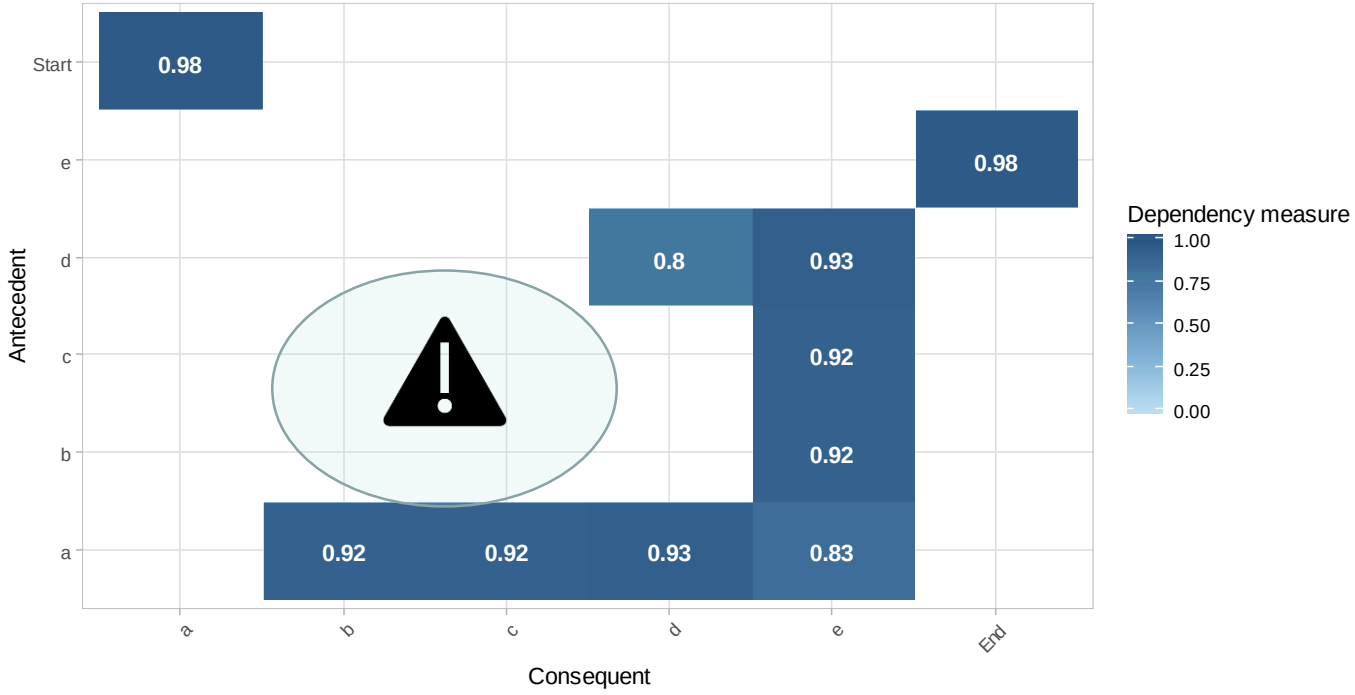
- $|a \Rightarrow^L b|$ goes towards 1 ➔ Causal
- $|a \Rightarrow^L b|$ goes towards 0 ➔ Parallel
- $|a \Rightarrow^L b|$ goes towards -1 ➔ Inverse Causal

Heuristic Miner

Christian-Albrechts-Universität zu Kiel



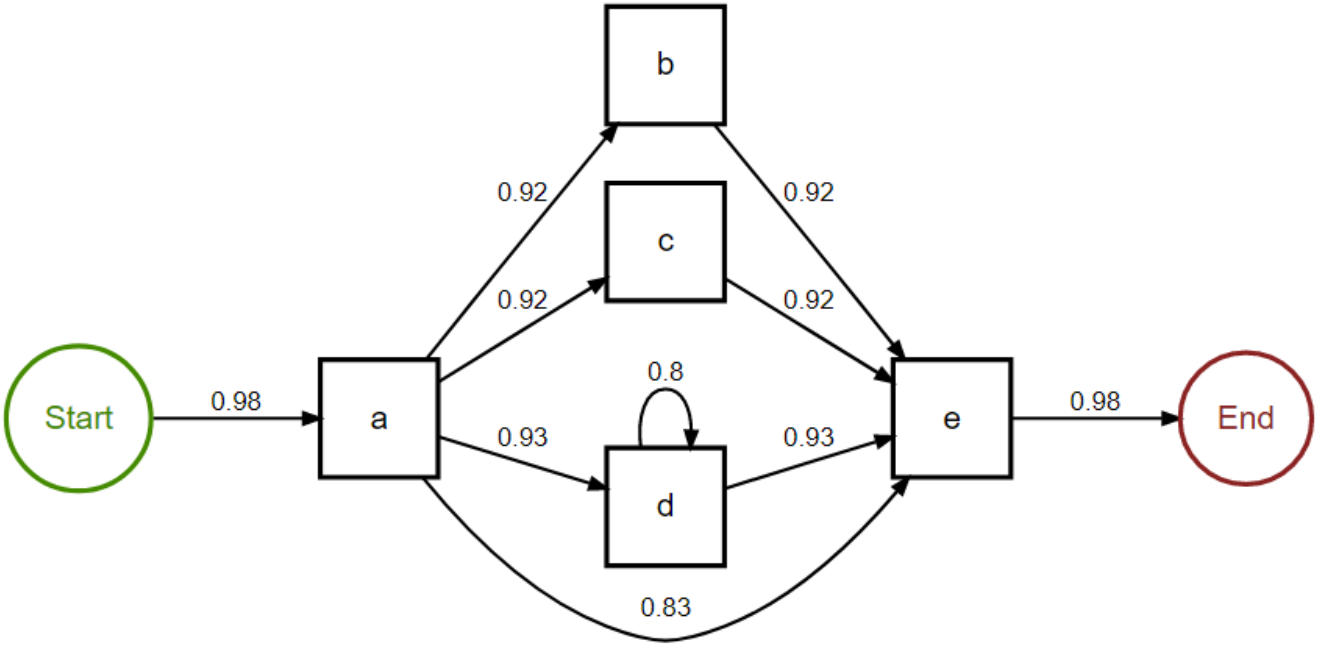
$$a \Rightarrow^{L} b = \begin{cases} \dfrac{|a >^{L} b| - |b >^{L} a|}{|a >^{L} b| + |b >^{L} a| + 1}, & \text{for } a \neq b \\ \dfrac{|a >^{L} a|}{|a >^{L} a| + 1} & \text{otherwise} \end{cases}$$



Heuristic Miner

Christian-Albrechts-Universität zu Kiel

Output – Dependency Graph



Heuristic Miner
- Example -

Christian-Albrechts-Universität zu Kiel

Let us assume L= $[\langle A,B,C,E,F\rangle^{20},\langle A,B,C,D,E,F\rangle^{10},\langle A,B,C,D,F,G\rangle^{5},\langle A,D,C,B,F,G\rangle^{2}]$

1. Construct Directly-Follows Matrix

$\mapsto_L$	A	B	C	D	E	F	G
A		35		2			
B			35			2	
C		2		15	20		
D			2		10	5	
E					30		
F							7
G							

Heuristic Miner
- Example -

Christian-Albrechts-Universität zu Kiel

Let us assume L= [<A,B,C,E,F>^{20},<A,B,C,D,E,F>^{10},<A,B,C,D,F,G>^{5},<A,D,C,B,F,G>^{2}]

2. Construct Dependency Matrix

$$a \Rightarrow^L b = \begin{cases} \dfrac{|a >^L b| - |b >^L a|}{|a >^L b| + |b >^L a| + 1}, & \text{for } a \neq b \\ \dfrac{|a >^L a|}{|a >^L a| + 1} & \text{otherwise} \end{cases}$$

$\Rightarrow^L$	A	B	C	D	E	F	G
A		0.97		0.67			
B	-0.97		0.87			0.67	
C	-0.97	-0.87		0.72	0.95		
D	-0.67		-0.72		0.91	0.83	
E			-0.95	-0.91		0.97	
F		-0.67		-0.83	-0.97		0.88
G						-0.88	

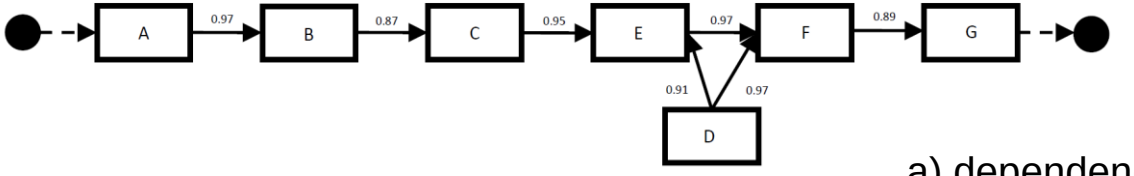
Heuristic Miner
- Example -

Christian-Albrechts-Universität zu Kiel

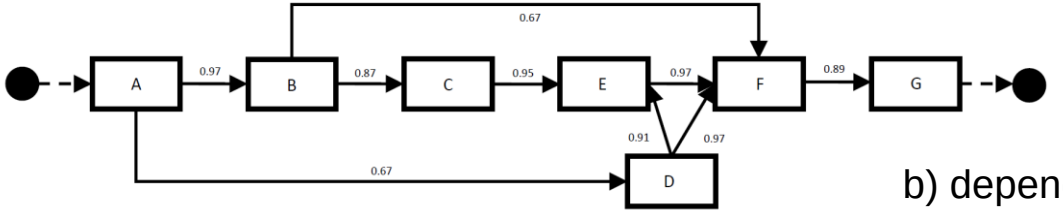
Let us assume L= [$<A,B,C,E,F>^{20}$,$<A,B,C,D,E,F>^{10}$,$<A,B,C,D,F,G>^{5}$,$<A,D,C,B,F,G>^{2}$]

3. Filter & Define threshold
   → reduce noise

$\Rightarrow^L$	A	B	C	D	E	F	G
A		0.97		0.67			
B	-0.97		0.87			0.67	
C	-0.97	-0.87		0.72	0.95		
D	-0.67		-0.72		0.91	0.83	
E			-0.95	-0.91		0.97	
F		-0.67		-0.83	-0.97		0.88
G						-0.88	



a) dependency graph with threshold [0.1,0.8]



b) dependency graph with threshold [0.1,0.65]

Heuristic Miner
- Example -

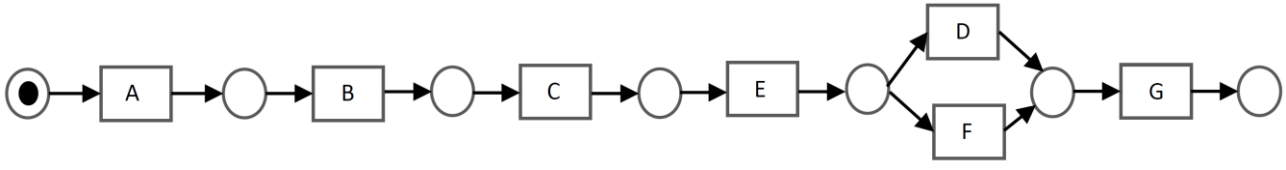
Christian-Albrechts-Universität zu Kiel

Let us assume L= [<A,B,C,E,F>^{20},<A,B,C,D,E,F>^{10},<A,B,C,D,F,G>^{5},<A,D,C,B,F,G>^{2}]

4. Construct Petri net

$\Rightarrow^L$	A	B	C	D	E	F	G
A		0.97		0.67			
B	-0.97		0.87			0.67	
C	-0.97	-0.87		0.72	0.95		
D	-0.67		-0.72		0.91	0.83	
E			-0.95	-0.91		0.97	
F		-0.67		-0.83	-0.97		0.88
G						-0.88	

$$a \mapsto_L (b \wedge c) = \left\{ \frac{|b \mapsto_L c| + |c \mapsto_L b|}{|a \mapsto_L b| + |a \mapsto_L c| + 1} \right.$$



Limitations of the Directly-Follows Graph

Christian-Albrechts-Universität zu Kiel

A practitioner's guide to process mining: Limitations of the directly-follows graph

Wil M.P. van der Aalst [a, b] ✉

Show more

https://doi.org/10.1016/j.procs.2019.12.189

Abstract

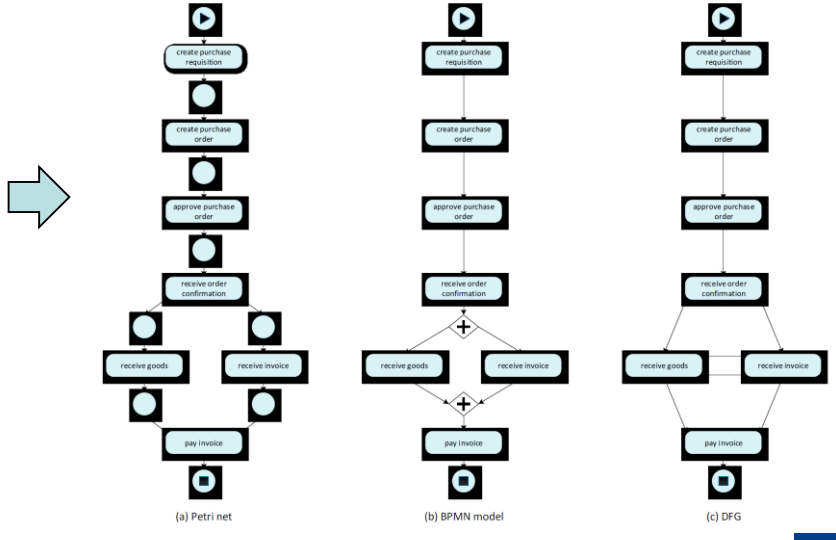
Process mining techniques use event data to show what people, machines, and organizations are really doing. Process mining provides novel insights that can be used to identify and address performance and compliance problems. In recent years, the adoption of process mining in practice increased rapidly. It is interesting to see how ideas first developed in open-source tools like ProM, get transferred to the dozens of available commercial process mining tools. However, these tools still resort to producing Directly-Follows Graphs (DFGs) based on event data rather than using more sophisticated notations also able to capture concurrency. Moreover, to tackle complexity, DFGs are seamlessly simplified by removing nodes and edges based on frequency thresholds. Process-mining practitioners tend to use such simplified DFGs actively. Despite their simplicity, these DFGs may be misleading and users need to know how these process models are generated before interpreting them. In this paper, we discuss the pitfalls of using simple DFGs generated by commercial tools. Practitioners conducting a process-mining project need to understand the risks associated with the (incorrect) use of DFGs and frequency-based simplification. Therefore, we put these risks in the spotlight.

Limitations of the Directly-Follows Graph

Christian-Albrechts-Universität zu Kiel

- to tackle complexity DFGs are seamlessly simplified by removing nodes and edges based on frequency thresholds
- DFGs may be misleading

case id (here an order)	activity	timestamp	resource	costs	customer
…	…	…	…	…	..
2019-88201	create purchase requisition	25-07-2019:09.15	John	€20.20	9950
2019-88201	create purchase order	25-07-2019:09.35	Mary	€48.30	9950
2019-88201	approve purchase order	25-07-2019:09.55	Sue	€30.70	9950
2019-88202	create purchase requisition	25-07-2019:10.15	John	€28.20	9955
2019-88202	create purchase order	25-07-2019:10.25	Mary	€29.30	9955
2019-88202	approve purchase order	25-07-2019:10.40	Sue	€37.60	9955
2019-88201	receive order confirmation	25-07-2019:11.50	Mary	€42.10	9950
2019-88201	receive goods	27-07-2019:09.35	Peter	€50.20	9950
2019-88202	receive order confirmation	27-07-2019:09.45	Mary	€42.30	9955
2019-88202	receive invoice	28-07-2019:10.10	Sue	€44.90	9955
2019-88201	receive invoice	28-07-2019:10.20	Sue	€30.80	9950
2019-88201	pay invoice	29-07-2019:11.05	Sue	€30.70	9950
2019-88202	receive goods	29-07-2019:11.35	Peter	€51.30	9955
2019-88202	pay invoice	29-07-2019:12.15	Sue	€29.20	9955
…	..	…	…	…	…



Chapter 1

Christian-Albrechts-Universität zu Kiel