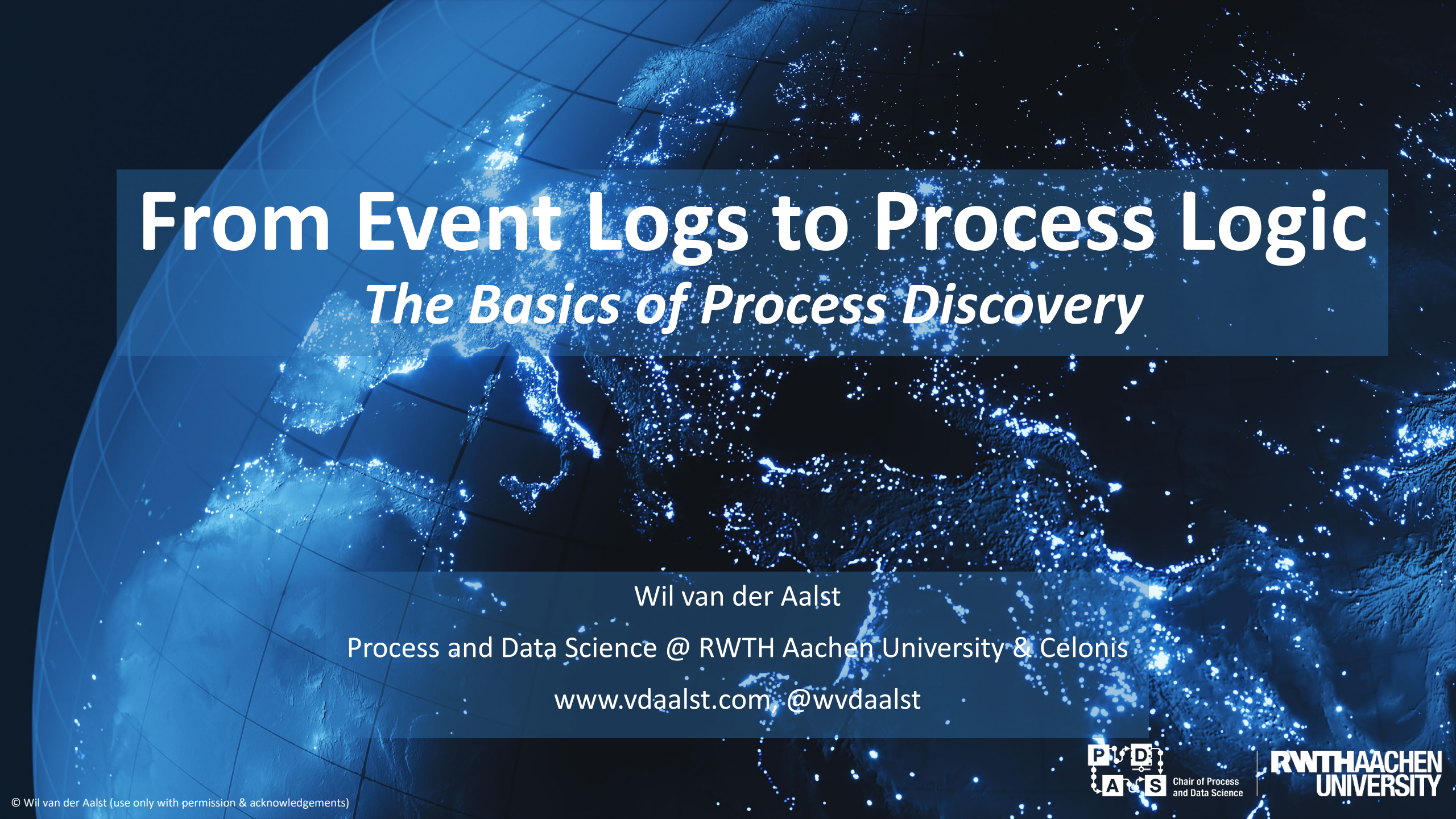


From Event Logs to Process Logic

The Basics of Process Discovery

Wil van der Aalst

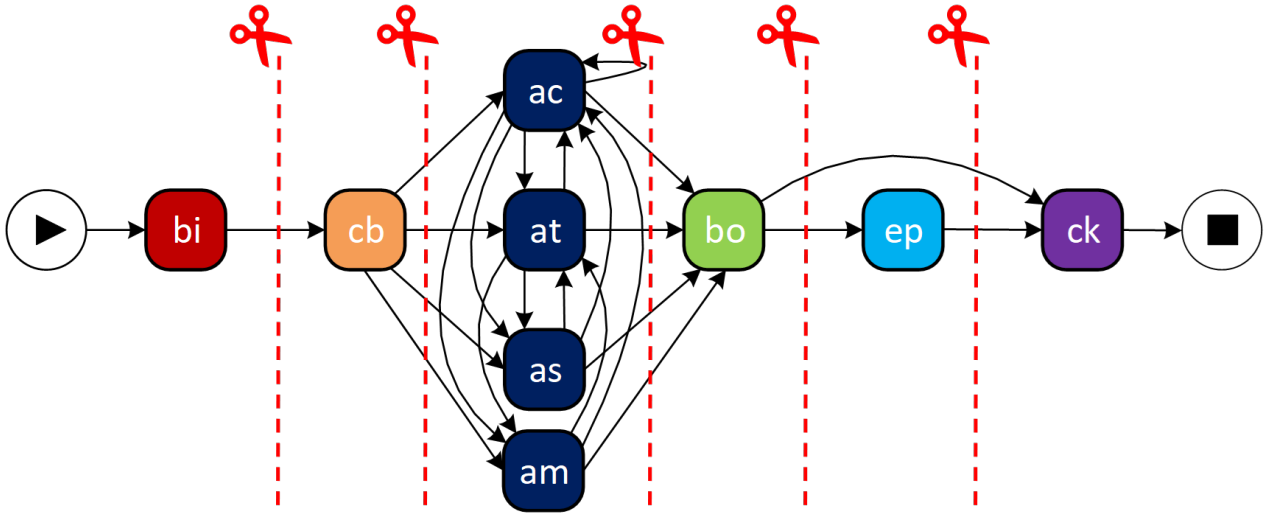
Process and Data Science @ RWTH Aachen University & Celonis

www.vdaalst.com @wvdaalst

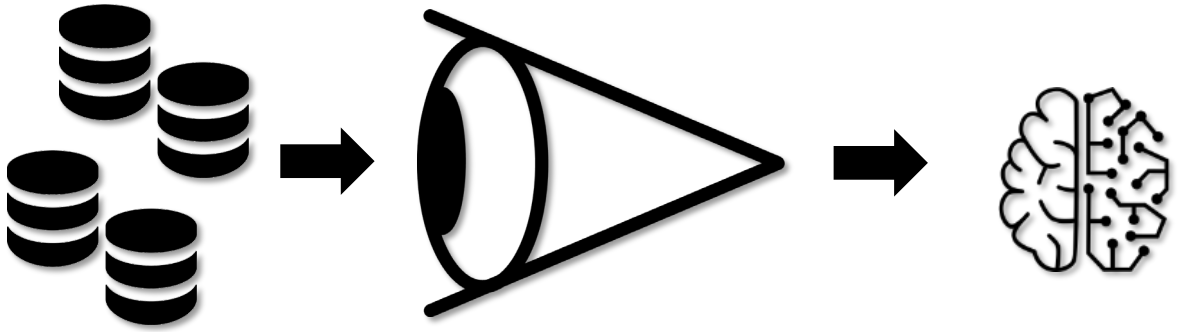
Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

About this second lecture



- ❑ Process Discovery
- ❑ Directly-Follows Graphs
- ❑ Bottom-Up Discovery: Alpha miner
- ❑ Top-Down Discovery: Inductive Mining
- ❑ “No Enterprise AI without OCPM”



complex reality | event data and process models | machine learning

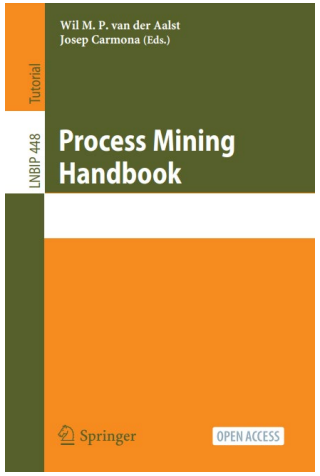
Chair of Process and Data Science

Outline: Foundations of Process Discovery

Baseline: Discovering DFG + filtering

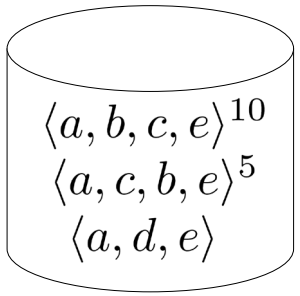
Bottom-up discovery	Top-down discovery
Alpha algorithm	Inductive mining

At times, I refer to the formal definitions in the Chapter 2 to show that with the right tools one can be precise and compact.

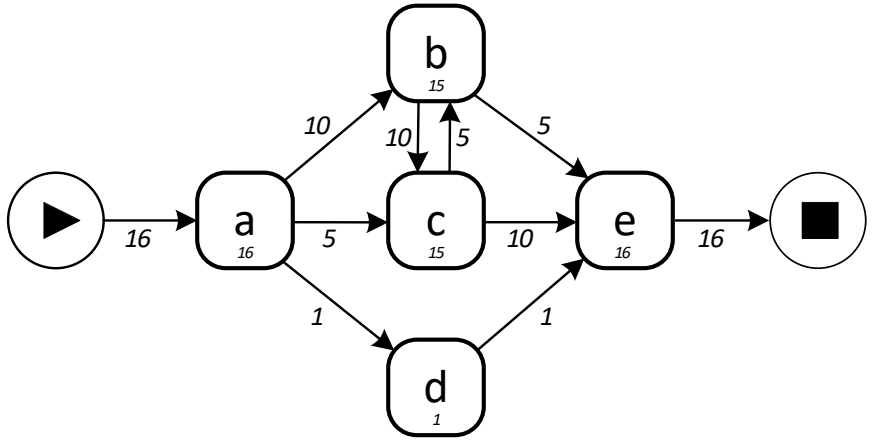


Chair of Process and Data Science

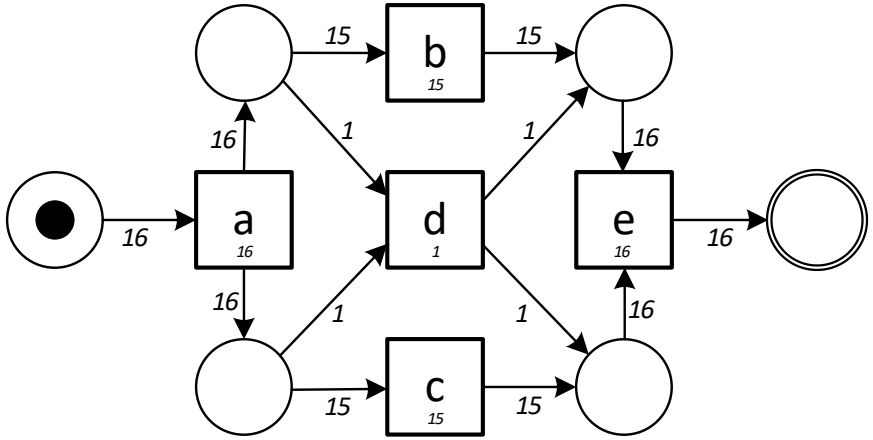
The main idea (informal)



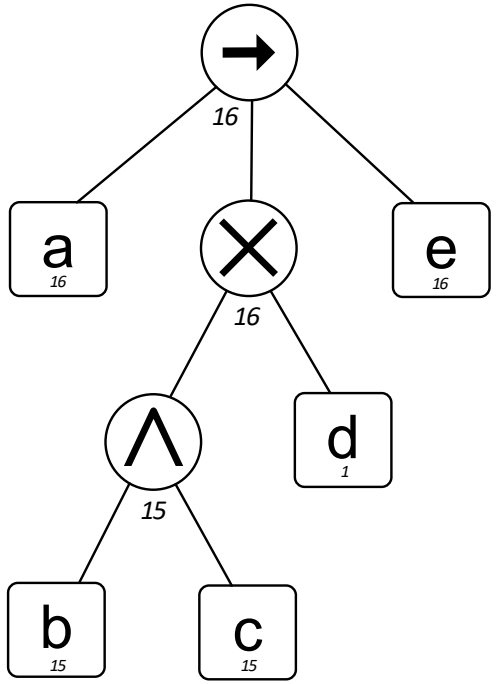
(a) Event log L_1



(b) Directly-Follows Graph (DFG): M_1



(c) Accepting Petri Net (APN): M_2



(d) Process Tree (PT): M_3

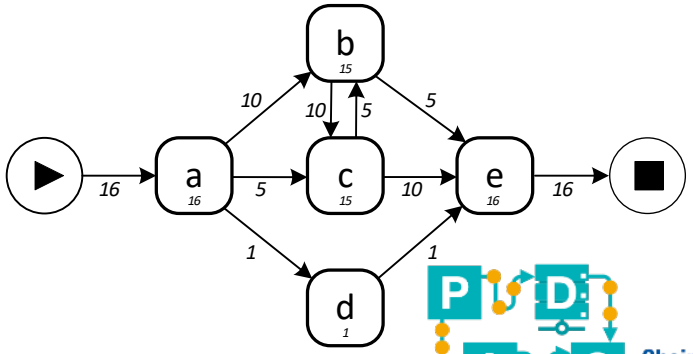
Chair of Process and Data Science

Chair of Process and Data Science
RWTH AACHEN UNIVERSITY

Baseline approach: DFGs

Definition 4 (Directly-Follows Graph). *A Directly-Follows Graph (DFG) is a pair* $G = (A, F)$ *where* $A \subseteq \mathcal{U}_{act}$ *is a set of activities and* $F \in \mathcal{B}((A \times A) \cup (\{\blacktriangleright\} \times A) \cup (A \times \{\blacksquare\}) \cup (\{\blacktriangleright\} \times \{\blacksquare\}))$ *is a multiset of arcs.* $\blacktriangleright$ *is the start node and* $\blacksquare$ *is the end node* $(\{\blacktriangleright, \blacksquare\} \cap \mathcal{U}_{act} = \emptyset)$. $\mathcal{U}_G \subseteq \mathcal{U}_M$ *is the set of all DFGs.*

- Graph with nodes representing activities and start ▶ and end ■.
- Behavior starts with dummy activity ▶ and ends with dummy activity ■. Node ▶ is a source node and ■ is a sink node.
- Arcs represent the directly-follows relation.
- Multisets to represent frequencies.
- Can be viewed as summary of the data!

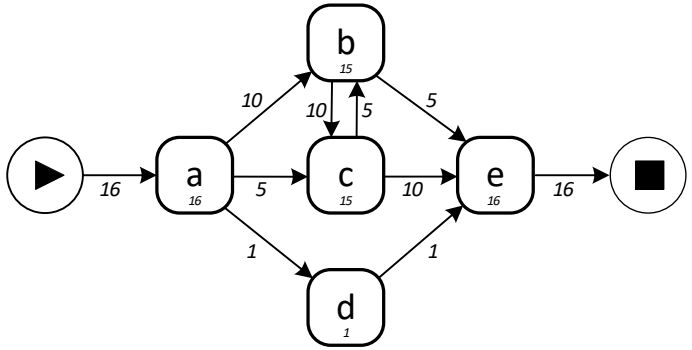


Chair of Process and Data Science

Language of a DFG

Definition 5 (Traces of a DFG). *Let* $G = (A, F) \in \mathcal{U}_G$ *be a DFG. The set of possible traces described by* G *is* $lang(G) = \{\langle a_2, a_3, \ldots, a_{n-1} \rangle \mid a_1 = \blacktriangleright \wedge a_n = \blacksquare \wedge \forall_{1 \leq i < n} (a_i, a_{i+1}) \in F\}$.

- **Possible traces: All paths possible according to the graph starting in node ▶ and ending in node ■.**
- **Recall: ▶ is a source node and ■ is a sink node.**



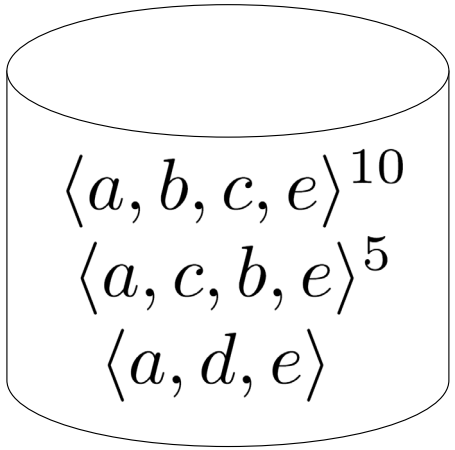
Chair of Process and Data Science

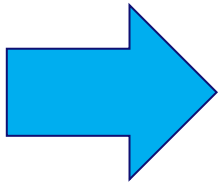
Baseline discovery

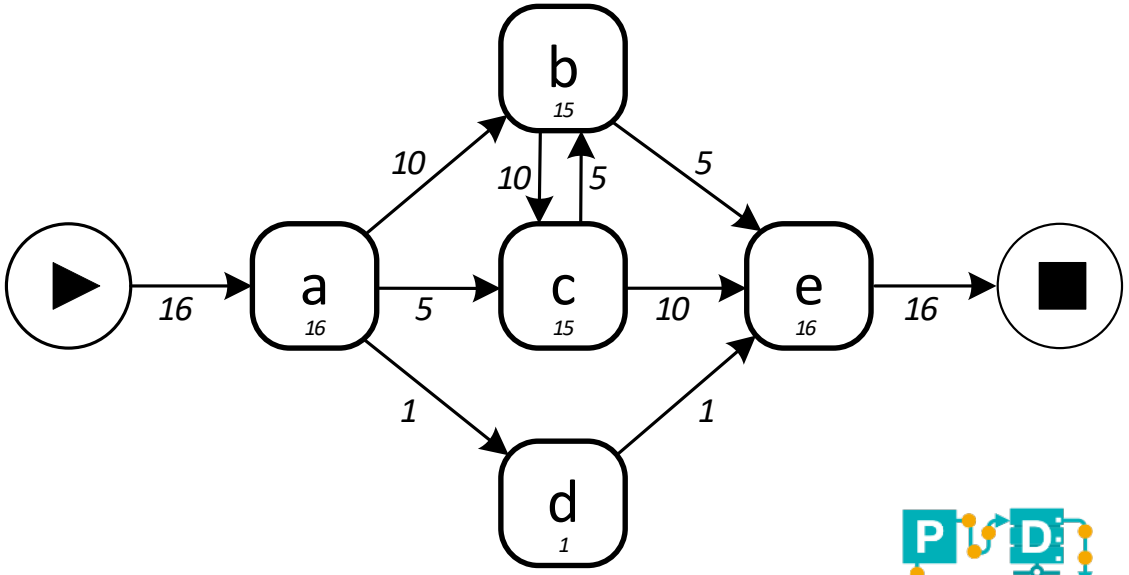
Your first discovery algorithm in just two lines of mathematics

Definition 6 (Baseline Discovery Algorithm). *Let* $L \in \mathcal{B}(\mathcal{U}_{act}{}^{*})$ *be an event log.* $disc_{DFG}(L) = (A, F)$ *is the DFG based on* L *with:*

- $A = \{a \in \sigma \mid \sigma \in L\}$ *and*
- $F = [(\sigma_i, \sigma_{i+1}) \mid \sigma \in L' \wedge 1 \leq i < |\sigma|]$ *with* $L' = [\langle \blacktriangleright \rangle \cdot \sigma \cdot \langle \blacksquare \rangle \mid \sigma \in L]$.

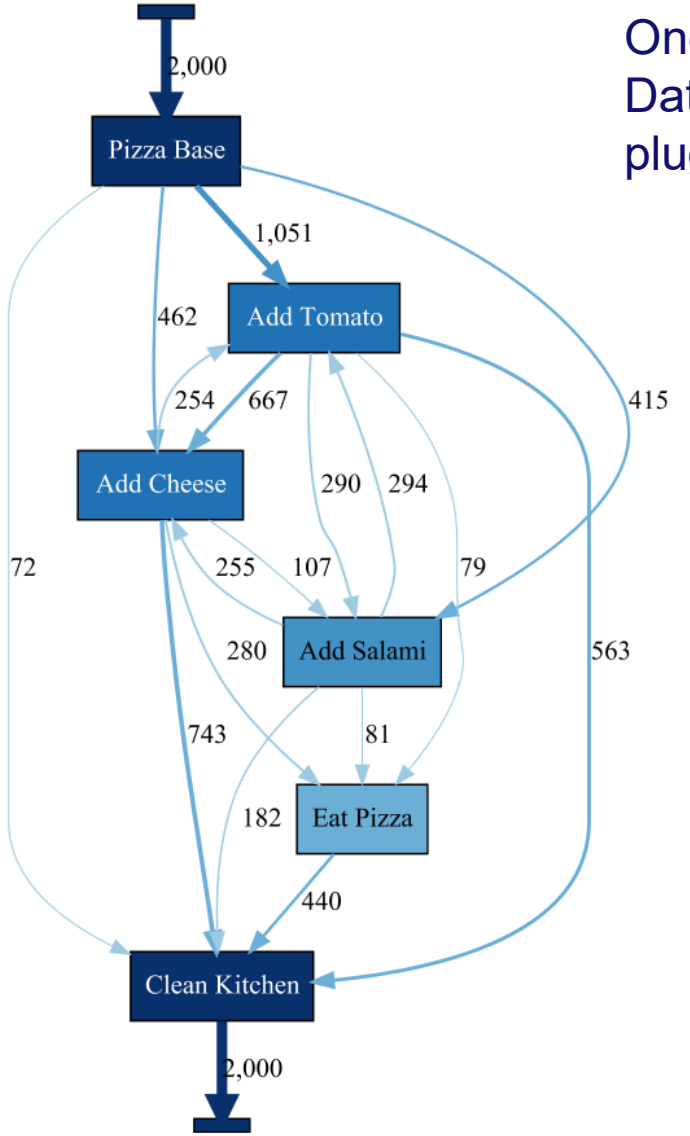






Chair of Process and Data Science

DFG discovery in ProM



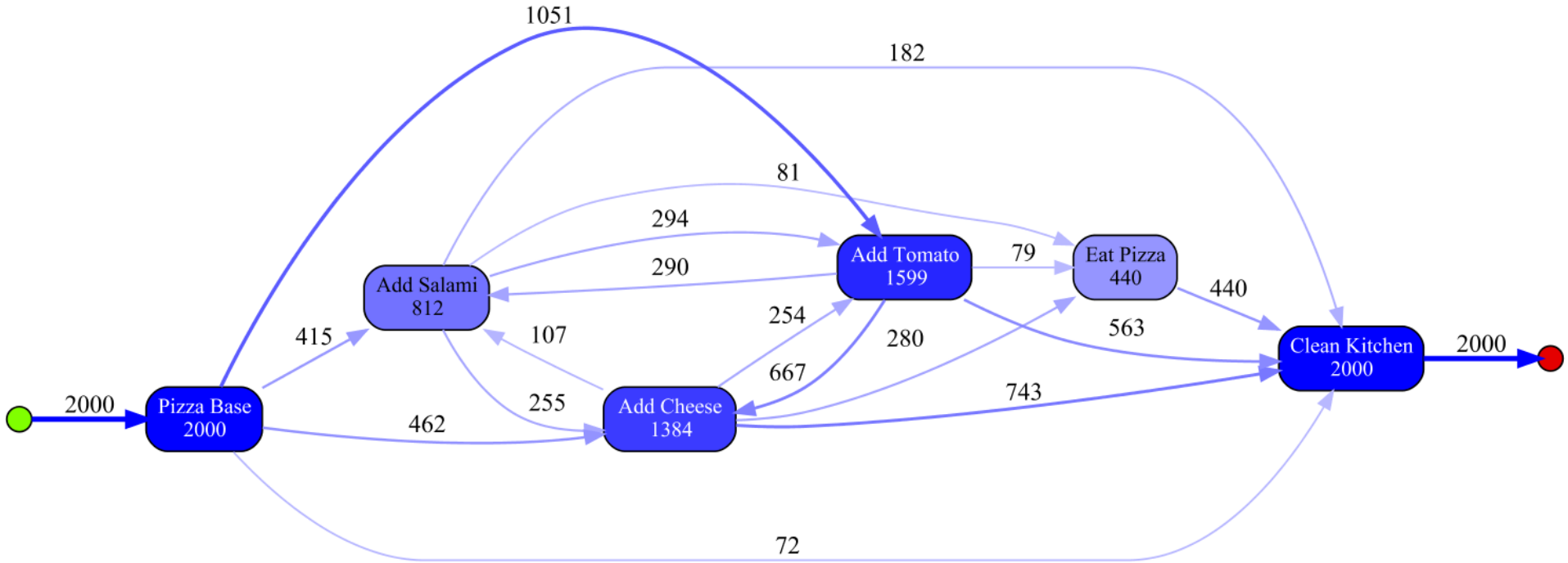
One of the views of the Data-aware heuristic miner plug-in (Felix Mannhardt)

ProM process mining workbench

Chair of Process and Data Science

DFG discovery in ProM

Directly-follows visual miner (Sander Leemans)

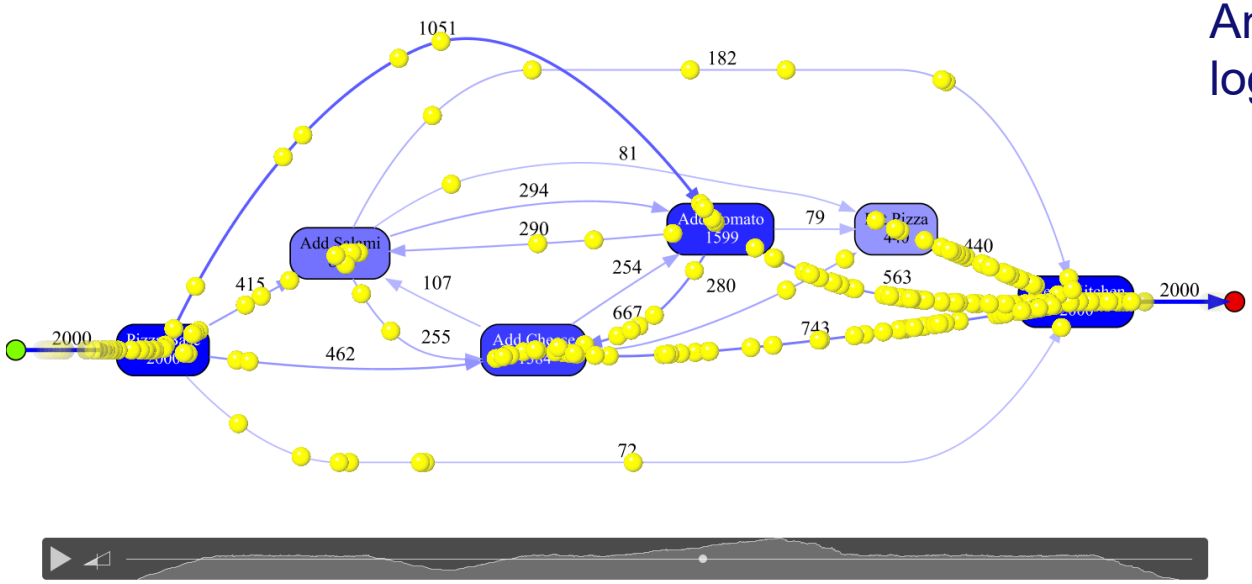


ProM process mining workbench

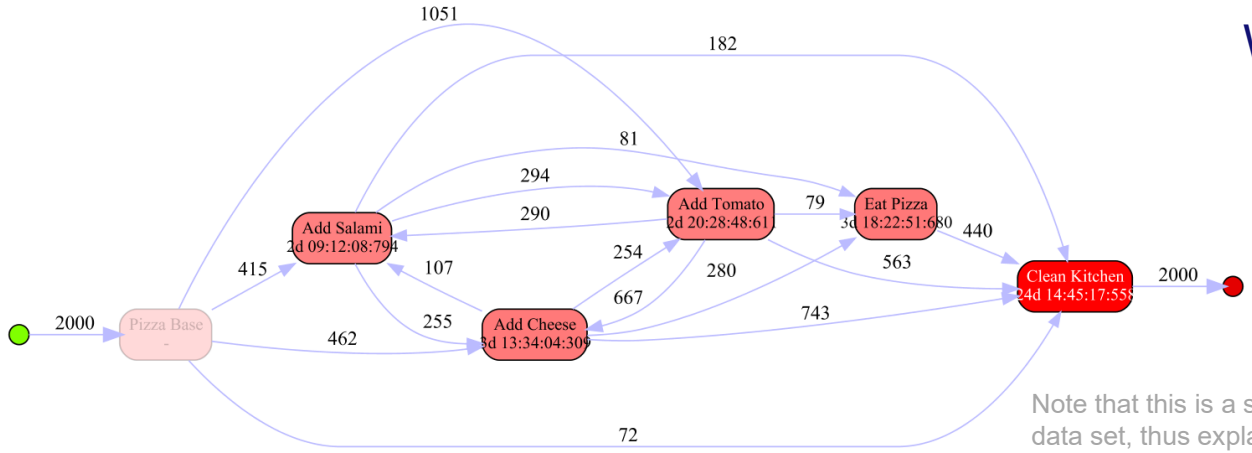
Chair of Process and Data Science

DFG discovery in ProM

Animation of the event log on top of the model



Waiting times

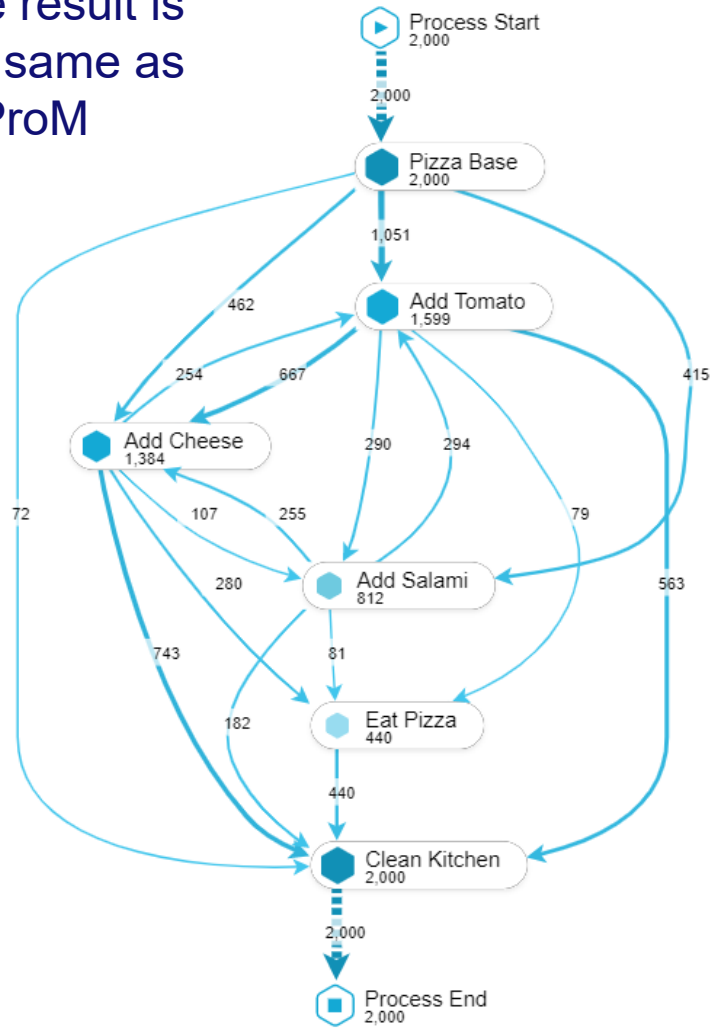


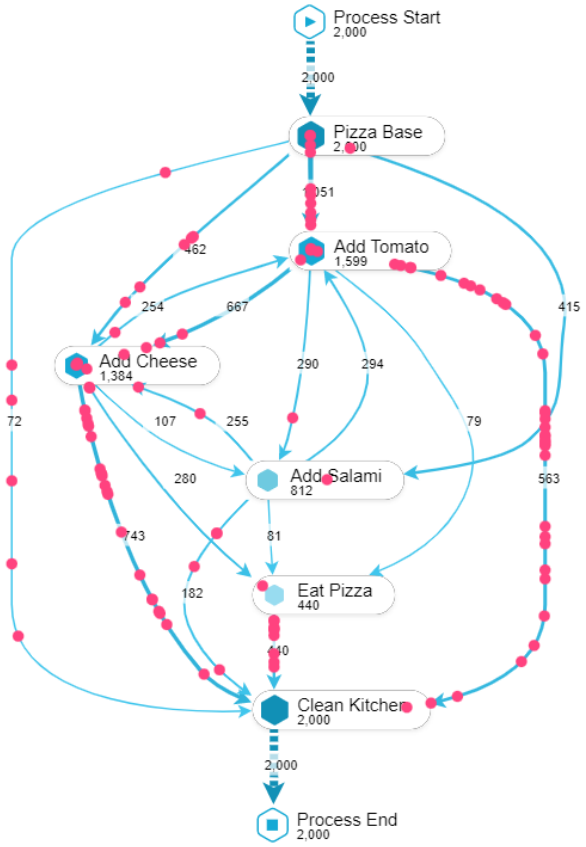
Note that this is a synthetic data set, thus explaining the long delays.

Chair of Process and Data Science

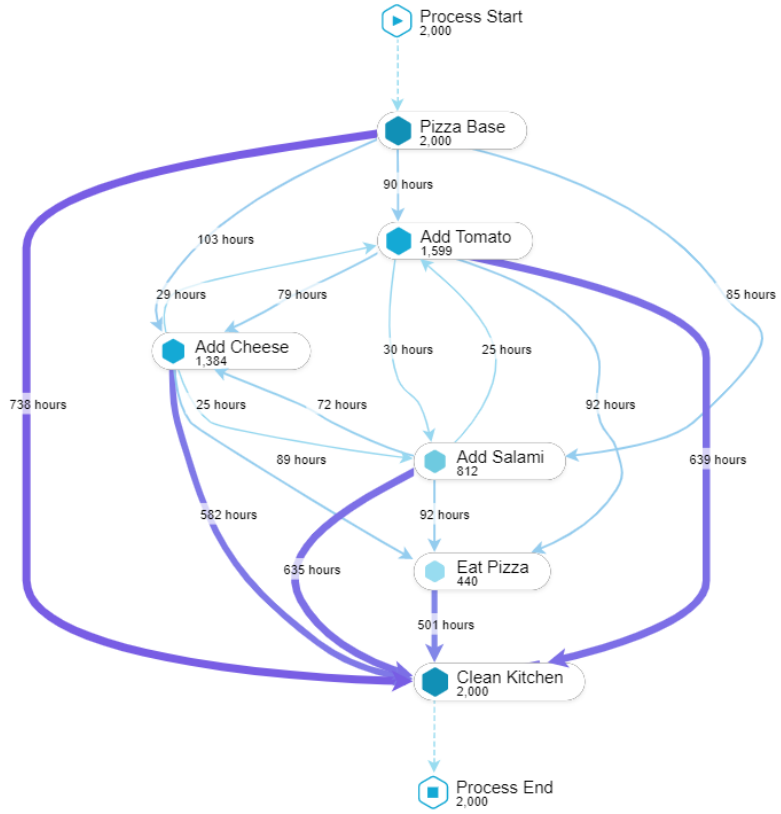
DFG Discovery in Celonis

The result is the same as in ProM





animation



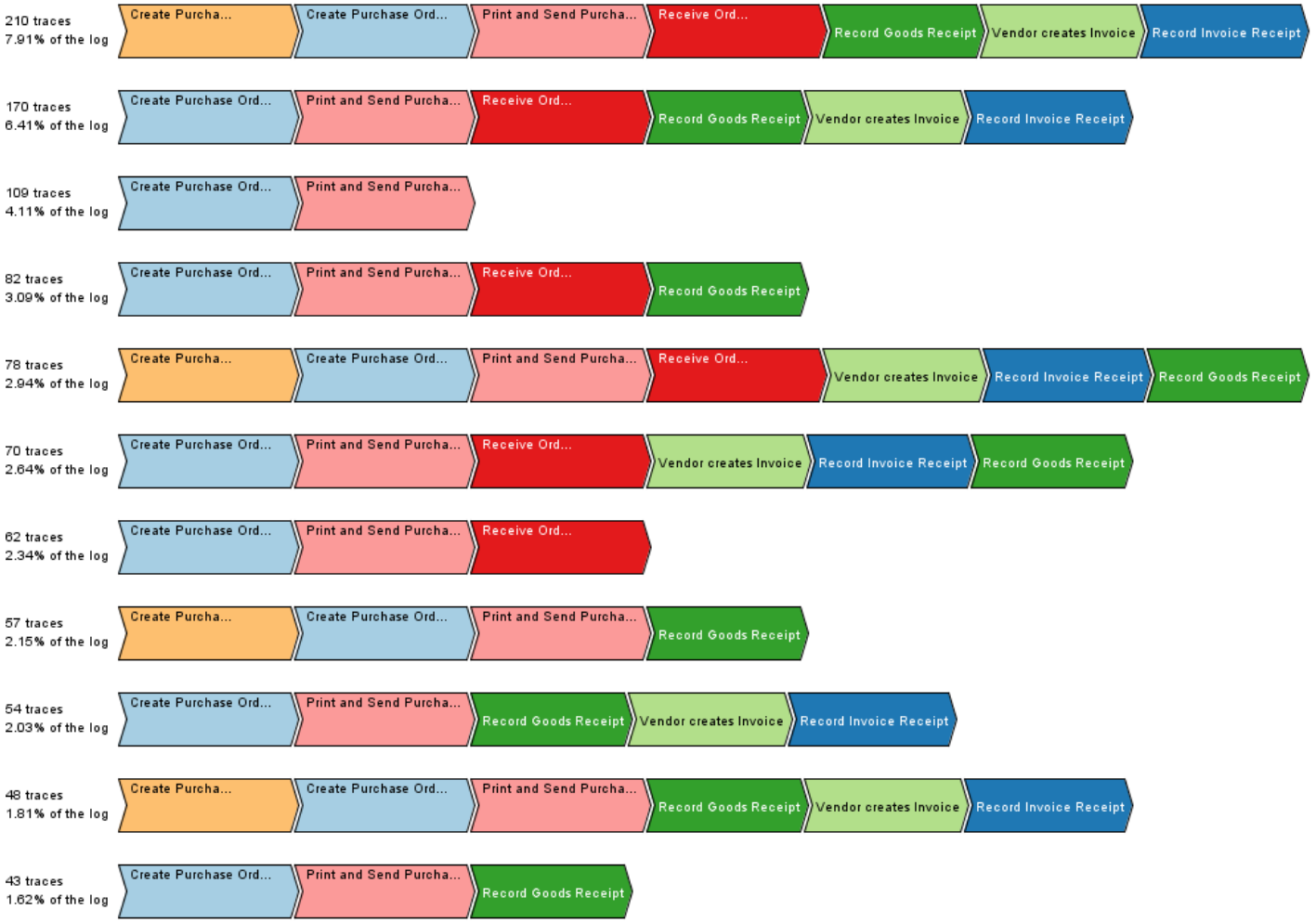
times

Note that this is a synthetic data set, thus explaining the long delays.

celonis

Chair of Process and Data Science

What if we get Spaghetti instead of Lasagna?

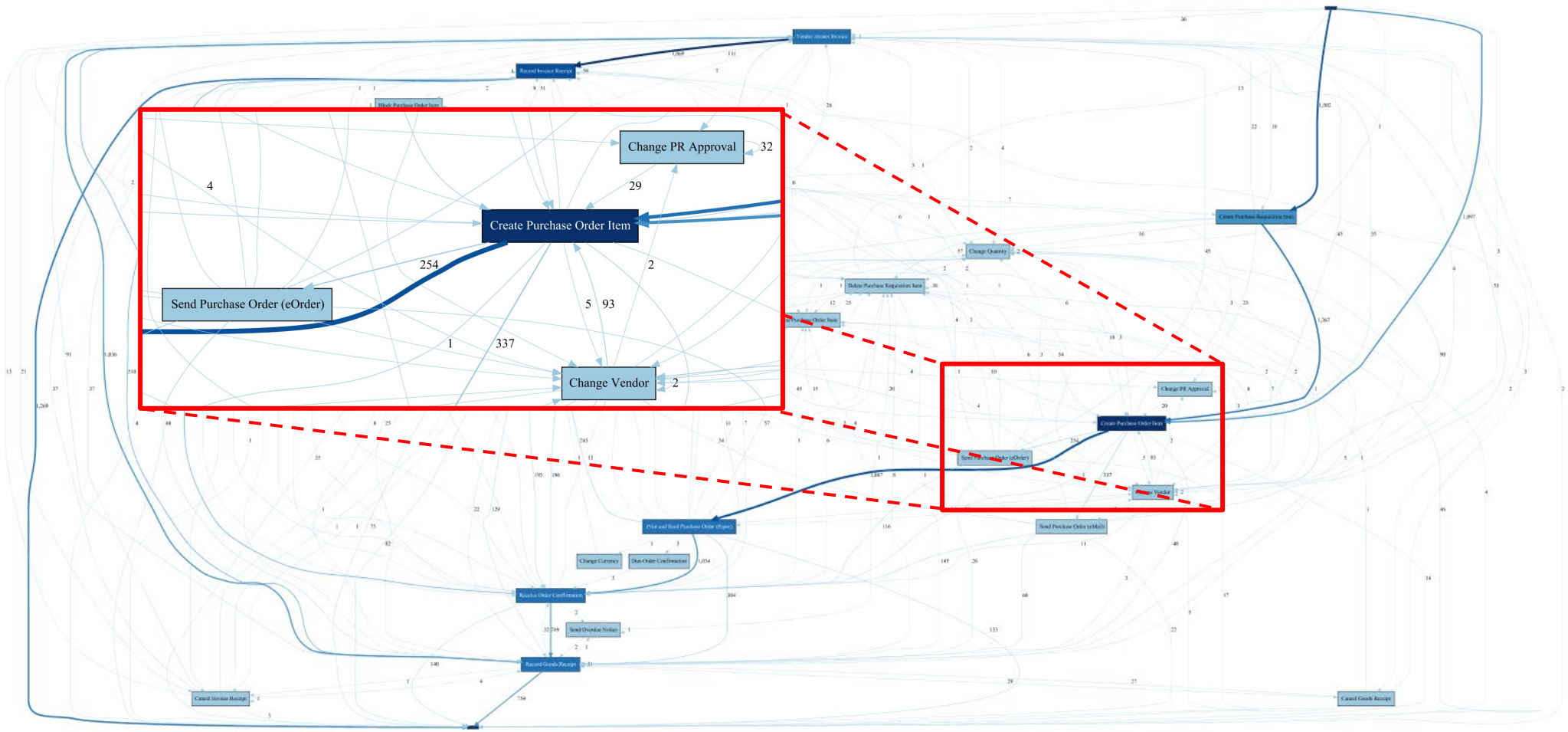


- Purchase to Pay (P2P).
- 2654 cases
- 16226 events
- 685 variants
- 24 unique activities

Still relatively simple, but …

Chair of Process and Data Science

What if we get Spaghetti instead of Lasagna?



Data-aware heuristic miner plug-in

Chair of Process and Data Science

Chair of Process and Data Science
RWTH AACHEN UNIVERSITY

Filtering

- **Activity-based filtering**: Rank the activities (e.g., based on frequency) and remove lower-ranked activities completely from your data.
- **Variant-based filtering**: Rank the variants (e.g., based on frequency) and remove lower-ranked variants. A variant is simply a sequence of activities and may occur multiple times.
- **Arc-based filtering** (not recommended!): Delete arcs in the DFG (e.g., based on frequency).

Hospital Billing - Event Log

DOI: 10.4121/uuid:76c46b83-c930-4798-a1c9-4be94dfeb741

Cite

DATASET

by Felix Mannhardt

The 'Hospital Billing' event log was obtained from the financial modules of the ERP system of a regional hospital. The event log contains events that are related to the billing of medical services that have been provided by the hospital. Each trace of the event log records the activities executed to bill a package of medical services that were bundled together. The event log does not contain information about the actual medical services provided by the hospital. The 100,000 traces in the event log are a random sample of process instances that were recorded over three years. Several attributes such as the 'state' of the process, the 'caseType', the underlying 'diagnosis' etc. are included in the event log. Events and attribute values have been anonymized. The time stamps of events have been randomized for this purpose, but the time between events within a trace has not been altered. More information about the event log can be found in the related publications.

Cases: 100,000
Events: 451,359
Event Types: 18
Variants: 1,020

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

History

2017-08-01 first online, published, posted

Publisher

Eindhoven University of Technology

Format

media types: application/x-gzip, text/plain, text/xml

References

https://doi.org/10.1007/978-3-319-59536-8_34

Organizations

Usage statistics

14090	21	5878
views	citations	downloads

Categories

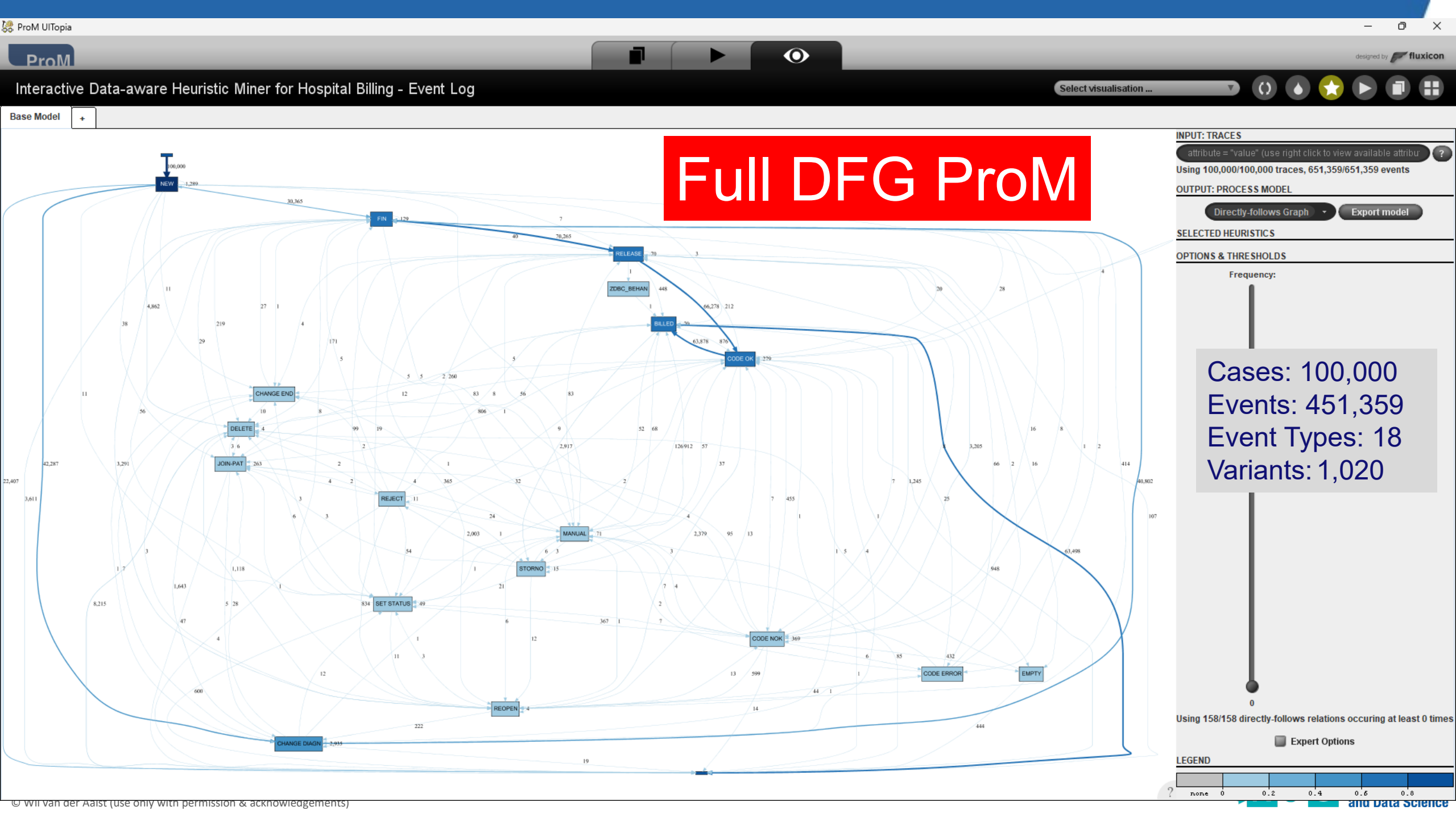
Business And Management

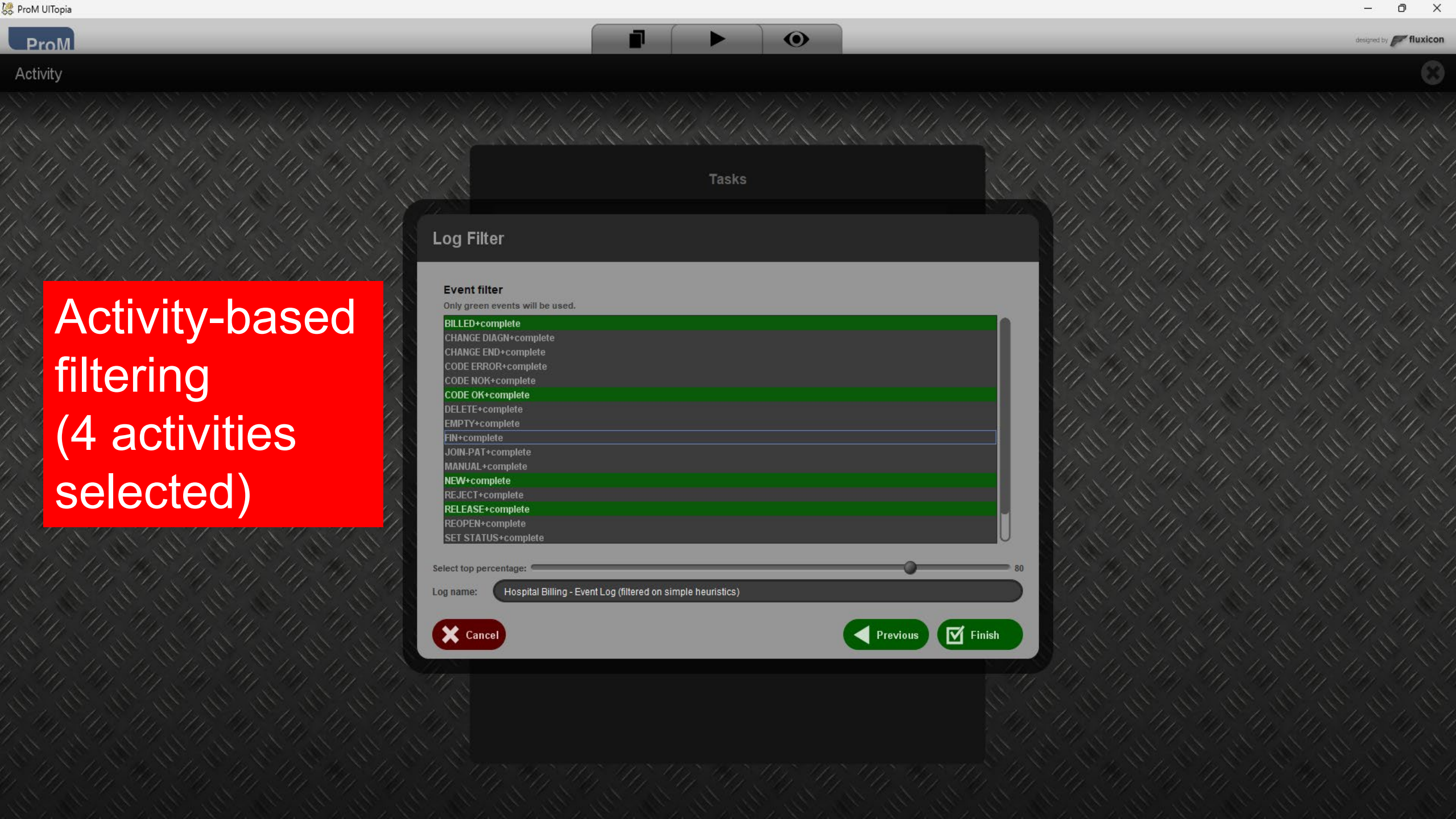
Keywords

000 Computer Science, Knowledge & Systems
Billing Process | Event Log | Hospital
IEEE Task Force On Process Mining
Process Mining | Real Life Event Logs

Time coverage

2012-12-13T10:13:18+01:00 / 2016-01-19T08:58:56+01:00

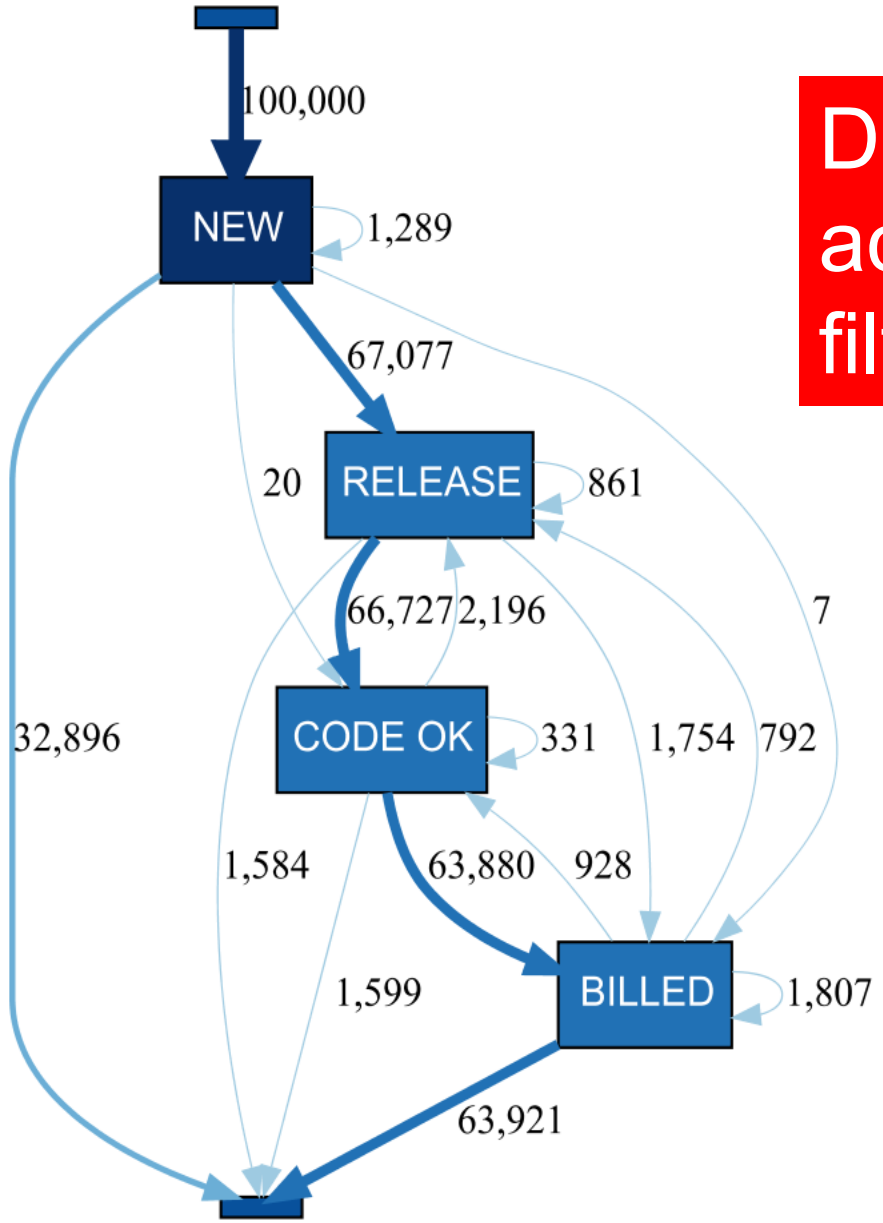



ProM UITopia
ProM
designed by fluxicon
Activity
Tasks
Log Filter
Event filter
Only green events will be used.
BILLED+complete
CHANGE DIAGN+complete
CHANGE END+complete
CODE ERROR+complete
CODE NOK+complete
CODE OK+complete
DELETE+complete
EMPTY+complete
FIN+complete
JOIN-PAT+complete
MANUAL+complete
NEW+complete
REJECT+complete
RELEASE+complete
REOPEN+complete
SET STATUS+complete
Select top percentage:
80
Log name:
Hospital Billing - Event Log (filtered on simple heuristics)
Cancel
Previous
Finish
Activity-based filtering (4 activities selected)

ProM UITopia

Interactive Data-aware Heuristic Miner for Hospital Billing - Event Log (filtered on simple heuristics)

Base Model



DFG after activity-based filtering

Cases: 100,000
Events: 307,669
Event Types: 4
Variants: 220

100% of cases
86% of events

INPUT: TRACES

attribute = "value" (use right click to view available attri

Using 100,000/100,000 traces, 507,669/507,669 events

OUTPUT: PROCESS MODEL

Directly-follows Graph | Export model

SELECTED HEURISTICS

OPTIONS & THRESHOLDS

Frequency:

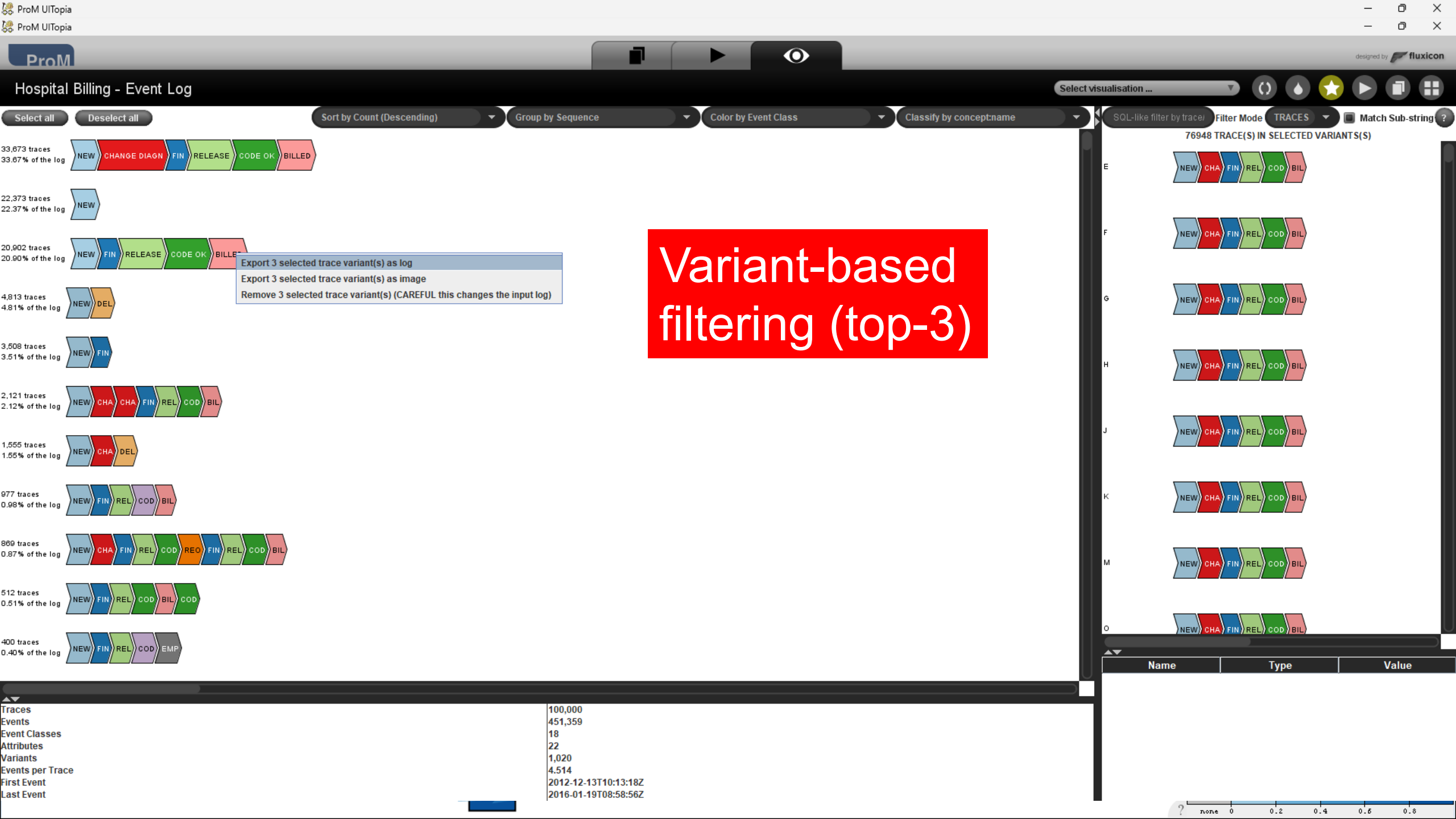
0

Using 18/18 directly-follows relations occuring at least 0 times

Expert Options

LEGEND

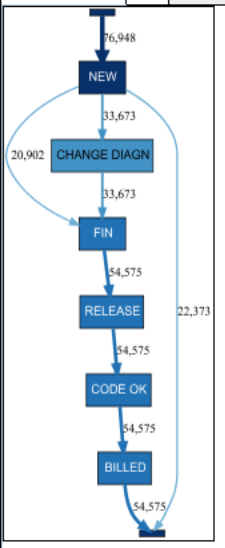
none 0 0.2 0.4 0.6 0.8

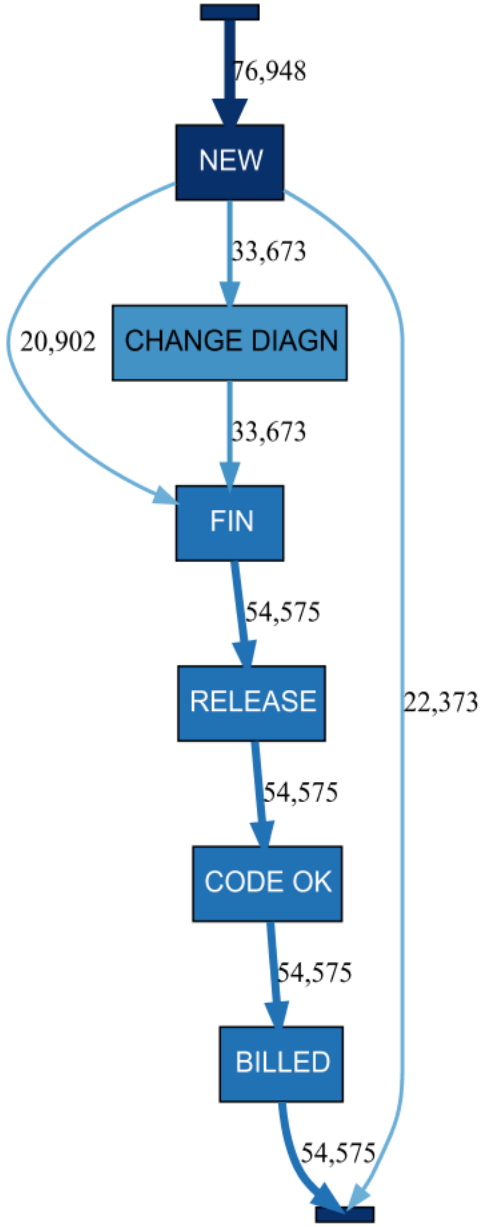

ProM UITopia
ProM UITopia
Hospital Billing - Event Log
Select visualisation ...
Select all
Deselect all
Sort by Count (Descending)
Group by Sequence
Color by Event Class
Classify by concept:name
SQL-like filter by trace)
Filter Mode
TRACES
Match Sub-string
76948 TRACE(S) IN SELECTED VARIANTS(S)
33,673 traces
33.67% of the log
NEW CHANGE DIAGN FIN RELEASE CODE OK BILLED
22,373 traces
22.37% of the log
NEW
20,902 traces
20.90% of the log
NEW FIN RELEASE CODE OK BILLE
Export 3 selected trace variant(s) as log
Export 3 selected trace variant(s) as image
Remove 3 selected trace variant(s) (CAREFUL this changes the input log)
4,813 traces
4.81% of the log
NEW DEL
3,508 traces
3.51% of the log
NEW FIN
2,121 traces
2.12% of the log
NEW CHA CHA FIN REL COD BIL
1,555 traces
1.55% of the log
NEW CHA DEL
977 traces
0.98% of the log
NEW FIN REL COD BIL
869 traces
0.87% of the log
NEW CHA FIN REL COD REO FIN REL COD BIL
512 traces
0.51% of the log
NEW FIN REL COD BIL COD
400 traces
0.40% of the log
NEW FIN REL COD EMP
Variant-based filtering (top-3)
E
F
G
H
J
K
M
O
NEW CHA FIN REL COD BIL
Name
Type
Value
Traces 100,000
Events 451,359
Event Classes 18
Attributes 22
Variants 1,020
Events per Trace 4.514
First Event 2012-12-13T10:13:18Z
Last Event 2016-01-19T08:58:56Z

ProM UITopia

Interactive Data-aware Heuristic Miner for Top 3

Base Model





DFG after variant-based filtering (top-3)

77% of cases
72% of events

INPUT: TRACES

attribute = "value" (use right click to view available a

Using 76,948/76,948 traces, 482,817/482,817 events

OUTPUT: PROCESS MODEL

Directly-follows Graph | Export model

SELECTED HEURISTICS

OPTIONS & THRESHOLDS

Frequency:

Cases: 76,948
Events: 328,92
Event Types: 6
Variants: 3

0

Using 9/9 directly-follows relations occuring at least 0 times

Expert Options

LEGEND

none 0 0.2 0.4 0.6 0.8

Celonis | DFG | def | hospital-bil

academic-workspace-wvdaalst.eu-2.celonis.cloud/package-manager/ui/studio/ui/spaces/02ea5e91-e05d-4e47-9d31-11295b57e66a/packages/fab8a9f4-4b20-47c7-a664-3474262d5481/nodes/eabcf8f2-08bc-4bef-...

Studio > hospital-billing > def

Search CTRL /

1

3 Create version

Deploy

Search

All (3) View (2) Know ...

New asset

Variants

DFG

Hospital-billing KM

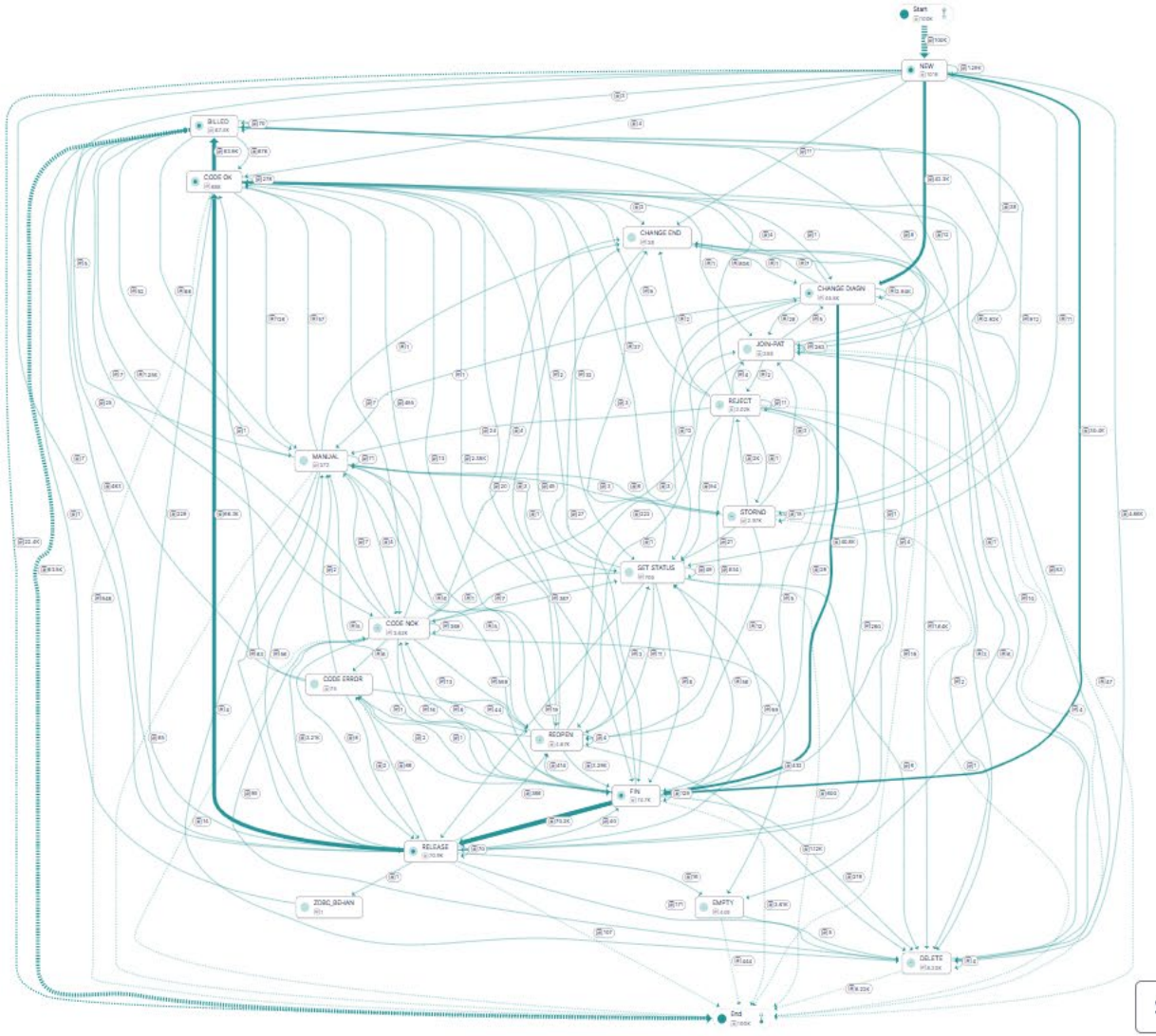
DFG

100%

0

Activity Frequency

Activities 18 of 18



DD

Legend

30%

Full DFG Celonis

Search

All (3) View (2) Know ...

New asset

Variants

DFG

Hospital-billing KM

DFG

100%

0

Activities 18 of 18

JOIN-PAT
358

REJECT
2.02K

STORNO
2.97K

SET STATUS
705

37 263 1 2 32 4 2 3 11 24 4 12 3 3 2.38K 2K 1 20 3 45 3 6 3 54 1 27 222 15 1 1 21 40.8K 4 1 49 834 29 83 6 1 7 367 5 10 369 5 12 260 1.64K 19 3 6 3 11 13 599 8 56 2

Celonis | DFG | def | hospital-bil

academic-workspace-wvdaalst.eu-2.celonis.cloud/package-manager/ui/studio/ui/spaces/02ea5e91-e05d-4e47-9d31-11295b57e66a/packages/fab8a9f4-4b20-47c7-a664-3474262d5481/nodes/eabcf8f2-08bc-4bef-...

Studio > hospital-billing > def

Search CTRL /

1

3 Create version

Deploy

Search

All (3) View (2) Know ...

New asset

Variants

DFG

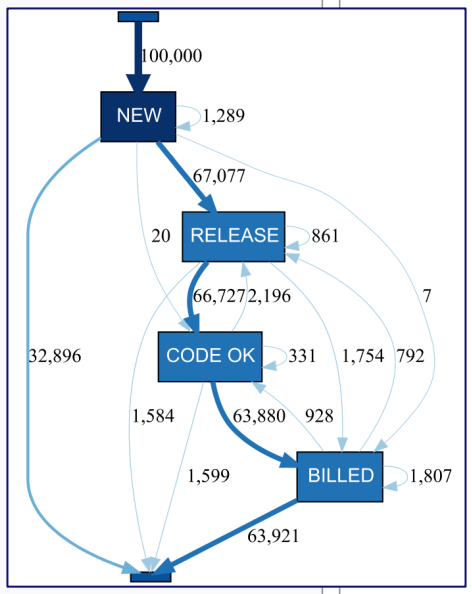
Hospital-billing KM

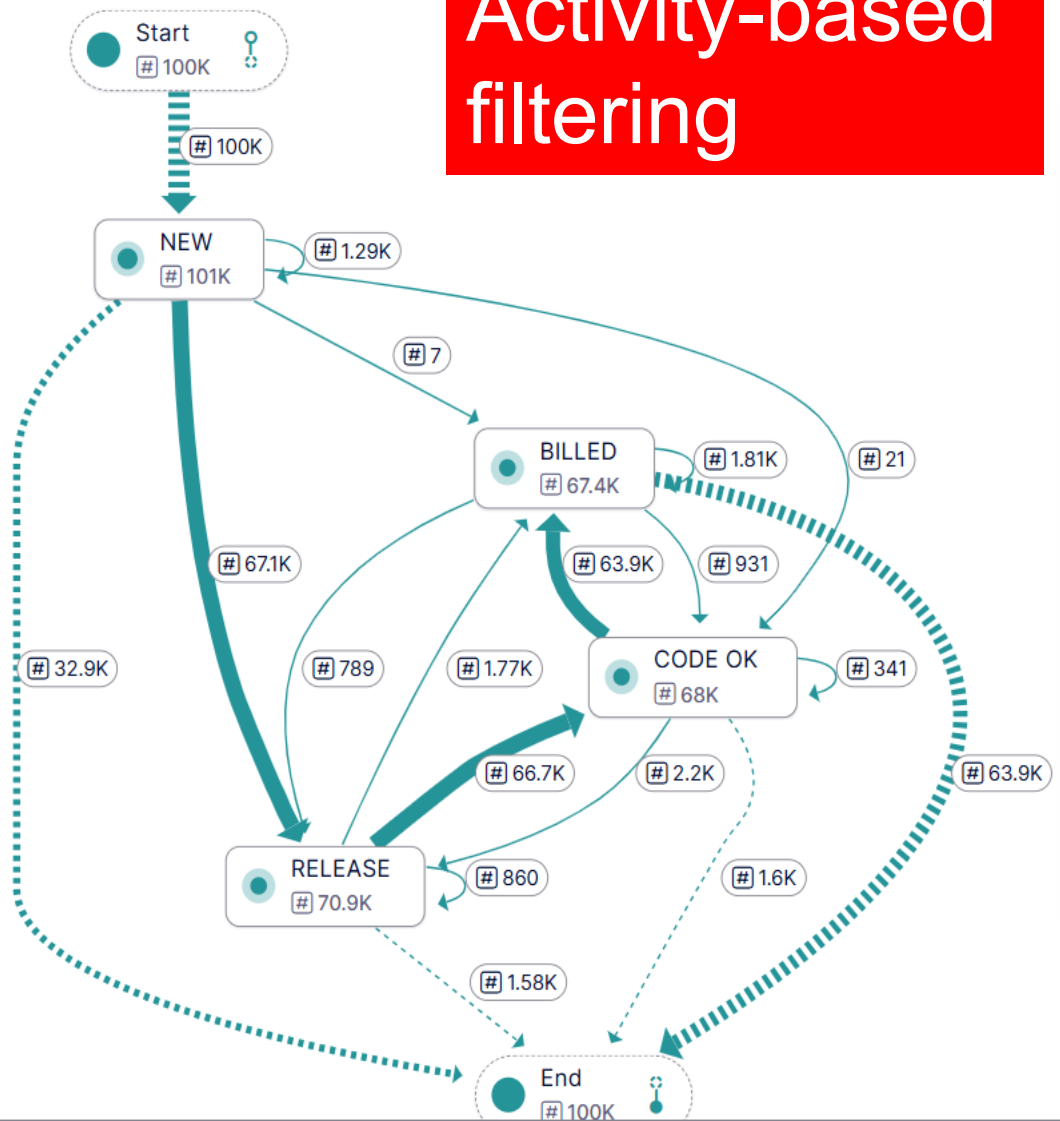
DFG

100%

0

Activity-based filtering





Activities 4 of 18

Search

Activity Select all Event logs

- [x] NEW
- [ ] FIN
- [x] RELEASE
- [x] CODE OK
- [x] BILLED
- [ ] CHANGE DIAGN
- [ ] DELETE
- [ ] REOPEN

Cancel Apply

Celonis | Variants | def | hospital

academic-workspace-wvdaalst.eu-2.celonis.cloud/package-manager/ui/studio/ui/spaces/02ea5e91-e05d-4e47-9d31-11295b57e66a/packages/fab8a9f4-4b20-47c7-a664-3474262d5481/nodes/bdc2e2b1-7bb4-484...

Studio > hospital-billing > def

Search CTRL /

1

3 Create version

Deploy

Search

All (3) View (2) Know ...

New asset

Variants

DFG

Hospital-billing KM

Variants

100%

0

Activities 18 of 18

Variant Explorer

34%

cases covered

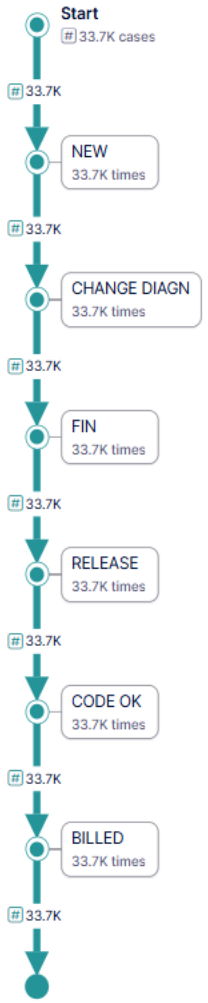
1 of 949 variant

33.7K of 100K cases

Apply Filter

Variant	Count	Coverage	Avg TPT
▶	33.7K	34%	131 d
#2	22.4K	22%	0 ms
#3	20.9K	21%	238 d
#4	4.81K	5%	51 d
#5	3.51K	4%	144 d
#6	2.3K	2%	135 d
#7	1.56K	2%	18 d
#8	977	1%	102 d
#9	866	1%	159 d
#10	512	1%	450 d
Others	8.52K	9%	210 d

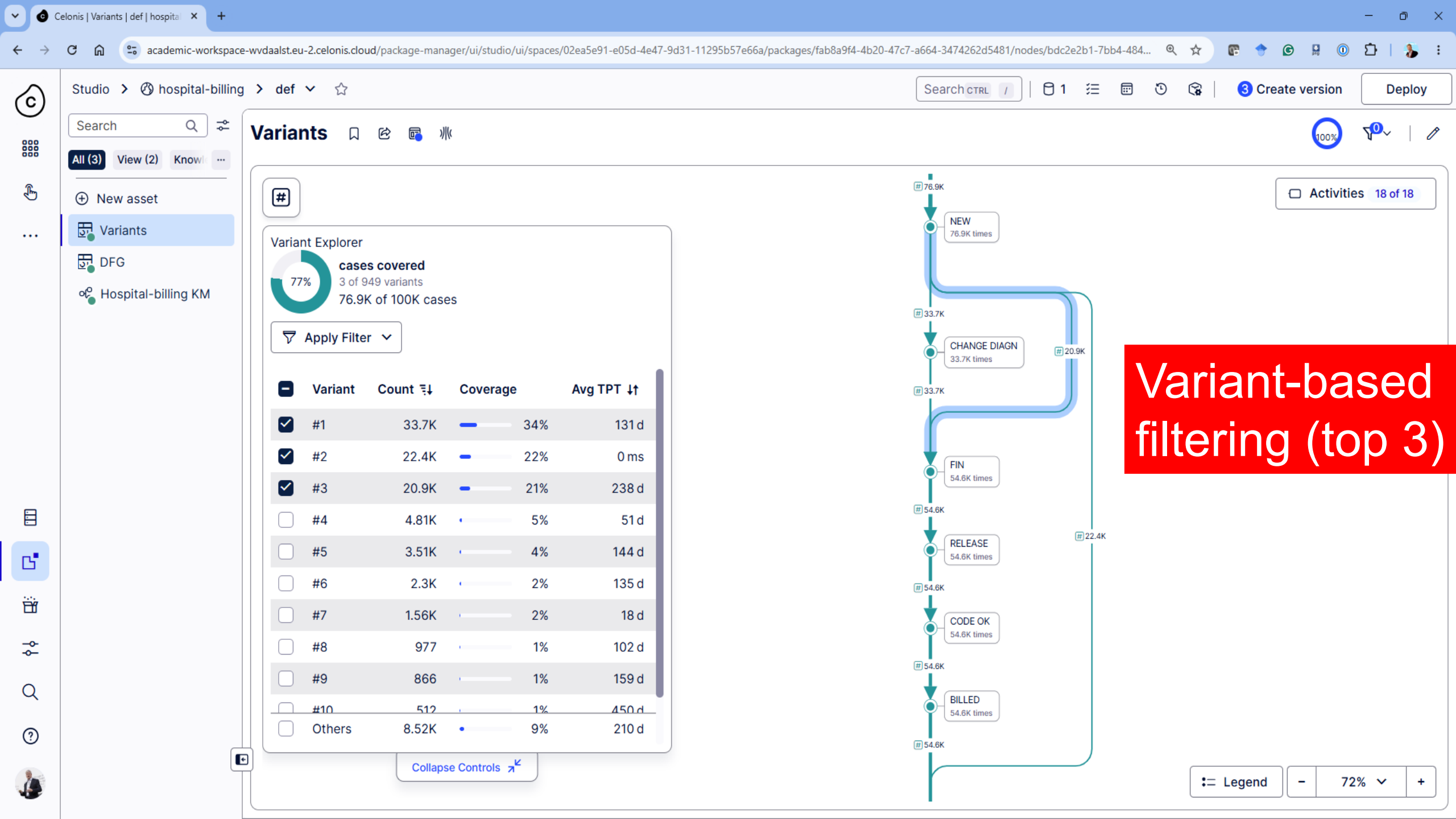
Collapse Controls



Variant-based filtering (top 1)

Legend

72%


Celonis | Variants | def | hospital
academic-workspace-wvdaalst.eu-2.celonis.cloud/package-manager/ui/studio/ui/spaces/02ea5e91-e05d-4e47-9d31-11295b57e66a/packages/fab8a9f4-4b20-47c7-a664-3474262d5481/nodes/bdc2e2b1-7bb4-484...
Studio
hospital-billing
def
Search CTRL /
1
3 Create version
Deploy
Search
All (3)
View (2)
Know
New asset
Variants
DFG
Hospital-billing KM
Variants
100%
0
Variant Explorer
77%
cases covered
3 of 949 variants
76.9K of 100K cases
Apply Filter
Variant
Count
Coverage
Avg TPT
#1 33.7K 34% 131 d
#2 22.4K 22% 0 ms
#3 20.9K 21% 238 d
#4 4.81K 5% 51 d
#5 3.51K 4% 144 d
#6 2.3K 2% 135 d
#7 1.56K 2% 18 d
#8 977 1% 102 d
#9 866 1% 159 d
#10 512 1% 450 d
Others 8.52K 9% 210 d
Collapse Controls
Activities 18 of 18
76.9K
NEW
76.9K times
33.7K
CHANGE DIAGN
33.7K times
20.9K
33.7K
FIN
54.6K times
54.6K
RELEASE
54.6K times
22.4K
54.6K
CODE OK
54.6K times
54.6K
BILLED
54.6K times
54.6K
Legend
72%
Variant-based filtering (top 3)

Celonis | Variants | def | hospital

academic-workspace-wvdaalst.eu-2.celonis.cloud/package-manager/ui/studio/ui/spaces/02ea5e91-e05d-4e47-9d31-11295b57e66a/packages/fab8a9f4-4b20-47c7-a664-3474262d5481/nodes/bdc2e2b1-7bb4-484...

Studio > hospital-billing > def

Search CTRL /

1

Create version

Deploy

Search

All (3) View (2) Know ...

New asset

Variants

DFG

Hospital-billing KM

Variants

77%

Variant
3 selected

Attribute filter

Process filters

Process flow filter

Activity filter

Throughput time filter

Activity count filter

Copy filters to View

Reset filters

Variant Explorer

100%

cases covered

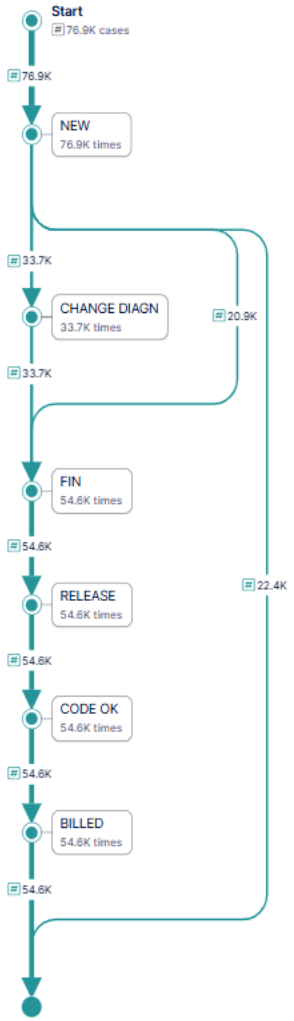
3 of 3 variants

76.9K of 76.9K cases

Apply Filter

Variant	Count	Coverage	Avg TPT
#1	33.7K	44%	131 d
#2	22.4K	29%	0 ms
#3	20.9K	27%	238 d

Collapse Controls



Legend

59%

Propogate filter that covers 77% of cases

Search
All (3) View (2) Know ...
New asset
Variants
DFG
Hospital-billing KM

Variants

77%
Variant
3 selected

Activities 18 of 18

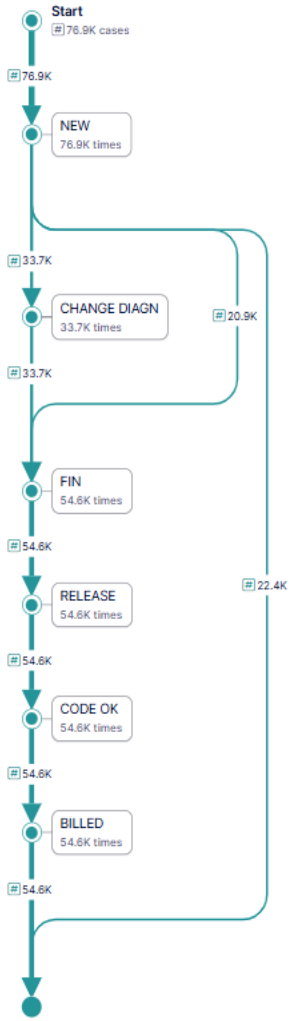
Variant Explorer

100%
cases covered
3 of 3 variants
76.9K of 76.9K cases

Apply Filter

Variant	Count	Coverage	Avg TPT
#1	33.7K	44%	131 d
#2	22.4K	29%	0 ms
#3	20.9K	27%	238 d

Collapse Controls



Legend
59%

Filters copied to the clipboard
Go to the View where you want to apply the filters and apply them by clicking on "apply filter". The filters will remain in the clipboard for 10 minutes.

Celonis | DFG | def | hospital-bil

academic-workspace-wvdaalst.eu-2.celonis.cloud/package-manager/ui/studio/ui/spaces/02ea5e91-e05d-4e47-9d31-11295b57e66a/packages/fab8a9f4-4b20-47c7-a664-3474262d5481/nodes/eabcf8f2-08bc-4bef-...

Google Chrome anpassen und verwalten

Studio > hospital-billing > def

Search CTRL /

1

3 Create version

Deploy

Search

All (3) View (2) Know ...

New asset

Variants

DFG

Hospital-billing KM

DFG

77%

Variant
3 selected

Activities 18 of 18

Start
76.9K

76.9K

NEW
76.9K

33.7K

CHANGE DIAGN
33.7K

20.9K

33.7K

FIN
54.6K

54.6K

22.4K

RELEASE
54.6K

54.6K

CODE OK
54.6K

54.6K

BILLED
54.6K

54.6K

End
76.9K

Variant-based filtering (top-3)

DD

Legend

73%

Celonis | Variants | def | hospital

academic-workspace-wvdaalst.eu-2.celonis.cloud/package-manager/ui/studio/ui/spaces/02ea5e91-e05d-4e47-9d31-11295b57e66a/packages/fab8a9f4-4b20-47c7-a664-3474262d5481/nodes/bdc2e2b1-7bb4-484...

Studio > hospital-billing > def

Search CTRL /

1

3 Create version

Deploy

Search

All (3) View (2) Know ...

New asset

Variants

DFG

Hospital-billing KM

Variants

100%

0

Variant-based filtering (top-10)

Variant Explorer

91%

cases covered

9 of 949 variants

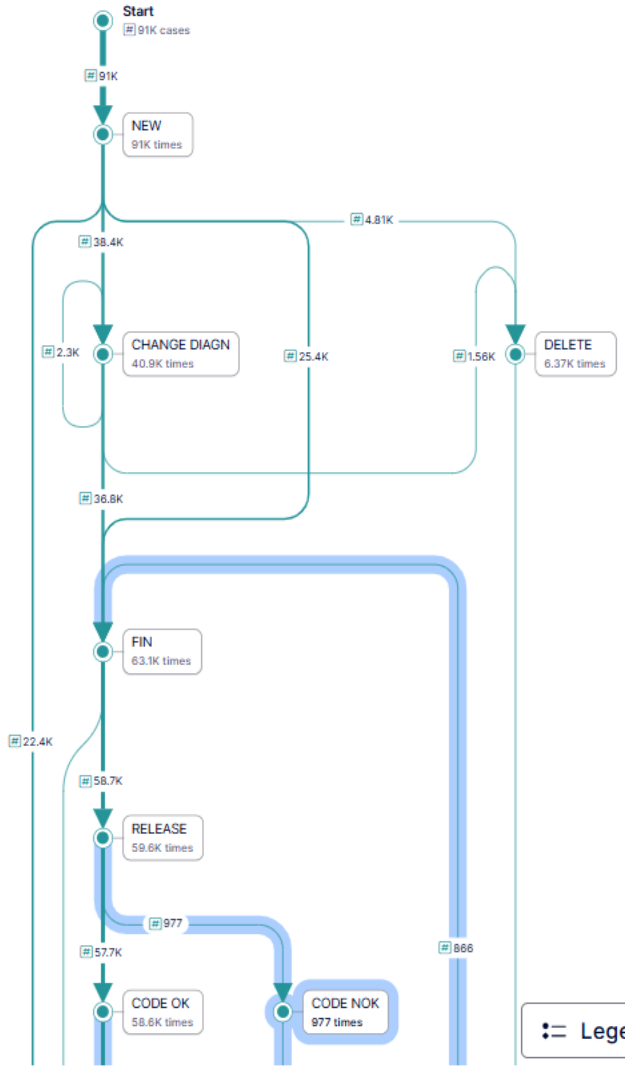
91K of 100K cases

Apply Filter

Variant	Count	Coverage	Avg TPT
#1	33.7K	34%	131 d
#2	22.4K	22%	0 ms
#3	20.9K	21%	238 d
#4	4.81K	5%	51 d
#5	3.51K	4%	144 d
#6	2.3K	2%	135 d
#7	1.56K	2%	18 d
#8	977	1%	102 d
#9	866	1%	159 d
#10	512	1%	450 d
Others	8.52K	9%	210 d

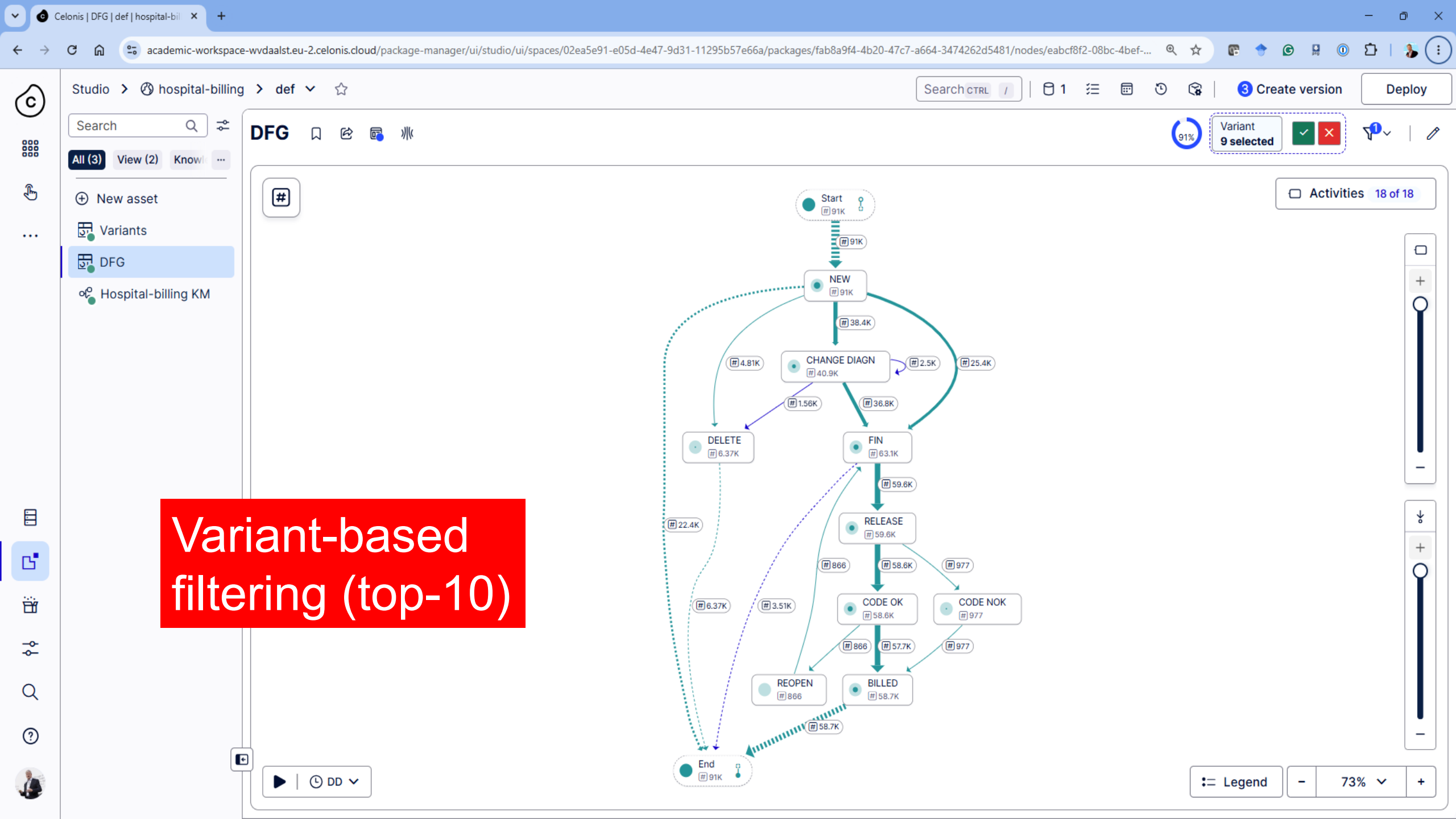
Collapse Controls

Activities 18 of 18



Legend

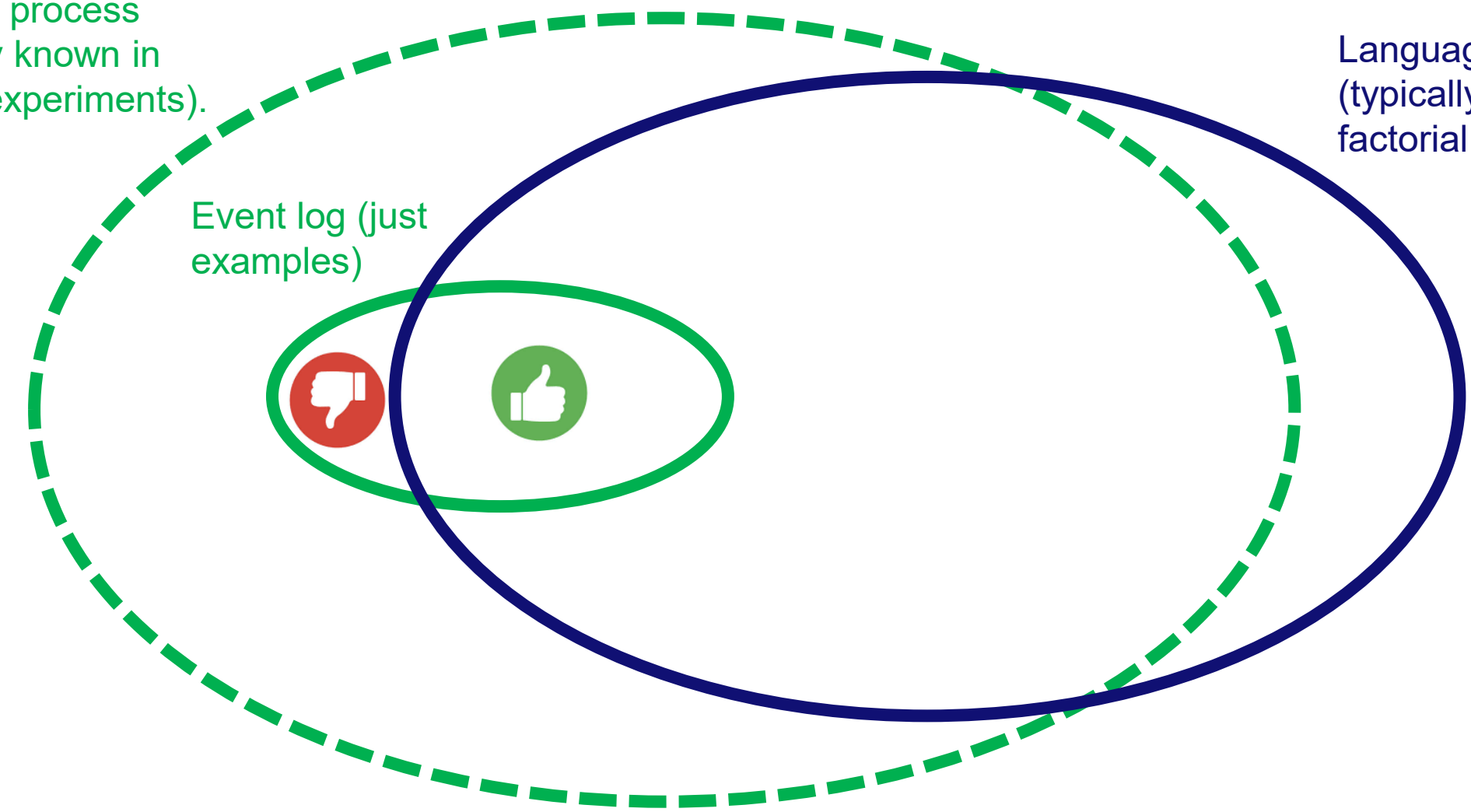
59%


Variant-based filtering (top-10)
Studio
hospital-billing
def
Search
DFG
Variant
9 selected
Activities 18 of 18
New asset
Variants
DFG
Hospital-billing KM
Start
NEW
CHANGE DIAGN
DELETE
FIN
RELEASE
CODE OK
CODE NOK
REOPEN
BILLED
End
Legend
73%

Challenges

- If the model allows for a loop, we have infinitely many possible traces. This can never be observed!
- The event log just shows examples, the fact that something did not happen does not mean it cannot.
- We do not have negative traces, i.e., it is not a classification problem.
- Hence, precision and recall cannot be defined in the usual manner.

Visualizing the challenges

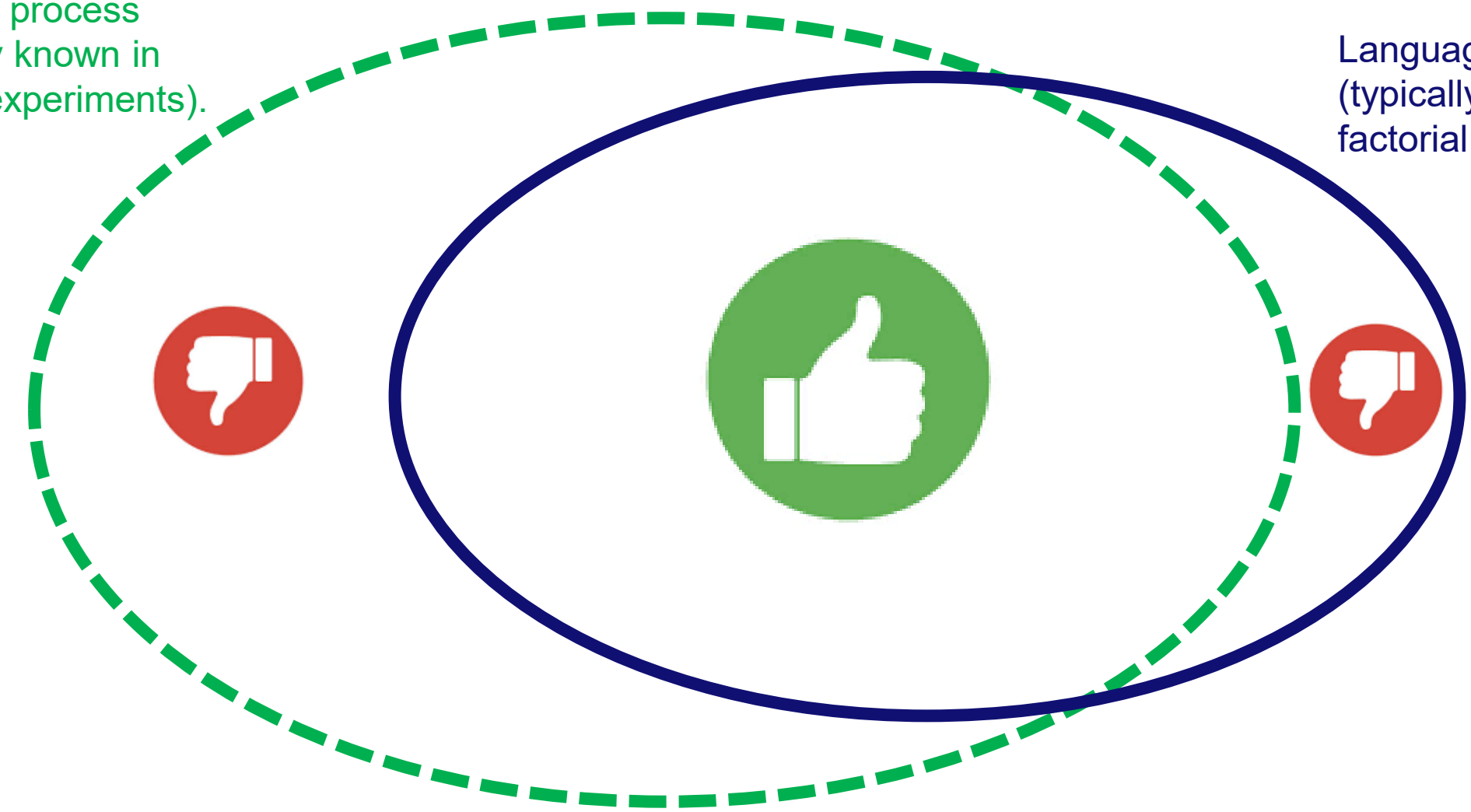


Chair of Process and Data Science

What we would like to know, but cannot know

Real process (only known in lab experiments).

Language of the model (typically infinitely or factorial many traces).



Chair of Process and Data Science

Therefore, there are many approximations (often using proxies)

See later lectures!

- **Replay fitness** (using the fraction of fitting traces on the event log, token-based, or alignment based).
- **Precision** (e.g., escaping edges).
- **Simplicity** (e.g., number of arcs).
- **Generalization** (e.g., likelihood that the next trace will fit given some assumptions about the distribution).

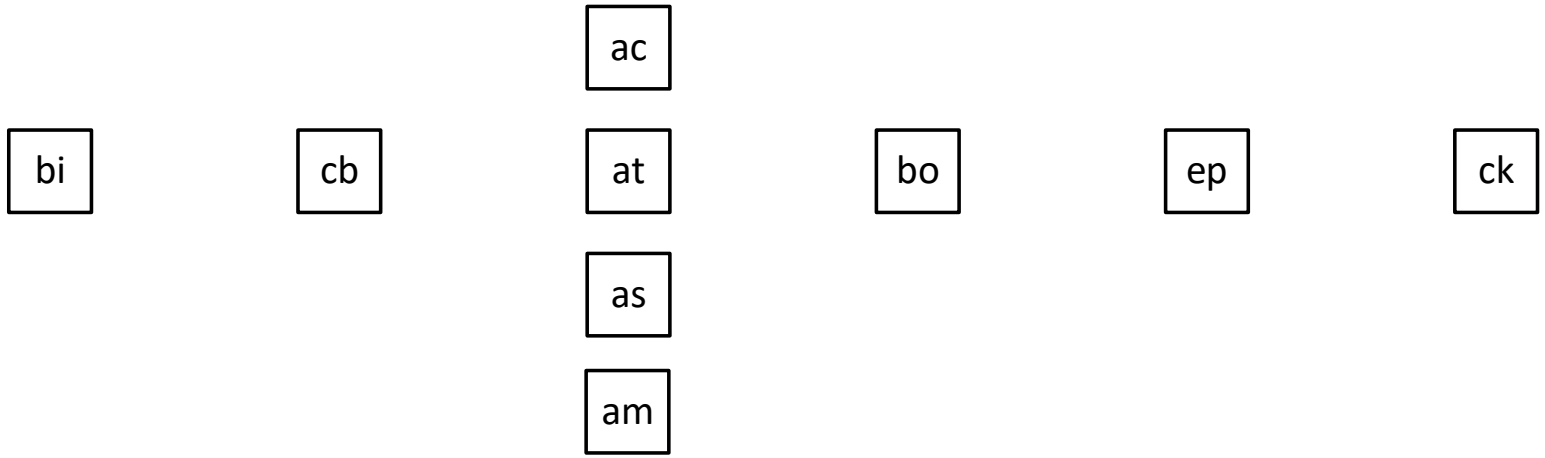
Check out stochastic conformance checking!

Sander Leemans, Wil van der Aalst, Tobias Brockhoff, Artem Polyvyanyy: Stochastic process mining: Earth movers' stochastic conformance. Inf. Syst. 102: 101724 (2021)

Chair of Process and Data Science
RWTH AACHEN UNIVERSITY

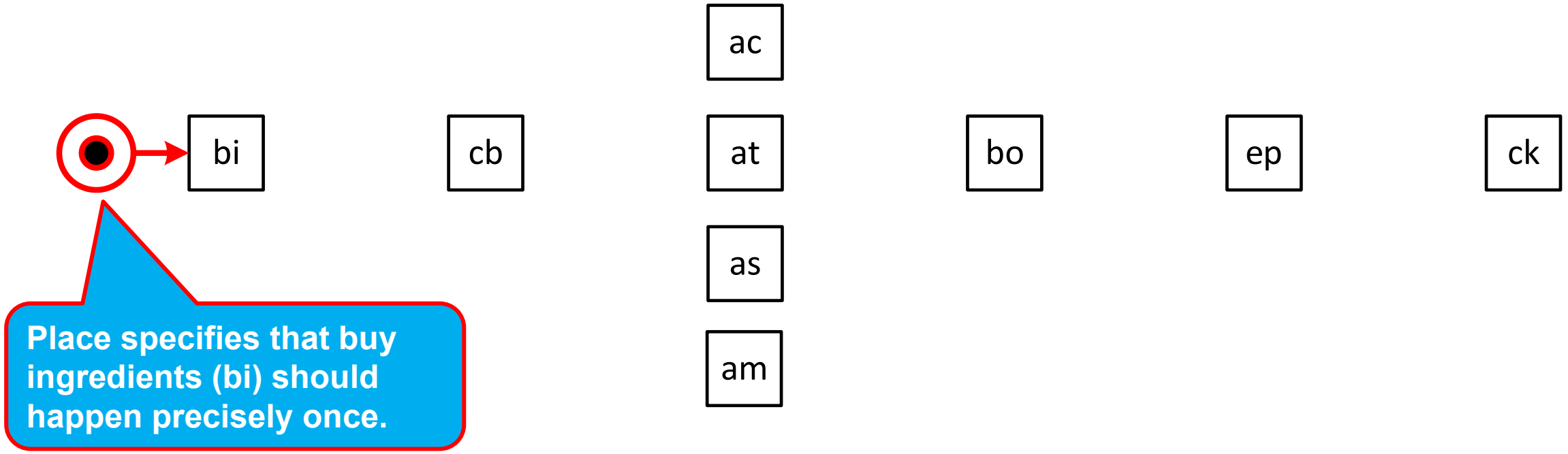
Bottom-up discovery

- Assume that anything is possible.
- Start adding constraints supported by the data.
- A Petri net place is a constraint.
- Accepting Petri-nets are surprisingly declarative.



Using short names: buy ingredients (bi), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

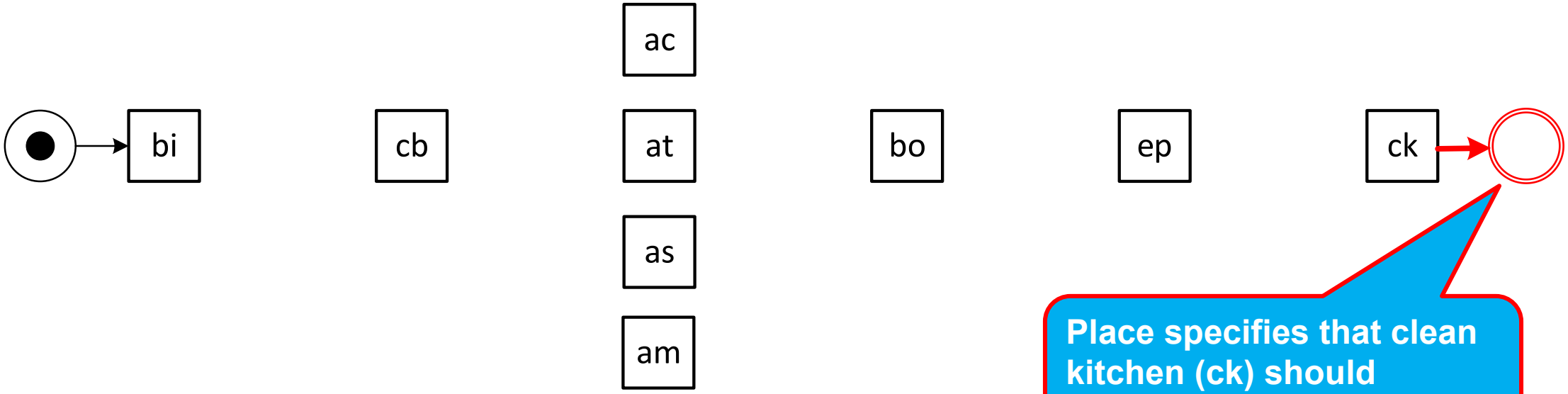
Places as constraints



An accepting Petri net has an initial and final marking (here the final marking is [ ], i.e., no tokens).

Using short names: buy ingredients (bi), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

Places as constraints

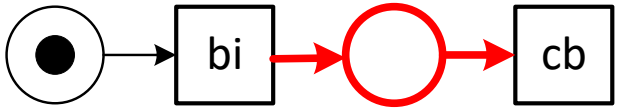


An accepting Petri net has an initial and final marking (here the final marking has one token in sink place).

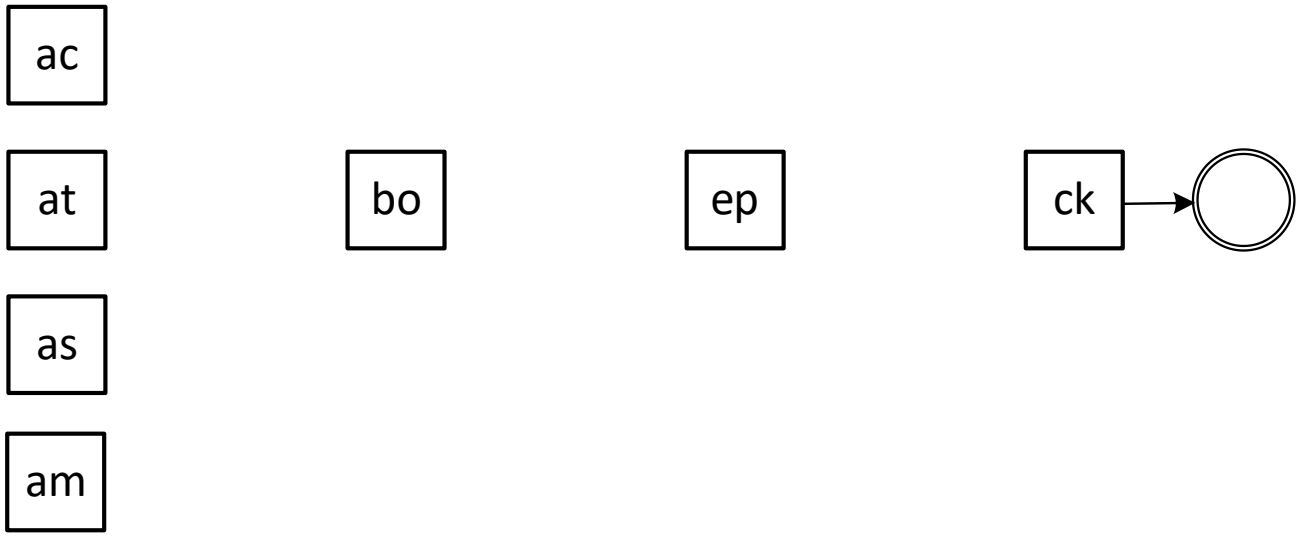
Using short names: buy ingredients (bi), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

Places as constraints

- #bi = #cb at the end of each case
- #bi ≥ #cb at any point in time



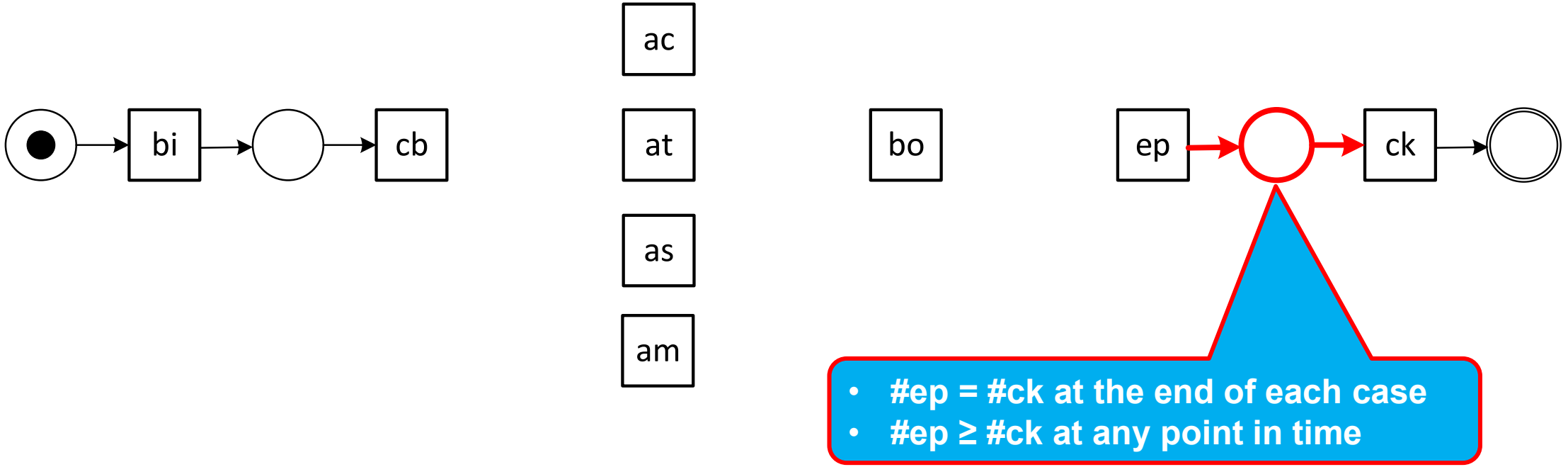
Place specifies that bi and cb should happen the same number of times. Moreover, at any stage the number of occurrences of cb should not exceed the number of occurrences of bi.



A place defines a local constraint.

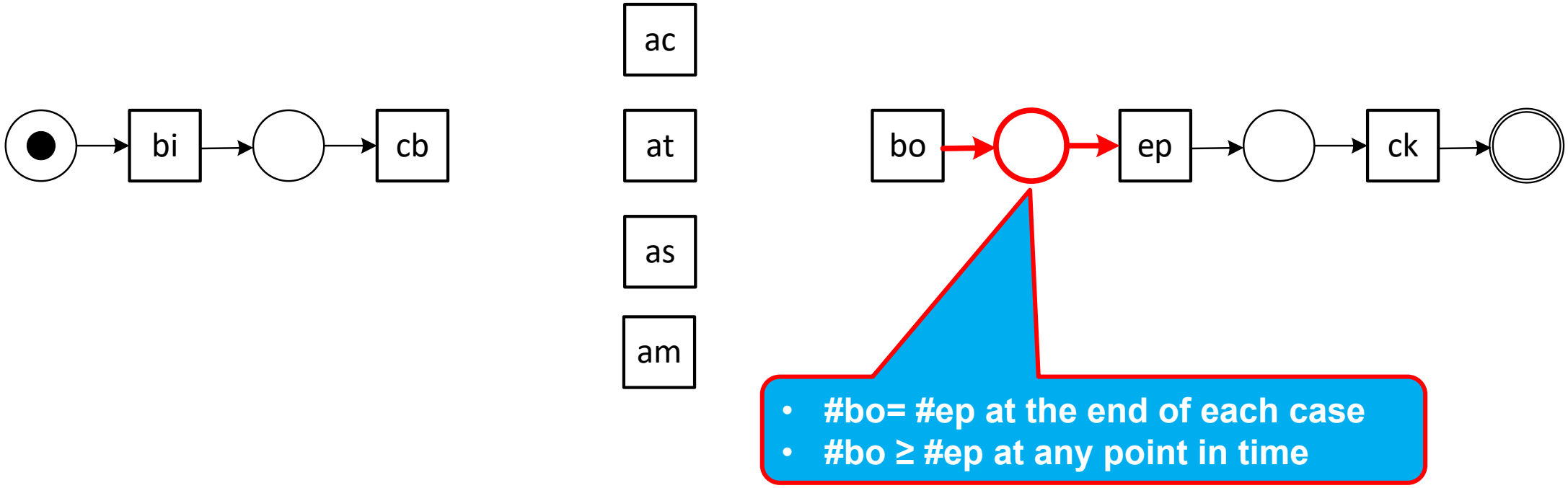
Using short names: buy ingredients (bi), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

Places as constraints



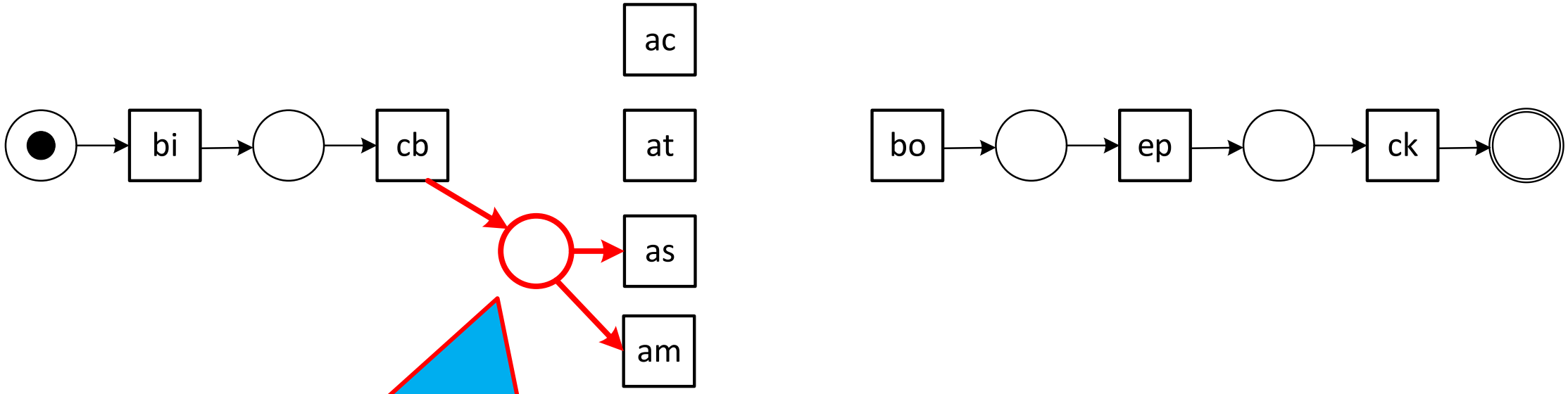
Using short names: buy ingredients (bi), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

Places as constraints



Using short names: buy ingredients (bi), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

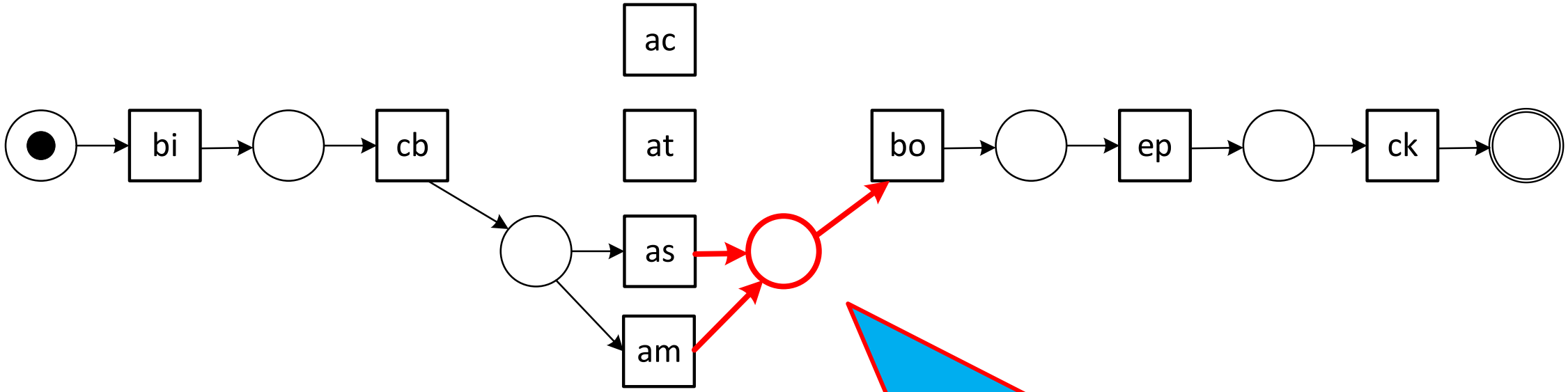
Places as constraints



- **#cb = #as+#am at the end of each case**
- **#cb ≥ #as+#am at any point in time**

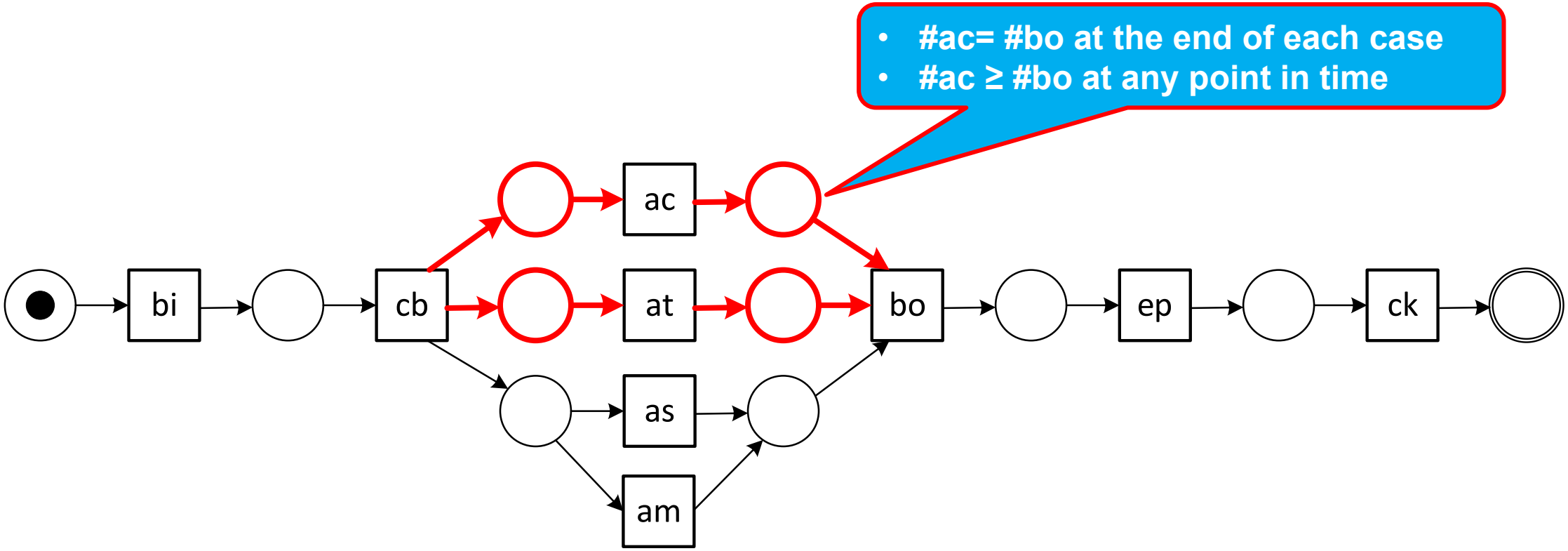
Using short names: buy ingredients (bi), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

Places as constraints



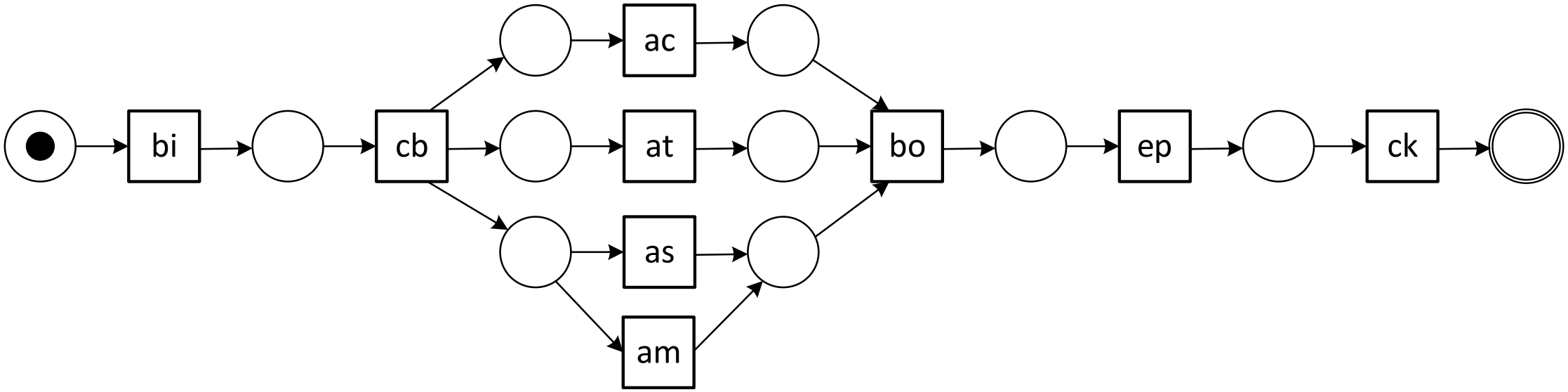
Using short names: buy ingredients (bi), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

Places as constraints



Using short names: buy ingredients (bi), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

Final accepting Petri net



Also every intermediate model was an accepting Petri net!

Bottom-up process discover uses this locality principle!

Using short names: buy ingredients (bi), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

Alpha Algorithm

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

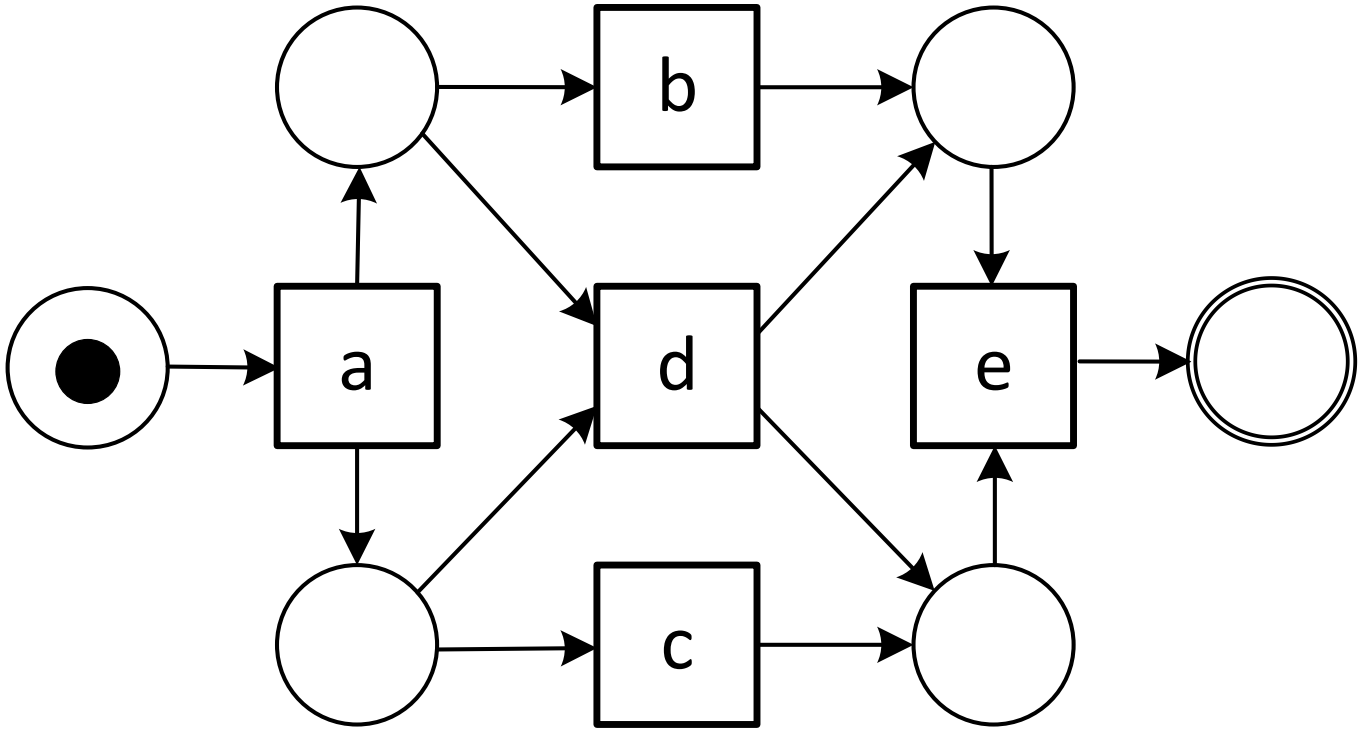
Just eight lines of mathematics, based on the DFG created before

Definition 22 (Alpha Algorithm). *The alpha algorithm* $disc_{alpha} \in \mathcal{B}(\mathcal{U}_{act}{}^*) \rightarrow \mathcal{U}_{AN}$ *returns an accepting Petri net* $disc_{alpha}(L)$ *for any event log* $L \in \mathcal{B}(\mathcal{U}_{act}{}^*)$. *Let* $A = act(L)$ *and* $fp(L) = fp(disc_{DFG}(L))$ *the footprint of event log* L. *This allows us to write* $a_1 \rightarrow_L a_2$ *if* $fp(L)((a_1, a_2)) = \rightarrow$ *and* $a_1 \#_L a_2$ *if* $fp(L)((a_1, a_2)) = \#$ *for any* $a_1, a_2 \in A' = A \cup \{\blacktriangleright, \blacksquare\}$.

1. $Cnd = \{(A_1, A_2) \mid A_1 \subseteq A' \wedge A_1 \neq \emptyset \wedge A_2 \subseteq A' \wedge A_2 \neq \emptyset \wedge \forall_{a_1 \in A_1} \forall_{a_2 \in A_2} a_1 \rightarrow_L a_2 \wedge \forall_{a_1, a_2 \in A_1} a_1 \#_L a_2 \wedge \forall_{a_1, a_2 \in A_2} a_1 \#_L a_2\}$ *are the candidate places,*
2. $Sel = \{(A_1, A_2) \in Cnd \mid \forall_{(A_1', A_2') \in Cnd} A_1 \subseteq A_1' \wedge A_2 \subseteq A_2' \Longrightarrow (A_1, A_2) = (A_1', A_2')\}$ *are the selected maximal places,*
3. $P = \{p_{(A_1, A_2)} \mid (A_1, A_2) \in Sel\} \cup \{p_{\blacktriangleright}, p_{\blacksquare}\}$ *is the set of all places,*
4. $T = \{t_a \mid a \in A'\}$ *is the set of transitions,*
5. $F = \{(t_a, p_{(A_1, A_2)}) \mid (A_1, A_2) \in Sel \wedge a \in A_1\} \cup \{(p_{(A_1, A_2)}, t_a) \mid (A_1, A_2) \in Sel \wedge a \in A_2\} \cup \{(p_{\blacktriangleright}, t_{\blacktriangleright}), (t_{\blacksquare}, p_{\blacksquare})\}$ *is the set of arcs,*
6. $l = \{(t_a, a) \mid a \in A\}$ *is the labeling function,*
7. $M_{init} = [p_{\blacktriangleright}]$ *is the initial marking,* $M_{final} = [p_{\blacksquare}]$ *is the final marking, and*
8. $disc_{alpha}(L) = ((P, T, F, l), M_{init}, M_{final})$ *is the discovered accepting Petri net.*

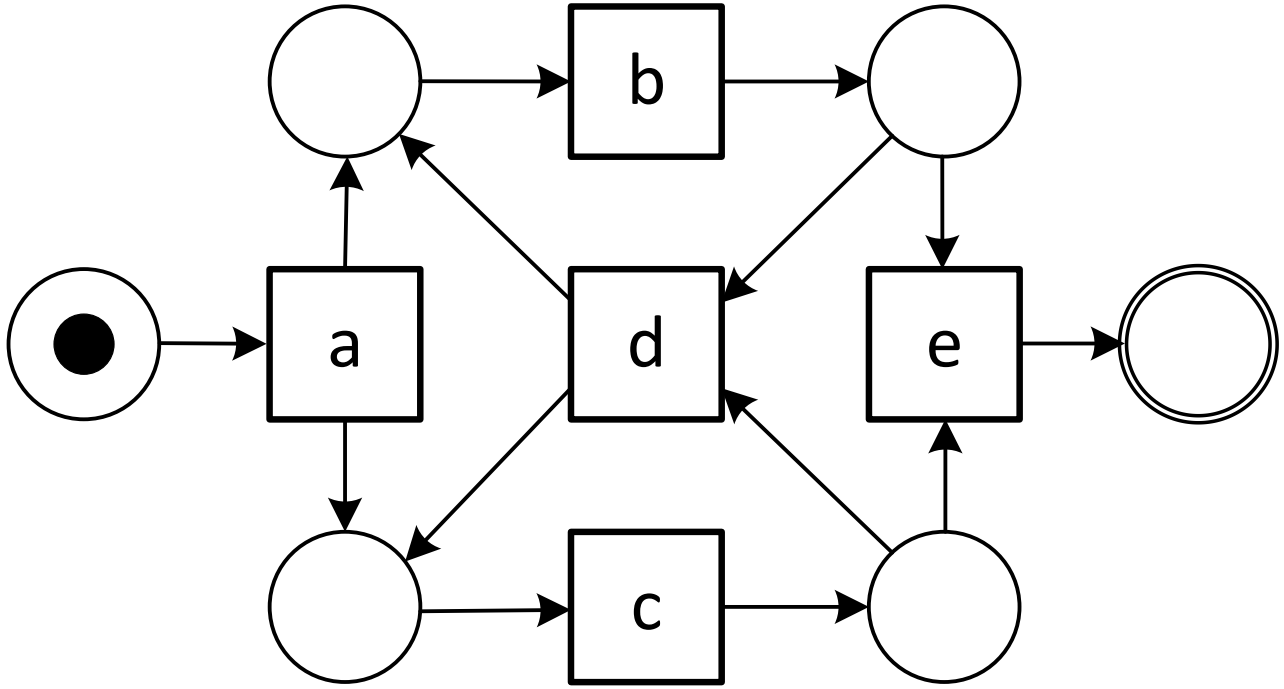
Example

$$L_1 = [\langle a, b, c, e\rangle^{10}, \langle a, c, b, e\rangle^{5}, \langle a, d, e\rangle] \in \mathcal{B}({\mathcal{U}_{act}}^*)$$



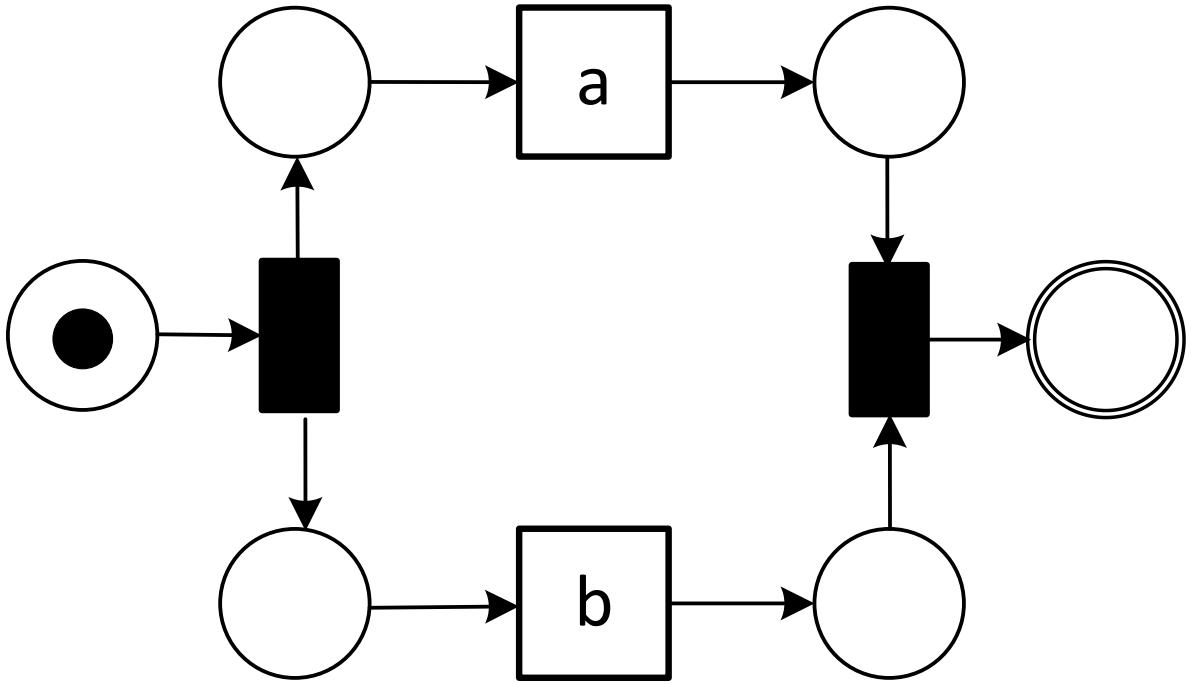
Another example

$$L_2 = [\langle a, b, c, e\rangle^{50}, \langle a, c, b, e\rangle^{40}, \langle a, b, c, d, b, c, e\rangle^{30}, \langle a, c, b, d, b, c, e\rangle^{20}, \langle a, b, c, d, c, b, e\rangle^{10}, \langle a, c, b, d, c, b, d, b, c, e\rangle^{10}]$$



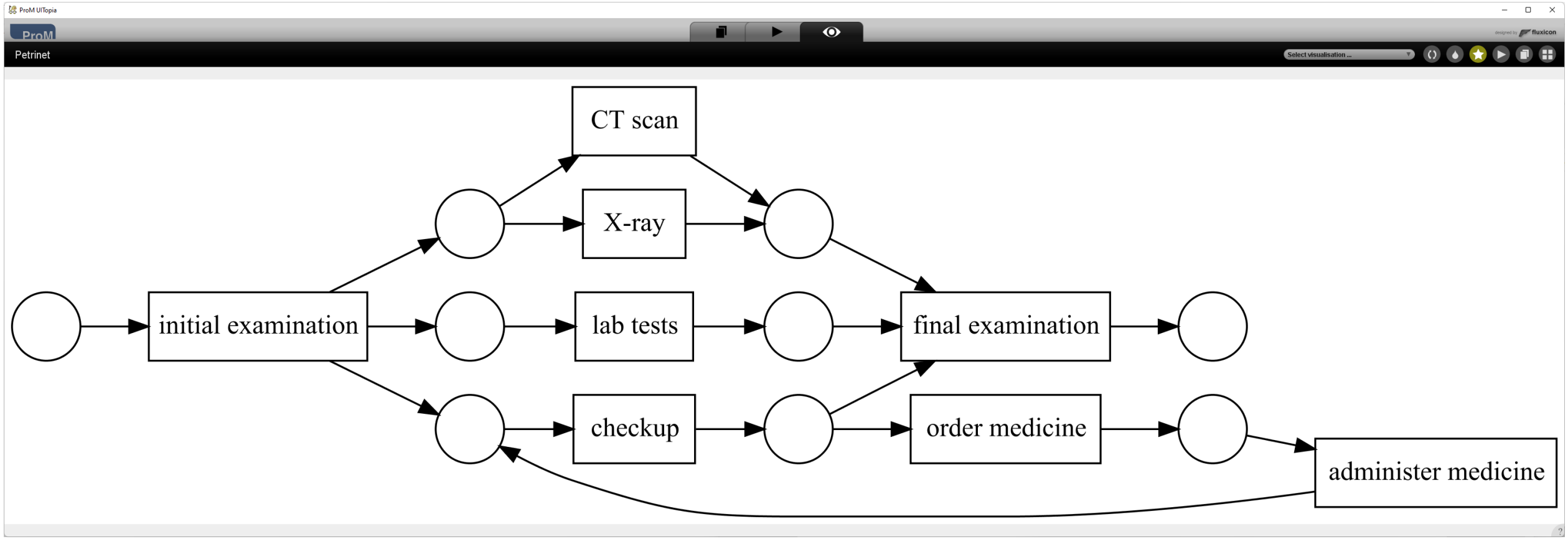
Another example

$$L_4 = [\langle a, b \rangle^{35}, \langle b, a \rangle^{15}]$$



Illustrates why it sometimes makes sense to add an artificial start and end.

Example in ProM
1856 cases, 11761 events, 197 variants



Chair of Process and Data Science

The Alpha algorithm is very simplistic
(mostly used for didactical reasons)

- Cannot handled certain constructs
- Weaker guarantees
- However, it illustrates the idea of process discovery.
- There exist various stronger bottom-up techniques:
  - State-based regions
  - Language-based regions
  - eST miner
  - Hybrid miner

Definition 22 (Alpha Algorithm). *The alpha algorithm* $disc_{alpha} \in \mathcal{B}(\mathcal{U}_{act}^*) \rightarrow \mathcal{U}_{AN}$ *returns an accepting Petri net* $disc_{alpha}(L)$ *for any event log* $L \in \mathcal{B}(\mathcal{U}_{act}^*)$. *Let* $A = act(L)$ *and* $fp(L) = fp(disc_{DFG}(L))$ *the footprint of event log* L. *This allows us to write* $a_1 \rightarrow_L a_2$ *if* $fp(L)((a_1, a_2)) = \rightarrow$ *and* $a_1 \#_L a_2$ *if* $fp(L)((a_1, a_2)) = \#$ *for any* $a_1, a_2 \in A' = A \cup \{\blacktriangleright, \blacksquare\}$.

1. $Cnd = \{(A_1, A_2) \mid A_1 \subseteq A' \wedge A_1 \neq \emptyset \wedge A_2 \subseteq A' \wedge A_2 \neq \emptyset \wedge \forall_{a_1 \in A_1} \forall_{a_2 \in A_2} a_1 \rightarrow_L a_2 \wedge \forall_{a_1, a_2 \in A_1} a_1 \#_L a_2 \wedge \forall_{a_1, a_2 \in A_2} a_1 \#_L a_2\}$ *are the candidate places,*
2. $Sel = \{(A_1, A_2) \in Cnd \mid \forall_{(A_1', A_2') \in Cnd} A_1 \subseteq A_1' \wedge A_2 \subseteq A_2' \Longrightarrow (A_1, A_2) = (A_1', A_2')\}$ *are the selected maximal places,*
3. $P = \{p_{(A_1, A_2)} \mid (A_1, A_2) \in Sel\} \cup \{p_{\blacktriangleright}, p_{\blacksquare}\}$ *is the set of all places,*
4. $T = \{t_a \mid a \in A'\}$ *is the set of transitions,*
5. $F = \{(t_a, p_{(A_1, A_2)}) \mid (A_1, A_2) \in Sel \wedge a \in A_1\} \cup \{(p_{(A_1, A_2)}, t_a) \mid (A_1, A_2) \in Sel \wedge a \in A_2\} \cup \{(p_{\blacktriangleright}, t_{\blacktriangleright}), (t_{\blacksquare}, p_{\blacksquare})\}$ *is the set of arcs,*
6. $l = \{(t_a, a) \mid a \in A\}$ *is the labeling function,*
7. $M_{init} = [p_{\blacktriangleright}]$ *is the initial marking,* $M_{final} = [p_{\blacksquare}]$ *is the final marking, and*
8. $disc_{alpha}(L) = ((P, T, F, l), M_{init}, M_{final})$ *is the discovered accepting Petri net.*

Chair of Process and Data Science
RWTH AACHEN UNIVERSITY

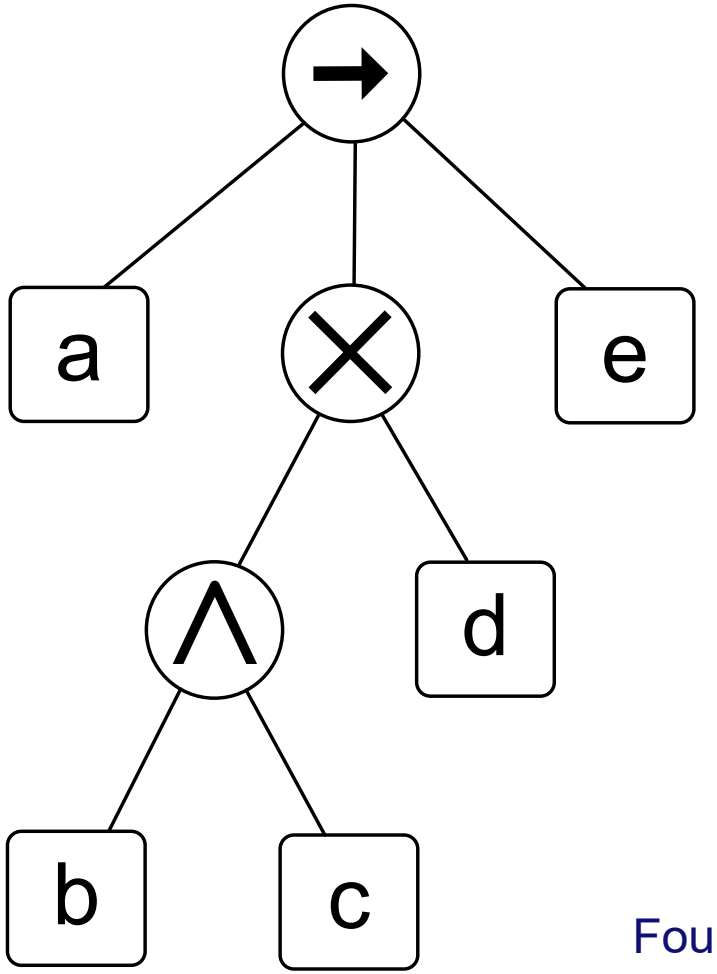
Top-down discovery

- **Divide and conquer.**
- **Split the problem recursively into smaller problems such that things get trivial.**
- **An example is the Inductive Mining (IM) technique:**
  - **Uses process trees.**
  - **The leading approach**
  - **Implemented in ProM, Celonis, and many other tools.**

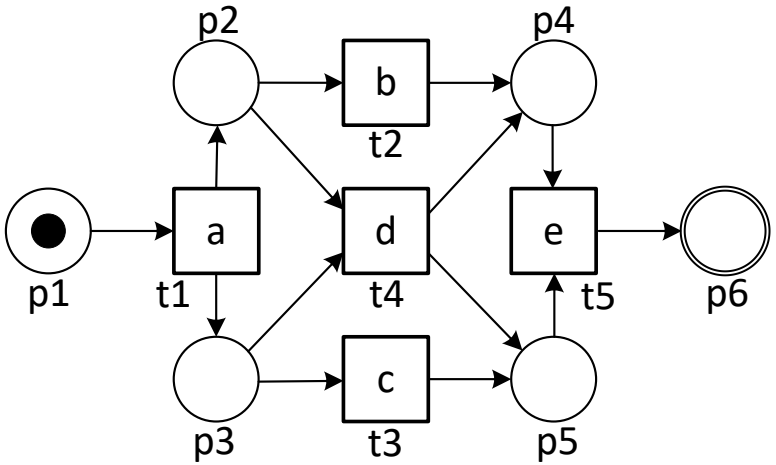
Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

A process tree

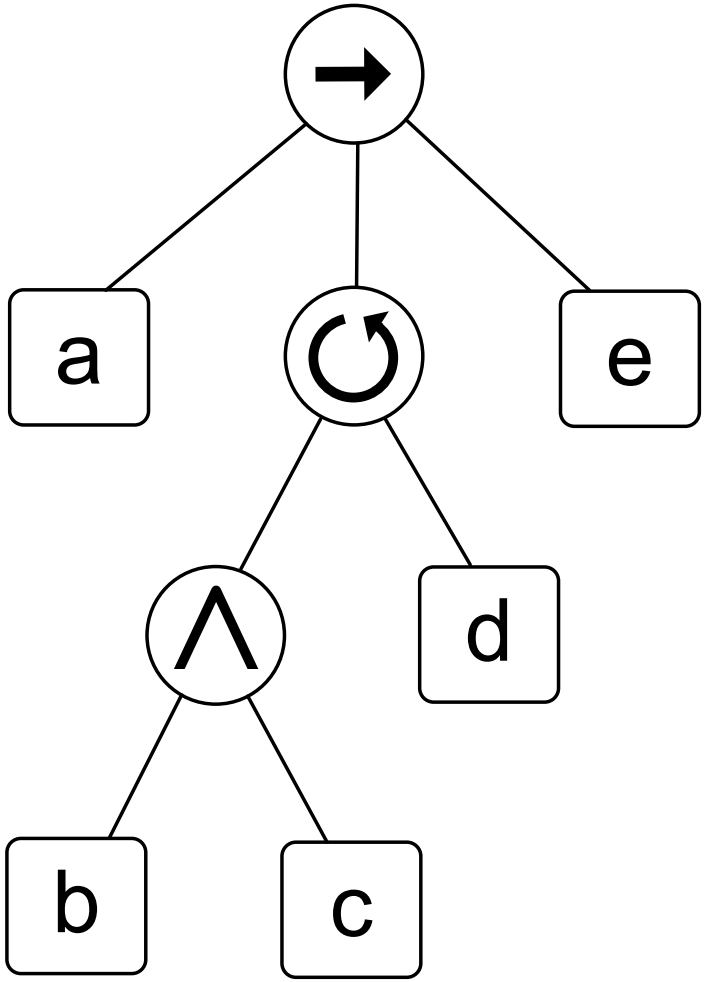


Semantics

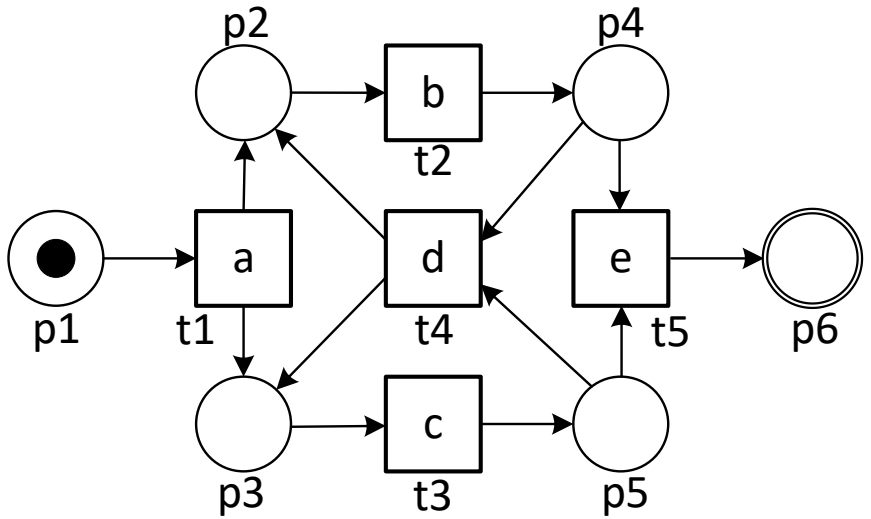


Four types of operators: → (sequential composition), × (exclusive choice), ∧ (parallel composition), and ↺ (redo loop).

Another process tree

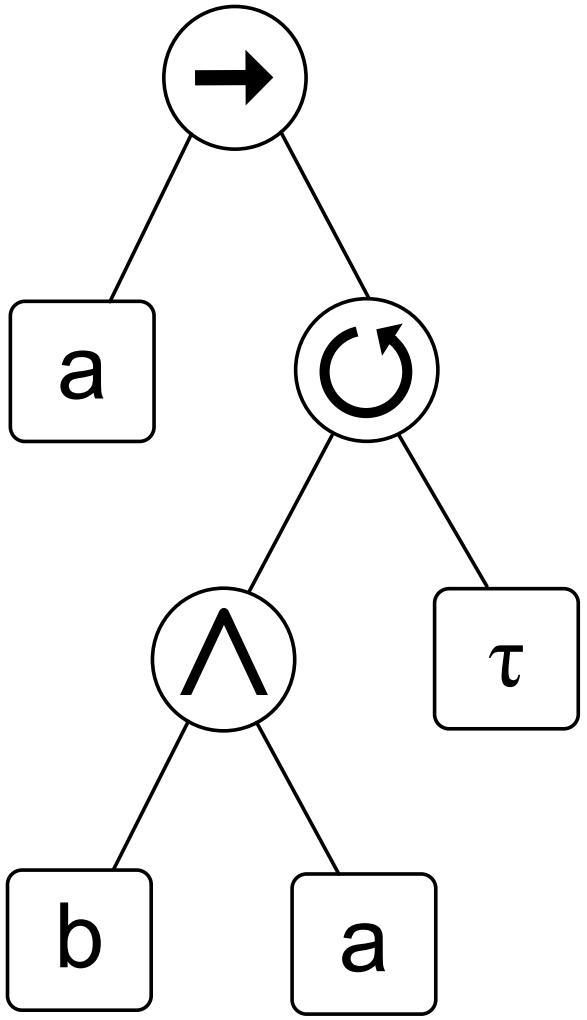


Semantics

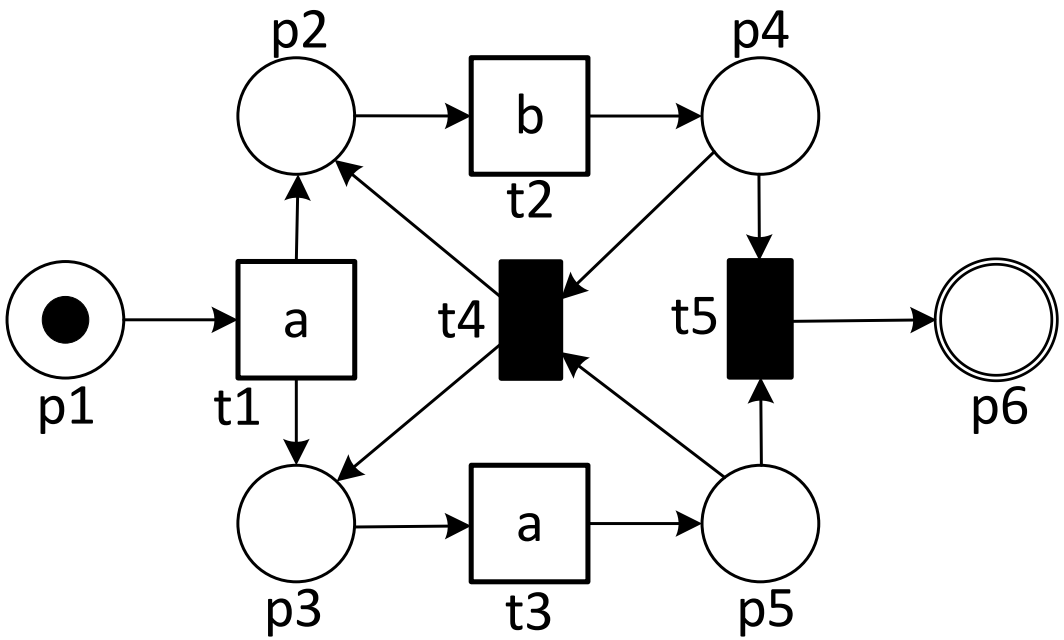


Chair of Process and Data Science

Another process tree



Semantics



Chair of Process and Data Science
RWTH AACHEN UNIVERSITY

Inductive Mining (IM)

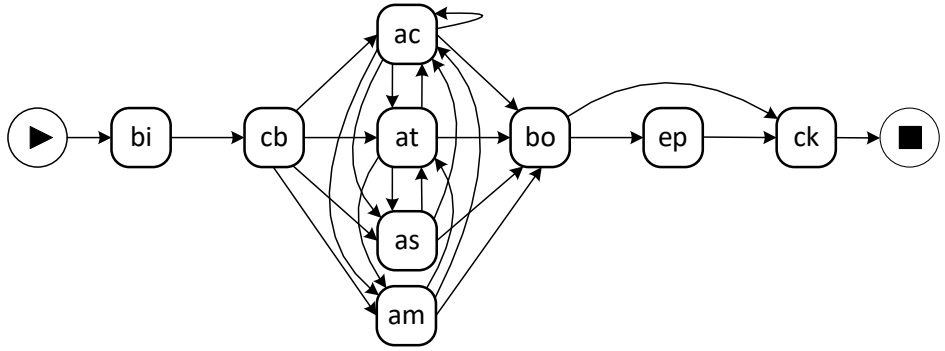
- Decompose the event log into smaller events logs until the problem get trivial.
- Four types of cuts corresponding to the operators: → (sequential composition), × (exclusive choice), ∧ (parallel composition), and ↺ (redo loop).
- In each step the activities are partitioned into subsets until they are singletons.
- Developed by Sander Leemans in the context of his PhD thesis (NWO project "Don't Search for the Undesirable! Avoiding "Blind Alleys" in Process Mining" 2012-2017)

Event log

Activities: buy ingredients (bi), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

Create a DFG for the whole event log

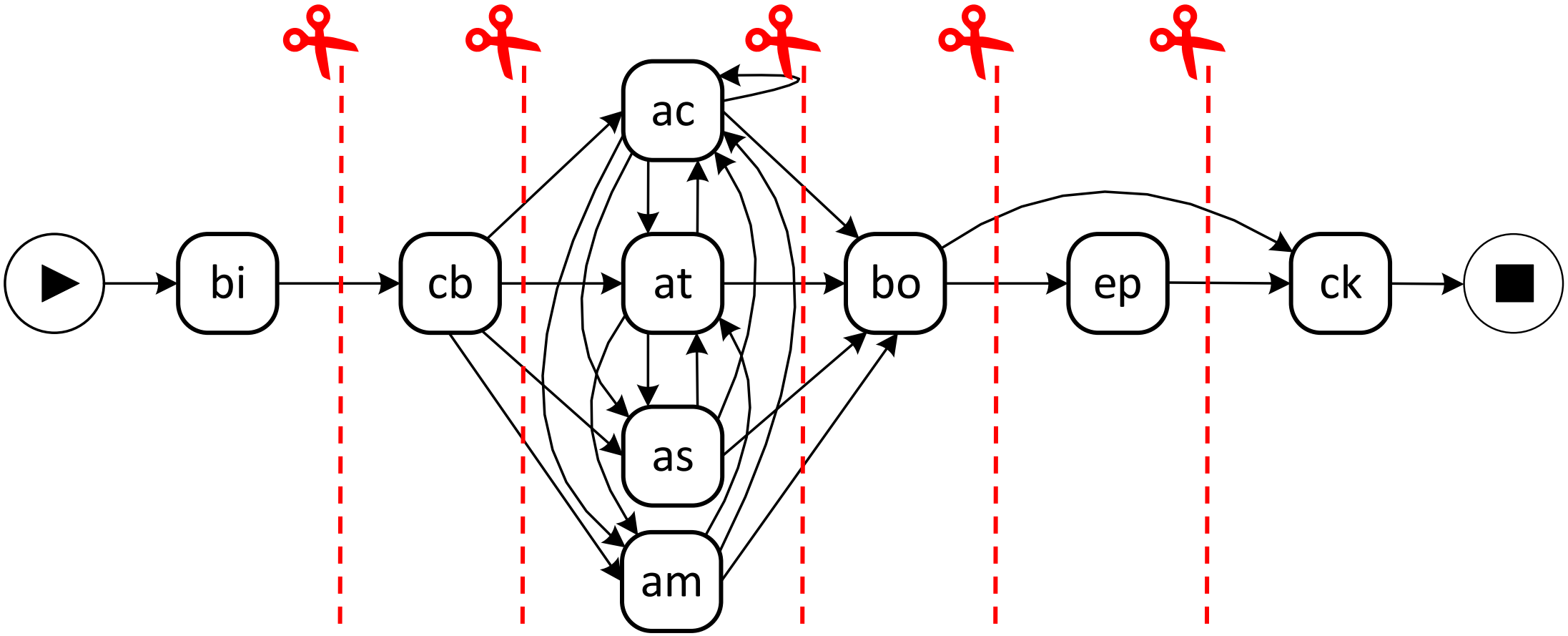
bi cb ac at as bo ep ck
bi cb ac as at bo ep ck
bi cb at ac as bo ep ck
bi cb at as ac bo ck
bi cb as ac at bo ep ck
bi cb as at ac bo ep ck
bi cb ac at am bo ep ck
bi cb ac am at bo ep ck
bi cb at ac am bo ck
bi cb at am ac bo ep ck
bi cb am ac at bo ep ck
bi cb am at ac bo ep ck
bi cb at as ac ac bo ep ck
bi cb as ac ac at ac bo ck
bi cb as at ac ac bo ep ck



Frequencies omitted for readability

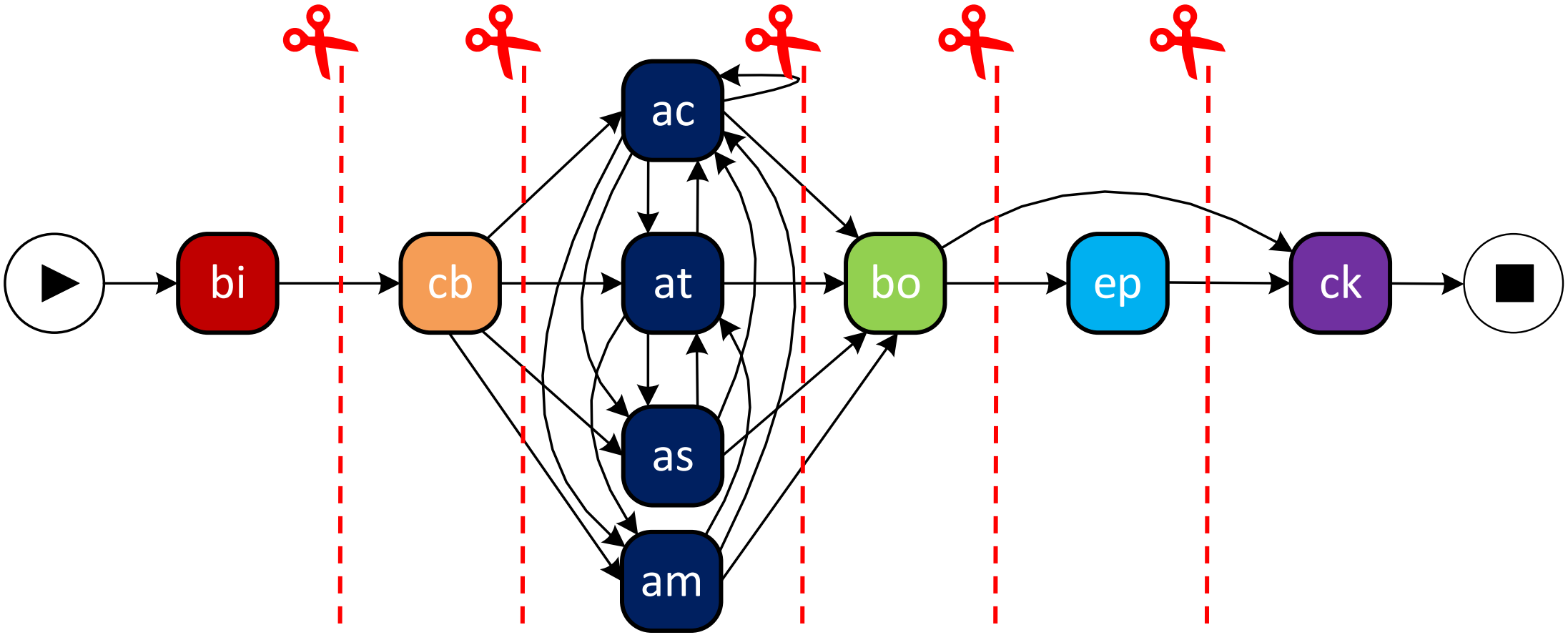
Chair of Process and Data Science

Apply a sequence cut



There is a sequence cut when the DFG can be split into sequential parts where only “forward connections” are possible. Note that we need to use the non-reflexive transitive closure of F.

Sequence cut partitions activities in six subsets

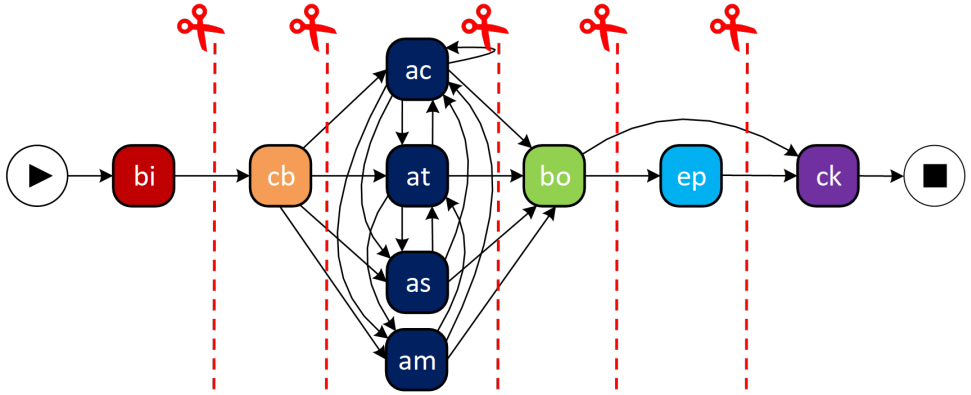


Chair of Process and Data Science

Color the events based on the partitioning

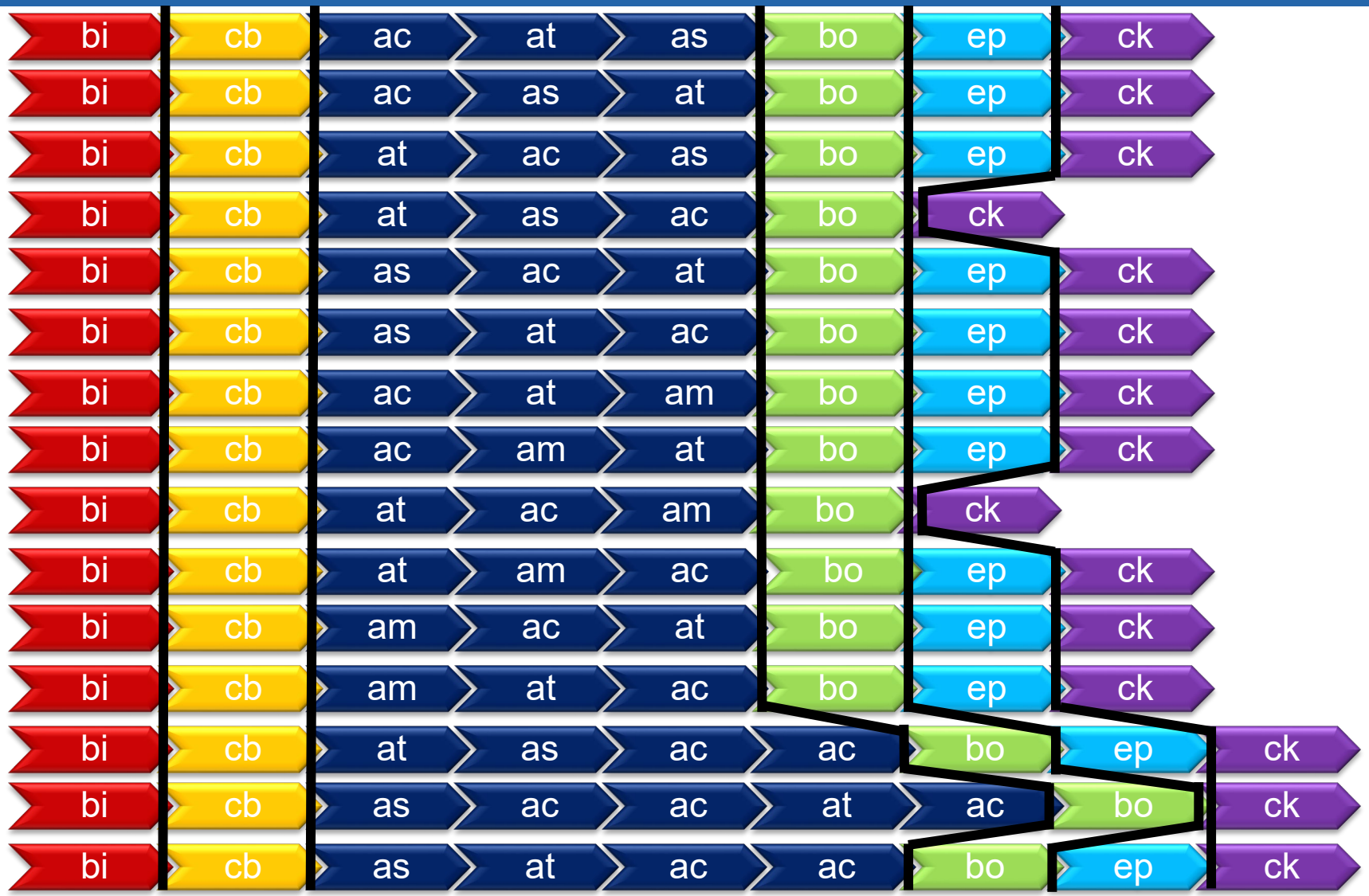
bi cb ac at as bo ep ck
bi cb ac as at bo ep ck
bi cb at ac as bo ep ck
bi cb at as ac bo ck
bi cb as ac at bo ep ck
bi cb as at ac bo ep ck
bi cb ac at am bo ep ck
bi cb ac am at bo ep ck
bi cb at ac am bo ck
bi cb at am ac bo ep ck
bi cb am ac at bo ep ck
bi cb am at ac bo ep ck
bi cb at as ac ac bo ep ck
bi cb as ac ac at ac bo ck
bi cb as at ac ac bo ep ck

Sequence cut



Chair of Process and Data Science

Split the event log based on the partitioning

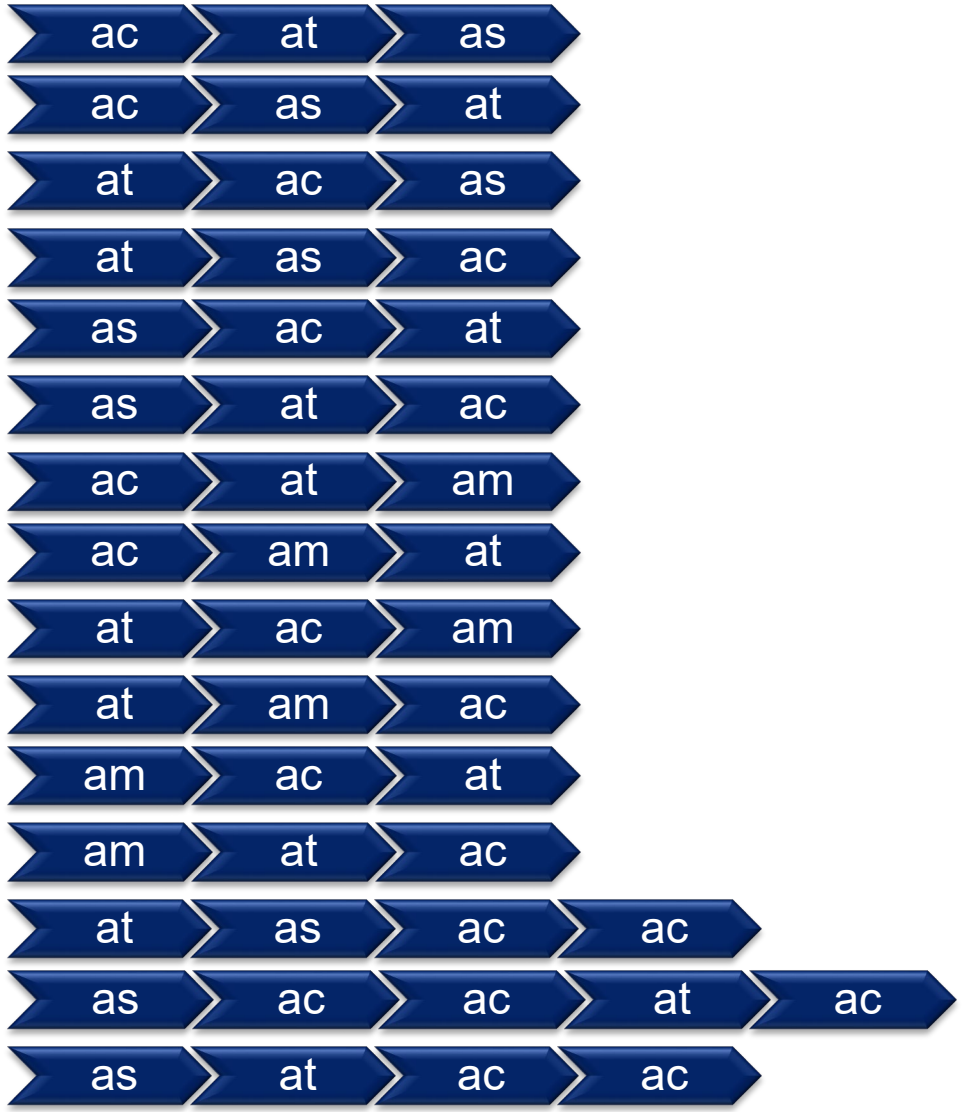


Chair of Process and Data Science

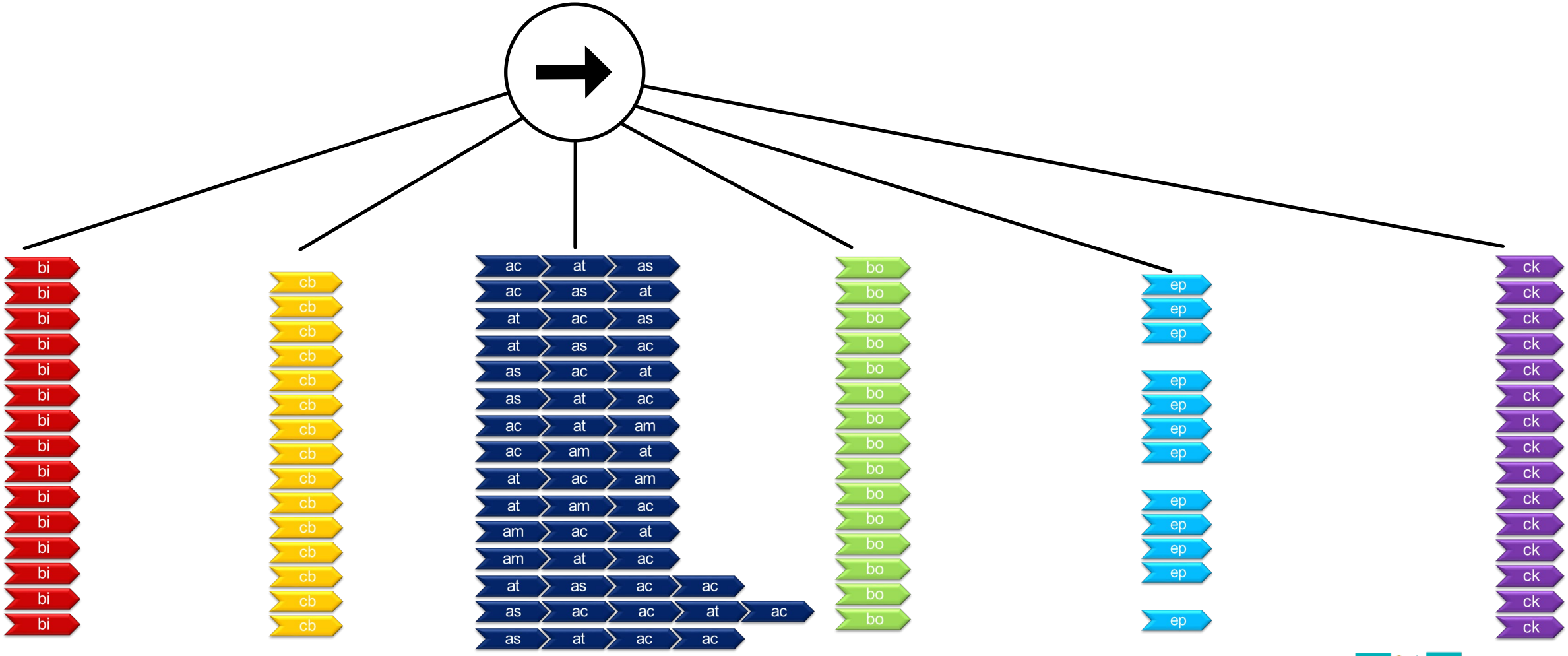
Five of the projected event logs refer to a single activity (base case)

Chair of Process and Data Science

The blue group has four activities



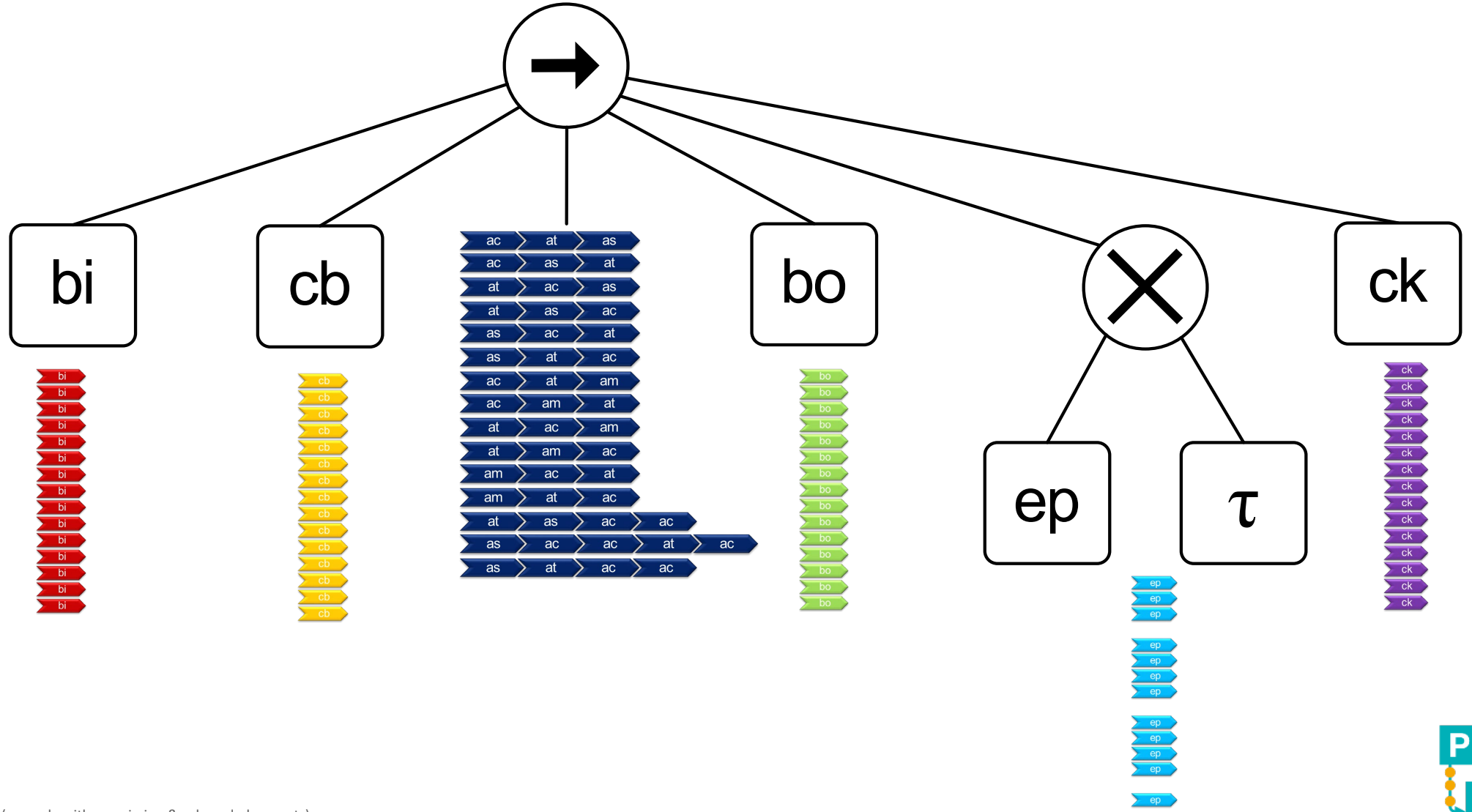
Recursion: Apply algorithm to all sublogs



Five of the projected event logs refer to a single activity (base case).

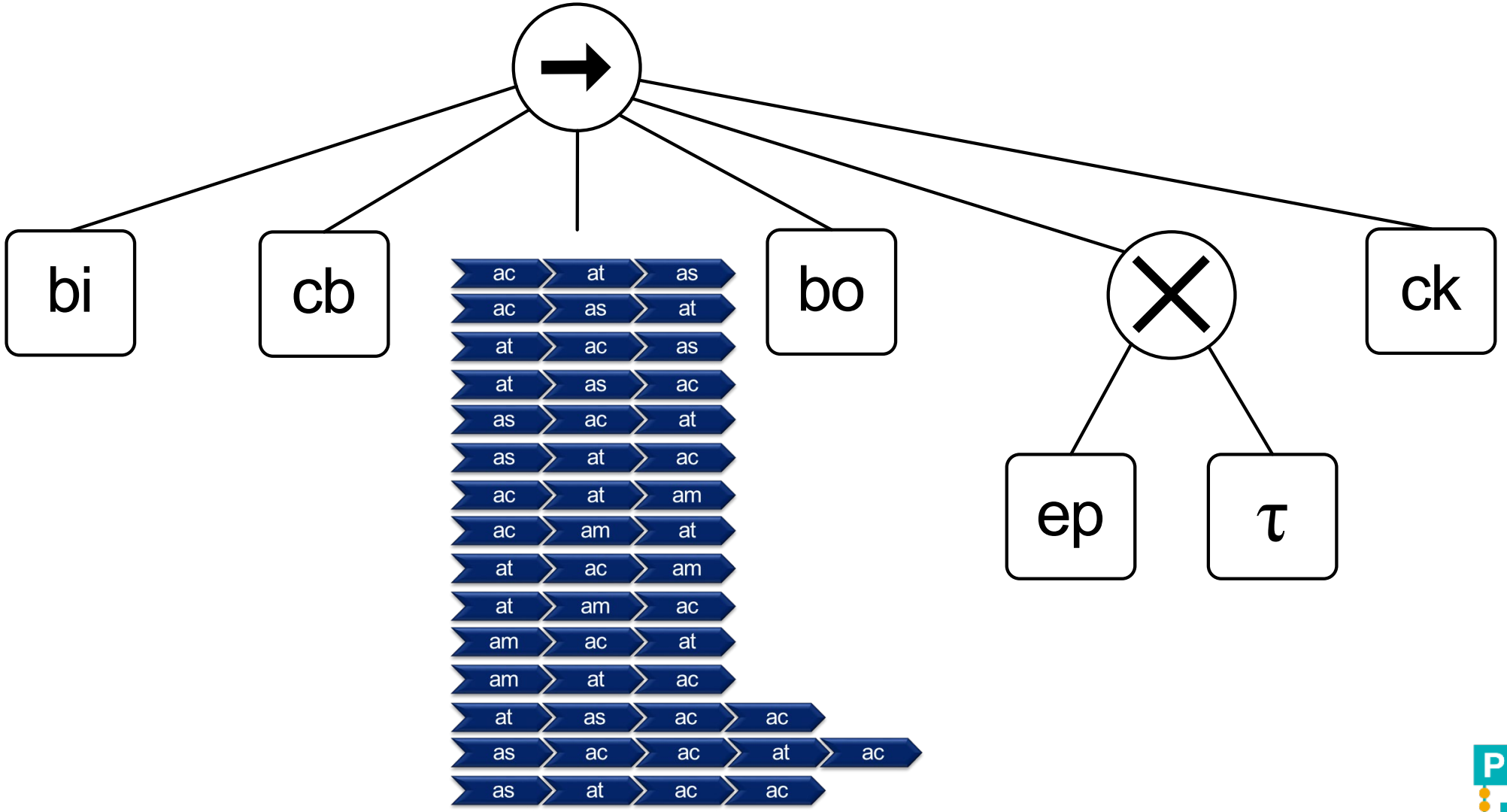
Chair of Process and Data Science

Handling the base cases (ep can be skipped)



Chair of Process and Data Science

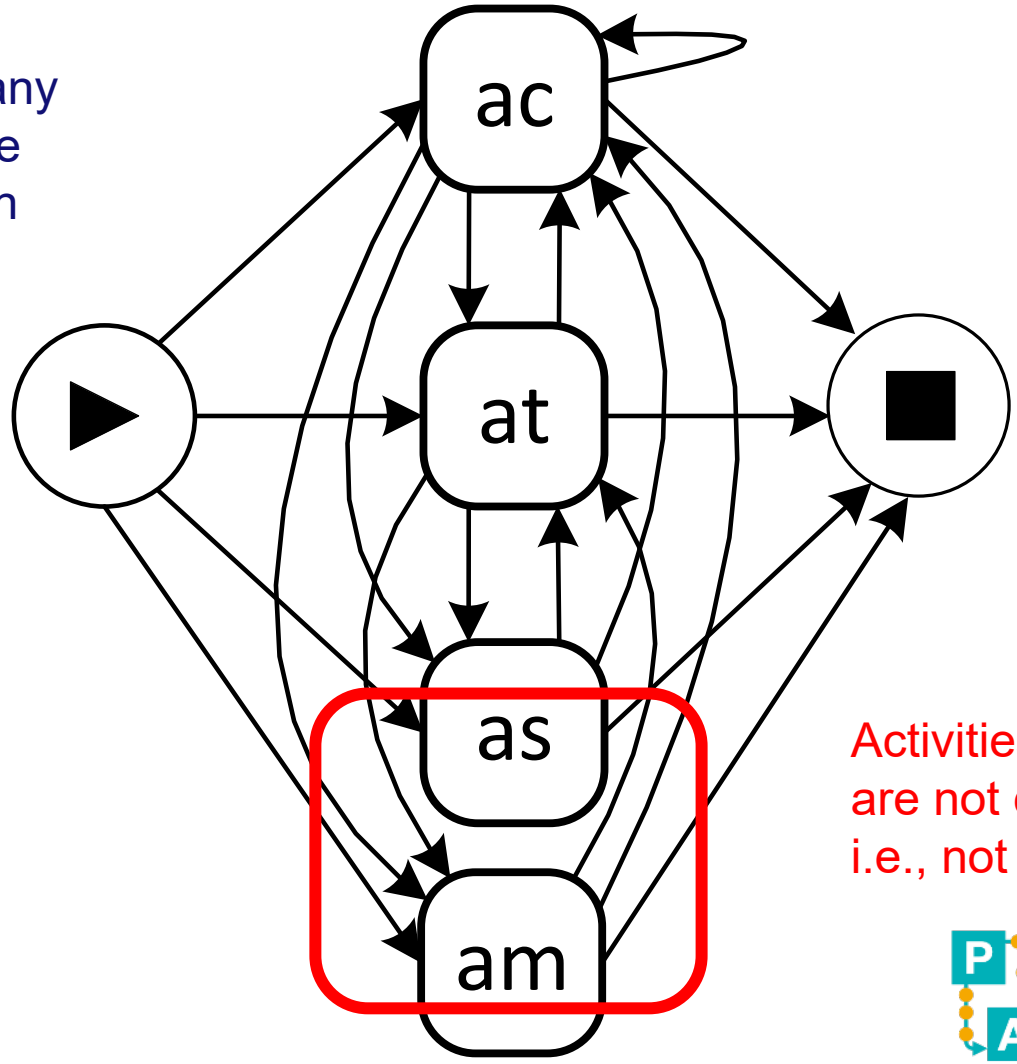
Only the blue event log remains



Continue with the blue event log

ac at as
ac as at
at ac as
at as ac
as ac at
as at ac
ac at am
ac am at
at ac am
at am ac
am ac at
am at ac
at as ac ac
as ac ac at ac
as at ac ac

There is a parallel cut when the DFG can be split into concurrent parts where any activity in one part can be followed by any activity in another part.



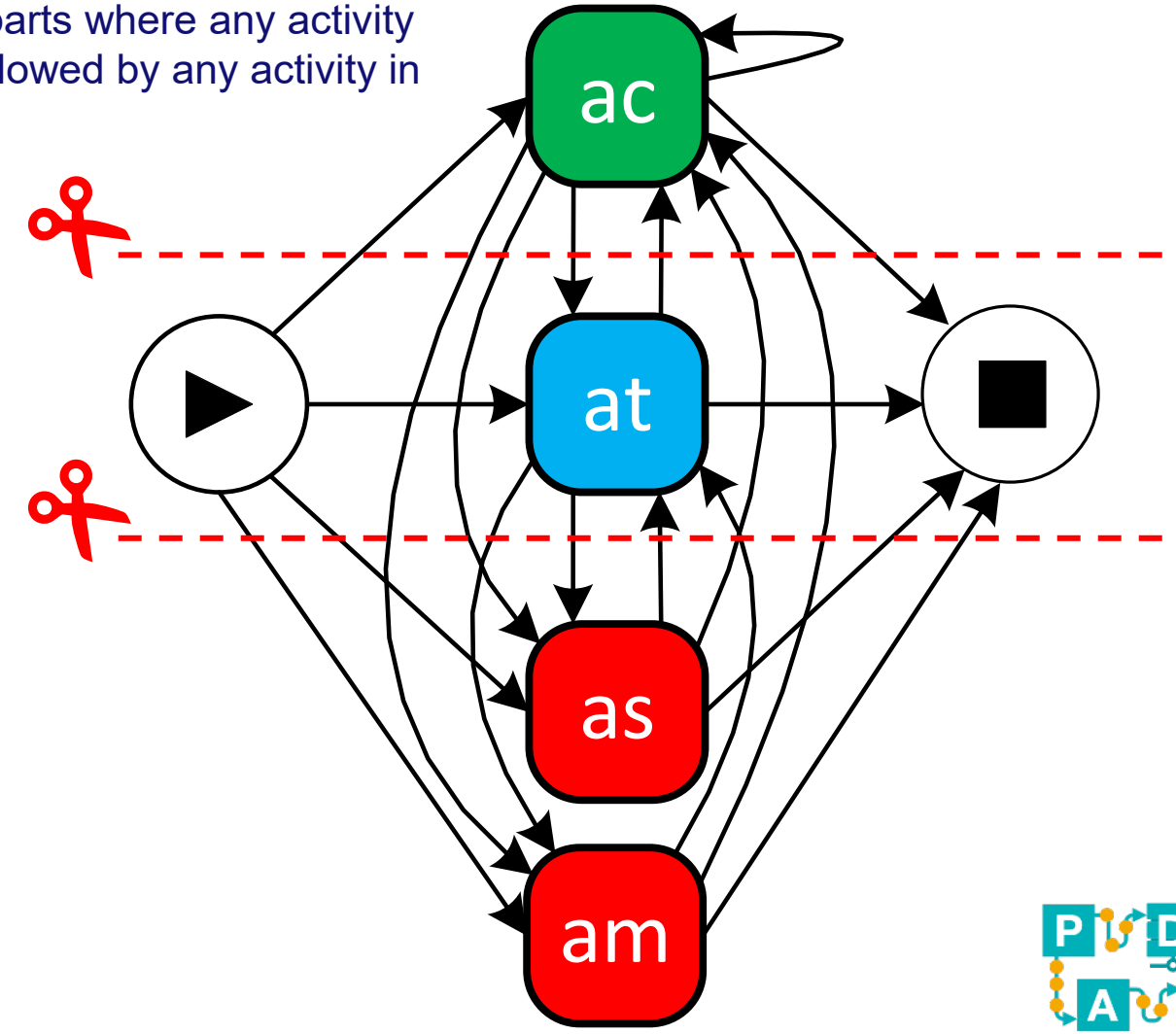
Activities as and am are not connected, i.e., not concurrent

Chair of Process and Data Science

Apply a parallel cut resulting in three activity groups

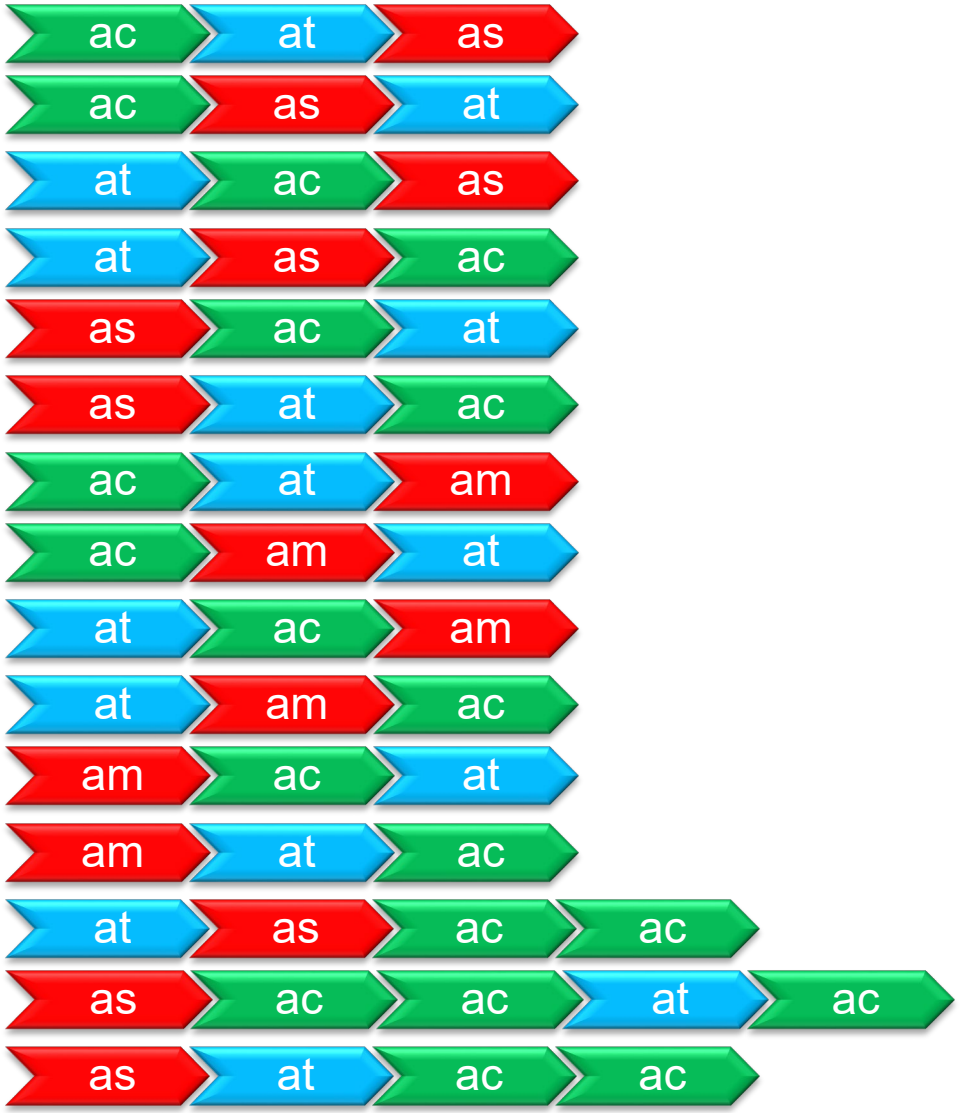
ac at as
ac as at
at ac as
at as ac
as ac at
as at ac
ac at am
ac am at
at ac am
at am ac
am ac at
am at ac
at as ac ac
as ac ac at ac
as at ac ac

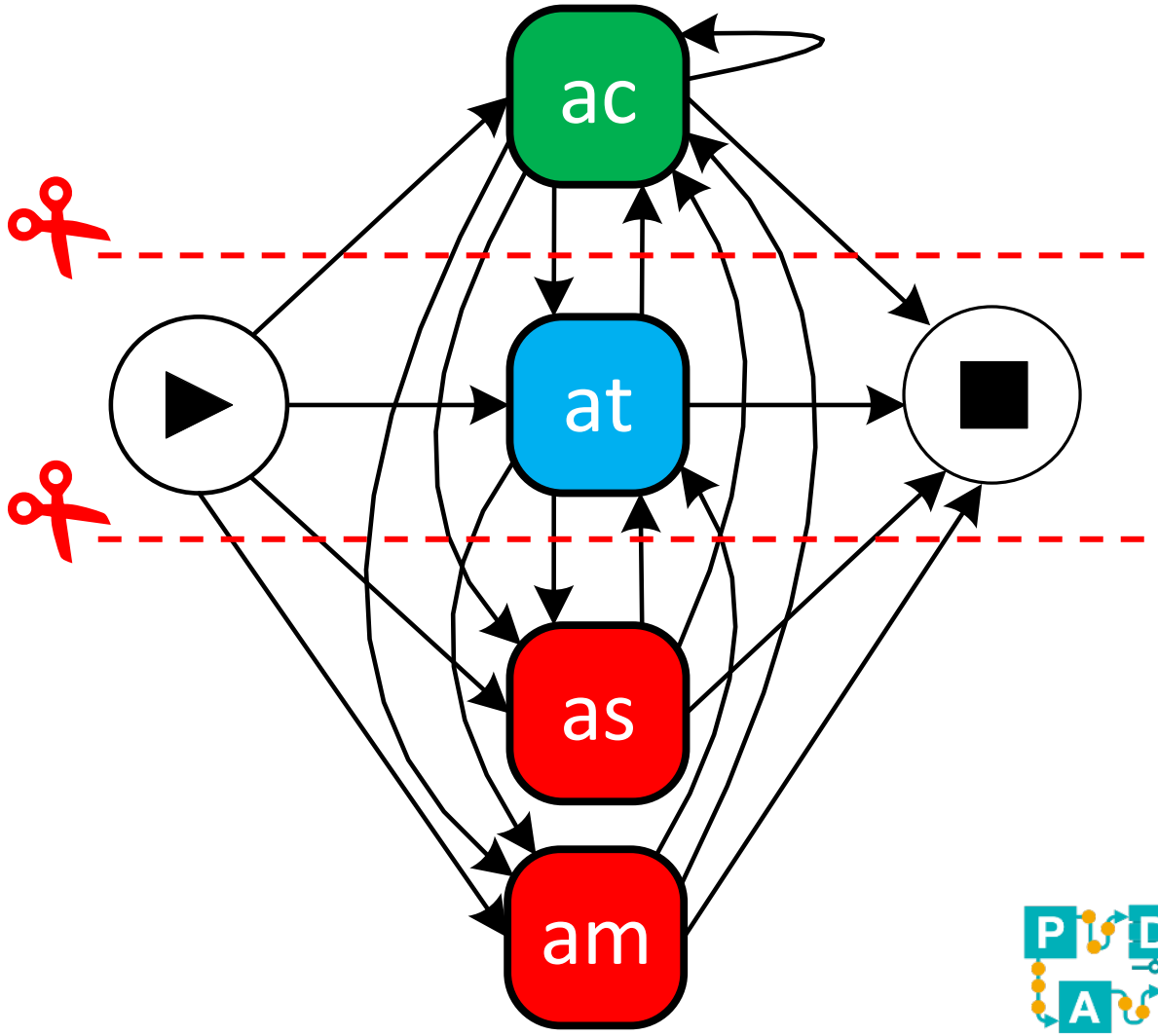
There is a parallel cut when the DFG can be split into concurrent parts where any activity in one part can be followed by any activity in another part.



Chair of Process and Data Science

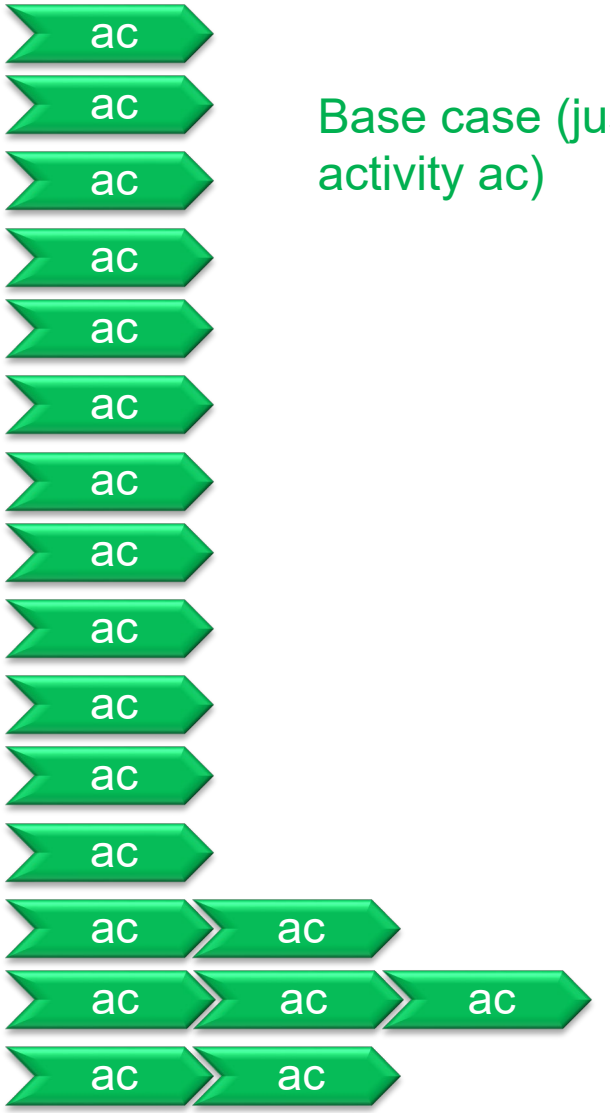
Apply a parallel cut resulting in three activity groups



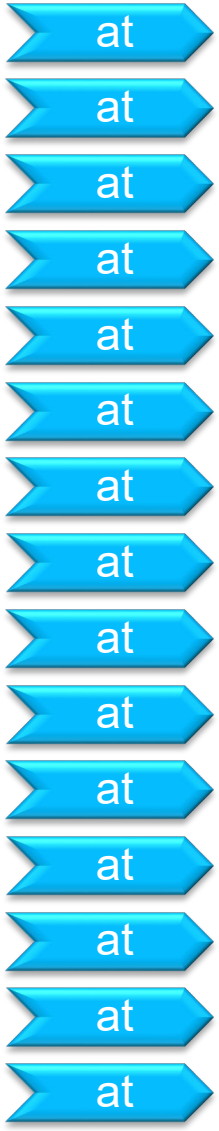


Chair of Process and Data Science

Three new event logs are created



Base case (just activity ac)

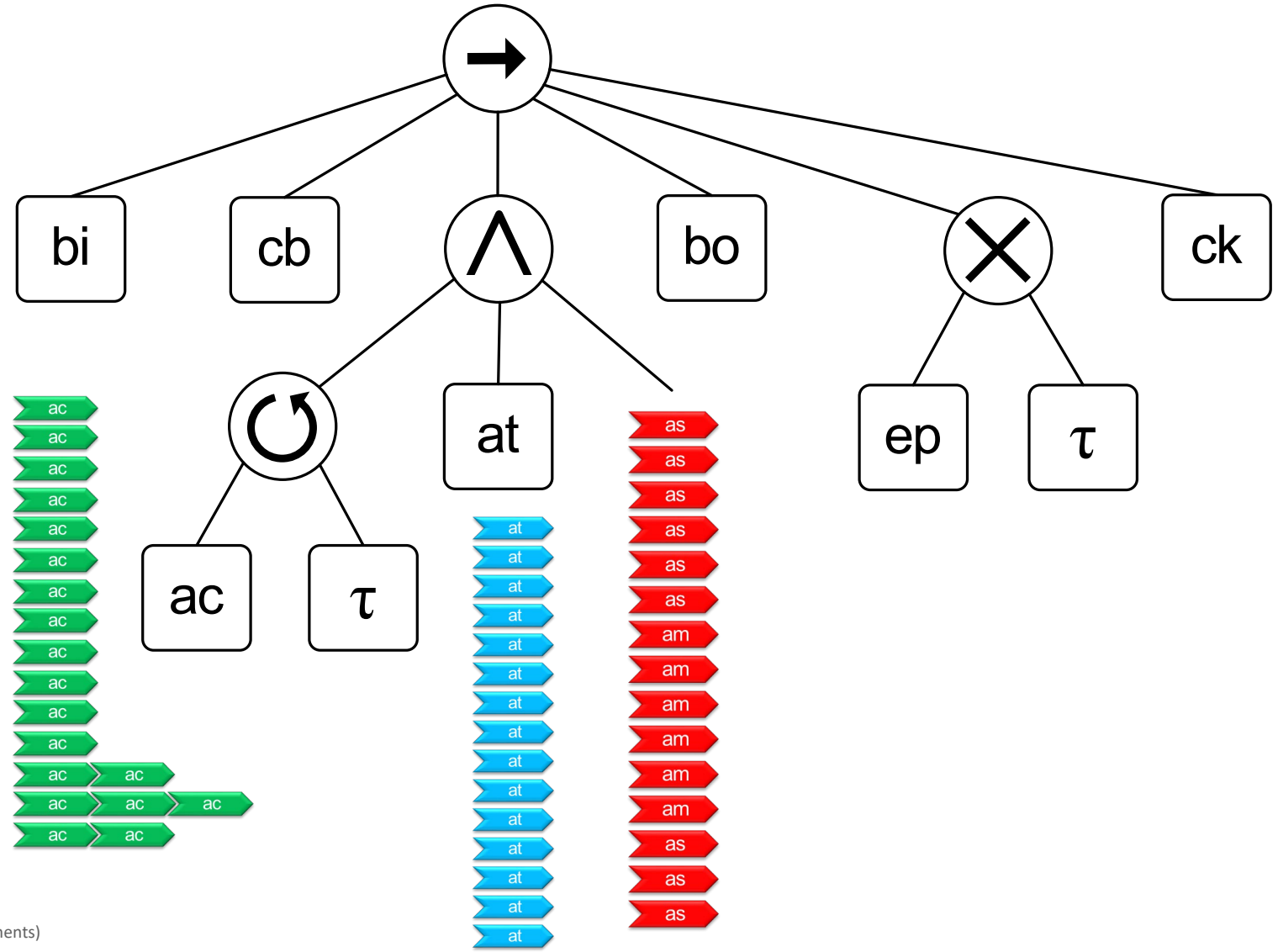


Base case (just activity at)

Not a base case, still two activities as and am.

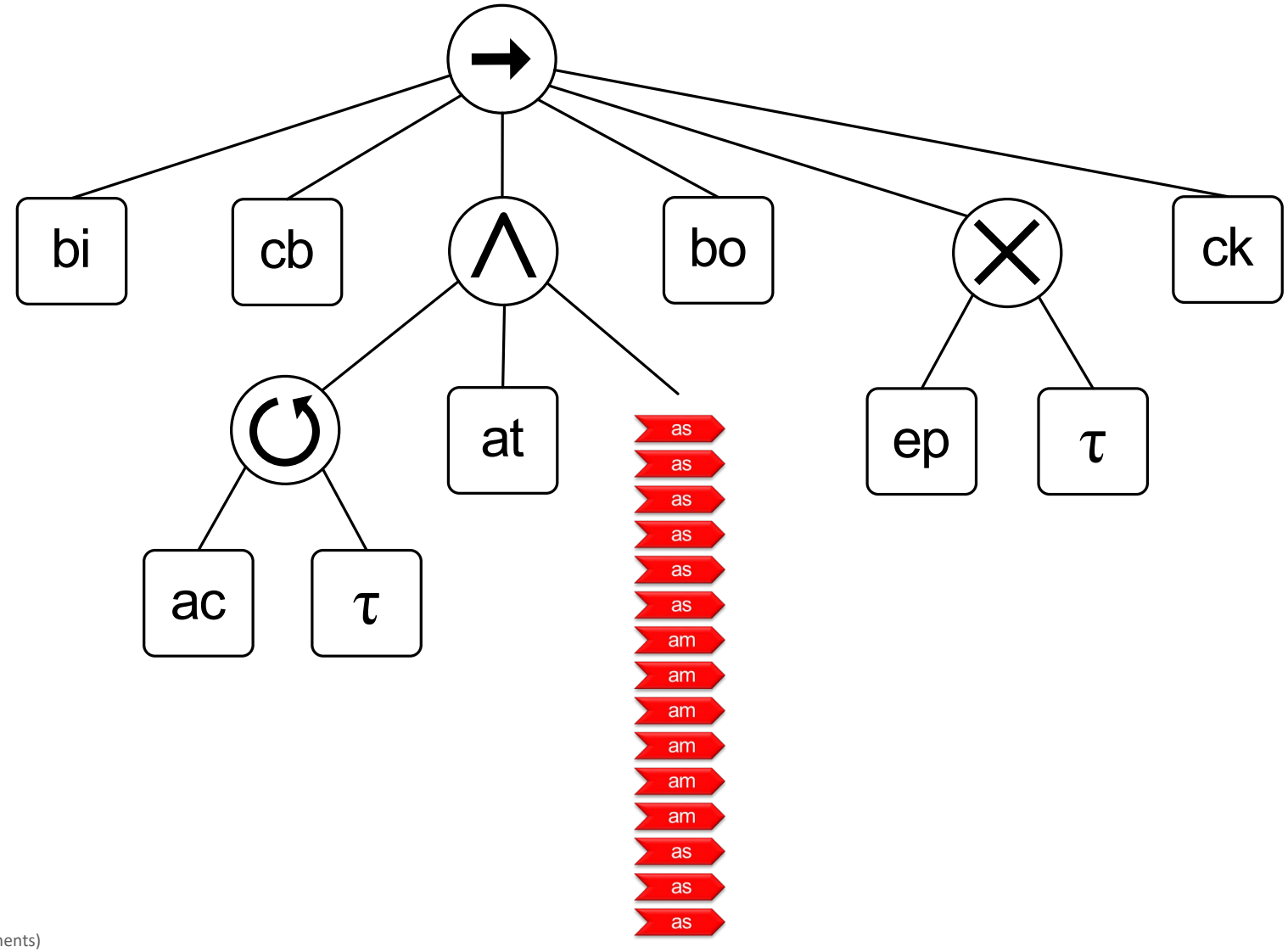
Chair of Process and Data Science

Handling the base cases (ac can be repeated)



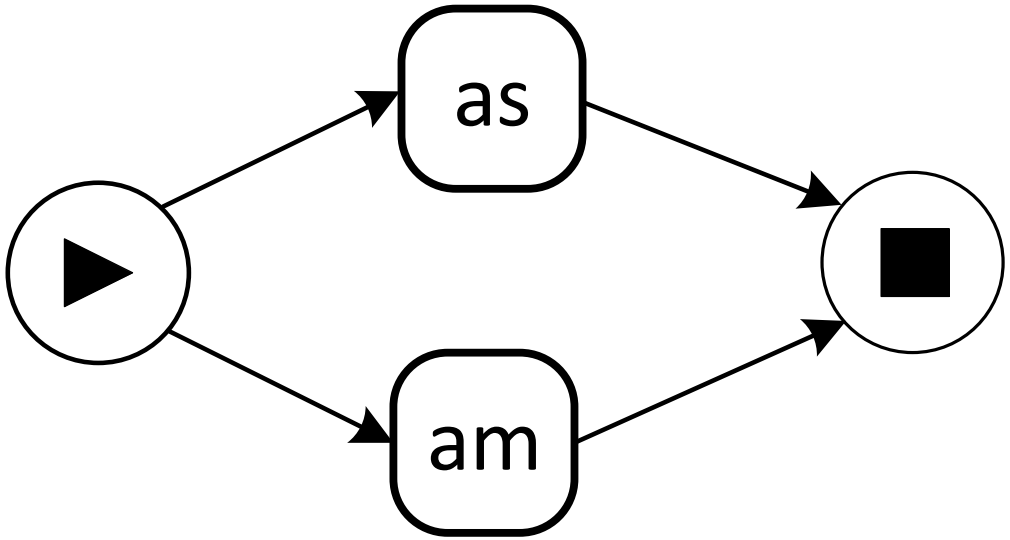
Chair of Process and Data Science

Only the red event log remains



Continue with the red event log

as
as
as
as
as
as
am
am
am
am
am
am
as
as
as

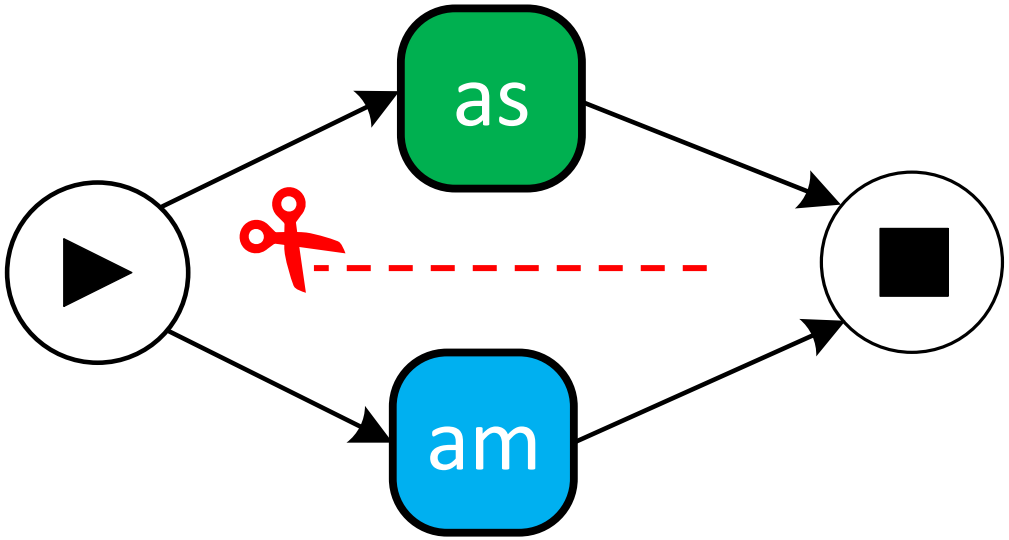


Chair of Process and Data Science

We find an exclusive-choice cut

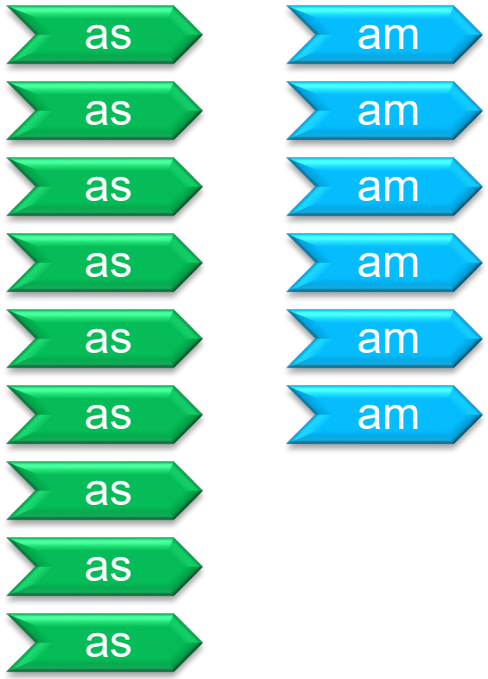
as
as
as
as
as
as
am
am
am
am
am
am
am
as
as
as

There is an exclusive-choice cut when the DFG can be split into disconnected parts after leaving out the artificial start and end.

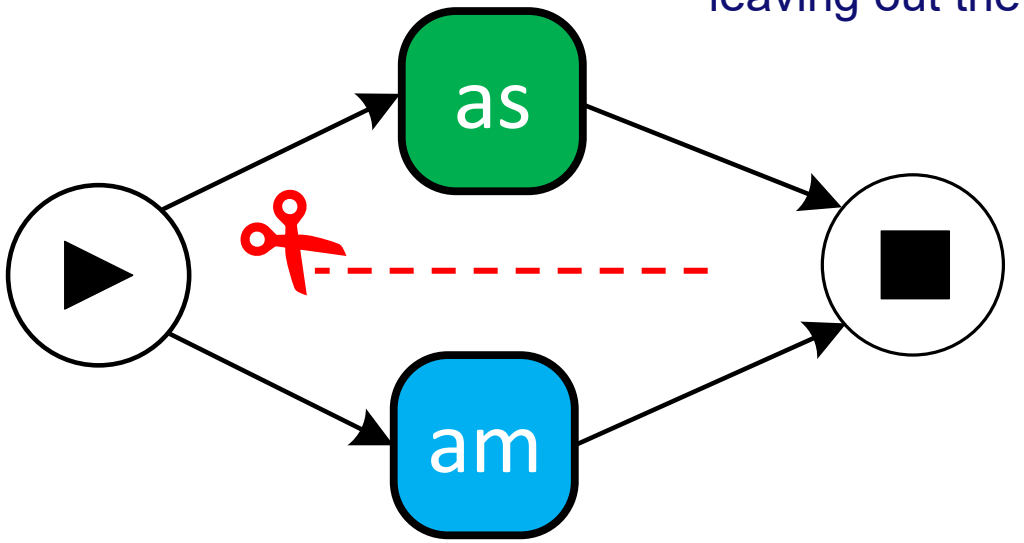


Chair of Process and Data Science

We find an exclusive-choice cut

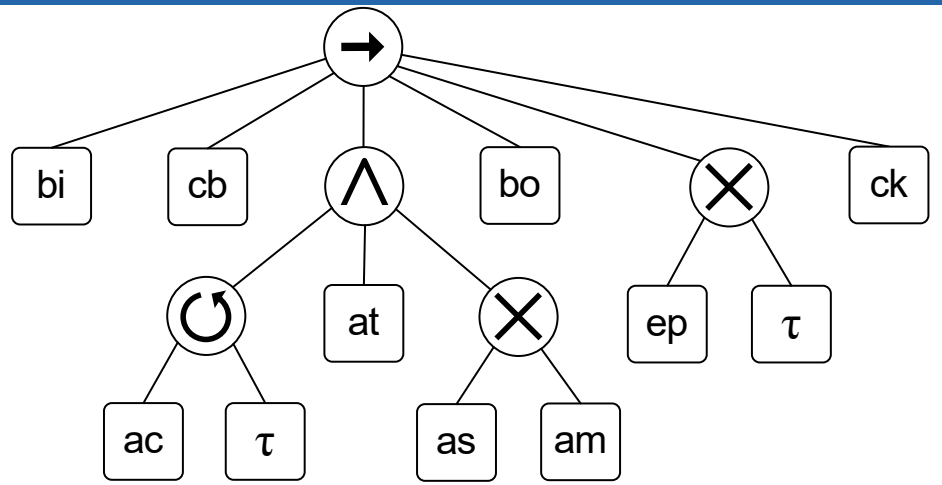


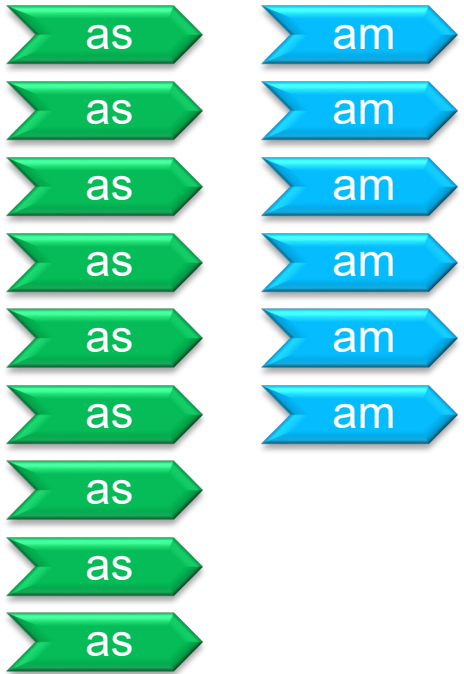
There is an exclusive-choice cut when the DFG can be split into disconnected parts after leaving out the artificial start and end.



Note that projection is now different than for the sequence and parallel cuts.

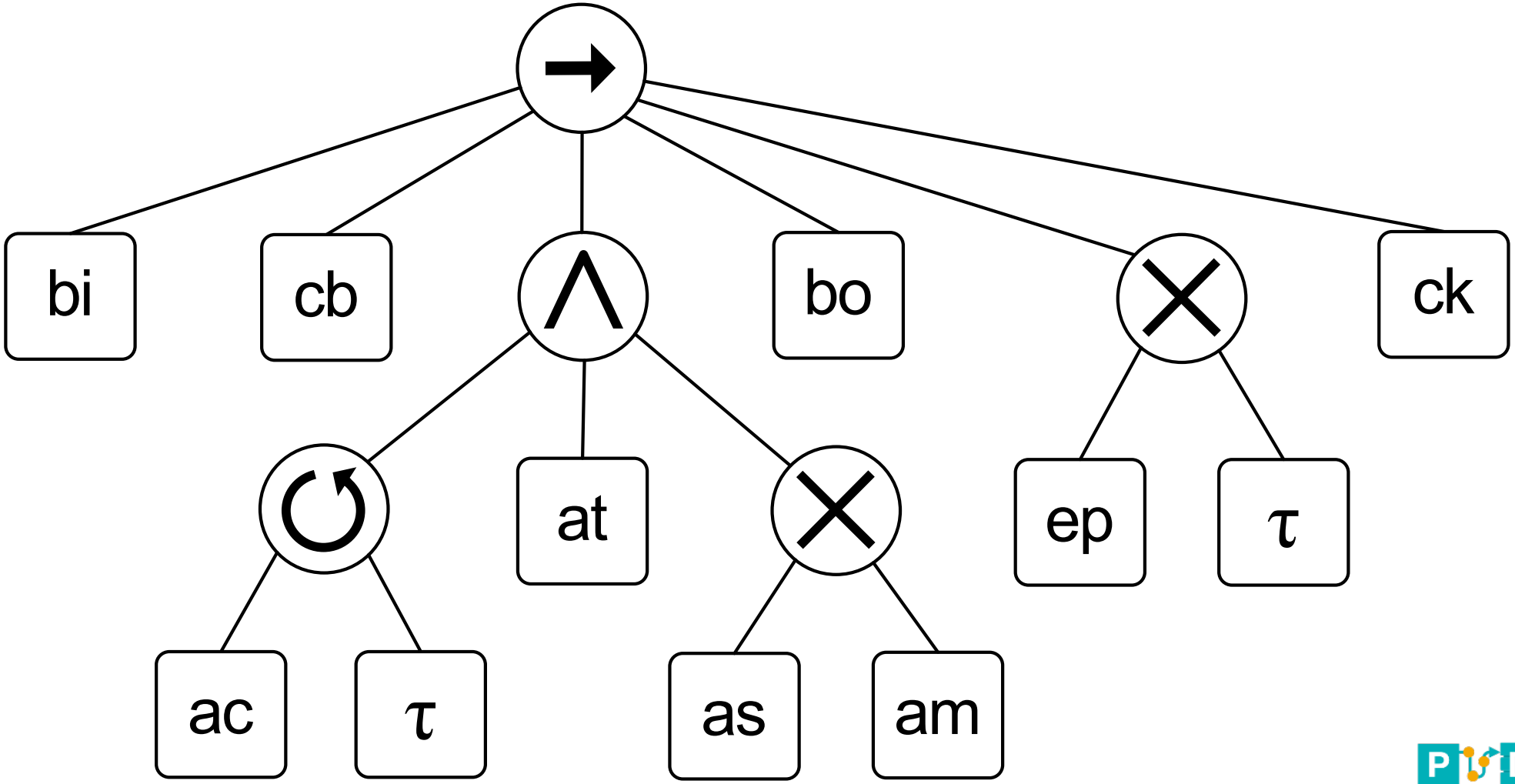
We end up with two base cases



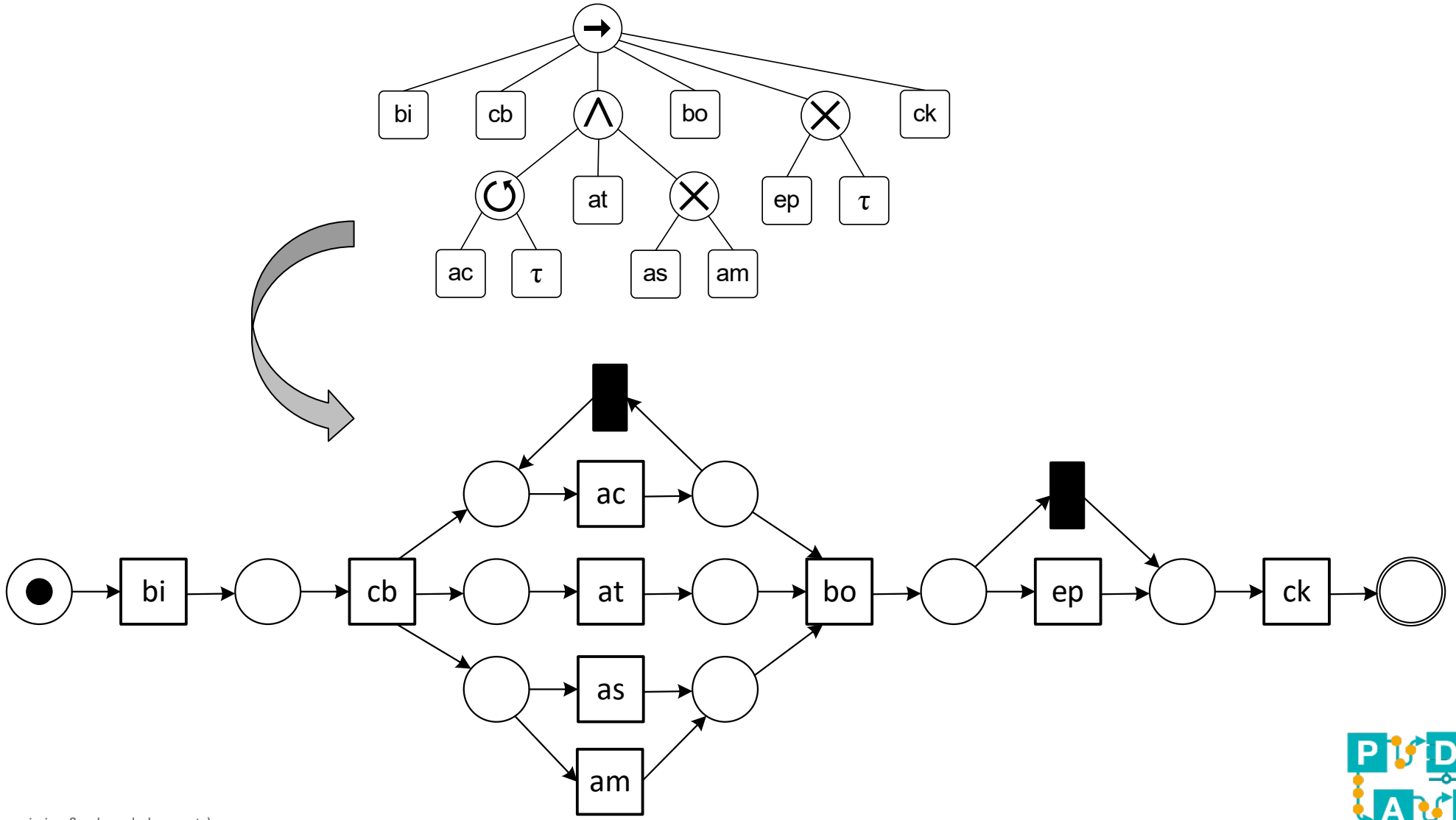


Chair of Process and Data Science

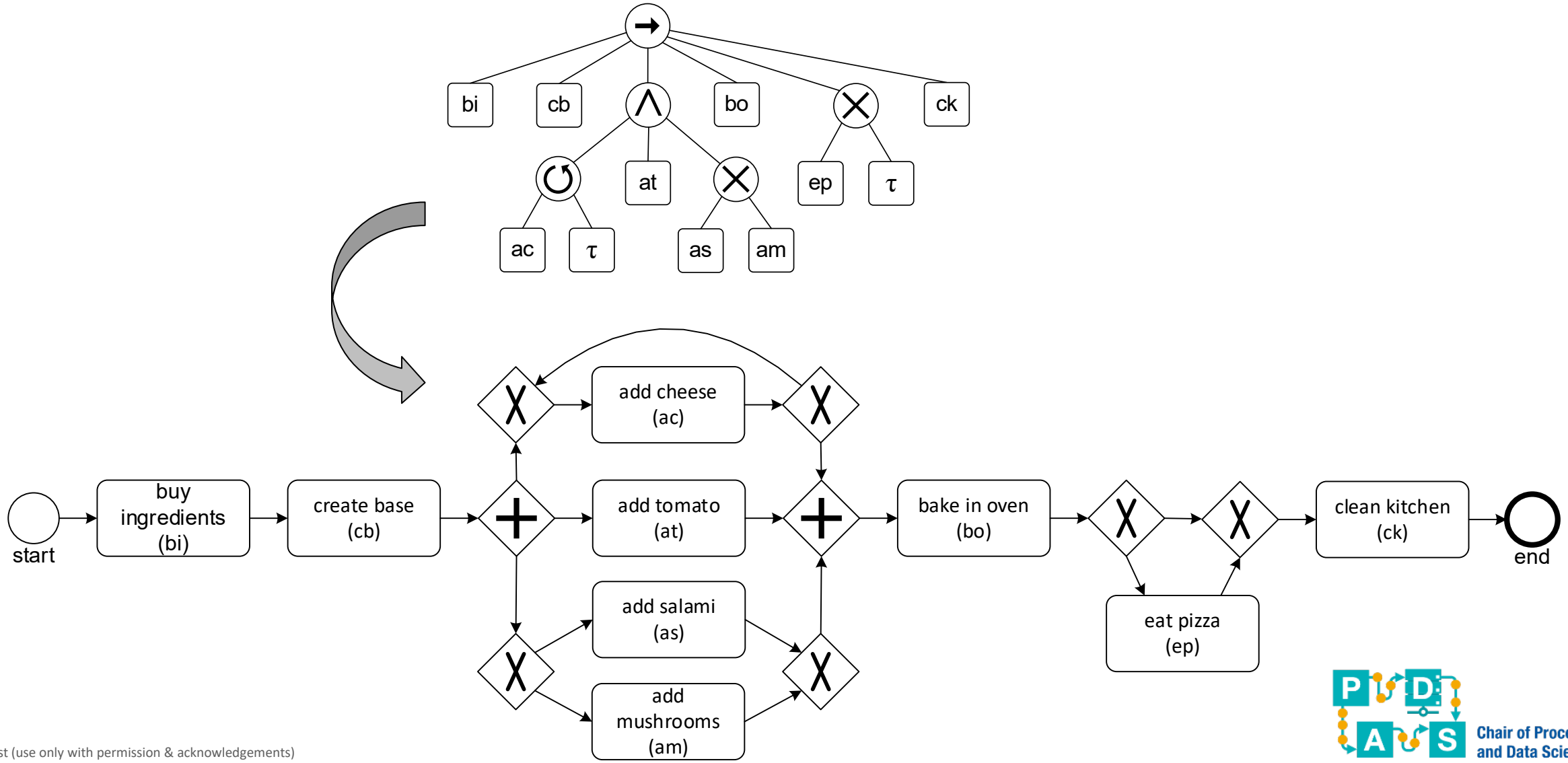
The process tree returned by the Inductive Mining algorithm



Can be visualized using Petri nets or BPMN



Can be visualized using Petri nets or BPMN

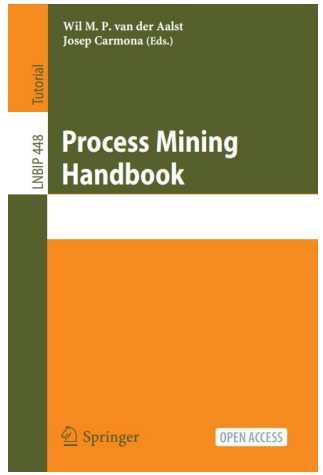


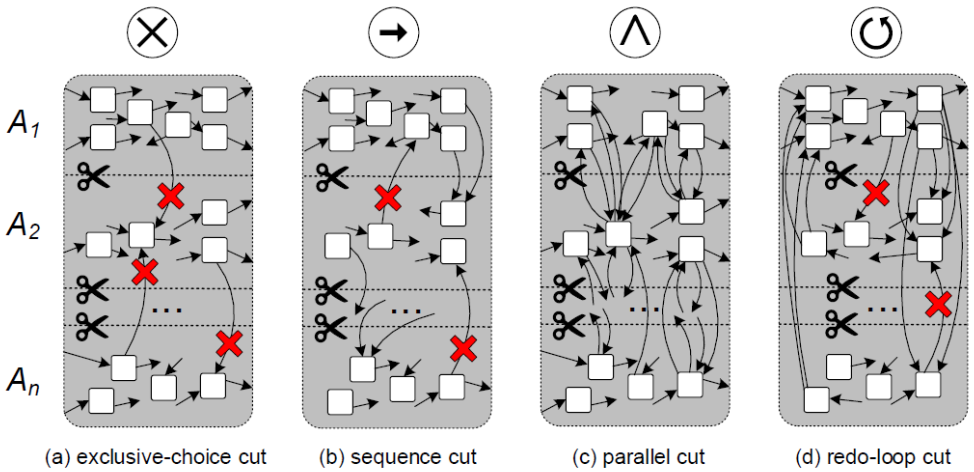
Chair of Process and Data Science

The details

Definition 23 (Sequence, Exclusive-Choice, Parallel, and Redo-Loop Cuts). *Let* $L \in \mathcal{B}(\mathcal{U}_{act}{}^*)$ *be an event log having a DFG* $disc_{DFG}(L) = (A, F)$ *based on* L *(note that* $A = act(L)$*) with start activities* $A^{start} = \{a \in A \mid (\blacktriangleright, a) \in F\}$ *and end activities* $A^{end} = \{a \in A \mid (a, \blacksquare) \in F\}$*. An* n*-ary* $\oplus$*-cut of* L *is a partition of* A *into* $n \geq 2$ *pairwise disjoint subsets* $A_1, A_2, \ldots, A_n$ *(i.e.,* $A = \bigcup_{i \in \{1,\ldots,n\}} A_i$ *and* $A_i \cap A_j = \emptyset$ *for* $i \neq j$*) with* $\oplus \in \{\rightarrow, \times, \wedge, \circlearrowleft\}$*. Such a* $\oplus$*-cut is denoted* $(\oplus, A_1, A_2, \ldots A_n)$*. For each type of operator* $\oplus \in \{\rightarrow, \times, \wedge, \circlearrowleft\}$ *specific conditions apply:*

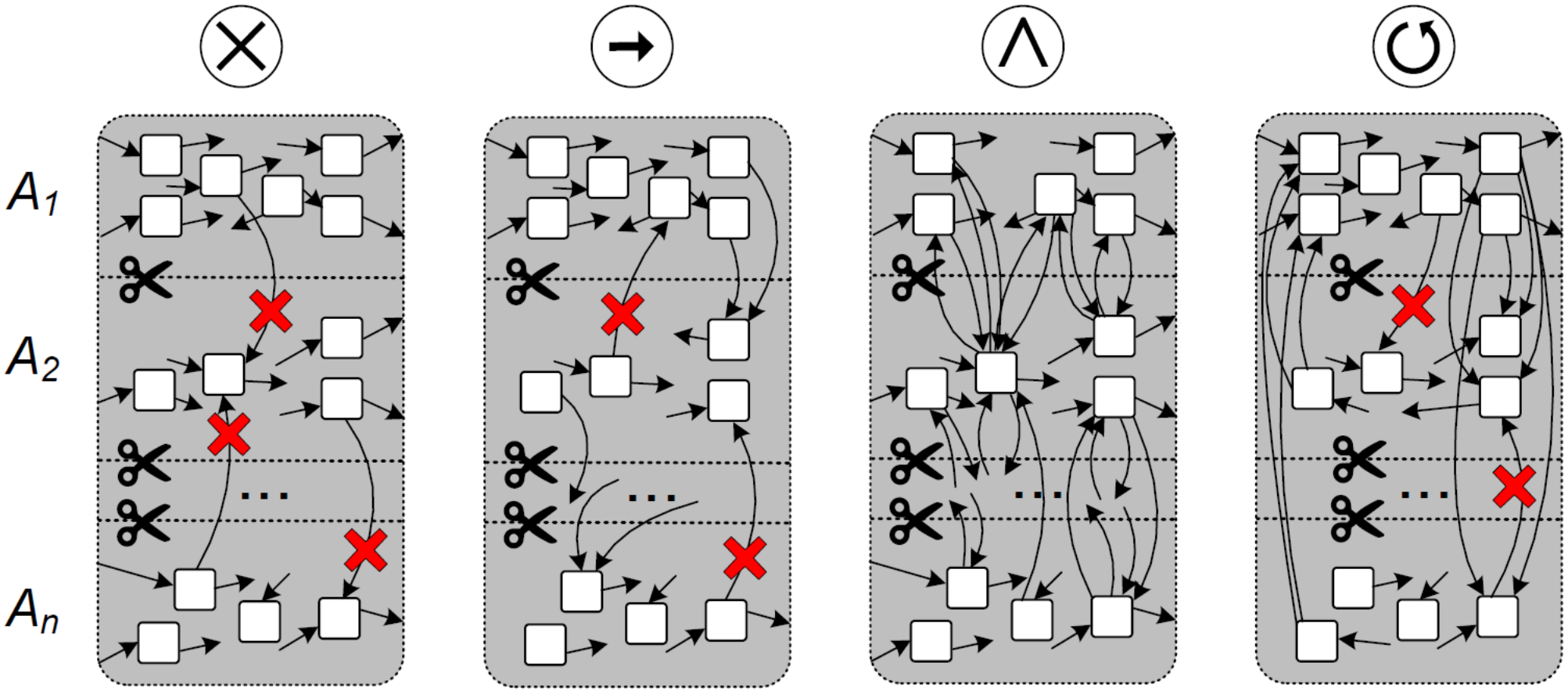
- *An* exclusive-choice cut *of* L *is a cut* $(\times, A_1, A_2, \ldots A_n)$ *such that*
  - $\forall_{i,j \in \{1,\ldots n\}} \forall_{a \in A_i} \forall_{b \in A_j}\ i \neq j \Rightarrow (a, b) \notin F$.
- *A* sequence cut *of* L *is a cut* $(\rightarrow, A_1, A_2, \ldots A_n)$ *such that*
  - $\forall_{i,j \in \{1,\ldots n\}} \forall_{a \in A_i} \forall_{b \in A_j}\ i < j \Rightarrow ((a, b) \in F^+ \wedge (b, a) \notin F^+)$. *(Note that* F^+ *is the non-reflexive transitive closure of* F*, i.e.,* $(a, b) \in F^+$ *means that there is a path from* a *to* b *in the DFG.)*
- *A* parallel cut *of* L *is a cut* $(\wedge, A_1, A_2, \ldots A_n)$ *such that*
  - $\forall_{i \in \{1,\ldots n\}}\ A_i \cap A^{start} \neq \emptyset \wedge A_i \cap A^{end} \neq \emptyset$ *and*
  - $\forall_{i,j \in \{1,\ldots n\}} \forall_{a \in A_i} \forall_{b \in A_j}\ i \neq j \Rightarrow (a, b) \in F$.
- *A* redo-loop cut *of* L *is a cut* $(\circlearrowleft, A_1, A_2, \ldots A_n)$ *such that*
  - $A^{start} \cup A^{end} \subseteq A_1$,
  - $\forall_{i,j \in \{2,\ldots n\}} \forall_{a \in A_i} \forall_{b \in A_j}\ i \neq j \Rightarrow (a, b) \notin F$,
  - $\{a \in A_1 \mid (a, b) \in F \wedge b \notin A_1\} = A^{end}$,
  - $\{a \in A_1 \mid (b, a) \in F \wedge b \notin A_1\} = A^{start}$,
  - $\forall_{(a,b) \in F}\ a \in A_1 \wedge b \notin A_1 \Rightarrow \forall_{a' \in A^{end}}\ (a', b) \in F$, *and*
  - $\forall_{(b,a) \in F}\ a \in A_1 \wedge b \notin A_1 \Rightarrow \forall_{a' \in A^{start}}\ (b, a') \in F$.





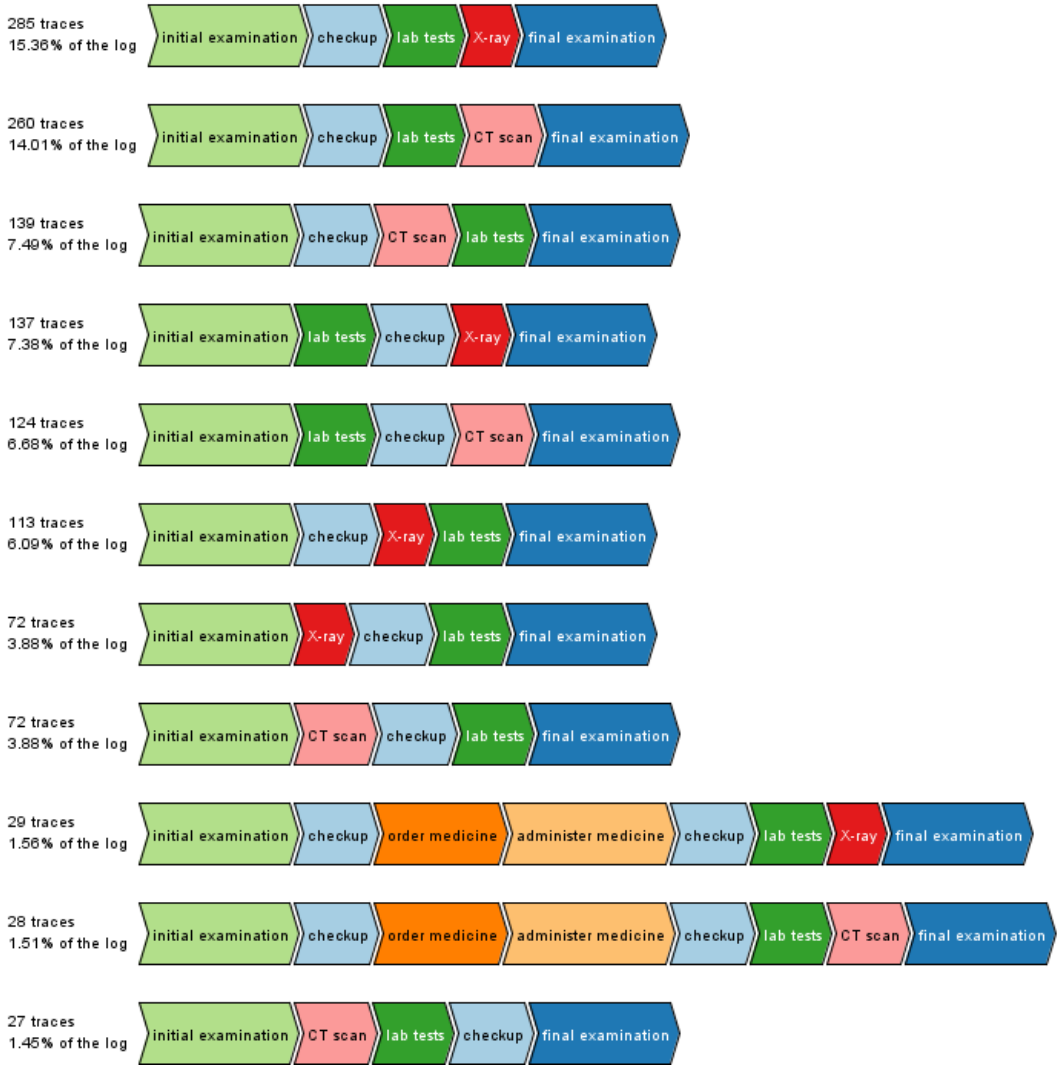
(a) exclusive-choice cut (b) sequence cut (c) parallel cut (d) redo-loop cut

Four types of cuts

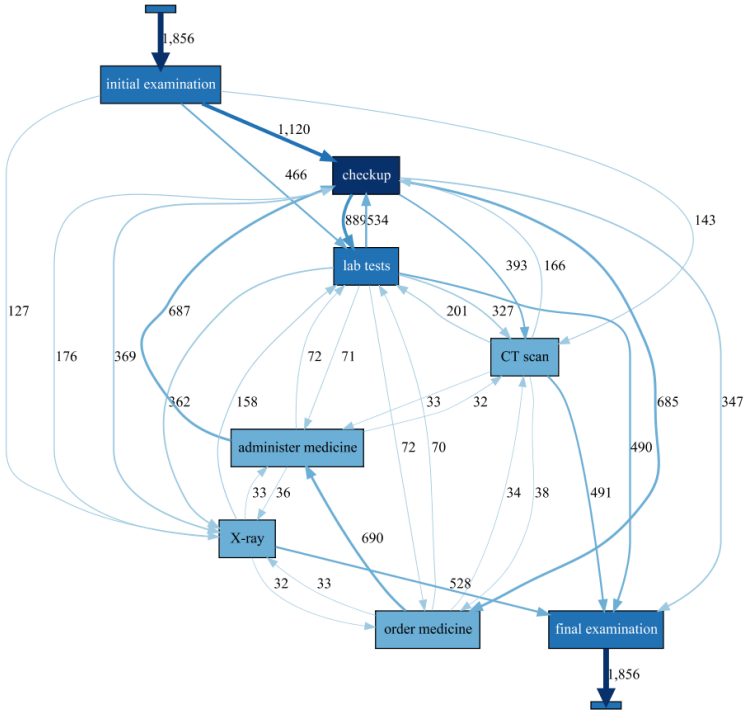


(a) exclusive-choice cut (b) sequence cut (c) parallel cut (d) redo-loop cut

Another example



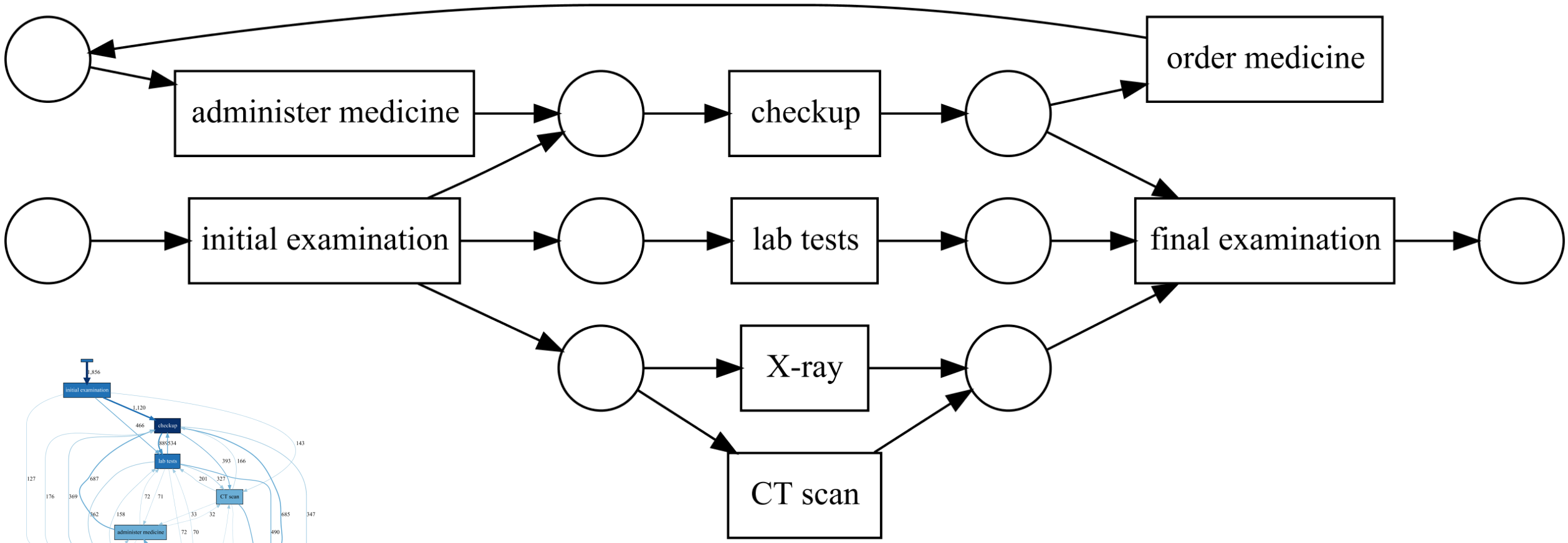
- 1856 cases, 197 variants
- 11761 events
- 8 unique activities



Just 11 of 197 variants

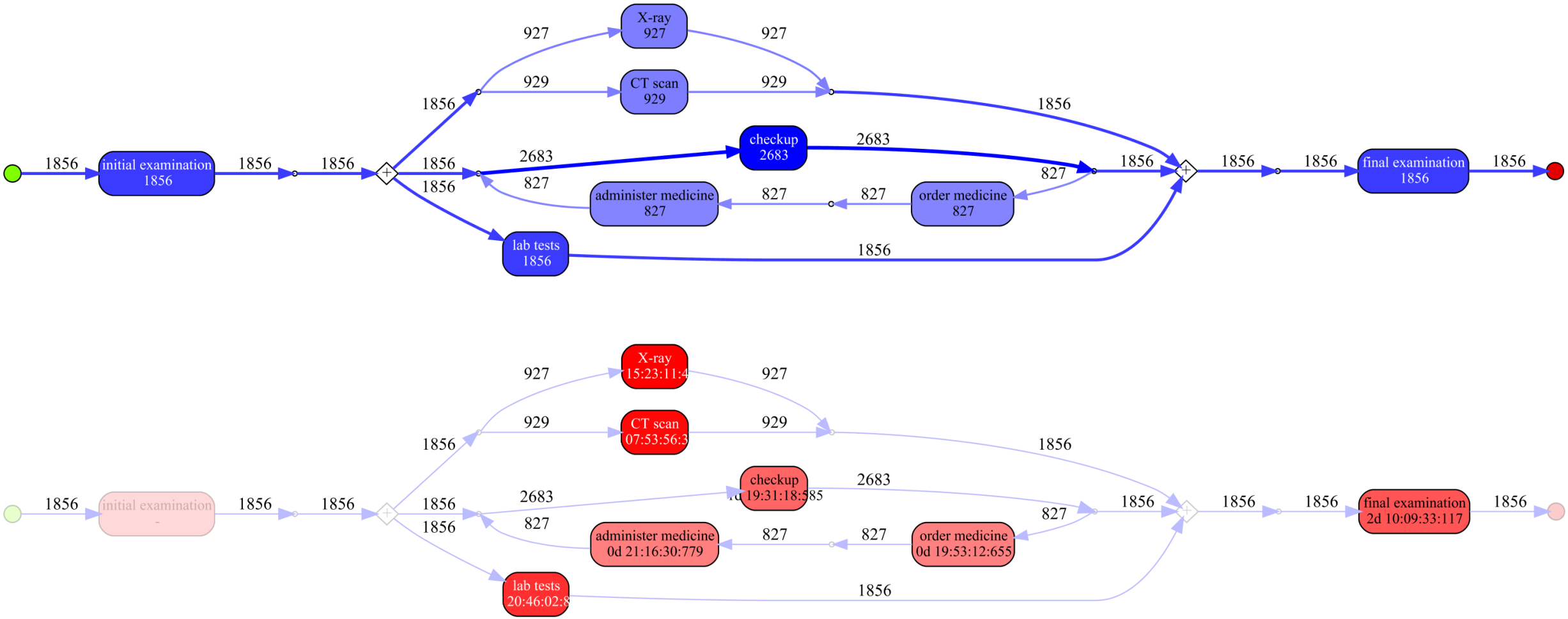
Chair of Process and Data Science

Alpha algorithm (ProM)



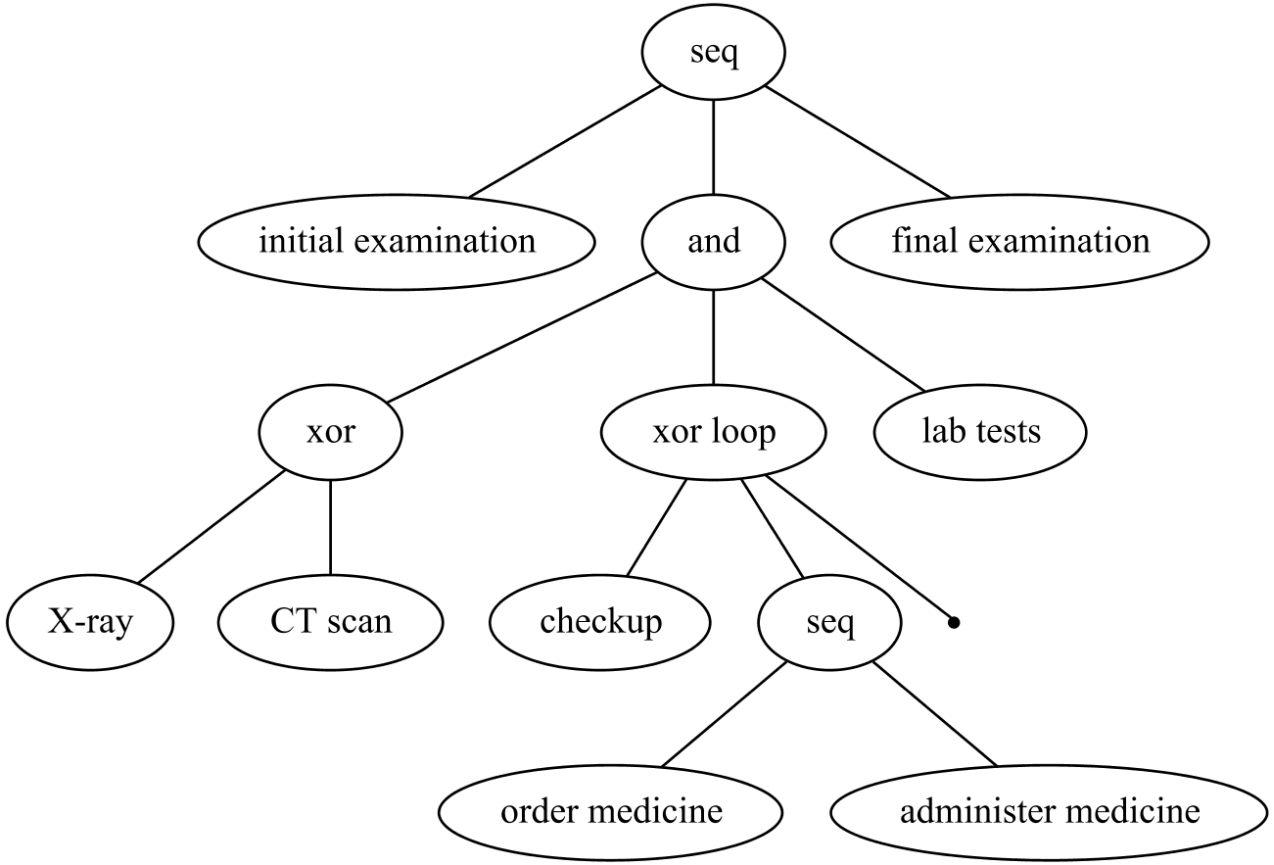
Chair of Process and Data Science

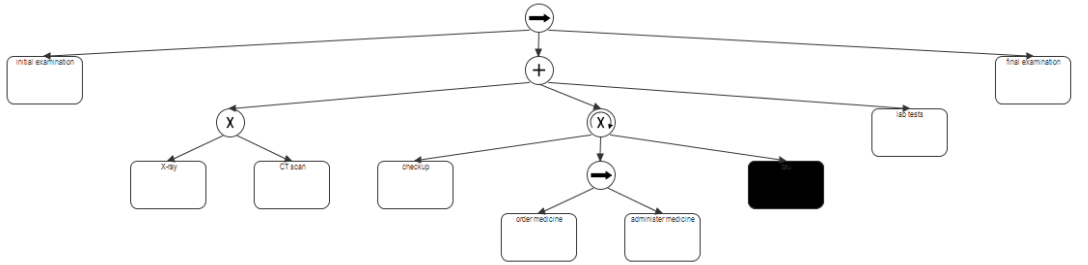
Inductive visual miner (ProM)

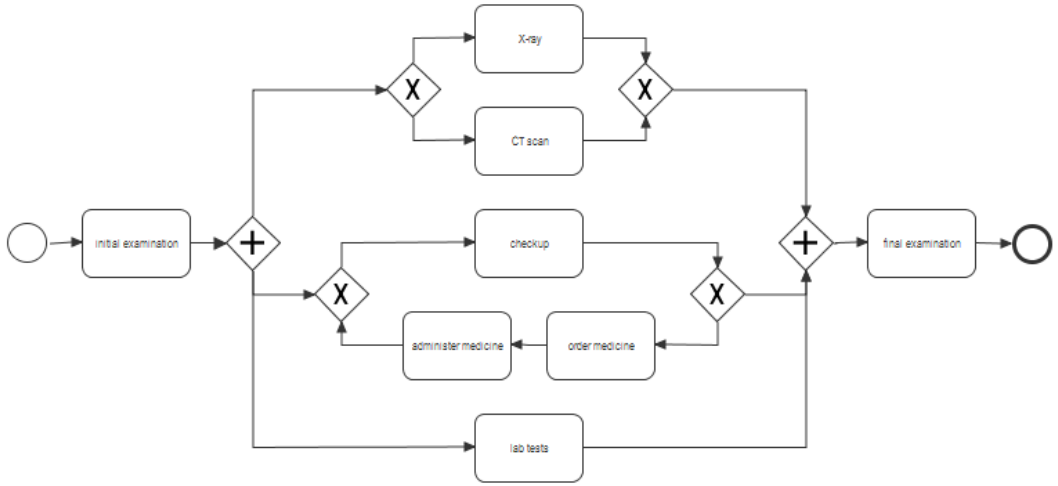


Chair of Process and Data Science

Different visualizations in ProM

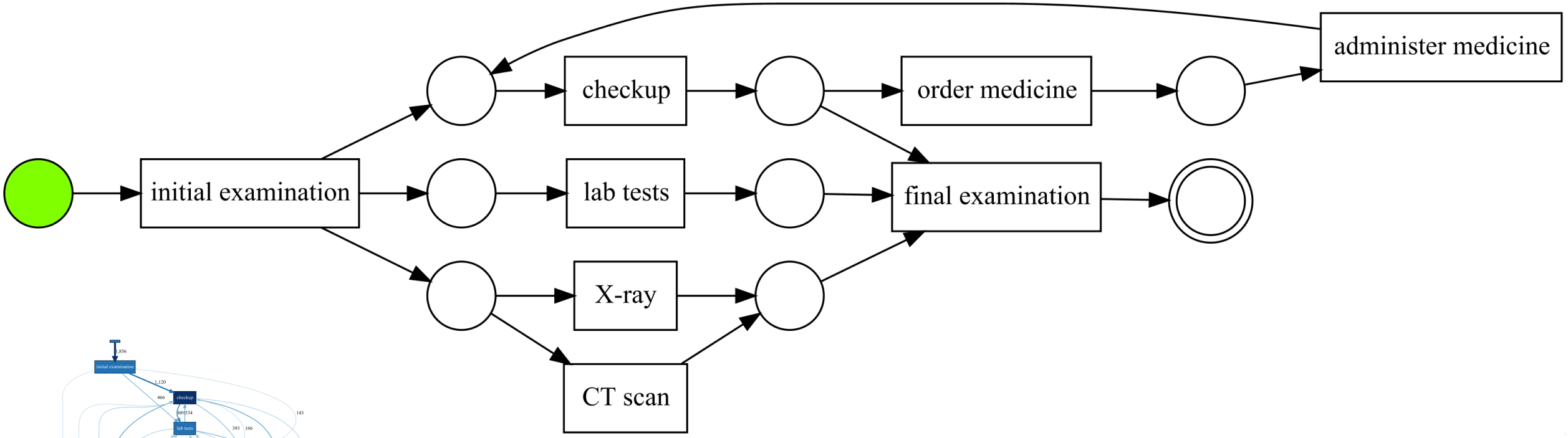


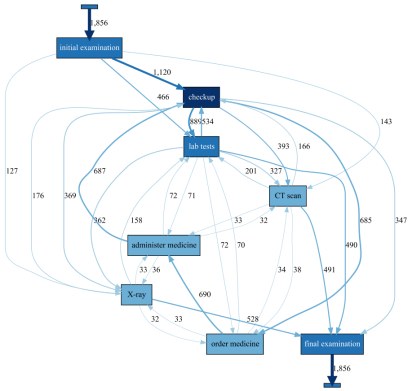




Chair of Process and Data Science

Mapped onto an accepting Petri net





Chair of Process and Data Science

Search
All (3) Knowledge Mode ...
New asset
BPMN
DFG View
synthetic-health-car...

BPMN Model Miner

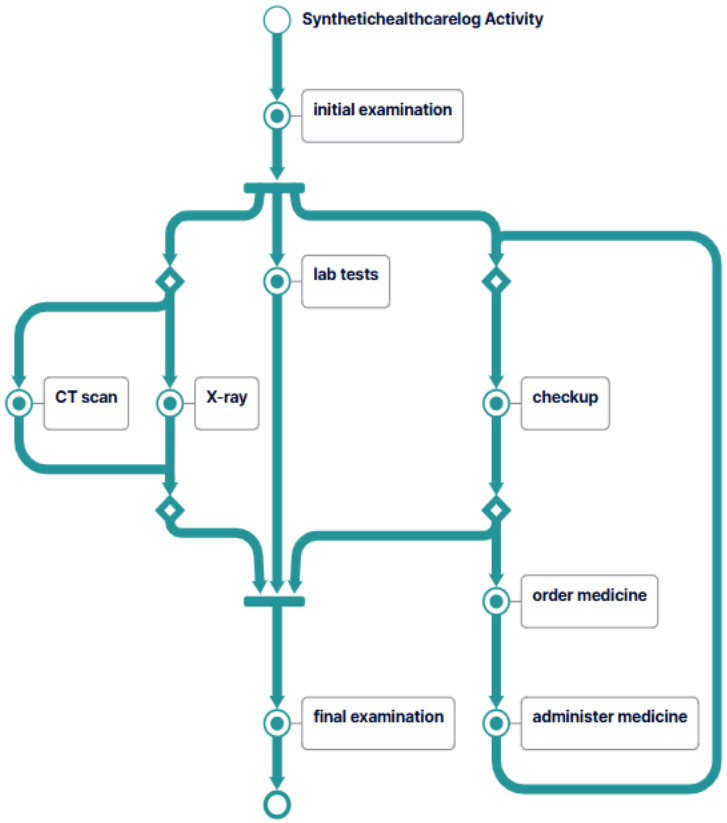
Continue Cancel

Custom process variants

Select variants to build your model. You can use the search criteria to find the relevant variants.

72 Selected variants Clear all

#	Count	Coverage	Avg TPT	#Step
#1	288	2.49%	10 days	5
#2	245	2.12%	10 days	5
#3	145	1.25%	9 days	5
#4	133	1.15%	9 days	5
#5	123	1.06%	10 days	5
#6	118	1.02%	9 days	5
#7	87	0.75%	9 days	5
#8	71	0.61%	9 days	5
#9	39	0.34%	11 days	8
#10	31	0.27%	6 days	5
#11	30	0.26%	6 days	5
#12	30	0.26%	9 days	8
#15	23	0.20%	6 days	5
#17	16	0.14%	9 days	8
#19	14	0.12%	10 days	8
#21	12	0.10%	10 days	8



Legend Zoom to fit 30%

Celonis | BPMN | DFG | synthetic

academic-workspace-wvdaalst.eu-2.celonis.cloud/package-manager/ui/studio/ui/spaces/a81eeceb-9152-4f59-ab67-7bae9a4e9293/packages/4842df8c-70e7-4133-95b5-d8c4ade0c589/nodes/5ba5f292-eecc-4d4e-...

Studio > synthetic-heal... > DFG

Search CTRL / | 1 | Create version | Deploy

BPMN Edit Mode

Deviations | Variants | Allowed deviations

Explore deviations

Conformance rate

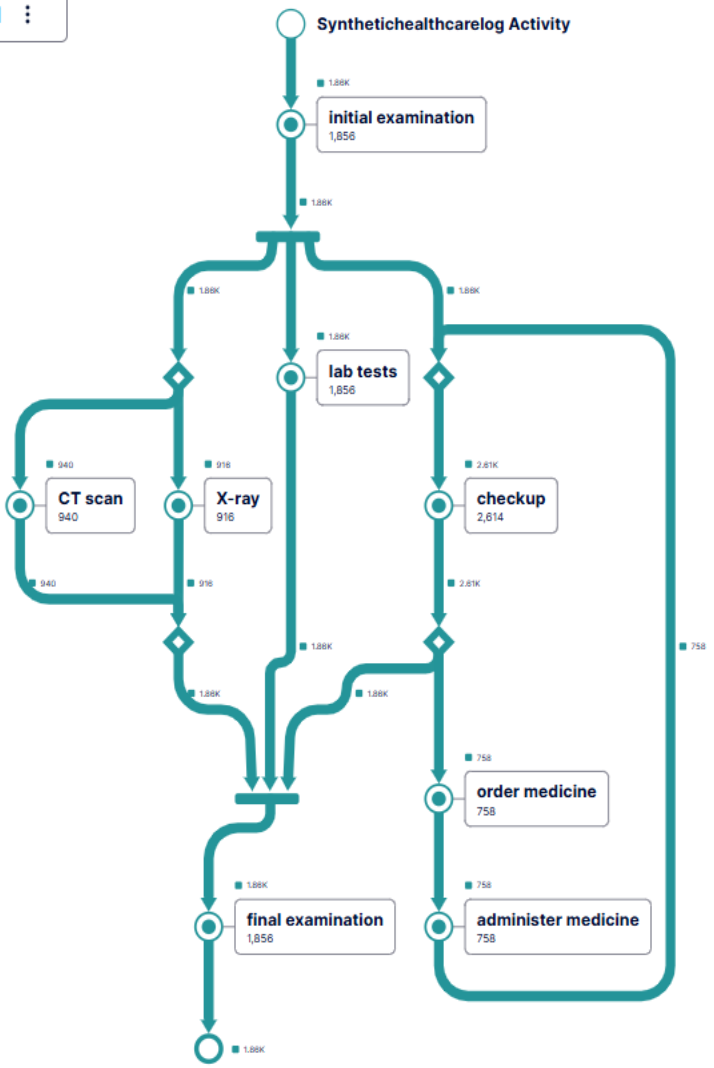
Synthetichealthcarelog Activity 100%

Deviations (0)

Congratulations! Your data has no deviations and conditions. It is 100% compliant.

BPMN export or edit with Celonis CPM

Target Model



Preset filters

Select the data you want to compare to the target model.

+ Add a filter

Data & Coverage Info

Synthetichealthcarelog Activity 100%

Legend | 30%

Throughput time filter

Event log

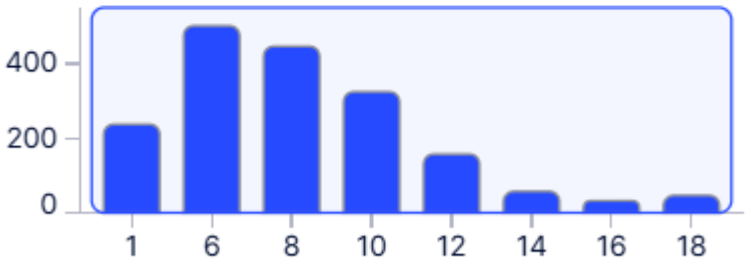
Syntheticehealthcarelog Activity

from: initial examination — First

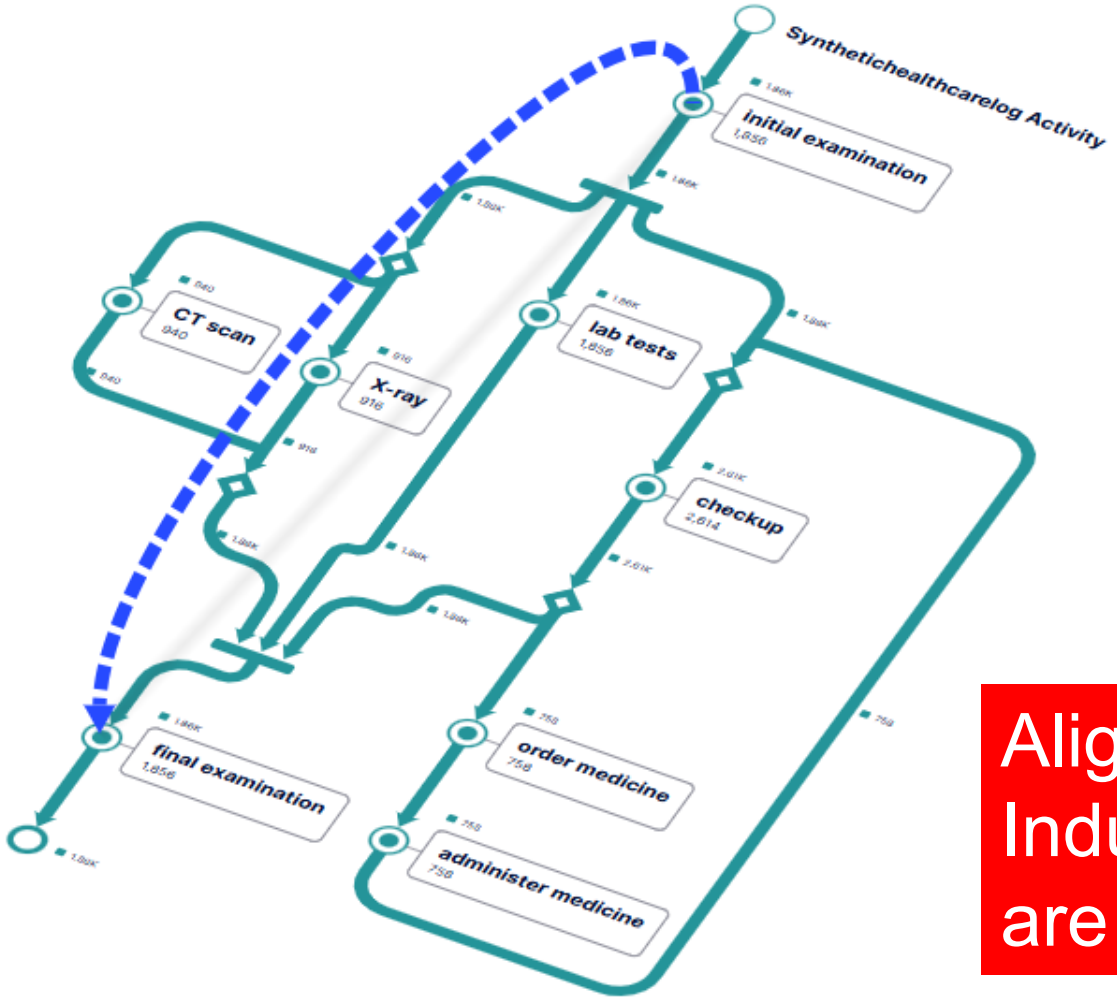
to: final examination — Last

avg.	med.	min.	max.
9.25 days	8 days	0 days	0 days

↑ Frequency



1 to 23 days



Alignments and Inductive Mining are supported !!

Preset filters

Select the data you want to the target model.

+ Add a filt

Data & Coverage Info

Syntheticehealthcarelo Activity

Legend

Summary: Inductive Mining

- The models are guaranteed to be sound, i.e., no deadlocks, no livelocks, and no other anomalies.
- The basic algorithm guarantees that the event log can be reproduce completely (of course one can filter if desired).
- The algorithm has good performance (and there are also more scalable variants) and implemented in several tools.
- There are various additional theoretical guarantees, i.e., rediscover the process tree used to create the event log.

Chair of Process and Data Science
RWTH AACHEN UNIVERSITY


The GenAI Divide
STATE OF AI IN BUSINESS 2025
MIT NANDA
Aditya Challapally
Chris Pease
Ramesh Raskar
Pradyumna Chari
July 2025
95% of AI projects fail
AI PROJECT
PROCESS MINING TOOLBOX - DO NOT OPEN
95% FAIL
Image generated using Google Gemini

“When Agents Hit the Walls” — CIO, April 2026
“Why enterprise AI initiatives keep dying before production” — InformationWeek, March 2026
“Most AI Investments Are Failing. The Problem Isn’t The Technology.” — Forbes, Feb. 2026
“The ‘Last Mile’ Problem Slowing AI Transformation” — Harvard Business Review, March 2026
“Why Most CEOs See No Return On AI Investments” — Forbes, March 2026
“Why Your Digital Investments Aren’t Creating Value” — Harvard Business Review, Feb. 2026
“The Hidden Costs That Are Undermining Enterprise AI ROI” — Forbes, March 2026
“Why AI Adoption Stalls, According to Industry Data” — Harvard Business Review, Feb. 2026
“From insights to action: Why enterprises are shifting to execution-driven AI systems” — Economic Times, March 2026
“Nearly 80% of UK firms have adopted AI — but barely any are seeing a positive ROI so far” — TechRadar, March 2026

AI fails when it is dropped into broken processes instead of fixing them

Without process visibility, AI optimizes only fragments

AI cannot automate a workflow that nobody has mapped or observed

If you automate chaos, you get automated chaos

AI pilots succeed in demos and fail in production because real processes are messy and fragmented

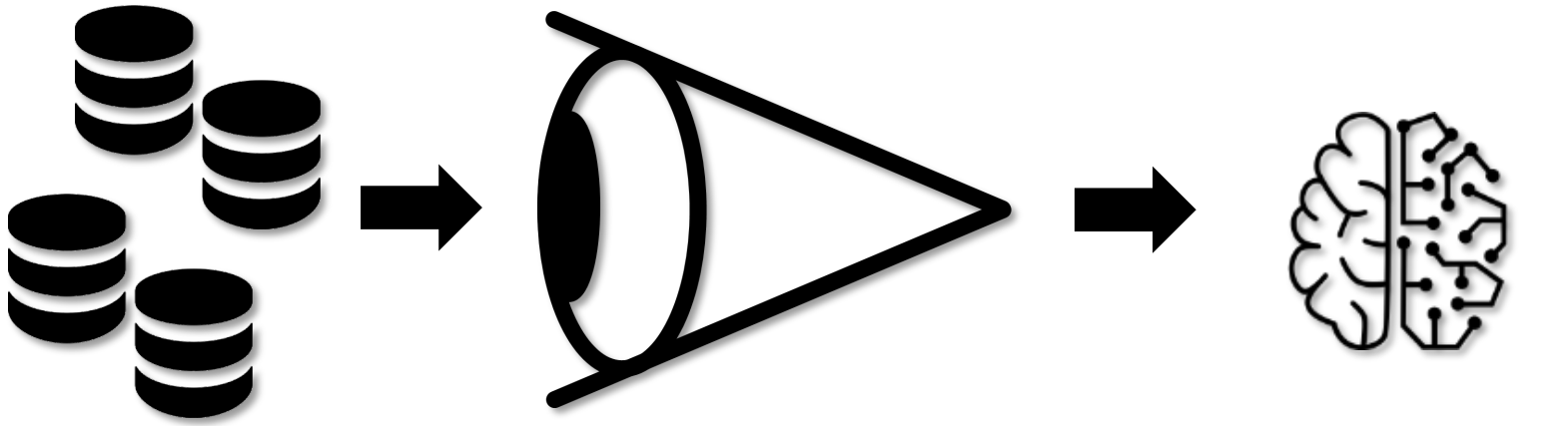
No ROI without process redesign

AI needs more than data; it needs process context

When AI hits the walls, those walls are usually process boundaries

Disconnected systems are a technical symptom; disconnected processes are the real disease

Process Mining Enables ML

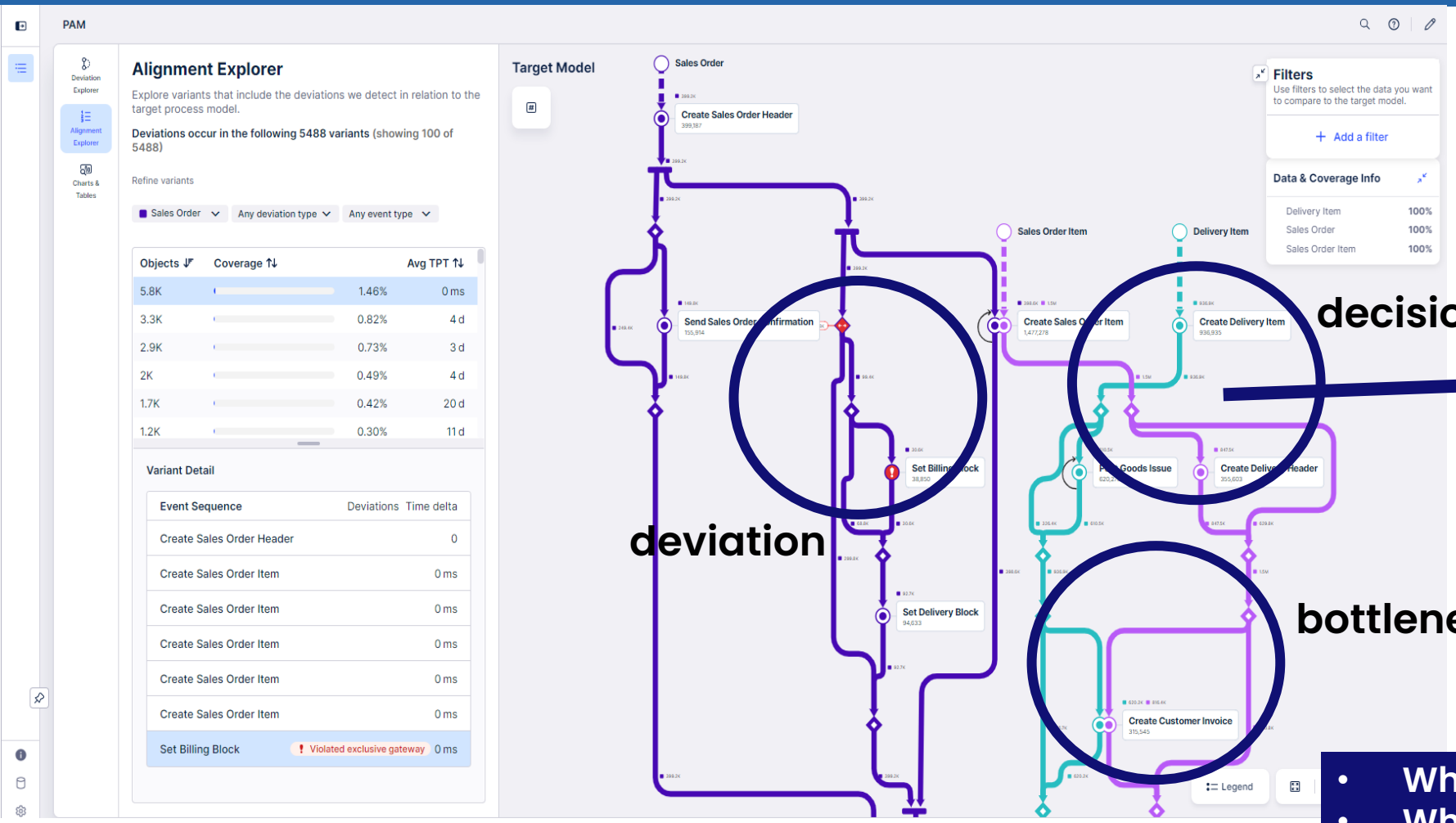


complex reality

event data and process models

machine learning

Generating “Machine Learning Problems” for “Process Problems”



deviation

decision

bottleneck

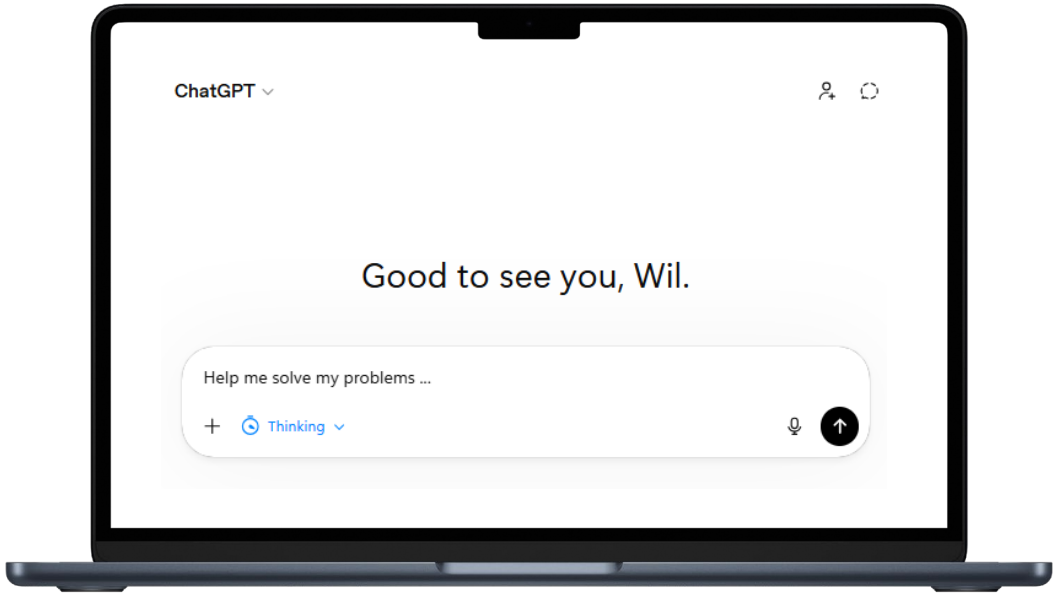
situation table

CASE	PRODUCT	ADDRESS	first resource	total throughput tim…	decision
1	SAMSUNG Galaxy J5	Munich	Caleb	239	pay
2	APPLE iPhone 6s 64…	Amsterdam	Lucas	201	pay
3	APPLE iPhone 5s 16…	New York	Sophia	503	cancel order
4	MOTOROLA Moto E …	New York	Sophia	498	cancel order
5	SAMSUNG Core Pri…	Aachen	Isabella	741	pay
6	SAMSUNG Galaxy S4	Munich	Emma	406	pay
7	MOTOROLA Moto G	Amsterdam	Lucas	598	pay
8	APPLE iPhone 6 16 …	Amsterdam	Sophia	209	pay
9	APPLE iPhone 5s 16…	Munich	Aiden	412	pay
10	HUAWEI P8 Lite	Amsterdam	Speedy	415	pay
11	MOTOROLA Moto G	Munich	Emma	508	pay
12	APPLE iPhone 5s 16…	Aachen	Jacob	460	pay
13	HUAWEI P8 Lite	Munich	RESOURCE	ACTIVITY	Activities count
14	SAMSUNG Core Pri…	Munich	Abigail	make delivery	592
15	SAMSUNG Galaxy S4	Aachen	Abigail	send reminder	4,347
			Aiden	place order	2,087
			Aiden	prepare delivery	1,659
			Alexander	cancel order	120
			Alexander	confirm payment	397
			Alexander	pay	417
			Alexander	send invoice	557
			Aubrey	make delivery	1,584
			Avery	make delivery	769
			Caleb	cancel order	68
			Caleb	confirm payment	244
			Caleb	pay	258
			Caleb	place order	180
			Caleb	prepare delivery	130
			Caleb	send invoice	327

- What is causing the bottleneck?
- Which orders are deviating?
- When will this product be delivered?
- Will we meet our SLA tomorrow?

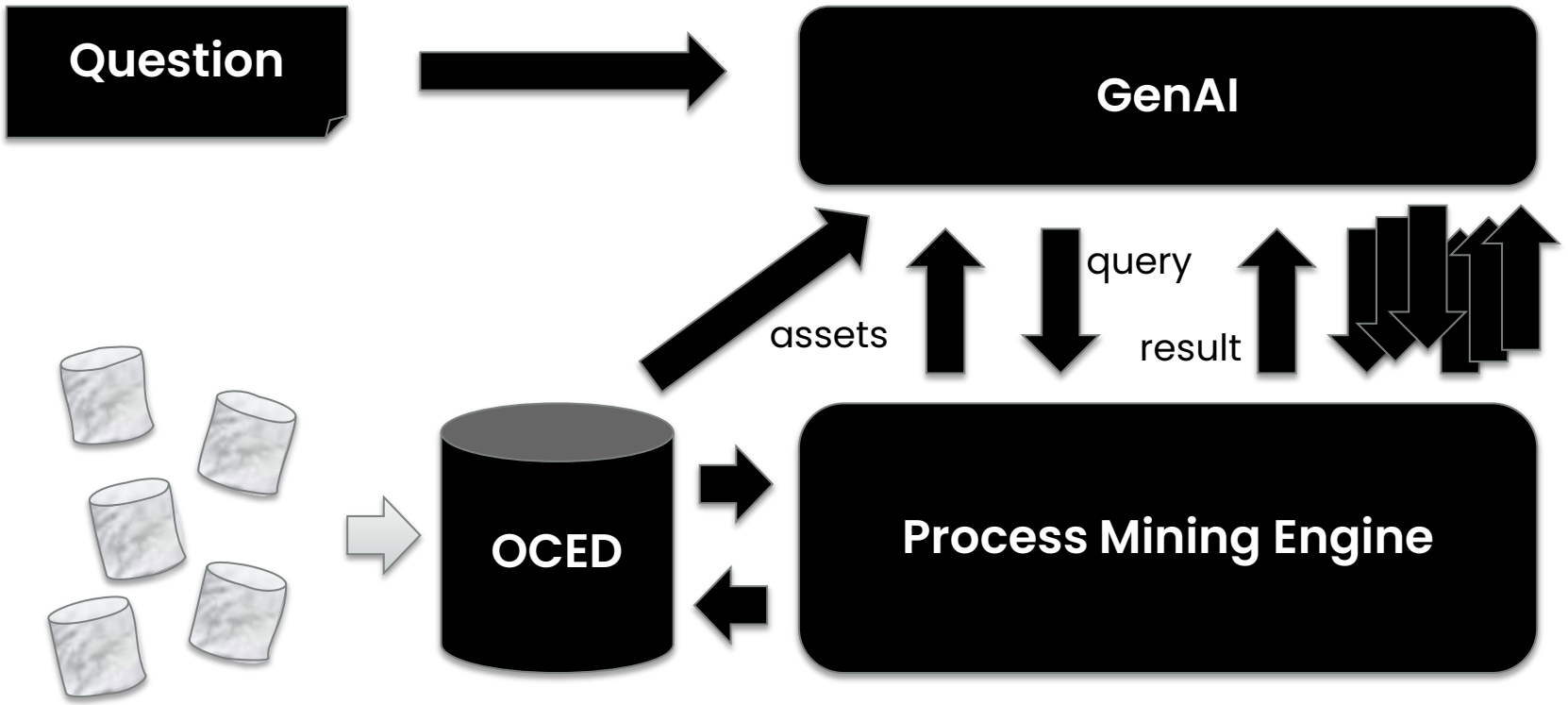
Guesswork Versus Computation

Question → GenAI ? Answers



Chair of Process and Data Science

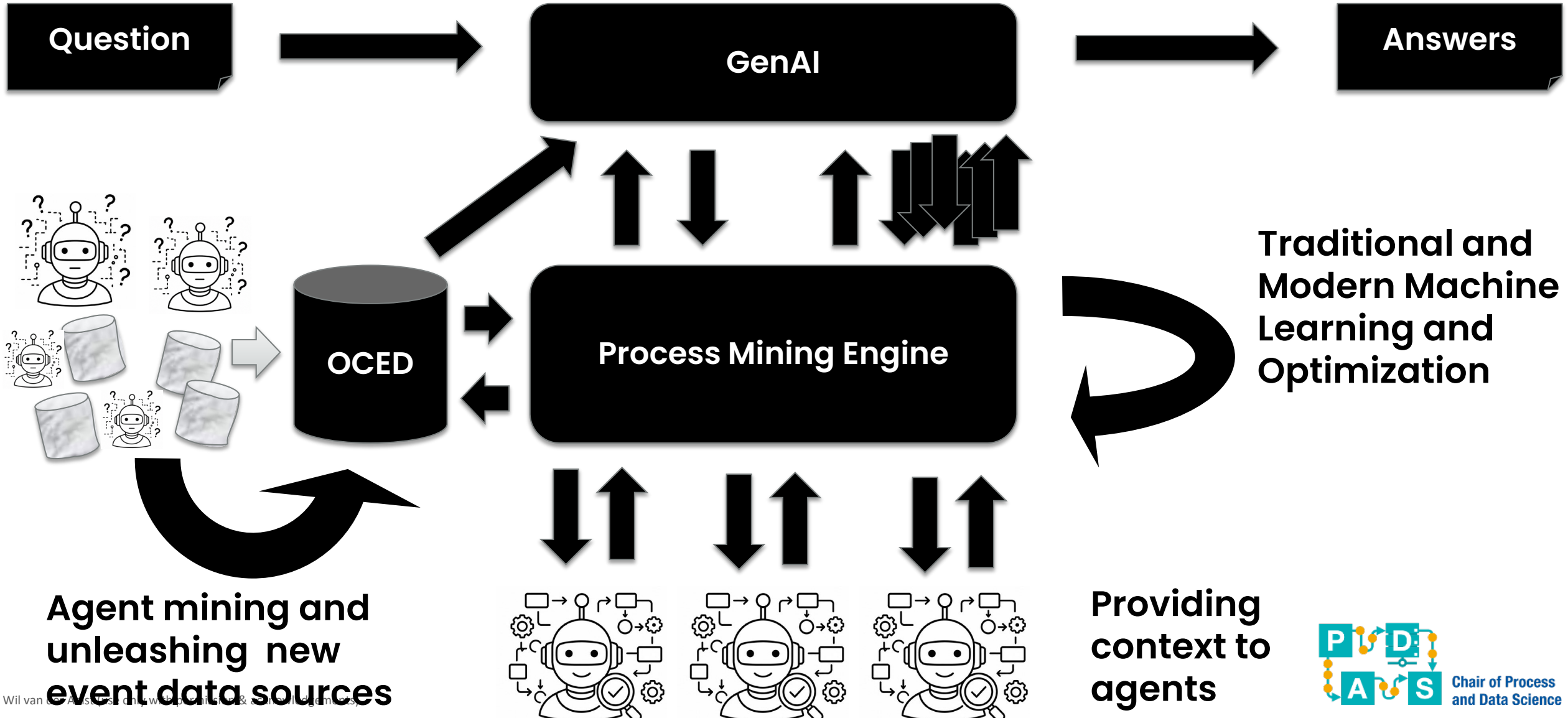
Guesswork Versus Computation



Based on facts and computation instead of Wikipedia & Co.

Chair of Process and Data Science

Not just copilots!



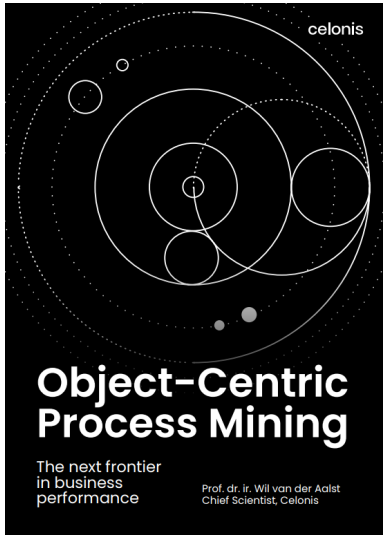
Chair of Process and Data Science

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

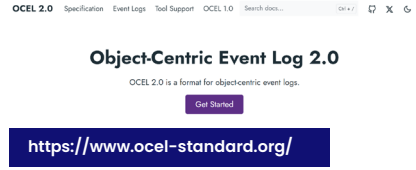
Object-Centric Process Mining (OCPM)

- **Out-of-scope for this introduction tutorial**
- **Like moving from 2D to 3D!**
- **See www.ocel-standard.org**



W. van der Aalst. Object-Centric Process Mining: The Next Frontier in Business Performance. 2023. celon.is/OCPM-Whitepaper

W.M.P. van der Aalst, Object-Centric Process Mining: Unraveling the Fabric of Real Processes. Mathematics 2023, 11, 2691. https://doi.org/10.3390/math11122691



https://www.ocel-standard.org/

01

Avoid repeatedly going back to your source systems

- Offers a single system-agnostic source of truth
- Saves time and helps to capture real-live events and objects

02

Avoid distortions due to the single-case assumption

- Squeezing reality into simple event logs creates distortions
- This includes the unintentional replication of events (convergence) and loss of causal relations (divergence)

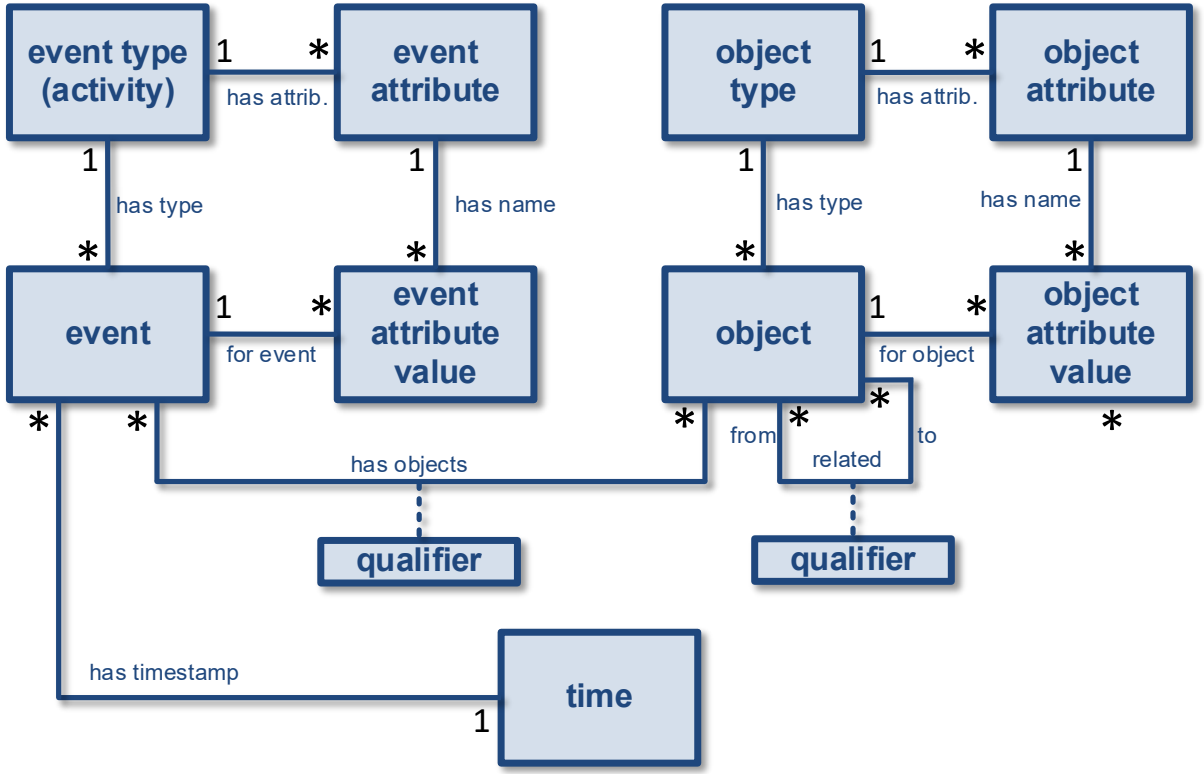
03

See and understand the interactions between different object types

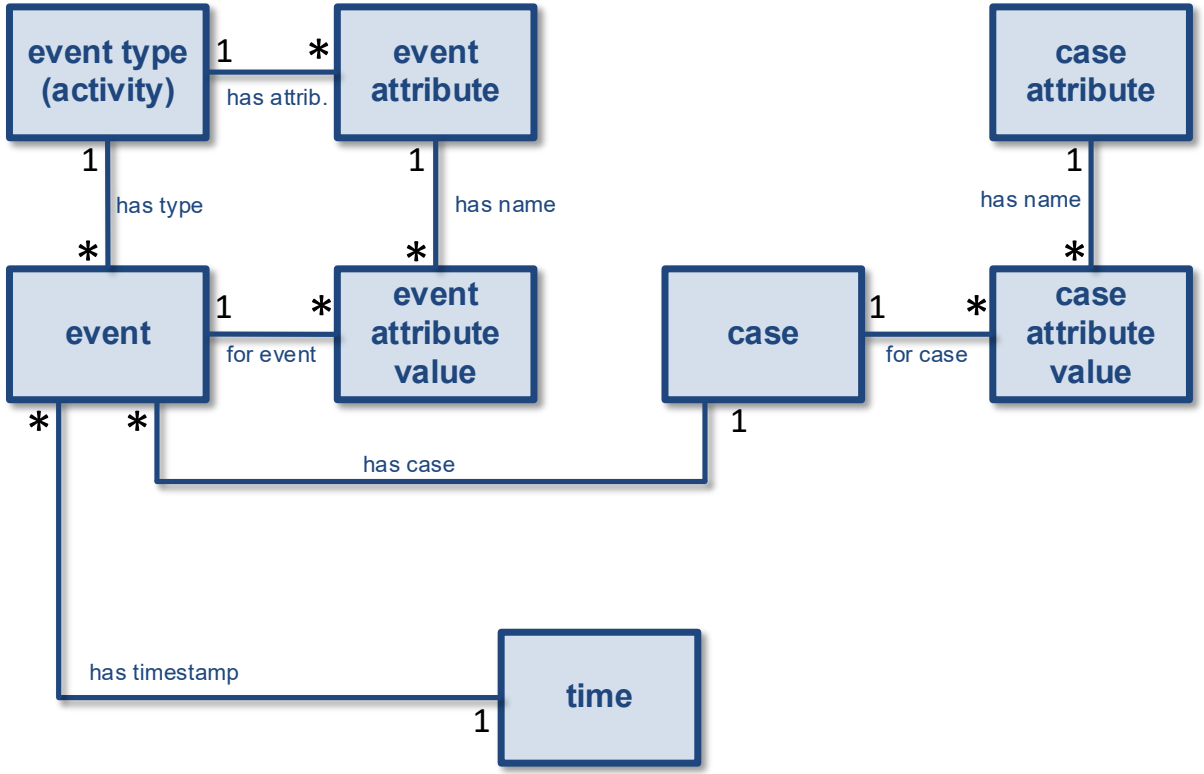
- Problems live at the intersections of processes and organizational entities
- E.g. low On-Time-In-Full (OTIF) scores may be caused by sales, production, procurement, logistics, etc.

Chair of Process and Data Science

Case-Centric Versus Object-Centric



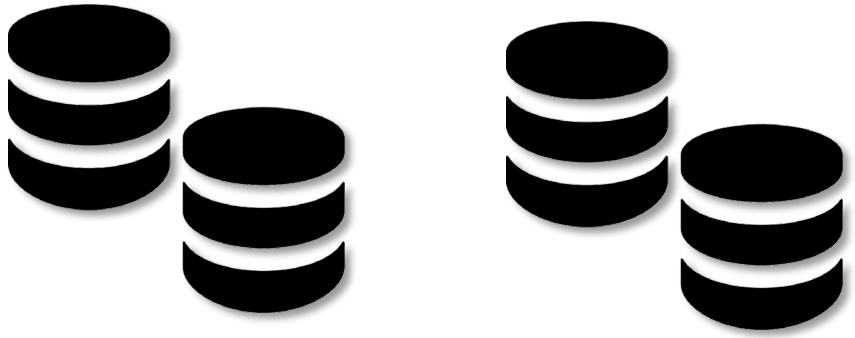
Object-Centric

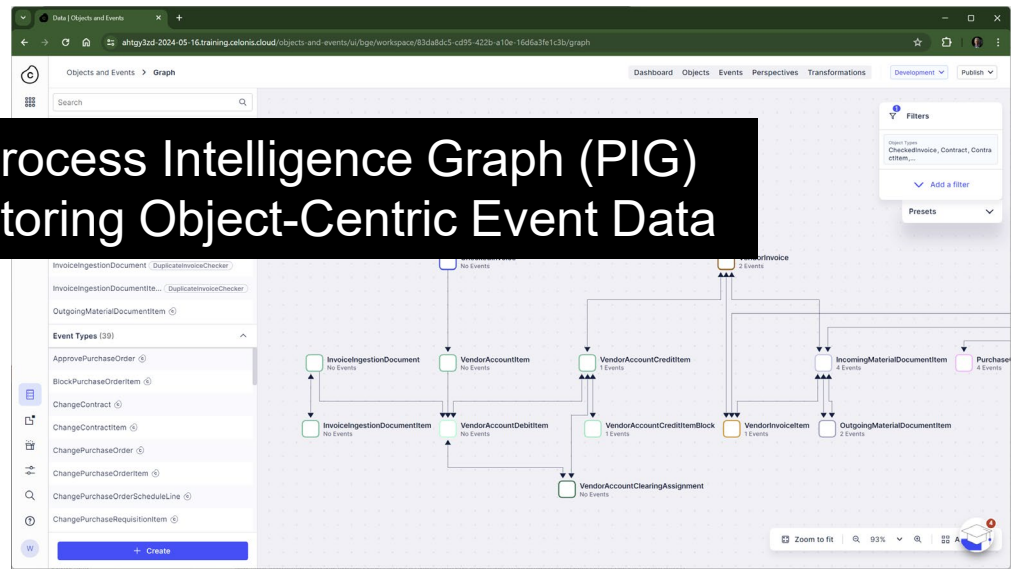


Case-Centric

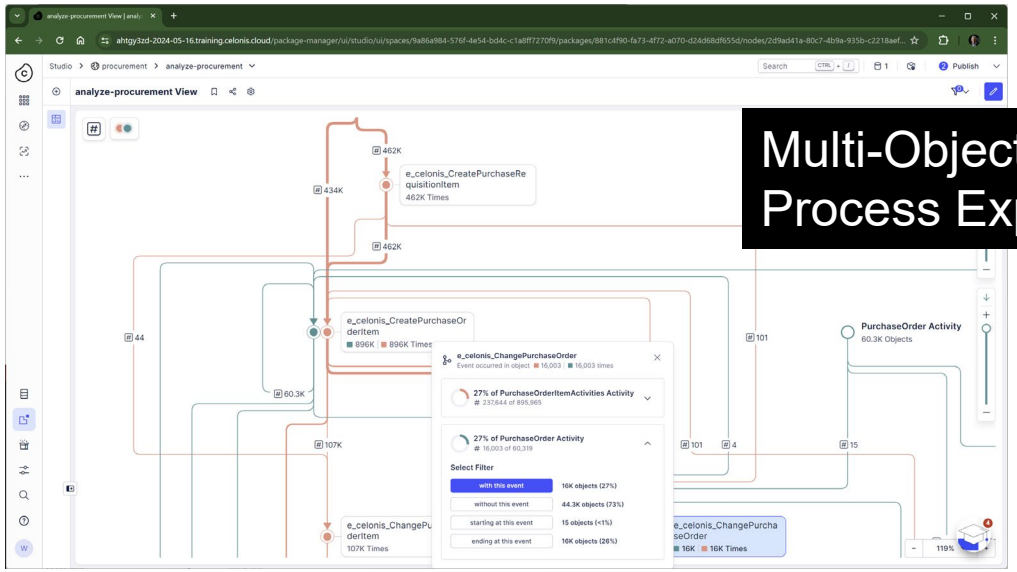
Chair of Process and Data Science

Celonis Supports OCPM

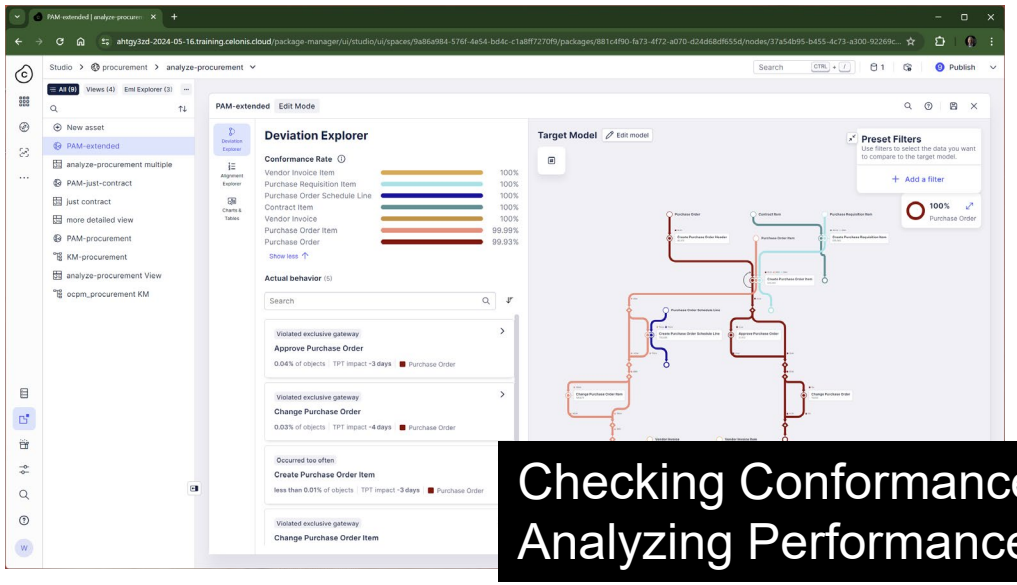




Process Intelligence Graph (PIG)
Storing Object-Centric Event Data

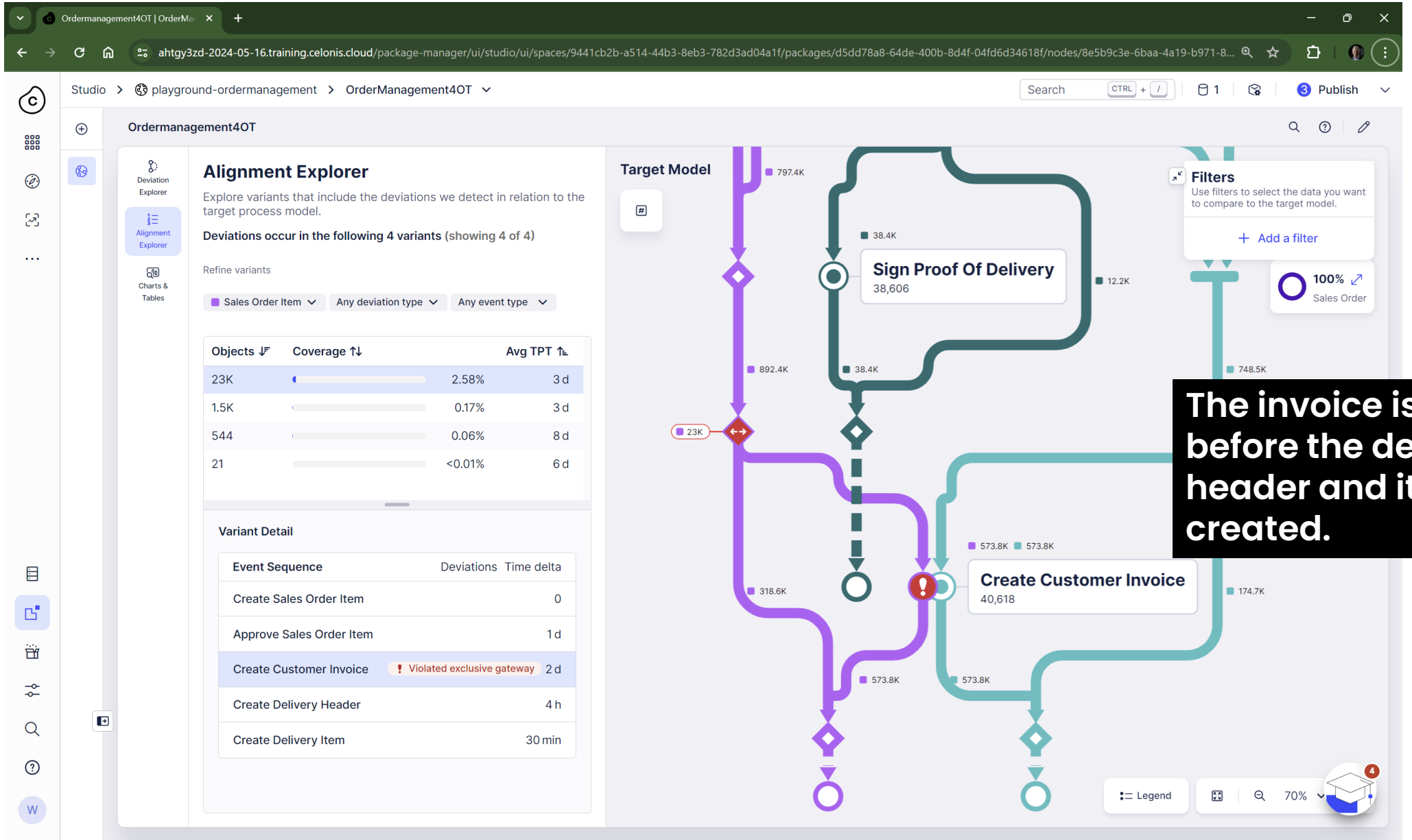


Multi-Object
Process Explorer



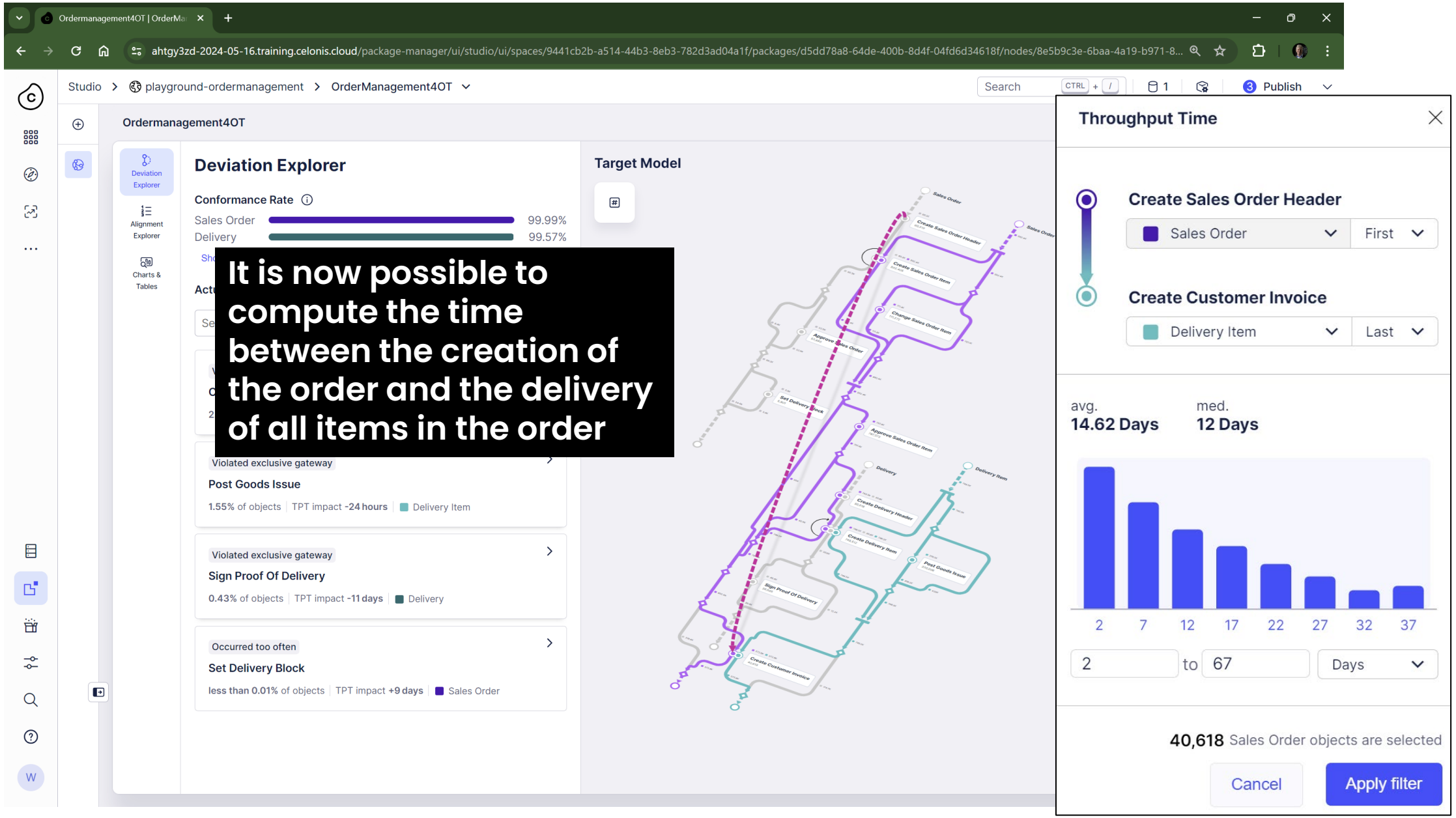
Checking Conformance and
Analyzing Performance

Conformance Checking Using Alignments



The invoice is created before the delivery header and item are created.

End-to-End Performance Analysis

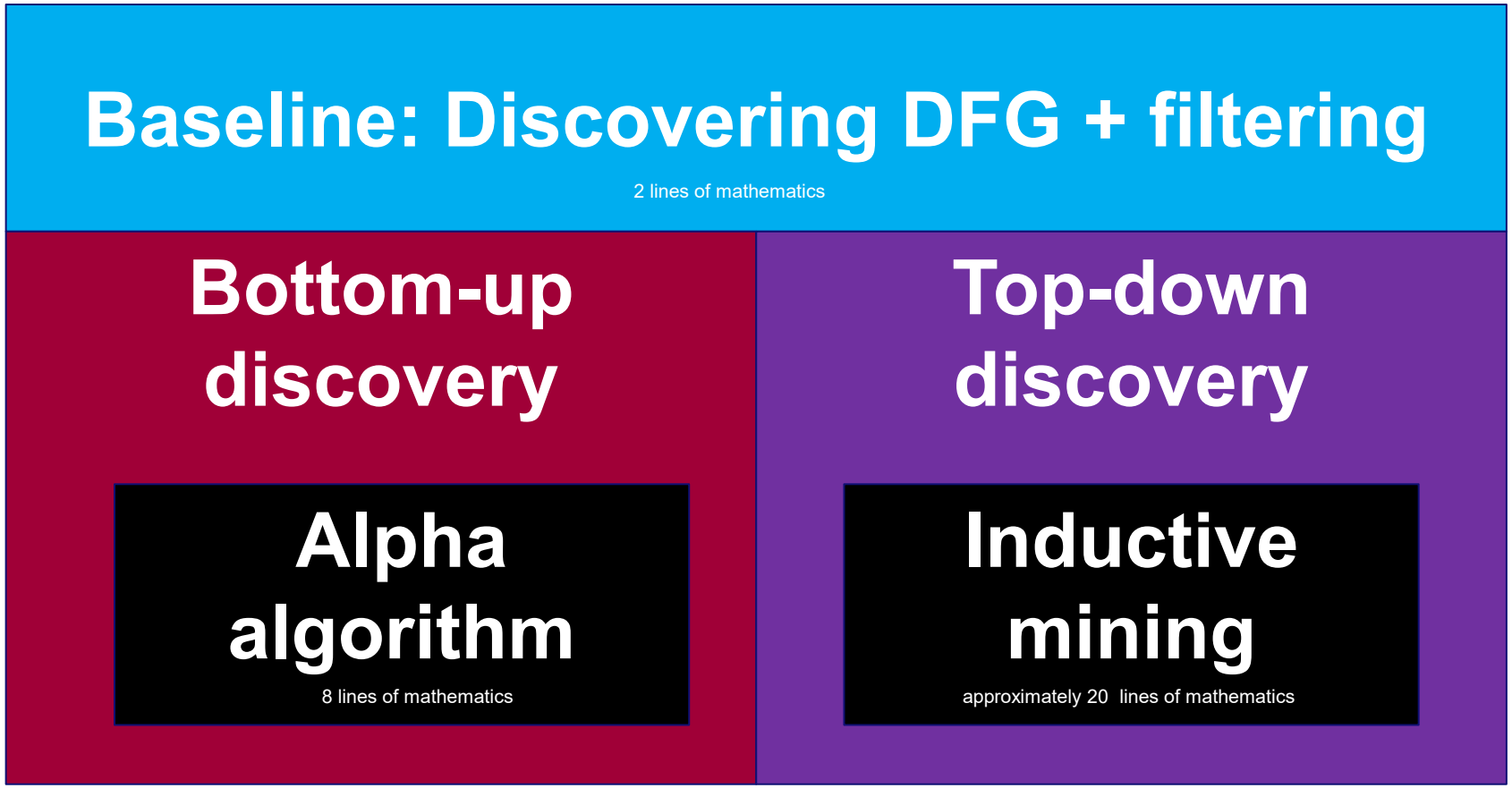


Conclusion

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

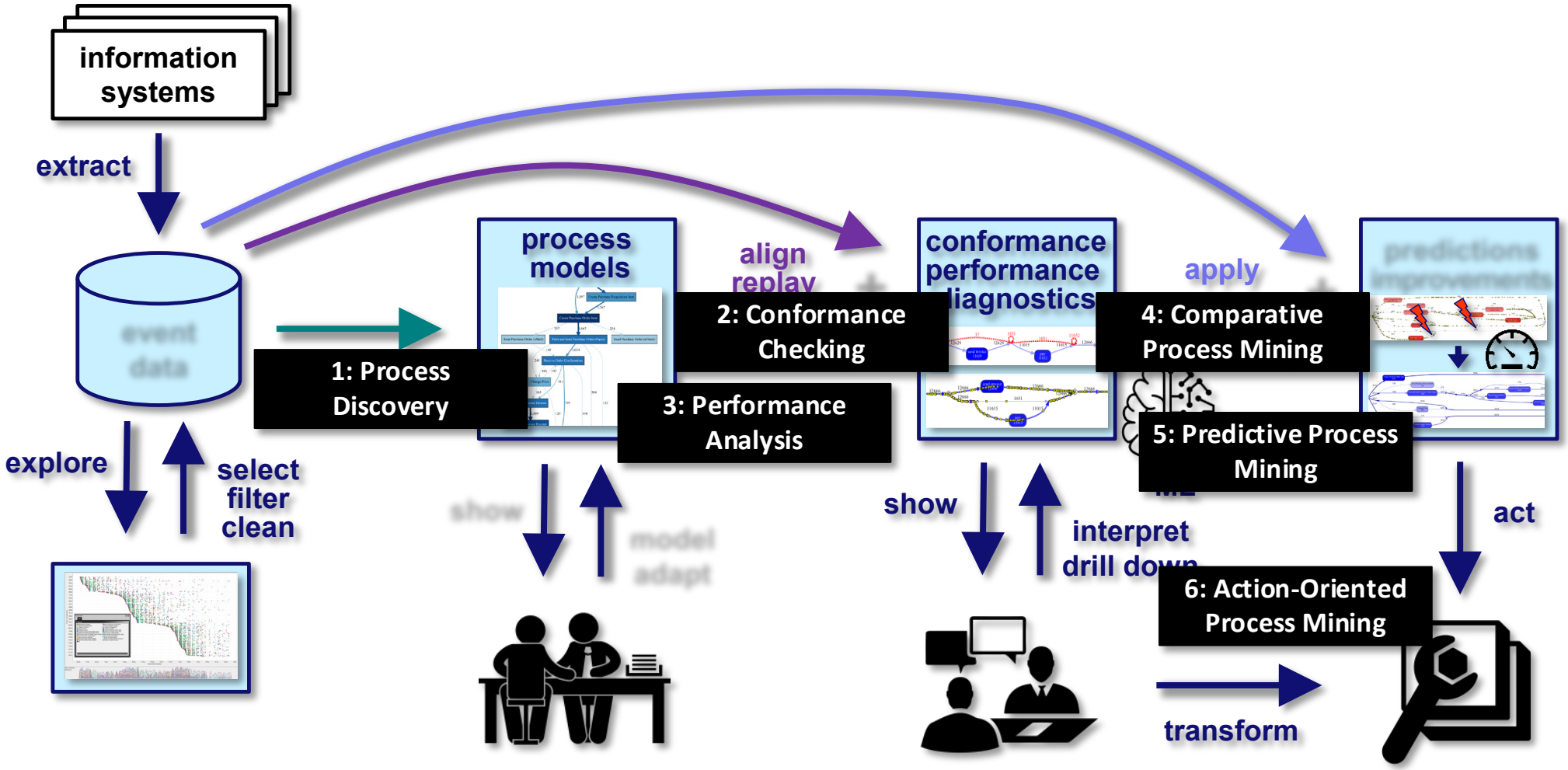
Foundations of Process Discovery



Not a solved problem!

Chair of Process and Data Science

Discovery is just one of many techniques



Chair of Process and Data Science

Websites

- www.processmining.org
- www.process-mining-summer-school.org
- www.tf-pm.org
- www.promtools.org
- www.celonis.com/academic-signup
- xes-standard.org
- ocel-standard.org
- www.pads.rwth-aachen.de
- www.vdaalst.com

Chair of Process and Data Science

Online courses

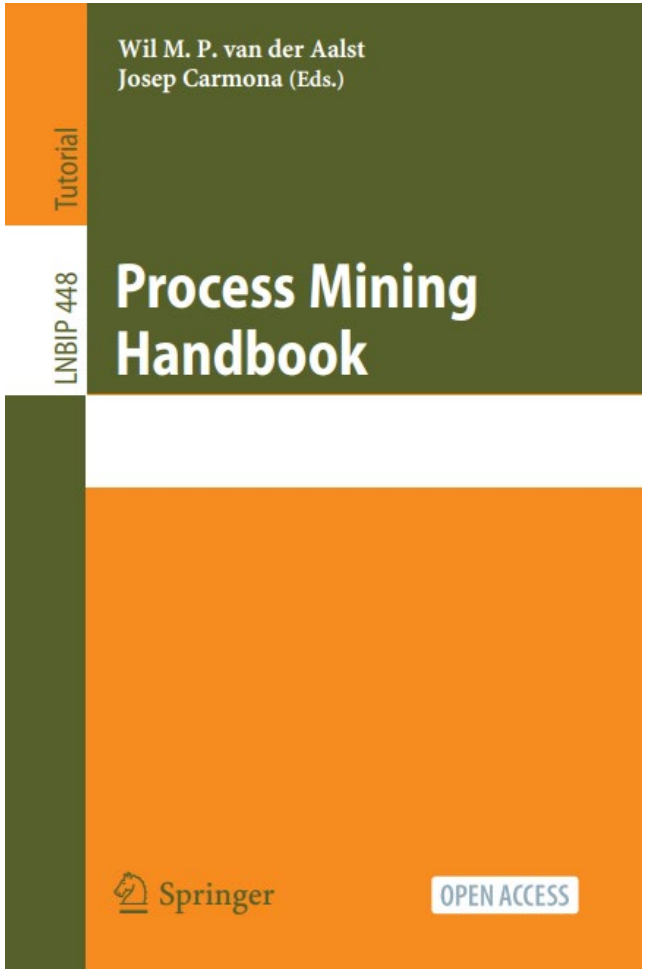
- **Coursera course “Process Mining: Data science in Action”**
  Register via coursera.org/learn/process-mining (>175.000 participants since 2015).
- **Celonis/RWTH course “Process Mining: From Theory to Execution”**
  Register via www.celonis.com/wils-process-mining-class.
- **edX course “A Hands-on Introduction to Process Mining”**
  Register via RWTHx: www.edx.org/school/rwthx.
- **edX course “Object Centric Process Mining”**
  Register via RWTHx: www.edx.org/school/rwthx.

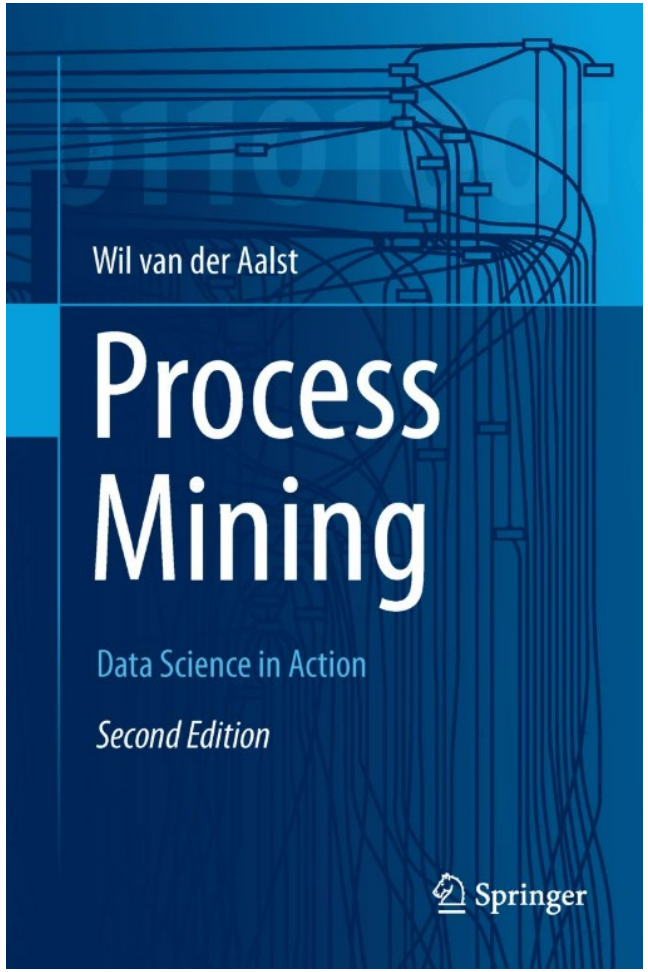
coursera TU/e

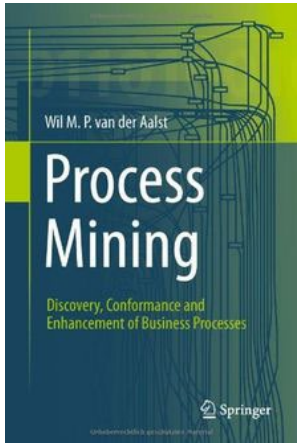
celonis RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

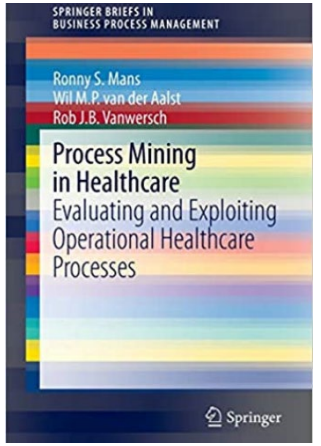
P D A S Chair of Process and Data Science

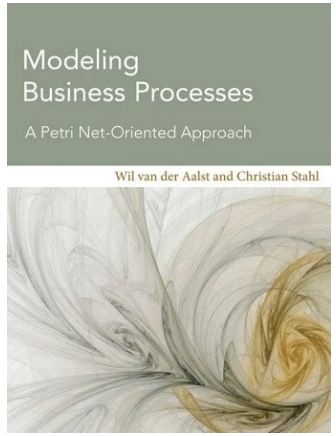
Books (not intended to be complete)

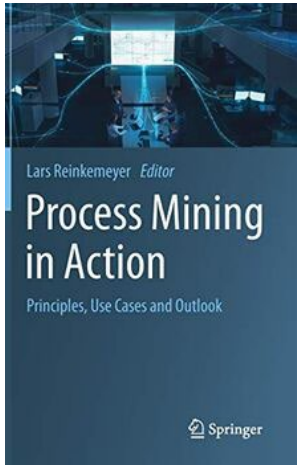


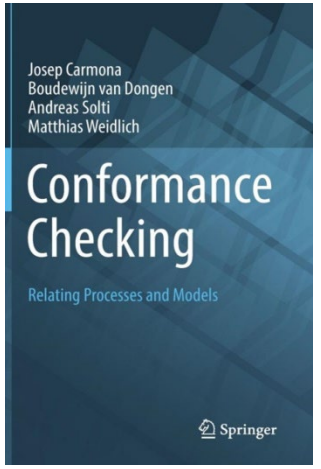












Chair of Process and Data Science