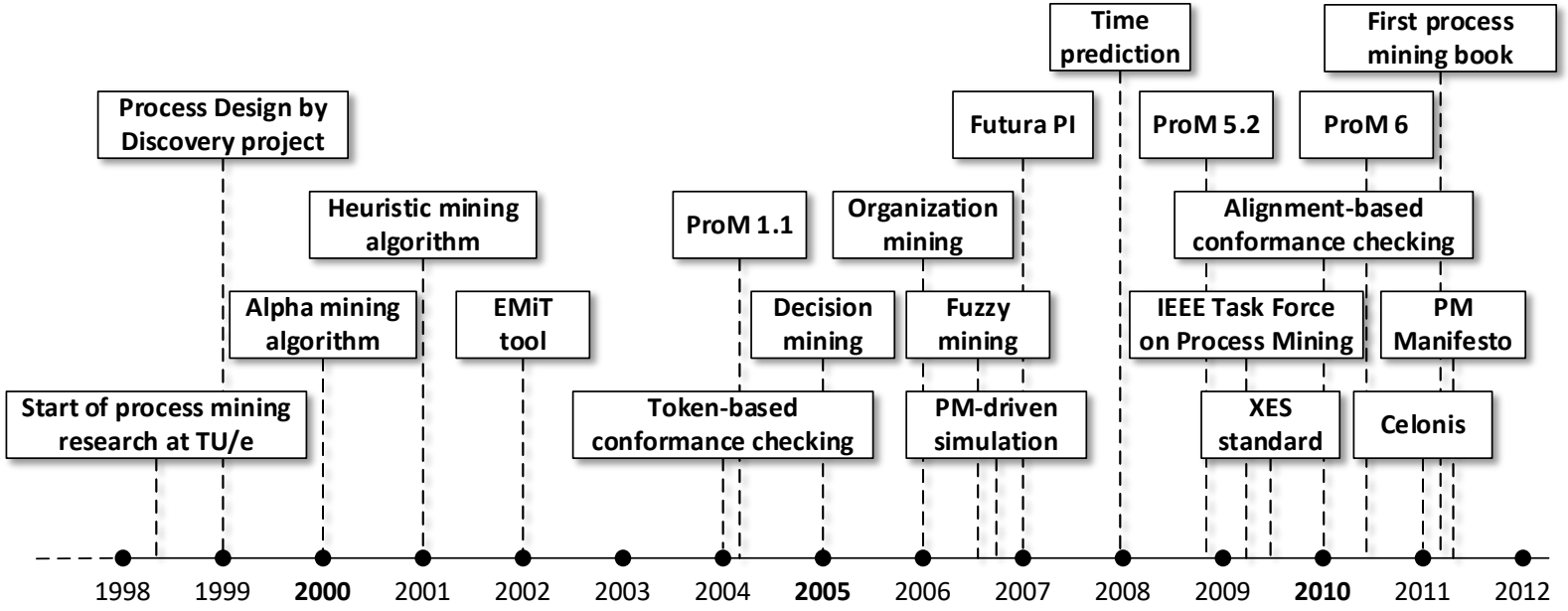
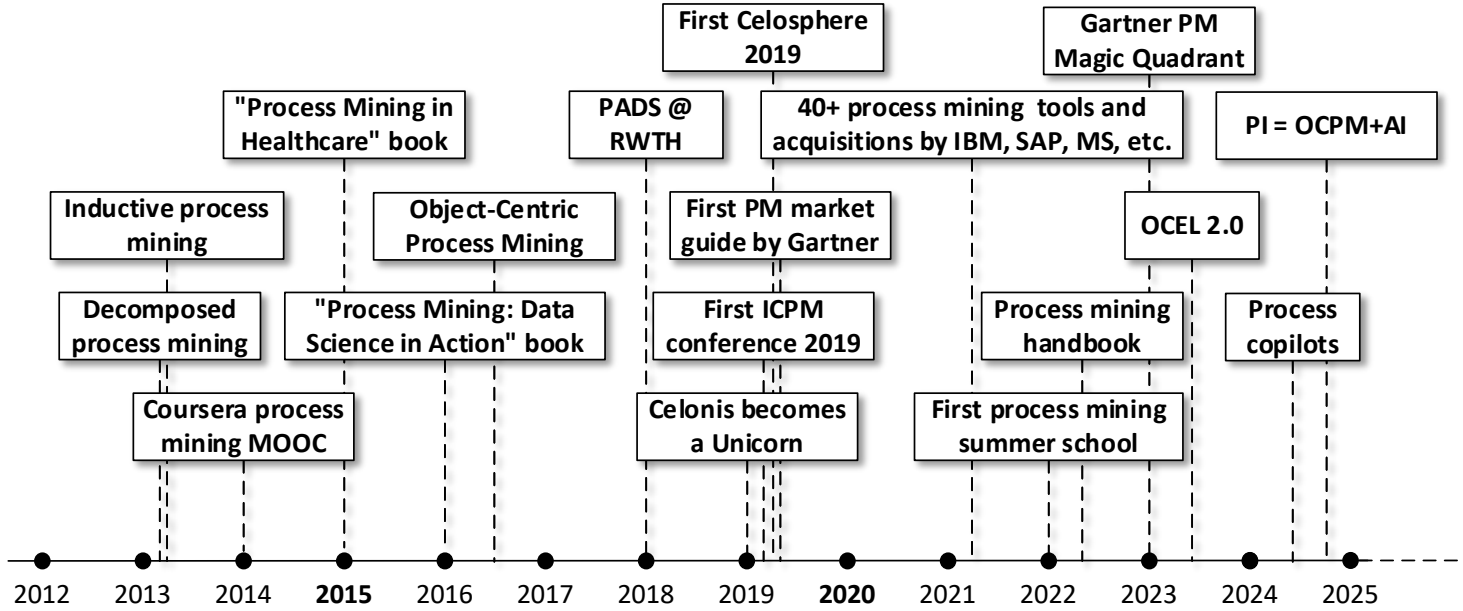
Process Mining Demystified

How Does it Really Work?

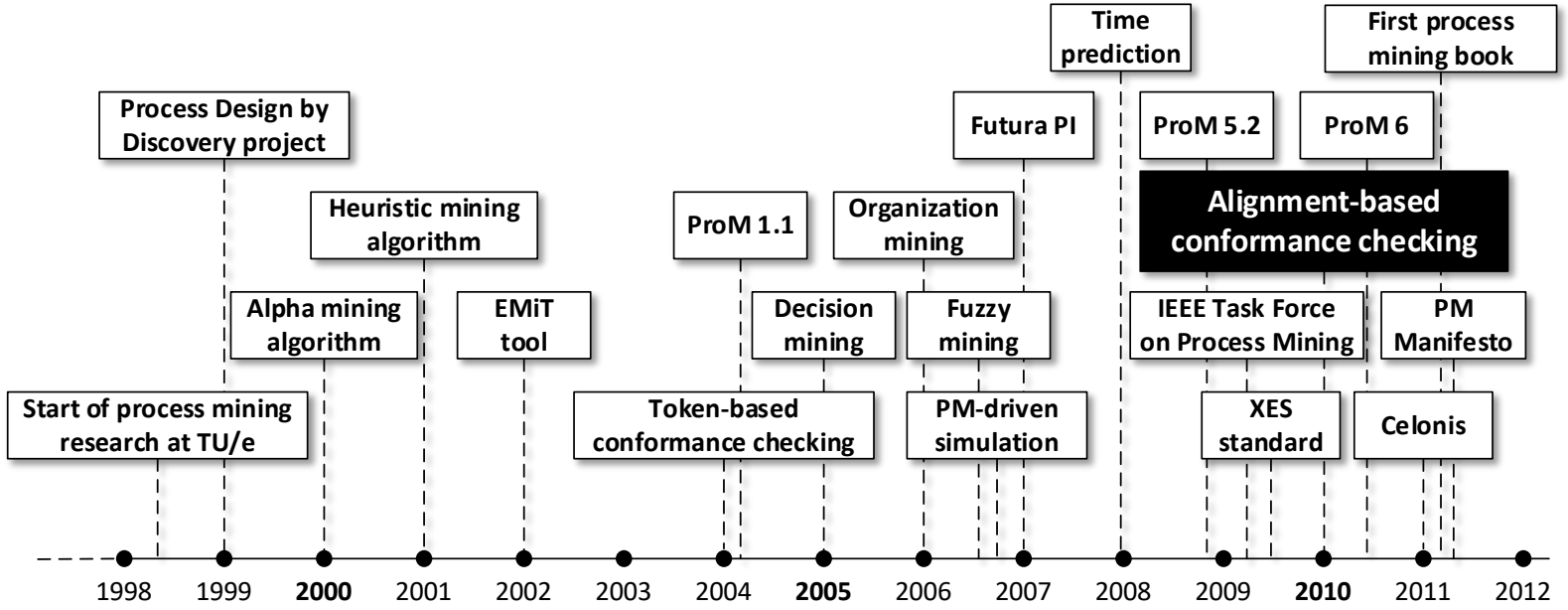
prof.dr.ir. Wil van der Aalst
Chief scientist Celonis & professor RWTH Aachen University

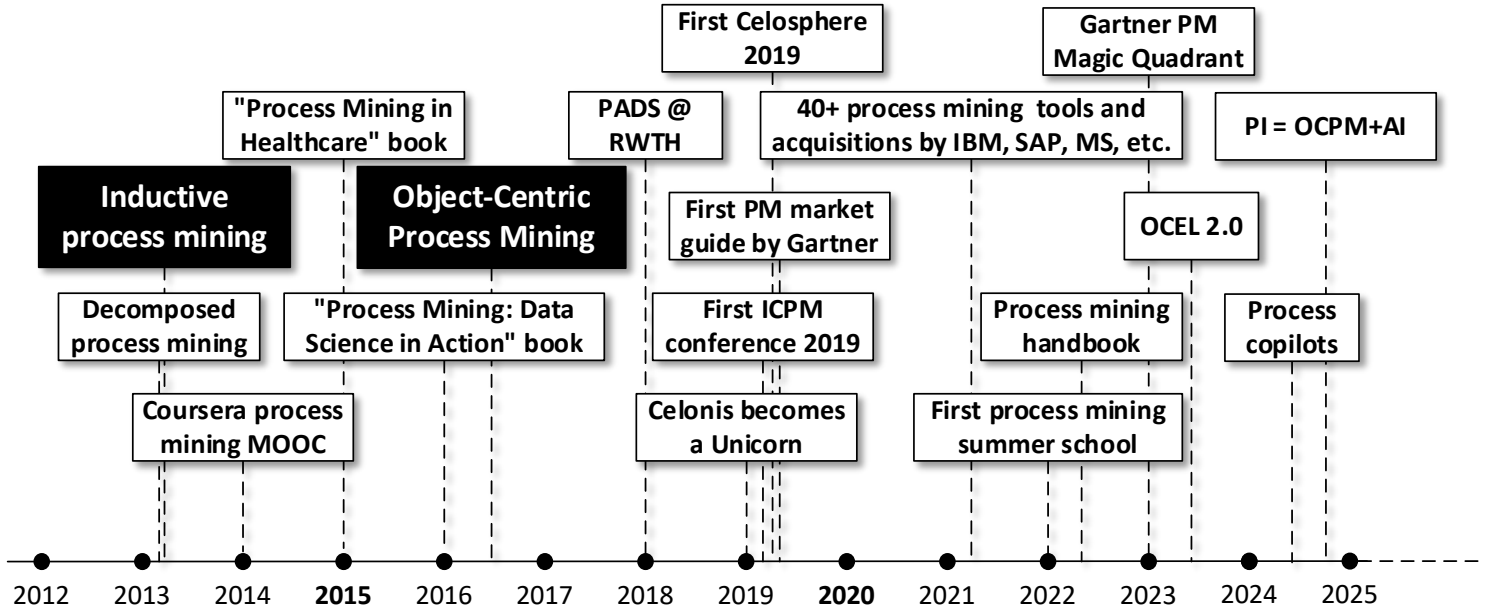
Timeline


Process Design by Discovery project
Heuristic mining algorithm
Alpha mining algorithm
EMiT tool
Start of process mining research at TU/e
ProM 1.1
Organization mining
Decision mining
Token-based conformance checking
Futura PI
Fuzzy mining
PM-driven simulation
Time prediction
ProM 5.2
Alignment-based conformance checking
IEEE Task Force on Process Mining
XES standard
ProM 6
First process mining book
PM Manifesto
Celonis
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012

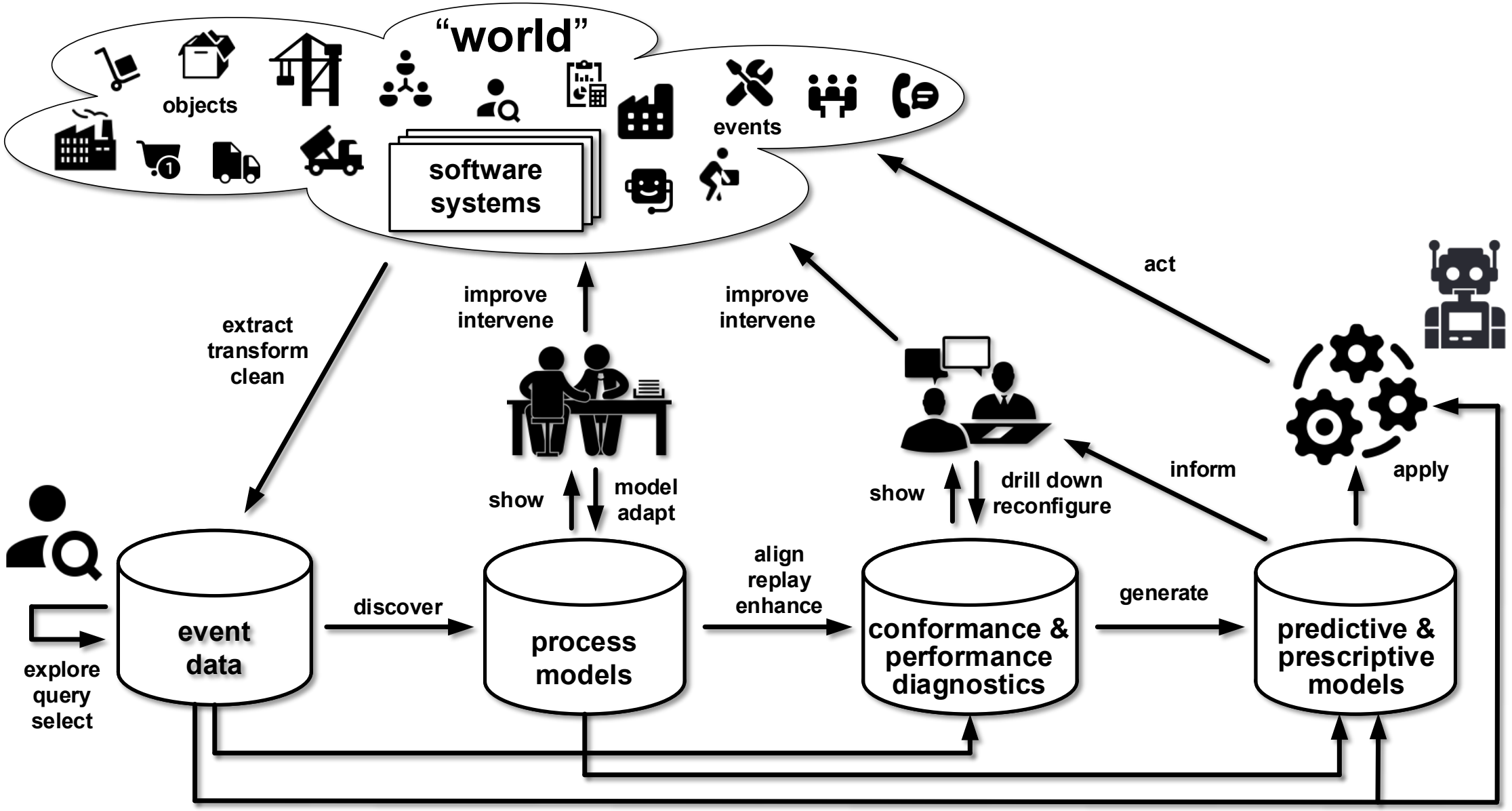

"Process Mining in Healthcare" book
Inductive process mining
Decomposed process mining
Coursera process mining MOOC
Object-Centric Process Mining
"Process Mining: Data Science in Action" book
PADS @ RWTH
First Celosphere 2019
First PM market guide by Gartner
First ICPM conference 2019
Celonis becomes a Unicorn
40+ process mining tools and acquisitions by IBM, SAP, MS, etc.
Gartner PM Magic Quadrant
Process mining handbook
First process mining summer school
PI = OCPM+AI
OCEL 2.0
Process copilots
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025

Timeline

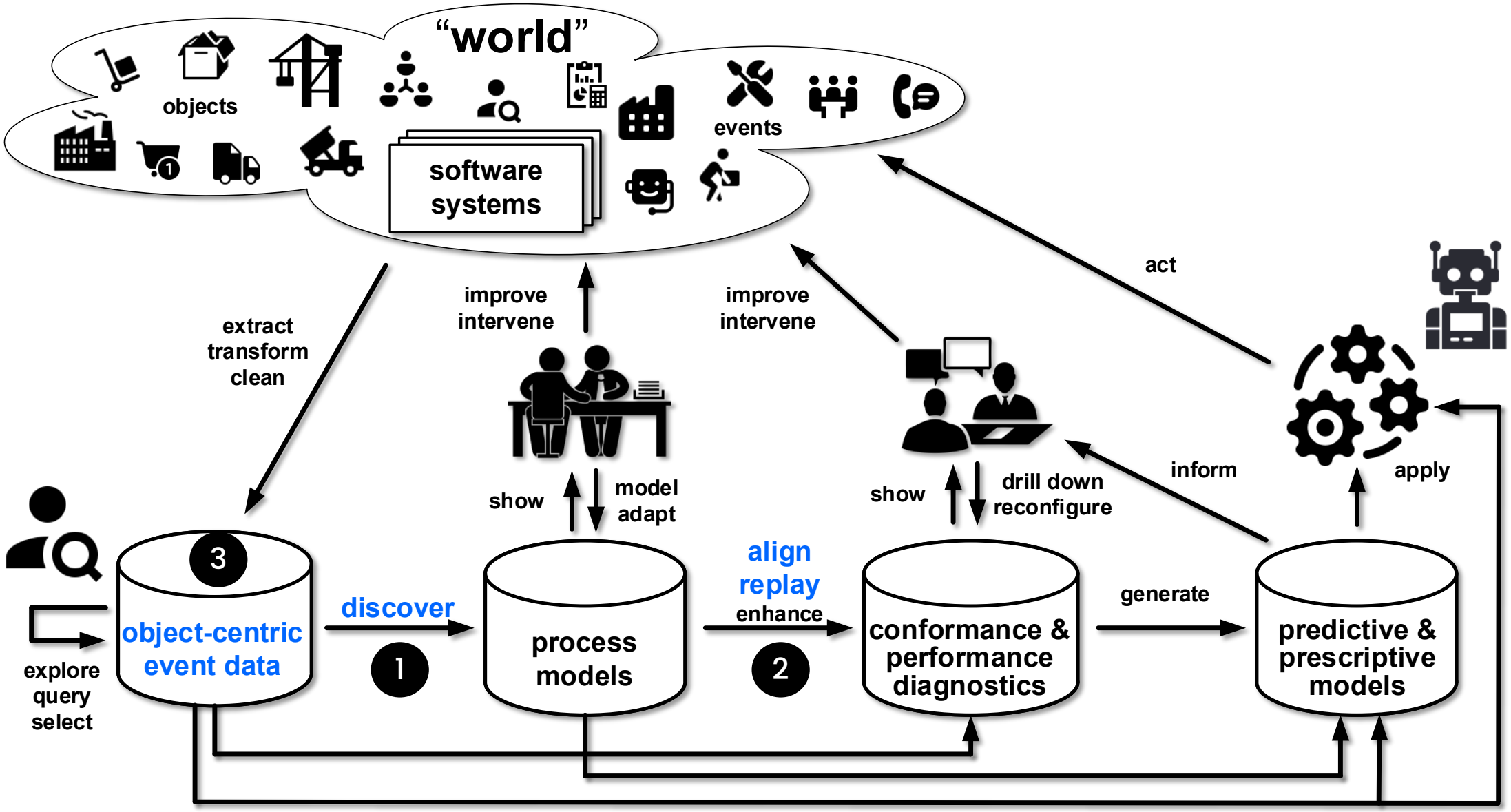

Process Design by Discovery project
Heuristic mining algorithm
Alpha mining algorithm
EMiT tool
ProM 1.1
Organization mining
Decision mining
Fuzzy mining
Futura PI
Time prediction
ProM 5.2
ProM 6
First process mining book
Alignment-based conformance checking
IEEE Task Force on Process Mining
PM Manifesto
Start of process mining research at TU/e
Token-based conformance checking
PM-driven simulation
XES standard
Celonis
1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012


First Celosphere 2019
Gartner PM Magic Quadrant
"Process Mining in Healthcare" book
PADS @ RWTH
40+ process mining tools and acquisitions by IBM, SAP, MS, etc.
PI = OCPM+AI
Inductive process mining
Object-Centric Process Mining
First PM market guide by Gartner
OCEL 2.0
Decomposed process mining
"Process Mining: Data Science in Action" book
First ICPM conference 2019
Process mining handbook
Process copilots
Coursera process mining MOOC
Celonis becomes a Unicorn
First process mining summer school
2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025

Overview


"world"
objects
events
software systems
extract
transform
clean
improve
intervene
improve
intervene
act
show
model
adapt
show
drill down
reconfigure
inform
apply
explore
query
select
event data
discover
process models
align
replay
enhance
conformance & performance diagnostics
generate
predictive & prescriptive models

Focus of today



Example Event Data (Case-Centric)

Case	Activity	Time
1	take order	2025-11-04 14:21
1	create base	2025-11-04 14:32
1	add tomato	2025-11-04 14:34
1	add salami	2025-11-04 14:34
1	add cheese	2025-11-04 14:35
1	bake in oven	2025-11-04 14:36
1	eat pizza	2025-11-04 14:40
1	clean kitchen	2025-11-04 14:48
2	take order	2025-11-04 19:07
2	create base	2025-11-04 19:18
2	add tomato	2025-11-04 19:19
2	add mushrooms	2025-11-04 19:19
2	add cheese	2025-11-04 19:19
2	bake in oven	2025-11-04 19:21
2	clean kitchen	2025-11-04 19:25
3	take order	2025-11-04 21:01
3	create base	2025-11-04 21:11
3	add cheese	2025-11-04 21:13
3	add mushrooms	2025-11-04 21:13
3	add tomato	2025-11-04 21:14
3	bake in oven	2025-11-04 21:16
3	eat pizza	2025-11-04 21:19
3	clean kitchen	2025-11-04 21:35
4	take order	2025-11-05 13:30
4	create base	2025-11-05 13:36
4	add cheese	2025-11-05 13:36
4	add tomato	2025-11-05 13:36
4	add salami	2025-11-05 13:37

...

Case	Activity	Time
1,275	add cheese	2026-08-15 20:39
1,275	bake in oven	2026-08-15 20:41
1,275	clean kitchen	2026-08-15 20:55
1,276	take order	2026-08-15 20:39
1,276	create base	2026-08-15 20:48
1,276	add mushrooms	2026-08-15 20:49
1,276	add cheese	2026-08-15 20:51
1,276	add tomato	2026-08-15 20:52
1,276	bake in oven	2026-08-15 20:55
1,276	eat pizza	2026-08-15 22:02
1,276	clean kitchen	2026-08-15 22:29
1,277	take order	2026-08-15 21:40
1,277	create base	2026-08-15 21:47
1,277	add salami	2026-08-15 21:49
1,277	add cheese	2026-08-15 21:49
1,277	add tomato	2026-08-15 21:51
1,277	bake in oven	2026-08-15 21:54
1,277	eat pizza	2026-08-15 23:01
1,277	clean kitchen	2026-08-15 23:13
1,278	take order	2026-08-15 23:38
1,278	create base	2026-08-15 23:56
1,278	add cheese	2026-08-15 23:57
1,278	add cheese	2026-08-15 23:58
1,278	add mushrooms	2026-08-15 23:58
1,278	add tomato	2026-08-16 00:56
1,278	bake in oven	2026-08-16 00:59
1,278	eat pizza	2026-08-16 02:04
1,278	clean kitchen	2026-08-16 08:13

1278
cases

10,404
events

9
activities

134
variants

Example Event Data (Case-Centric)

Case	Activity	Time
1	take order	2025-11-04 14:21
1	create base	2025-11-04 14:32
1	add tomato	2025-11-04 14:34
1	add salami	2025-11-04 14:34
1	add cheese	2025-11-04 14:35
1	bake in oven	2025-11-04 14:36
1	eat pizza	2025-11-04 14:40
1	clean kitchen	2025-11-04 14:48
2	take order	2025-11-04 19:07
2	create base	2025-11-04 19:18
2	add tomato	2025-11-04 19:19
2	add mushrooms	2025-11-04 19:19
2	add cheese	2025-11-04 19:19
2	bake in oven	2025-11-04 19:21
3	add cheese	2025-11-04 21:13
3	add mushrooms	2025-11-04 21:13
3	add tomato	2025-11-04 21:14
3	bake in oven	2025-11-04 21:16
3	eat pizza	2025-11-04 21:19
3	clean kitchen	2025-11-04 21:35
4	take order	2025-11-05 13:30
4	create base	2025-11-05 13:36
4	add cheese	2025-11-05 13:36
4	add tomato	2025-11-05 13:36
4	add salami	2025-11-05 13:37

...

Case	Activity	Time
1,275	add cheese	2026-08-15 20:39
1,275	bake in oven	2026-08-15 20:41
1,275	clean kitchen	2026-08-15 20:55
1,276	take order	2026-08-15 20:39
1,276	create base	2026-08-15 20:48
1,276	add mushrooms	2026-08-15 20:49
1,276	add cheese	2026-08-15 20:51
1,276	add tomato	2026-08-15 20:52
1,276	bake in oven	2026-08-15 20:55
1,276	eat pizza	2026-08-15 22:02
1,276	clean kitchen	2026-08-15 22:29
1,277	take order	2026-08-15 21:40
1,277	create base	2026-08-15 21:47
1,277	add salami	2026-08-15 21:49
1,277	clean kitchen	2026-08-15 23:13
1,278	take order	2026-08-15 23:38
1,278	create base	2026-08-15 23:56
1,278	add cheese	2026-08-15 23:57
1,278	add cheese	2026-08-15 23:58
1,278	add mushrooms	2026-08-15 23:58
1,278	add tomato	2026-08-16 00:56
1,278	bake in oven	2026-08-16 00:59
1,278	eat pizza	2026-08-16 02:04
1,278	clean kitchen	2026-08-16 08:13

Event = Case + Activity + Timestamp + …

Example Event Data (Case-Centric)

Case Id	# of Activities	Throughput Time	First Activity	First Activity Timestamp	Last Activity	Last Activity Timestamp
1	8	27 min	take order	11/04/25 14:21:34	clean kitchen	11/04/25 14:48:27
2	7	19 min	take order	11/04/25 19:07:09	clean kitchen	11/04/25 19:25:48
3	8	34 min	take order	11/04/25 21:01:05	clean kitchen	11/04/25 21:35:02
4	7	11 min	take order	11/05/25 13:30:25	clean kitchen	11/05/25 13:41:11
5	8	1 h	take order	11/05/25 18:35:38	clean kitchen	11/05/25 19:50:34
6	9	3 h	take order	11/05/25 20:58:03	clean kitchen	11/06/25 00:08:20
7	8	21 min	take order	11/06/25 14:03:57	clean kitchen	11/06/25 14:24:52
8	8	24 min	take order	11/06/25 18:05:07	clean kitchen	11/06/25 18:29:00
9	8	28 min	take order	11/06/25 19:21:25	clean kitchen	11/06/25 19:48:58
10	7	1 h	take order	11/06/25 20:54:02	clean kitchen	11/06/25 22:06:13
11	7	14 min	take order	11/07/25 13:16:53	clean kitchen	11/07/25 13:31:00
12	8	1 h	take order	11/07/25 14:37:26	clean kitchen	11/07/25 16:05:09
13	9	21 min	take order	11/07/25 18:37:33	clean kitchen	11/07/25 18:58:57
14	8	28 min	take order	11/07/25 20:08:29	clean kitchen	11/07/25 20:36:52
15	8	18 min	take order	11/07/25 21:12:15	clean kitchen	11/07/25 21:30:01
16	9	25 min	take order	11/08/25 12:14:16	clean kitchen	11/08/25 12:39:10
17	9	1 h	take order	11/08/25 12:56:59	clean kitchen	11/08/25 14:11:16
18	8	10 min	take order	11/08/25 14:16:50	clean kitchen	11/08/25 14:27:01
19	10	2 h	take order	11/08/25 14:59:33	clean kitchen	11/08/25 16:56:56
20	8	19 min	take order	11/08/25 18:17:49	clean kitchen	11/08/25 18:36:30
21	8	19 min	take order	11/08/25 18:39:10	clean kitchen	11/08/25 18:58:26
22	8	18 min	take order	11/08/25 19:39:42	clean kitchen	11/08/25 19:57:18
23	8	18 min	take order	11/08/25 20:40:45	clean kitchen	11/08/25 20:58:25

Case details: 1

Search

Activities **8** Items

- take order — 11/04/25 14:21:34
- create base — 11/04/25 14:32:37 — +11m
- add tomato — 11/04/25 14:34:02 — +12m
- add salami — 11/04/25 14:34:36 — +13m
- add cheese — 11/04/25 14:35:01 — +13m
- bake in oven — 11/04/25 14:36:41 — +15m
- eat pizza — 11/04/25 14:40:08 — +19m
- clean kitchen — 11/04/25 14:48:27 — +27m

Three example cases

Case details: 1090

Search

Activities **7** Items

take order	7/09/26 19:29:11	
create base	7/09/26 19:42:11	+13m
add tomato	7/09/26 19:43:51	+15m
add cheese	7/09/26 19:43:55	+15m
add salami	7/09/26 19:43:58	+15m
bake in oven	7/09/26 19:46:26	+17m
clean kitchen	7/09/26 19:52:06	+23m

Case details: 1262

Search

Activities **8** Items

take order	8/14/26 13:02:42	
create base	8/14/26 13:09:36	+7m
add tomato	8/14/26 13:11:18	+9m
add cheese	8/14/26 13:11:57	+9m
add mushrooms	8/14/26 13:12:37	+10m
bake in oven	8/14/26 13:15:24	+13m
eat pizza	8/14/26 13:25:49	+23m
clean kitchen	8/14/26 13:36:07	+33m

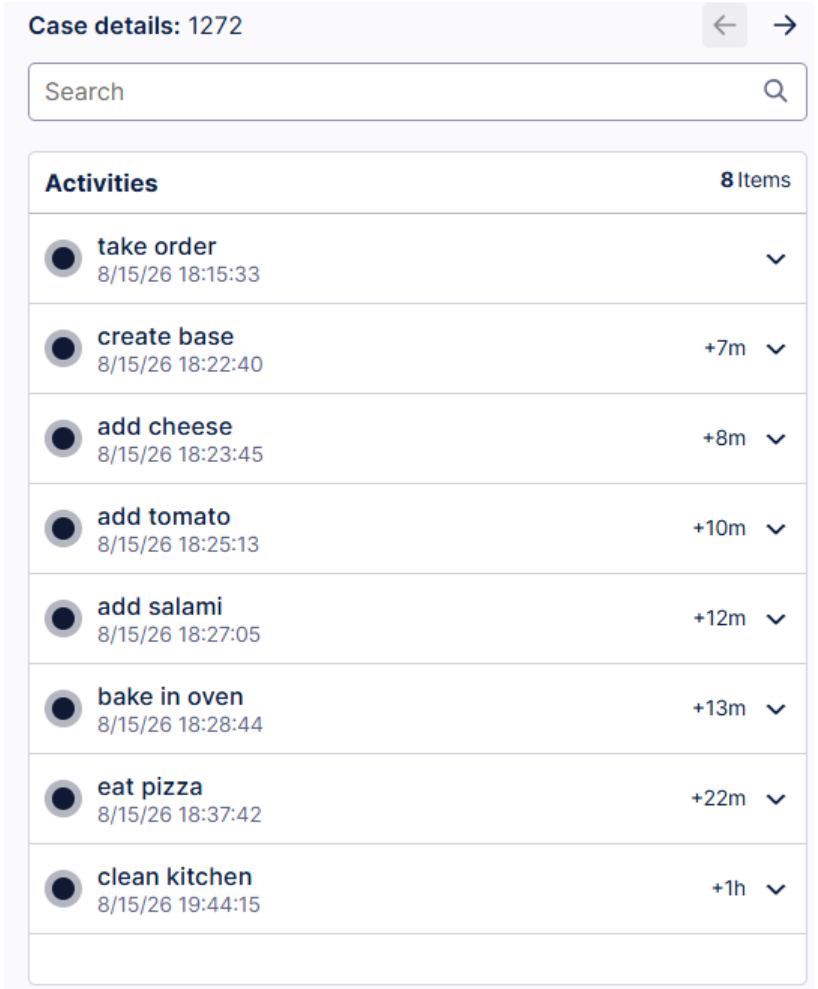
Case details: 1278

Search

Activities **9** Items

take order	8/15/26 23:38:26	
create base	8/15/26 23:56:24	+18m
add cheese	8/15/26 23:57:34	+19m
add cheese	8/15/26 23:58:42	+20m
add mushrooms	8/15/26 23:58:48	+20m
add tomato	8/16/26 00:56:48	+1h
bake in oven	8/16/26 00:59:46	+1h
eat pizza	8/16/26 02:04:50	+2h
clean kitchen		+9h

A case corresponds to a sequence of activities (trace)

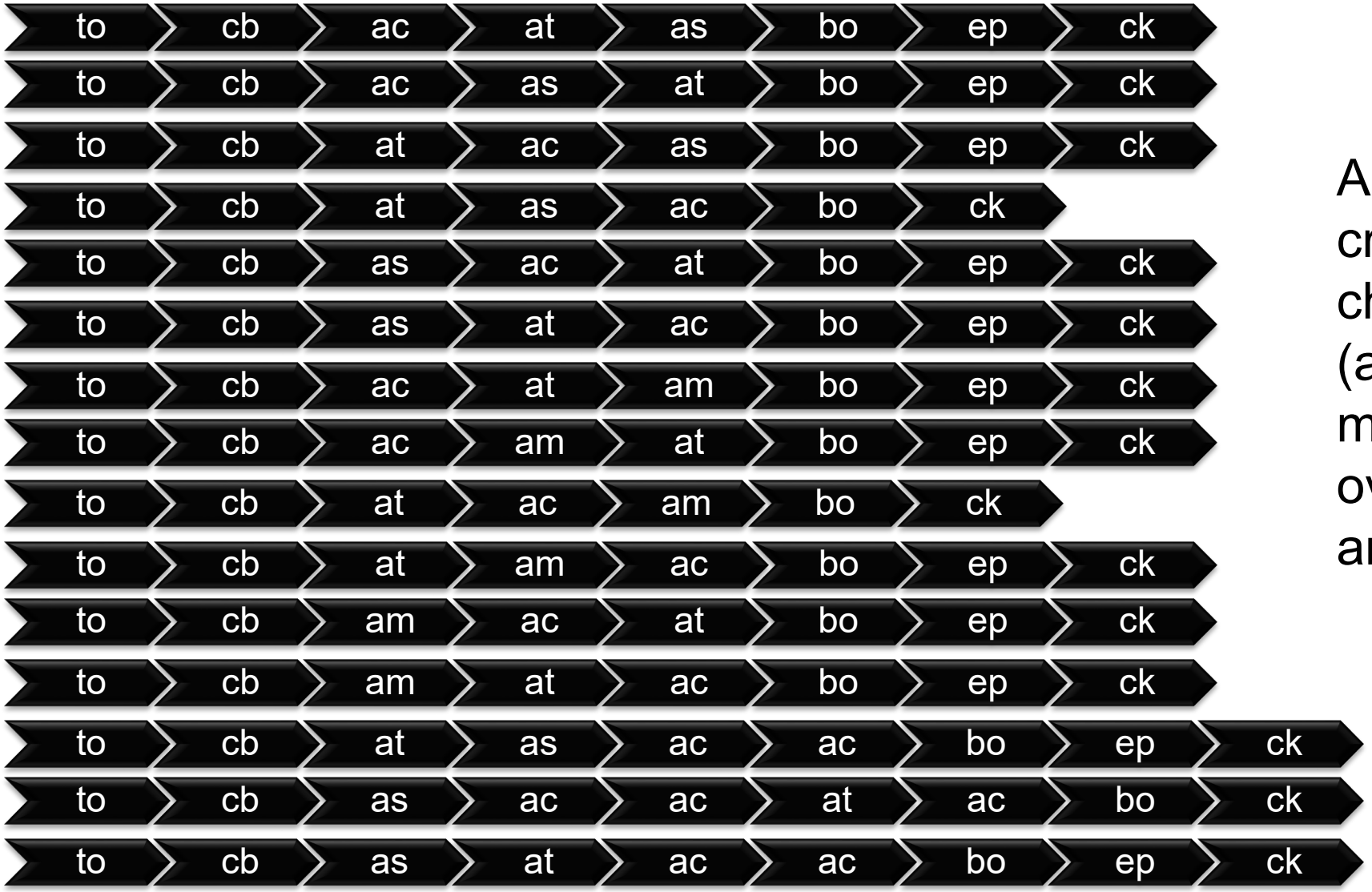


to
cb
ac
at
as
bo
ep
ck

Activities: take order (to), 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).

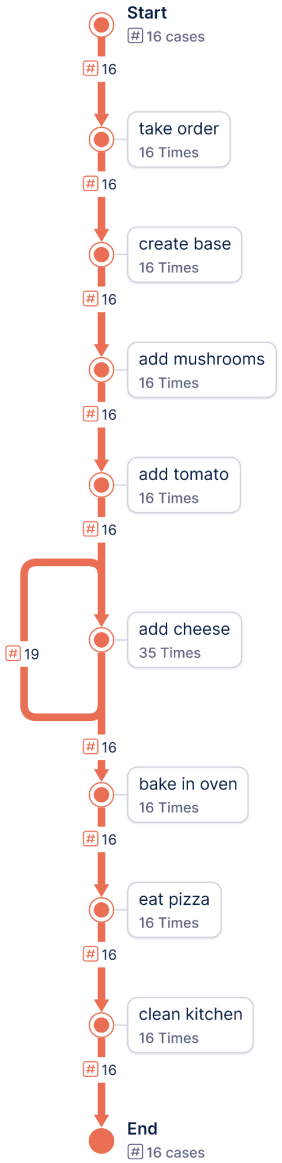
to cb ac at as bo ep ck

Some of the 134 unique variants

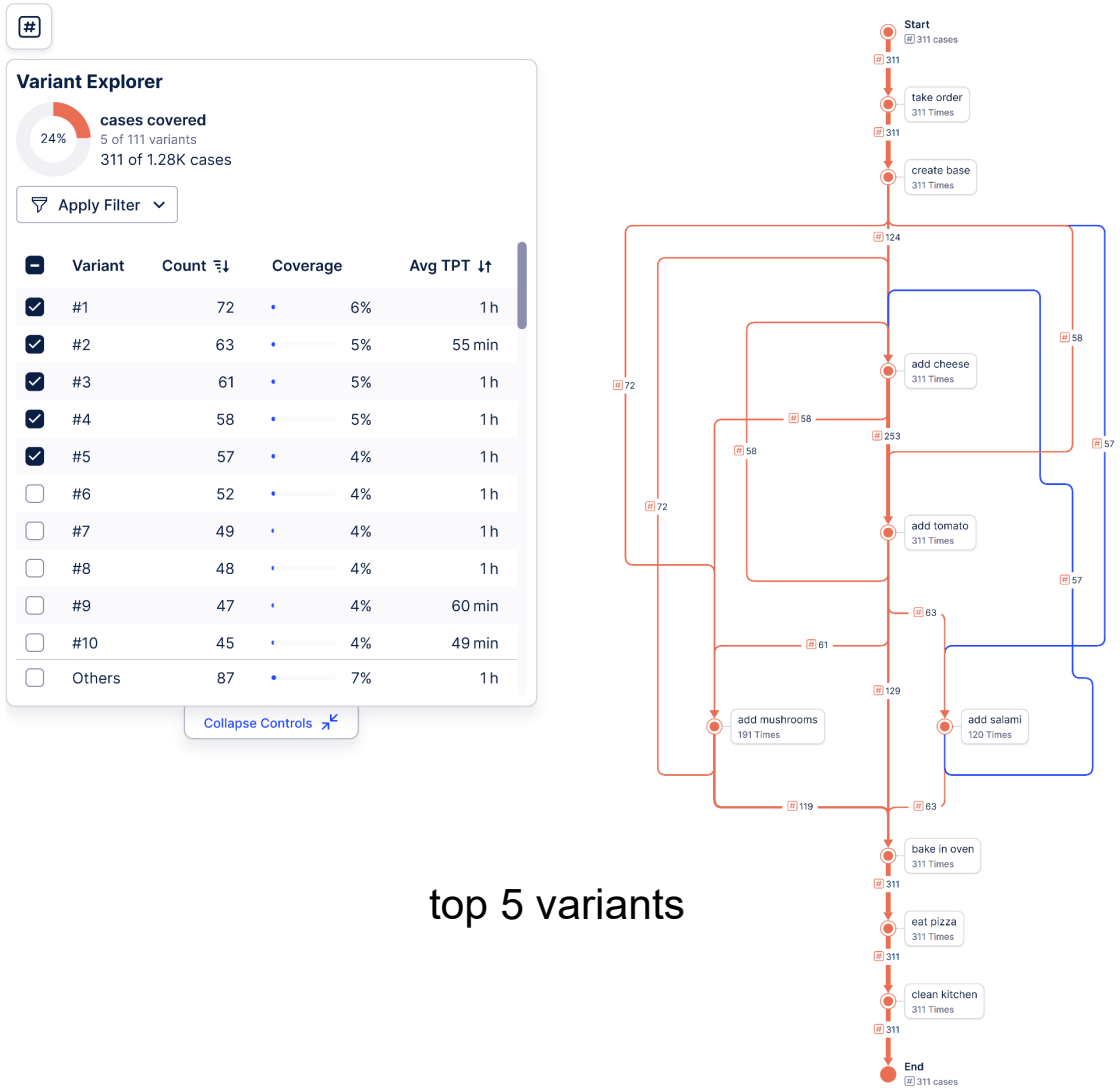


Activities: take order (to), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

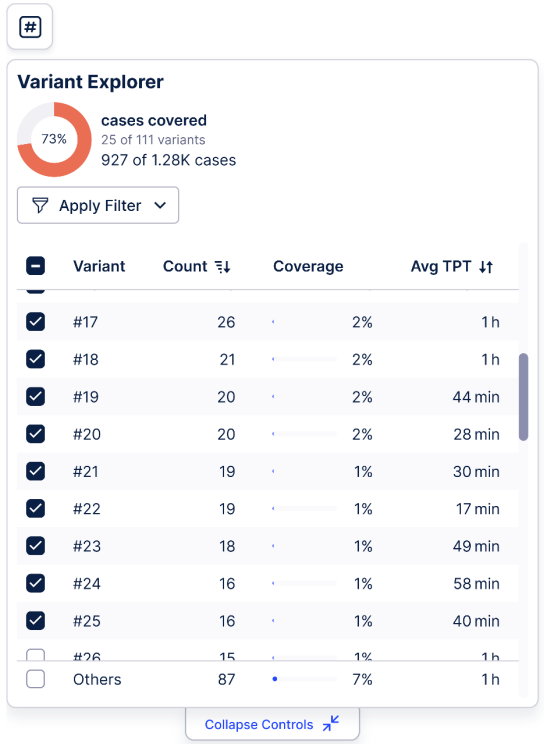
Variant explorer only shows 111 variants

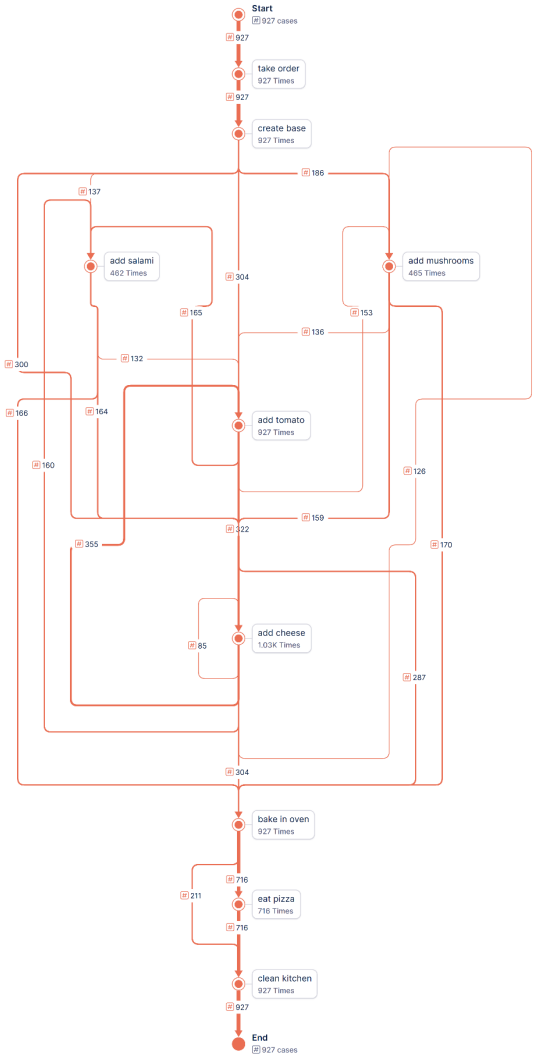


Visualizing multiple variants



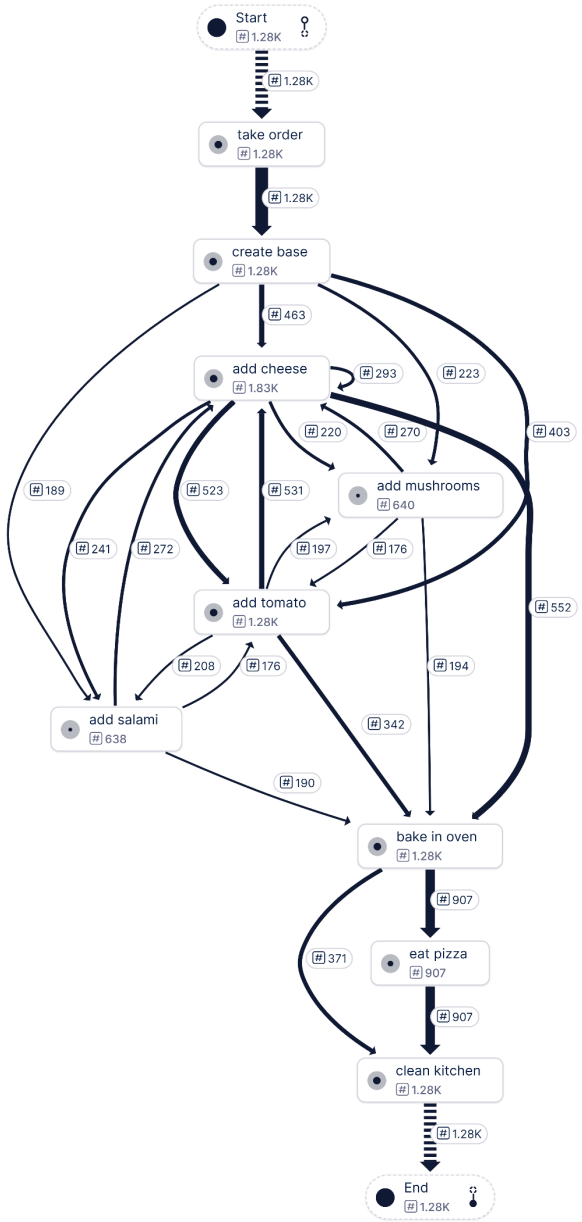
top 5 variants



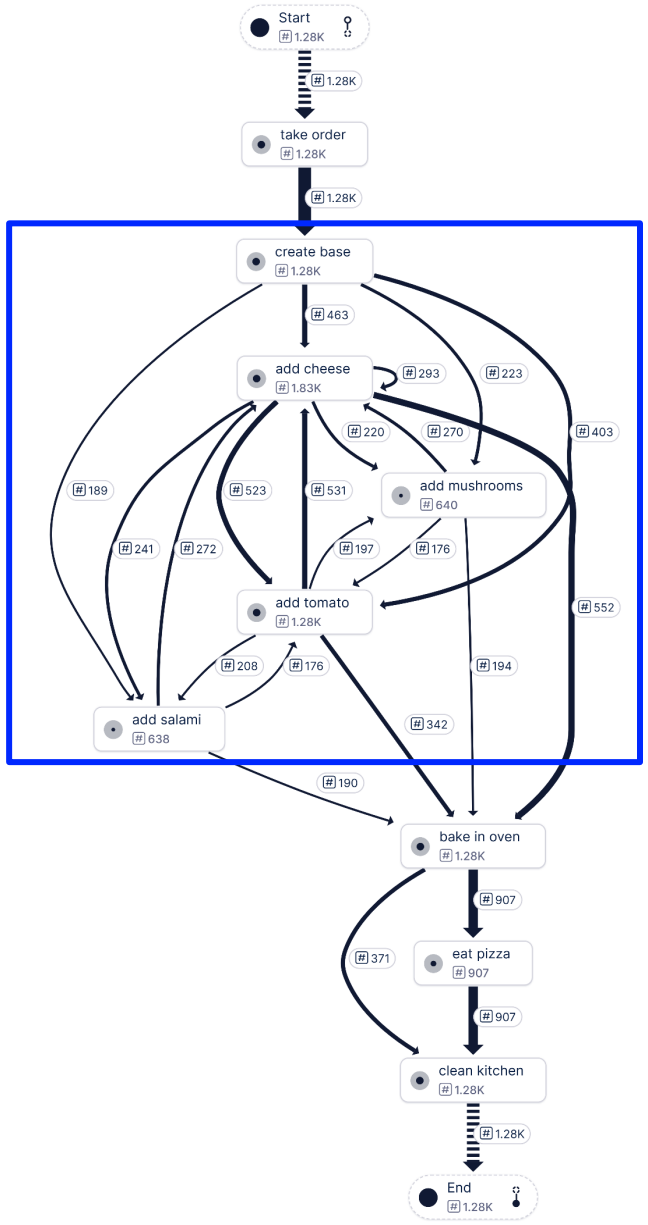


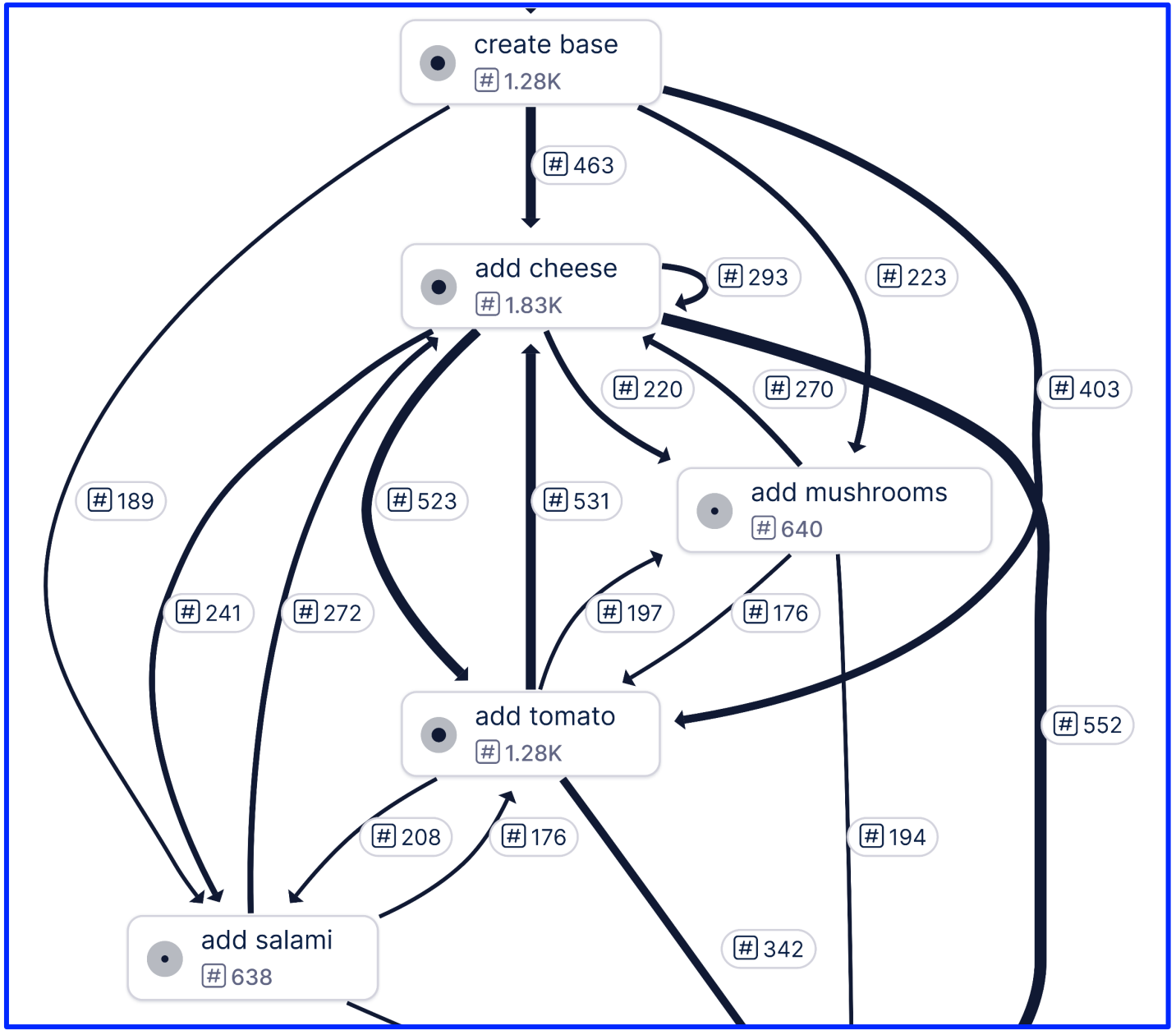
top 25 variants

The corresponding Directly-Follows Graph (DFG)

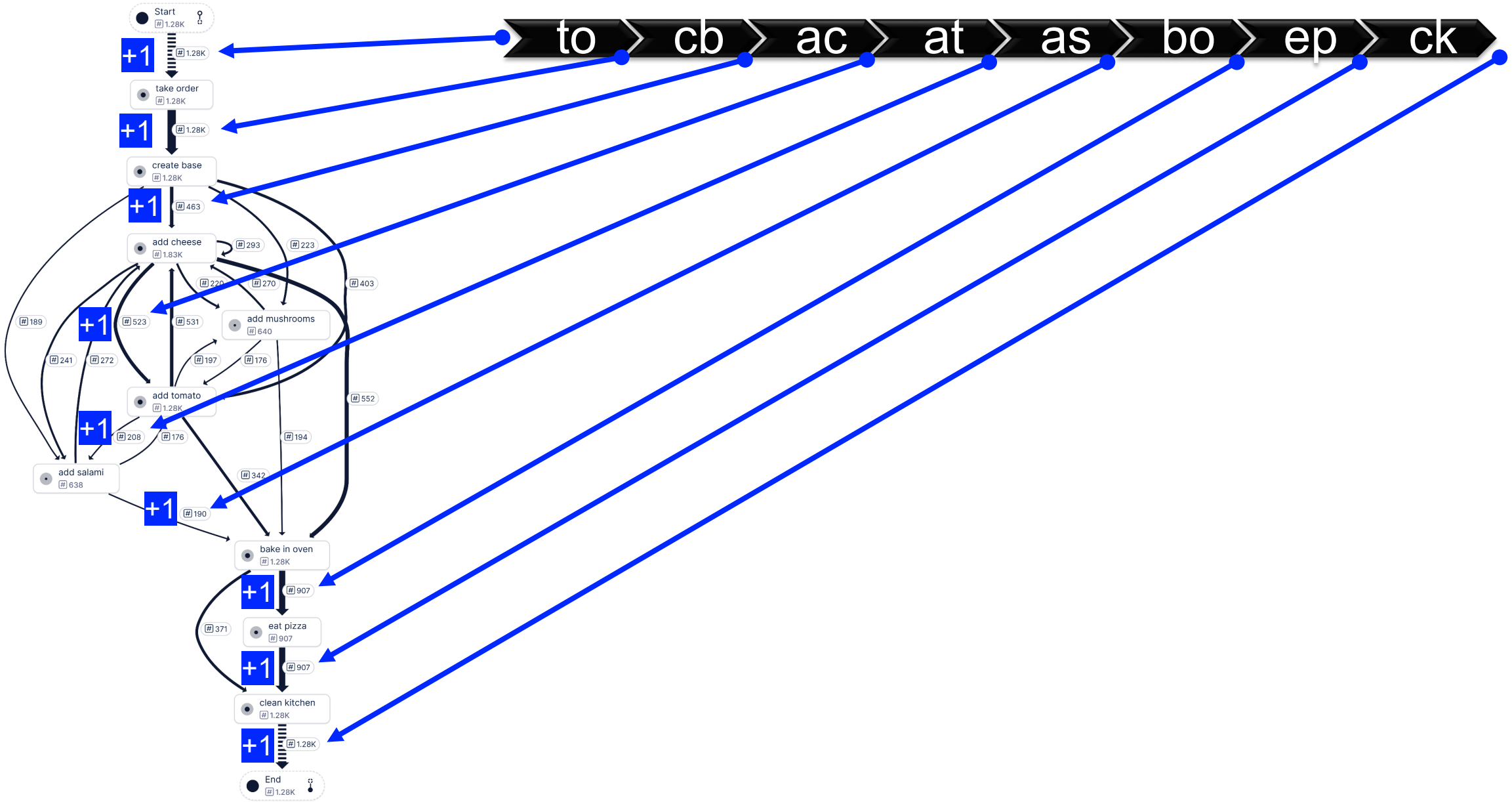


The corresponding Directly-Follows Graph (DFG)



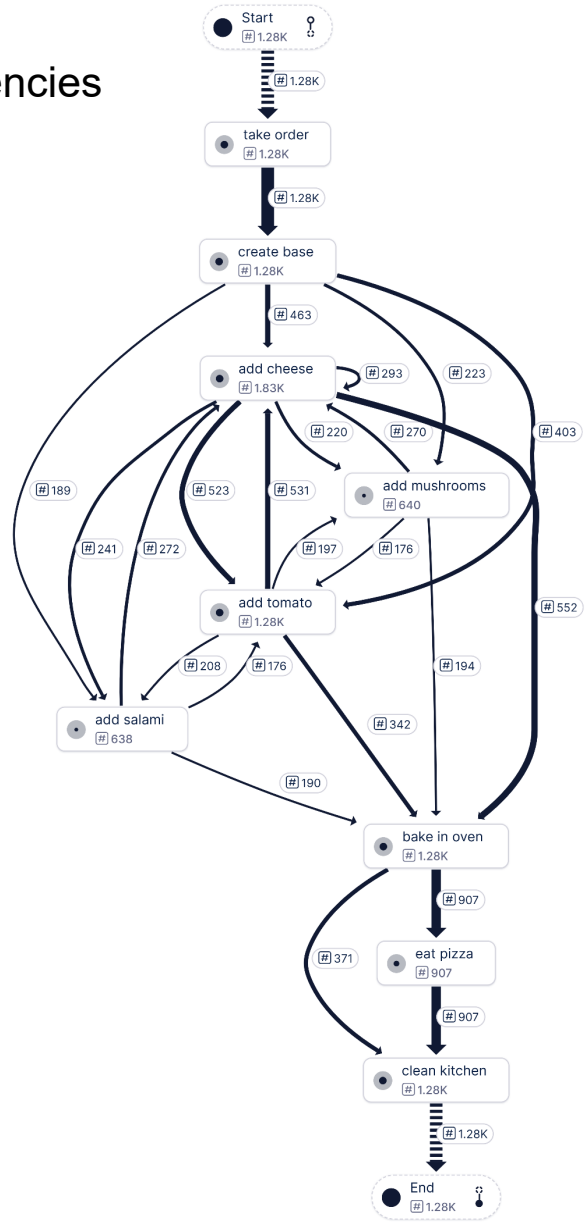


The corresponding Directly-Follows Graph (DFG)

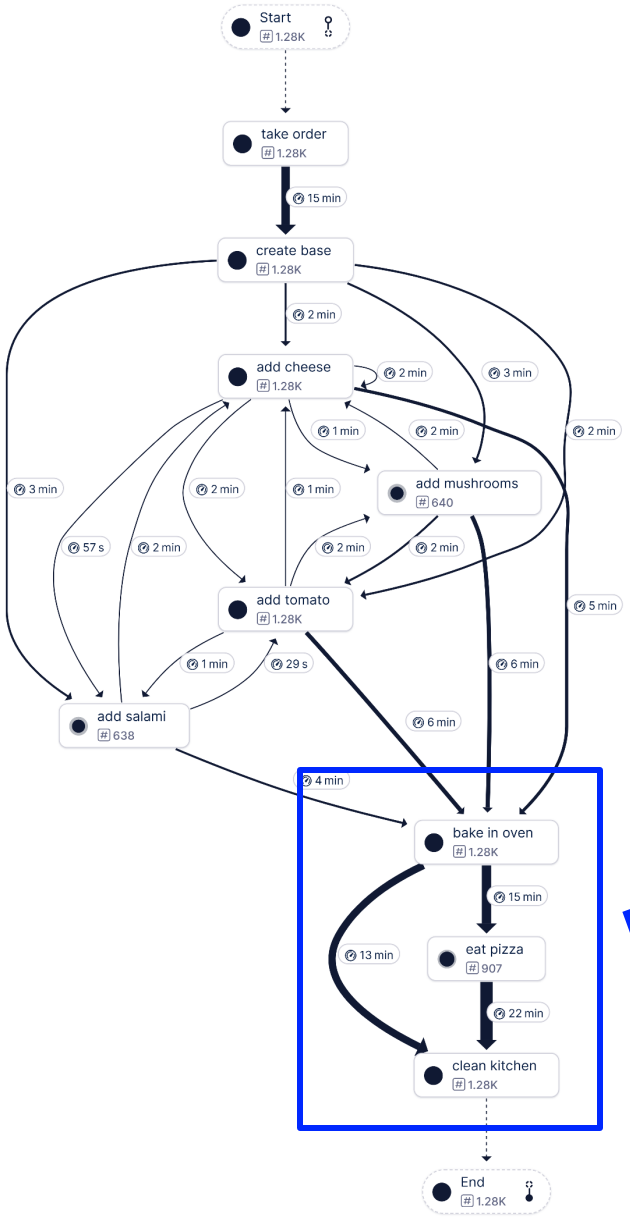


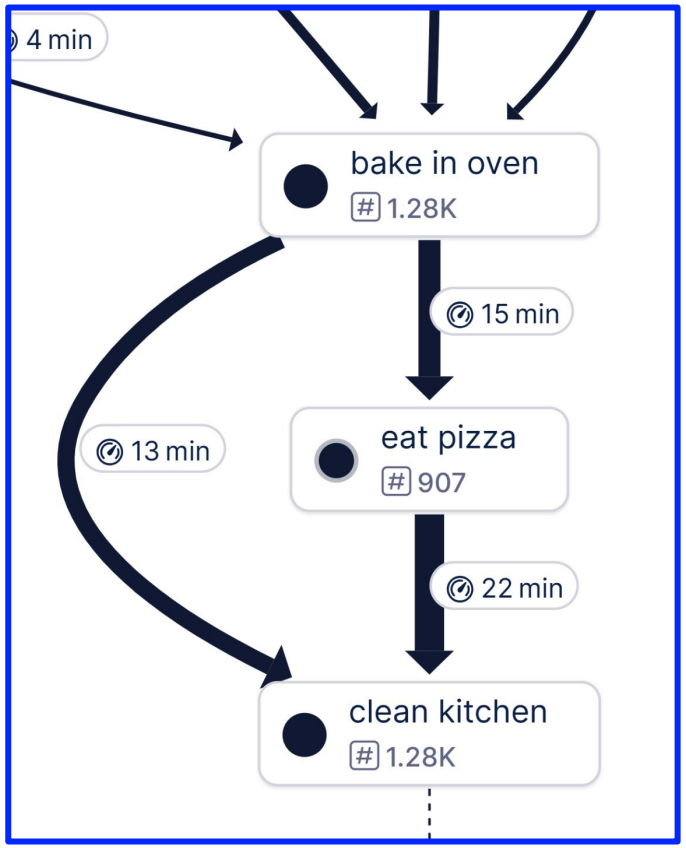
The corresponding Directly-Follows Graph (DFG)

frequencies

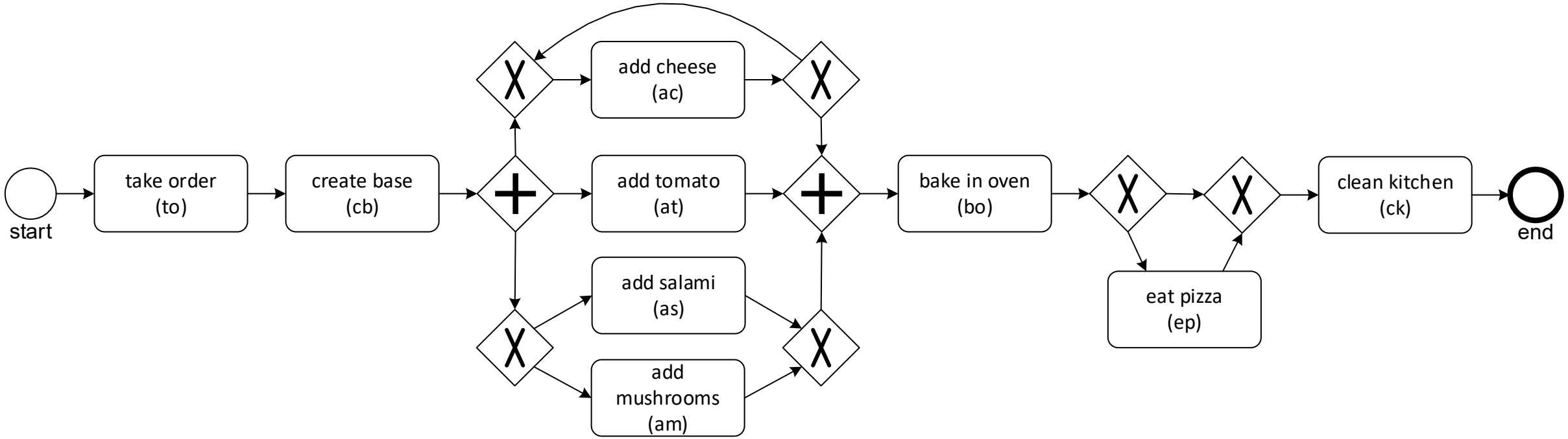


time

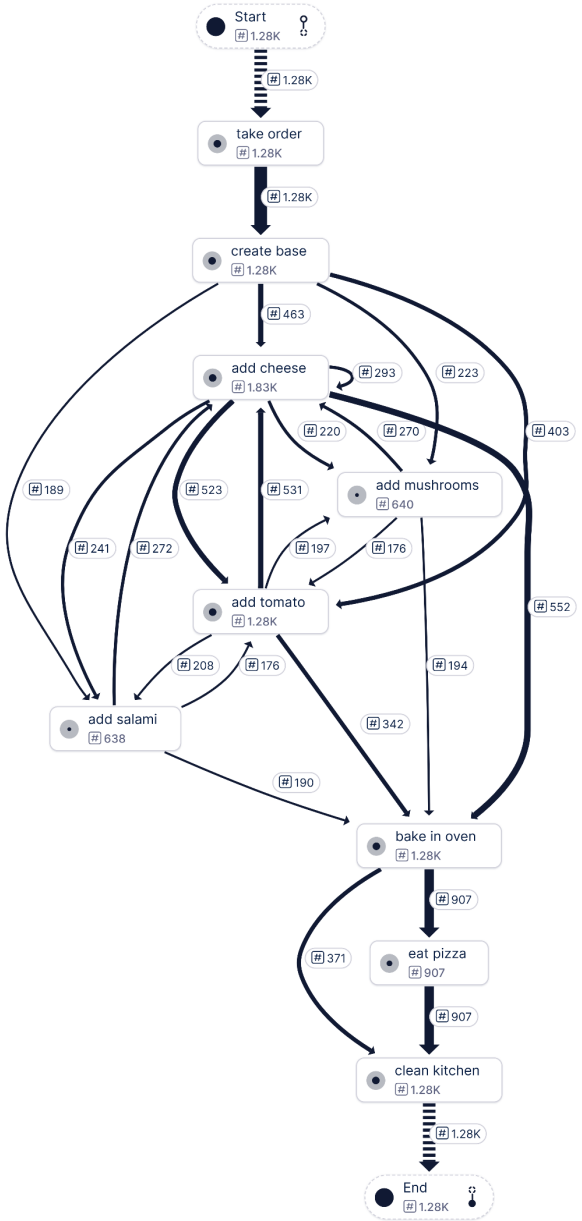


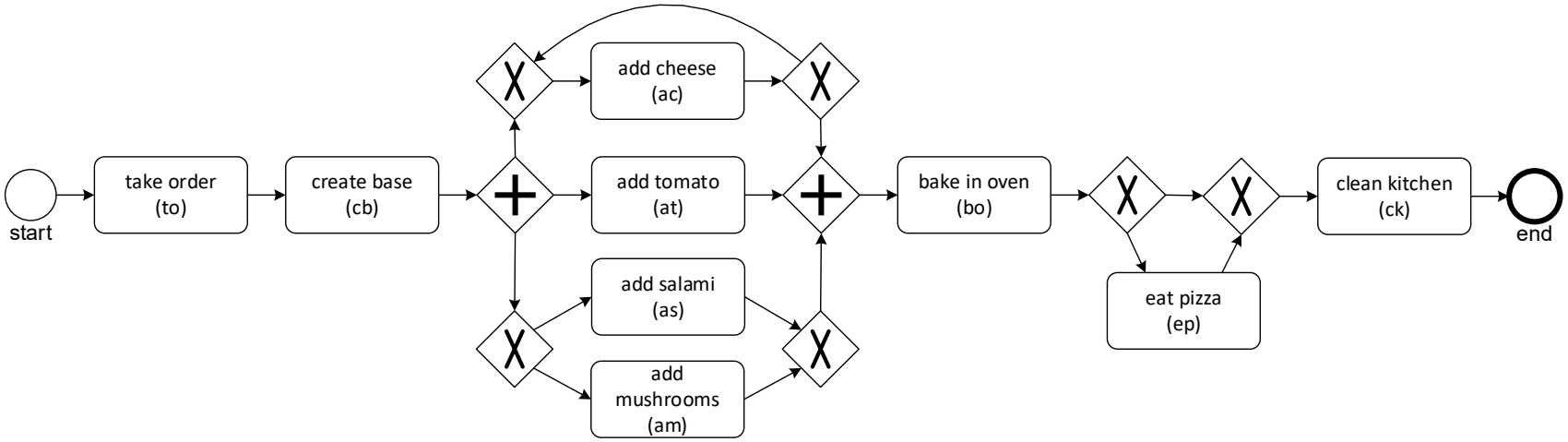


Using Inductive Mining (IM) we can automatically discover this BPMN model !



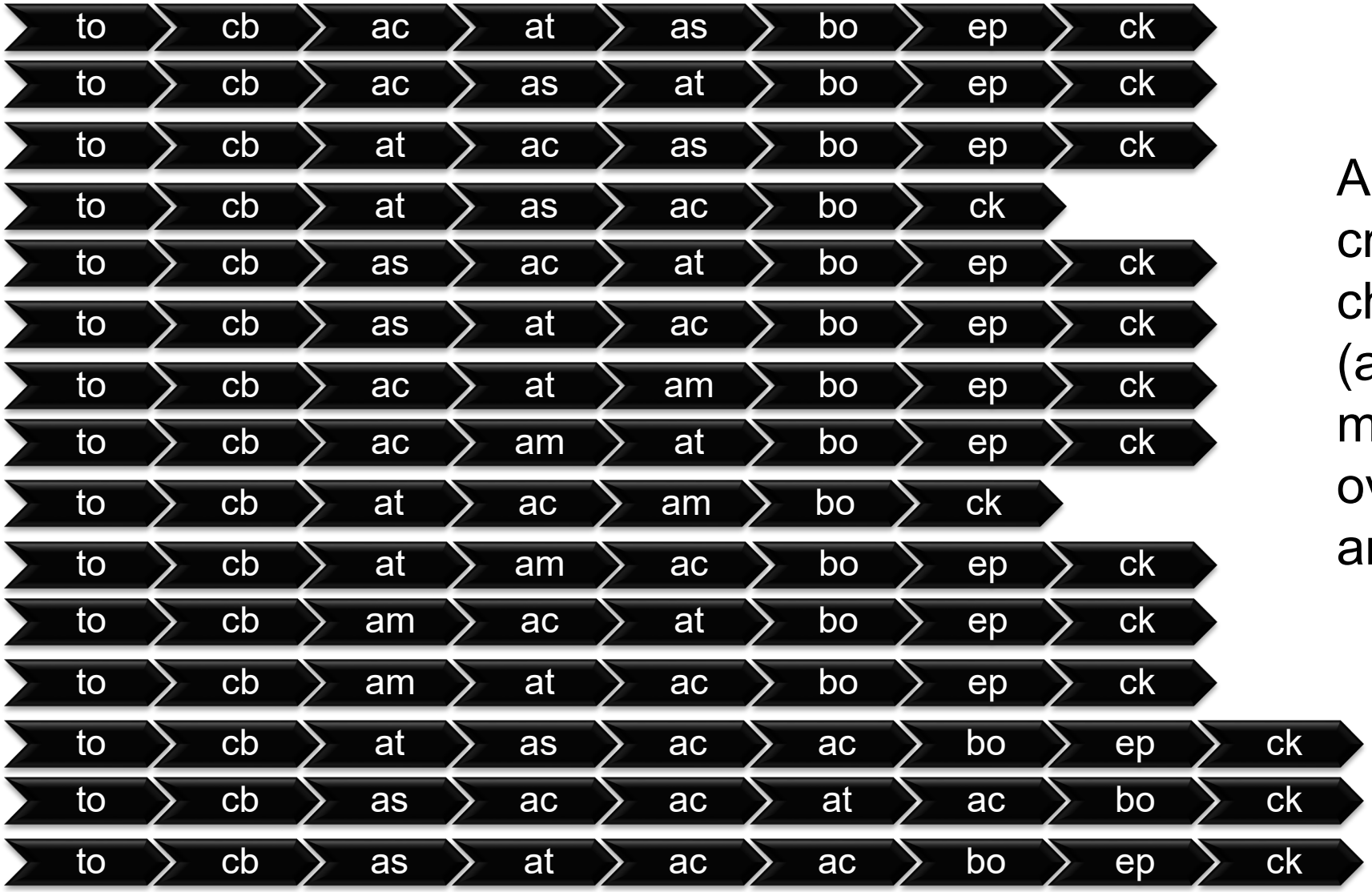
The BPMN model is easier to understand and more precise!





- There was no pizza with both salami and mushrooms.
- There was no pizza with multiple portions of tomato, salami, or mushrooms.
- There was no pizza without tomato.
- Etc.

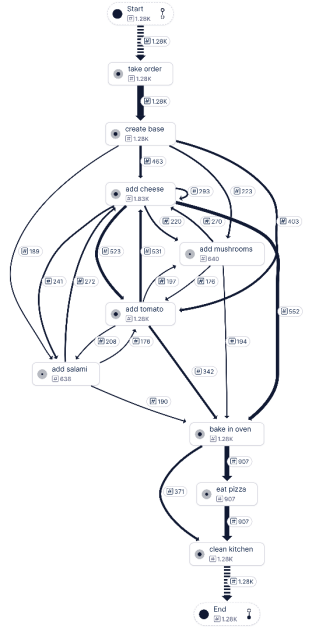
How does it work? Start with the event data.

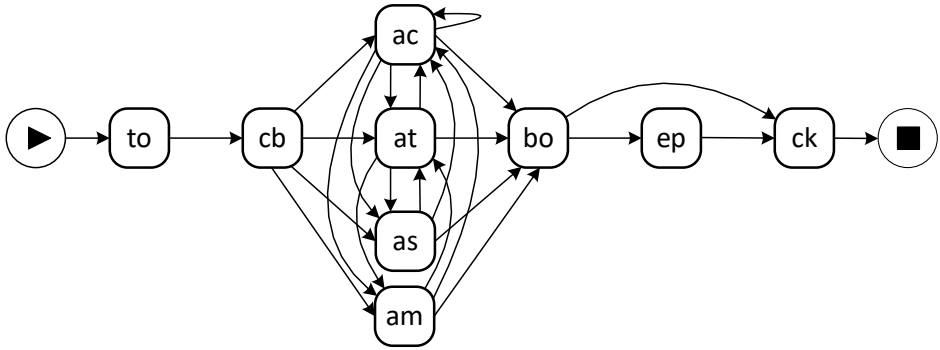


Activities: take order (to), create base (cb), add cheese (ac), add tomato (at), add salami (as), add mushrooms (am), bake in oven (bo), eat pizza (ep), and clean kitchen (ck).

Create the DFG

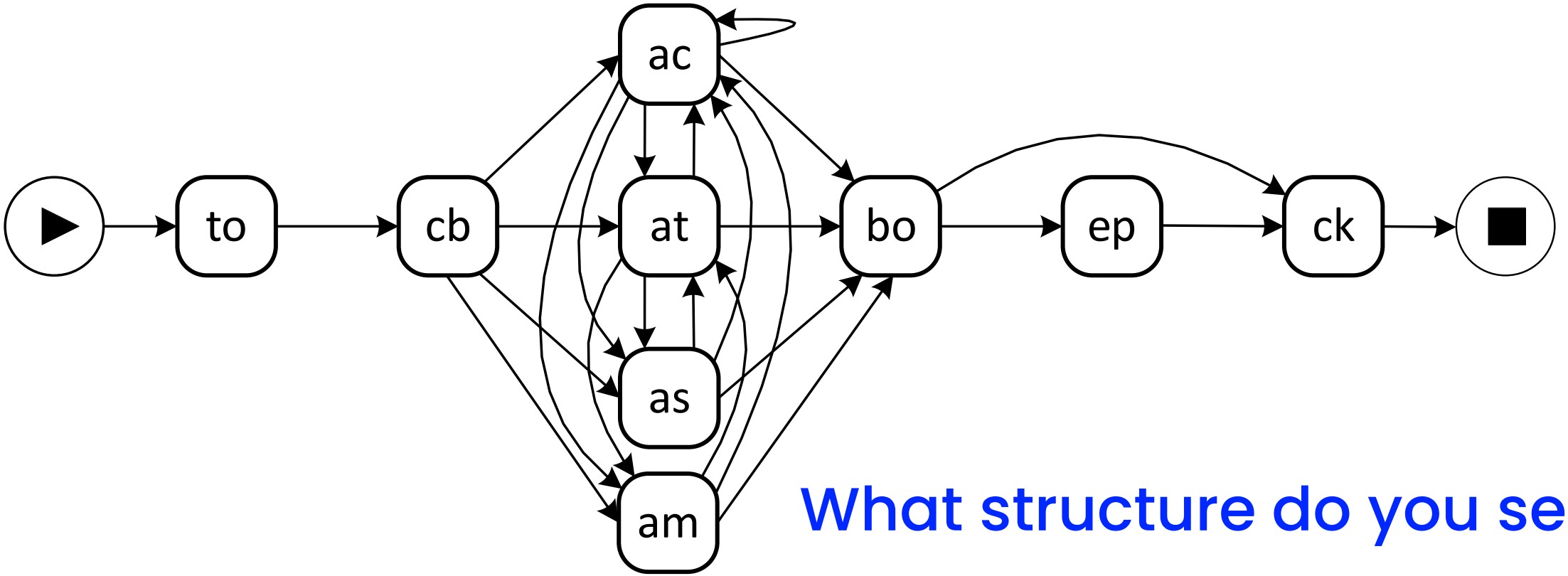
to cb ac at as bo ep ck
to cb ac as at bo ep ck
to cb at ac as bo ep ck
to cb at as ac bo ck
to cb as ac at bo ep ck
to cb as at ac bo ep ck
to cb ac at am bo ep ck
to cb ac am at bo ep ck
to cb at ac am bo ck
to cb at am ac bo ep ck
to cb am ac at bo ep ck
to cb am at ac bo ep ck
to cb at as ac ac bo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





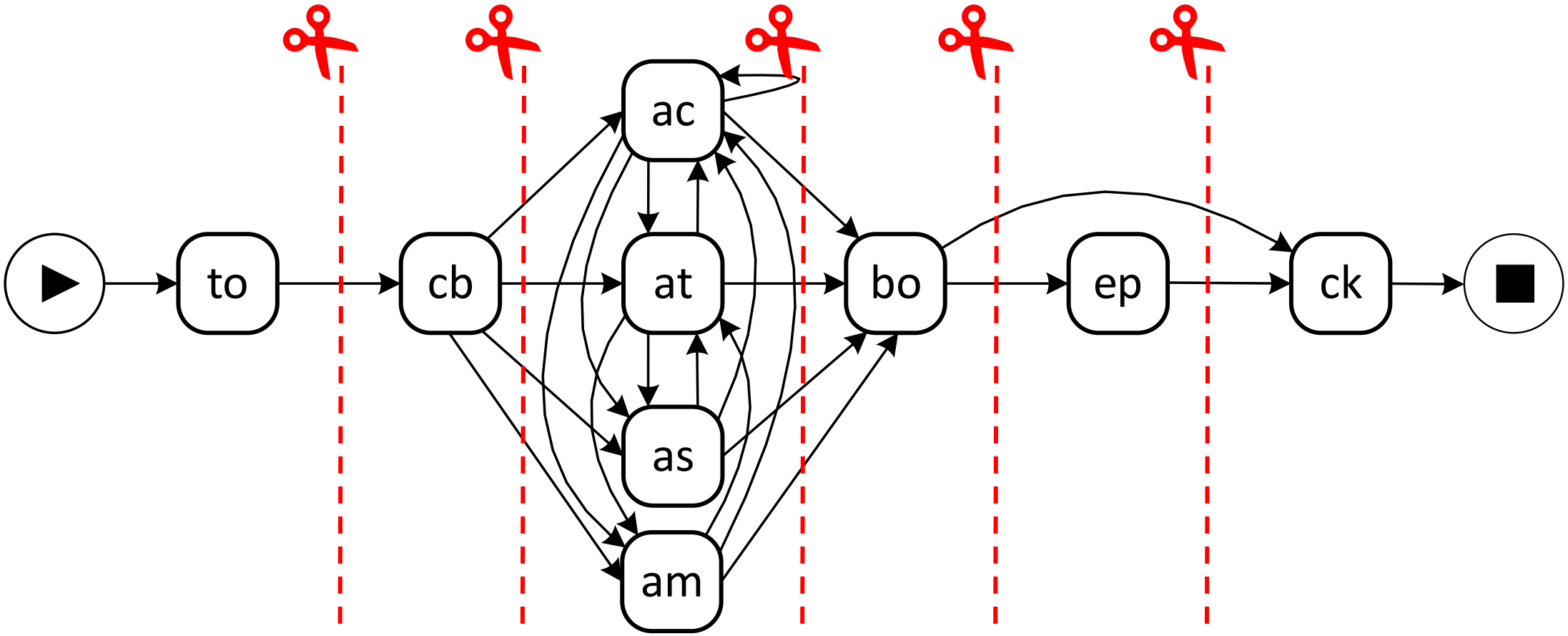
frequencies omitted and layout improved for readability

We want to use a divide-and-conquer strategy.



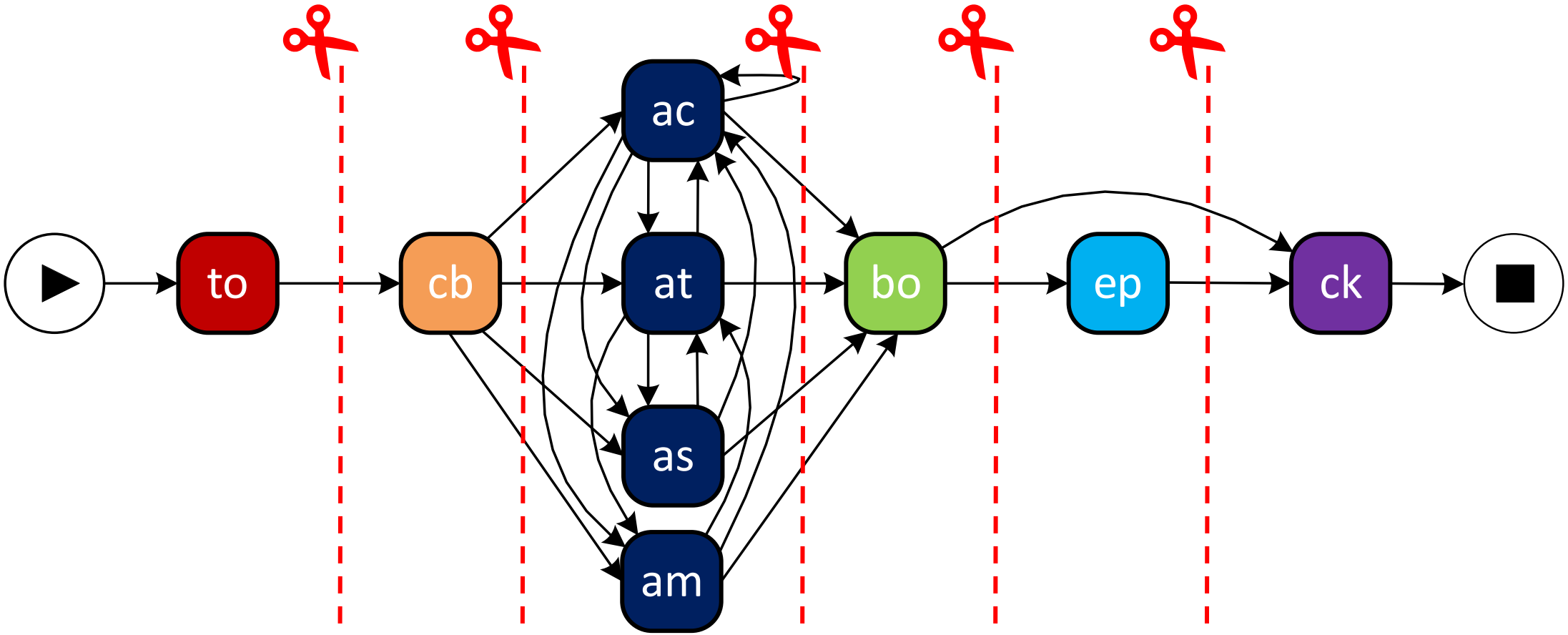
What structure do you see?

Apply a sequence cut



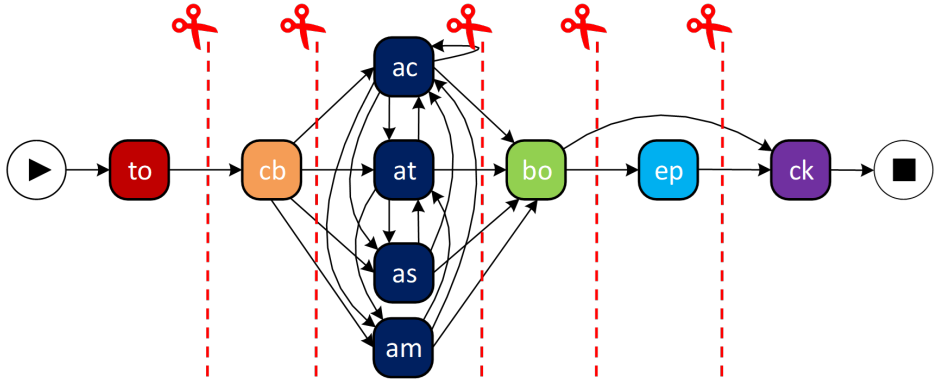
There is a sequence cut when the DFG can be split into sequential parts where only "forward connections" are possible. Note that we need to use the non-reflexive transitive closure.

Sequence cut partitions activities in six subsets



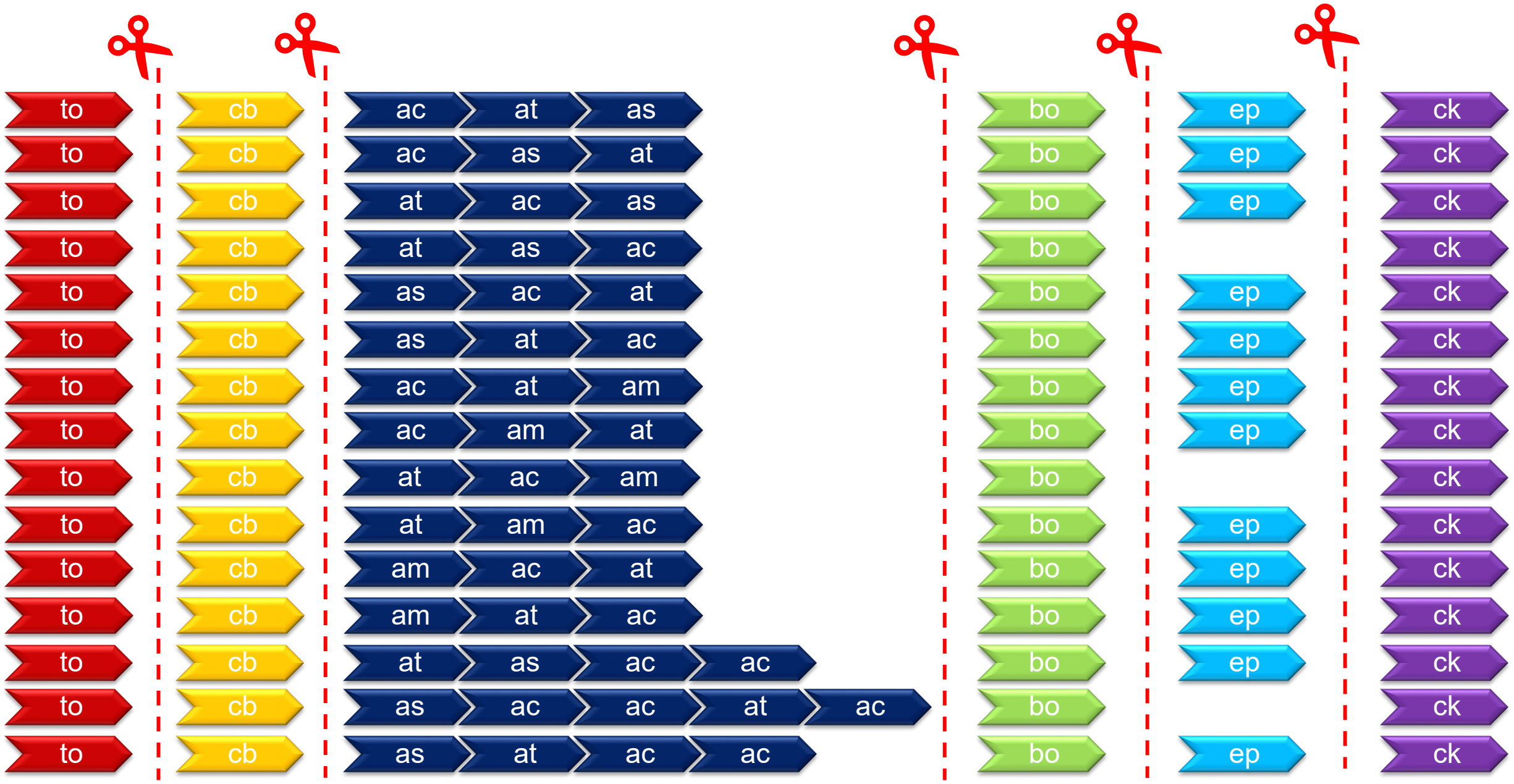
Color the events based on the partitioning

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

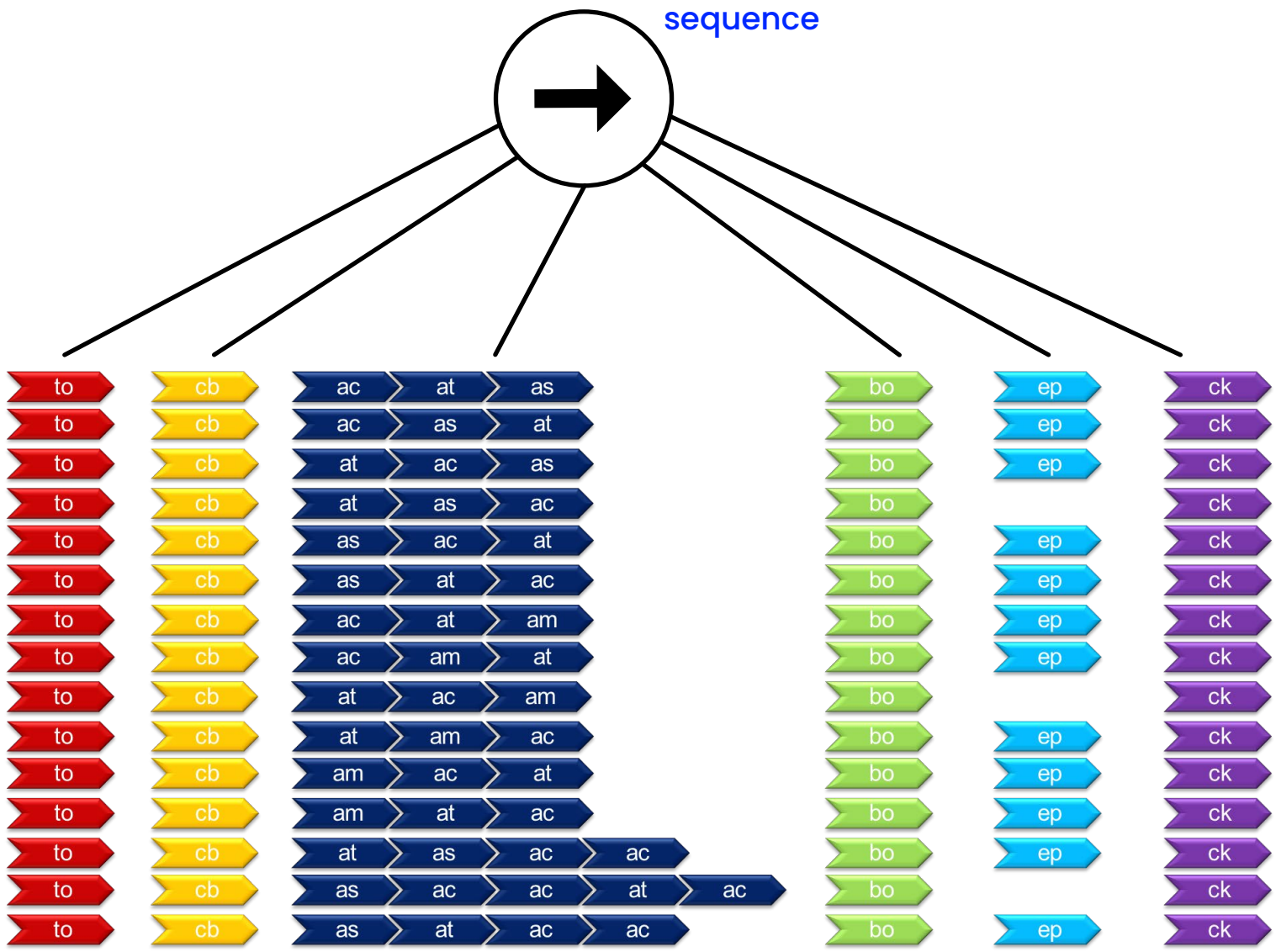


Split the events based on the coloring

Create six new event logs

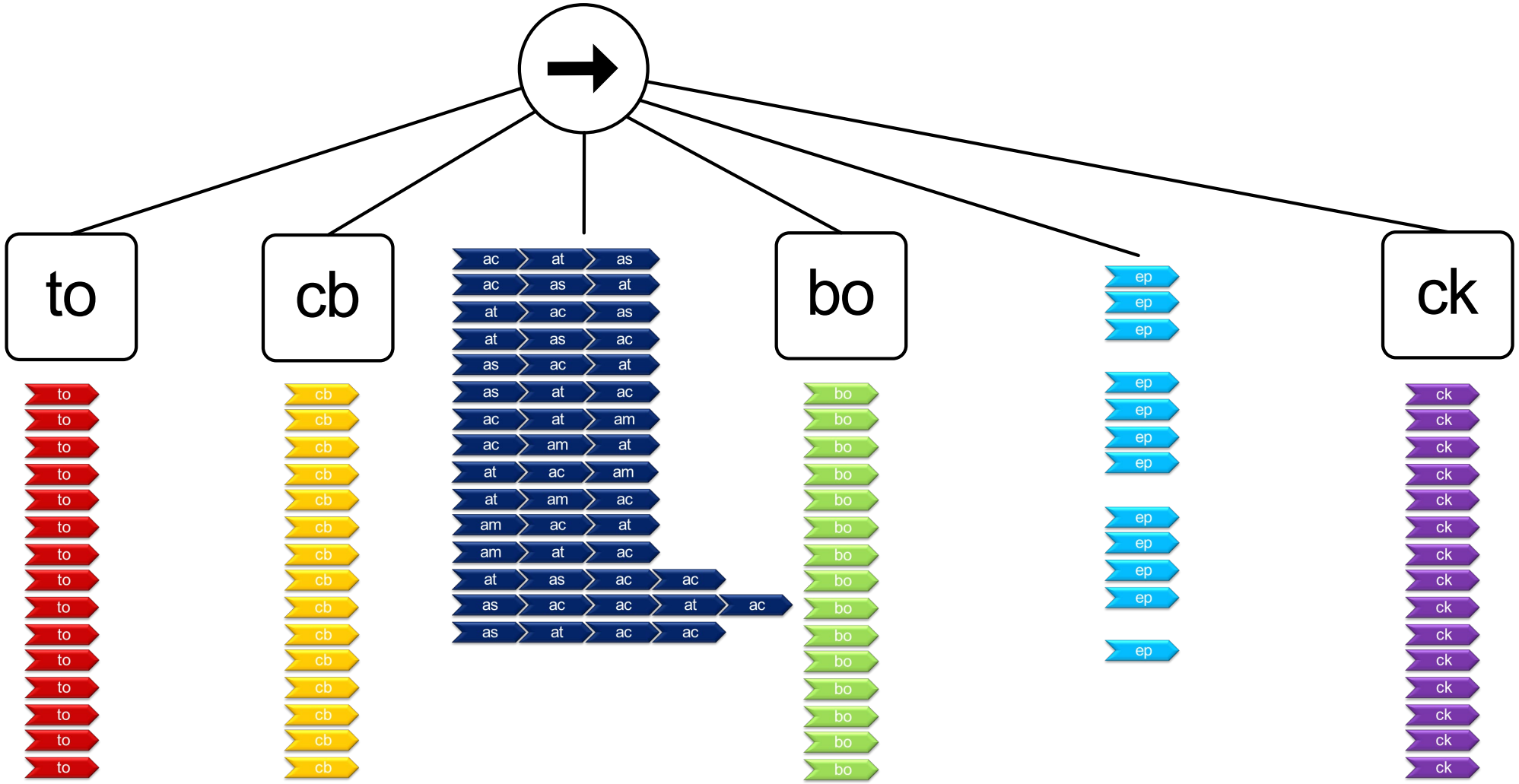


Connect the subprocesses using a process tree starting with a sequence node

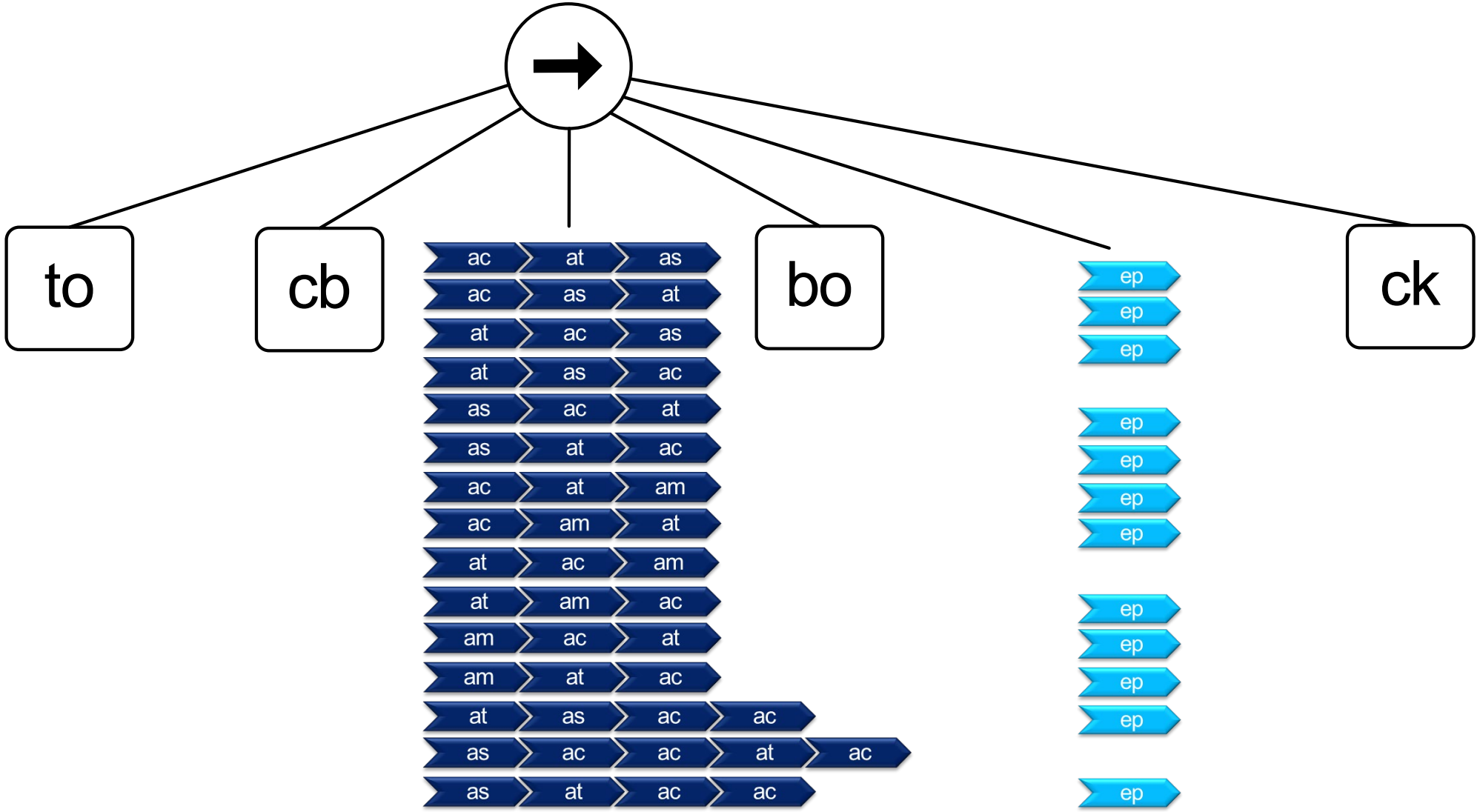


Five of the projected event logs refer to a single activity (base case).

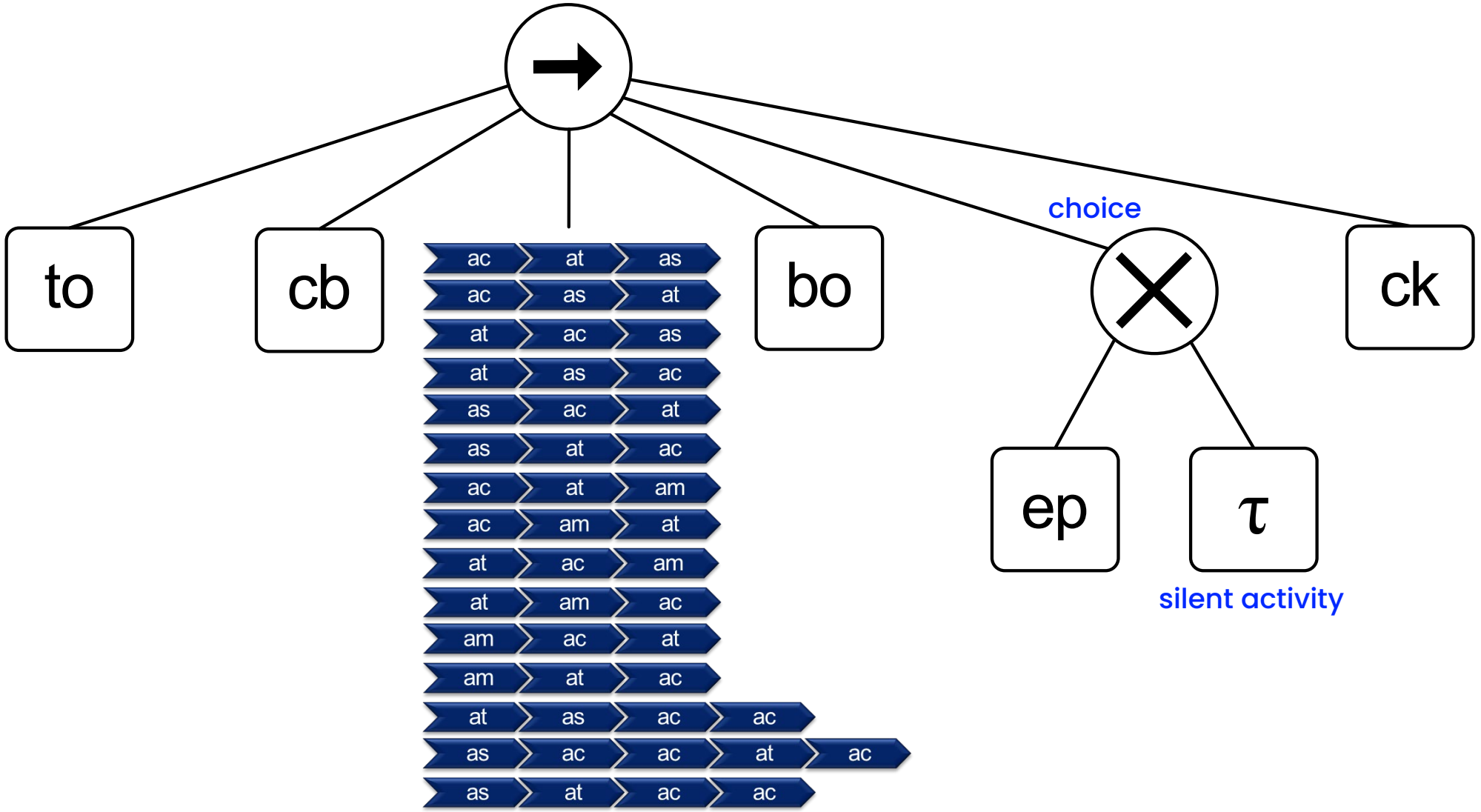
First, handle the event logs where one activity happens precisely once



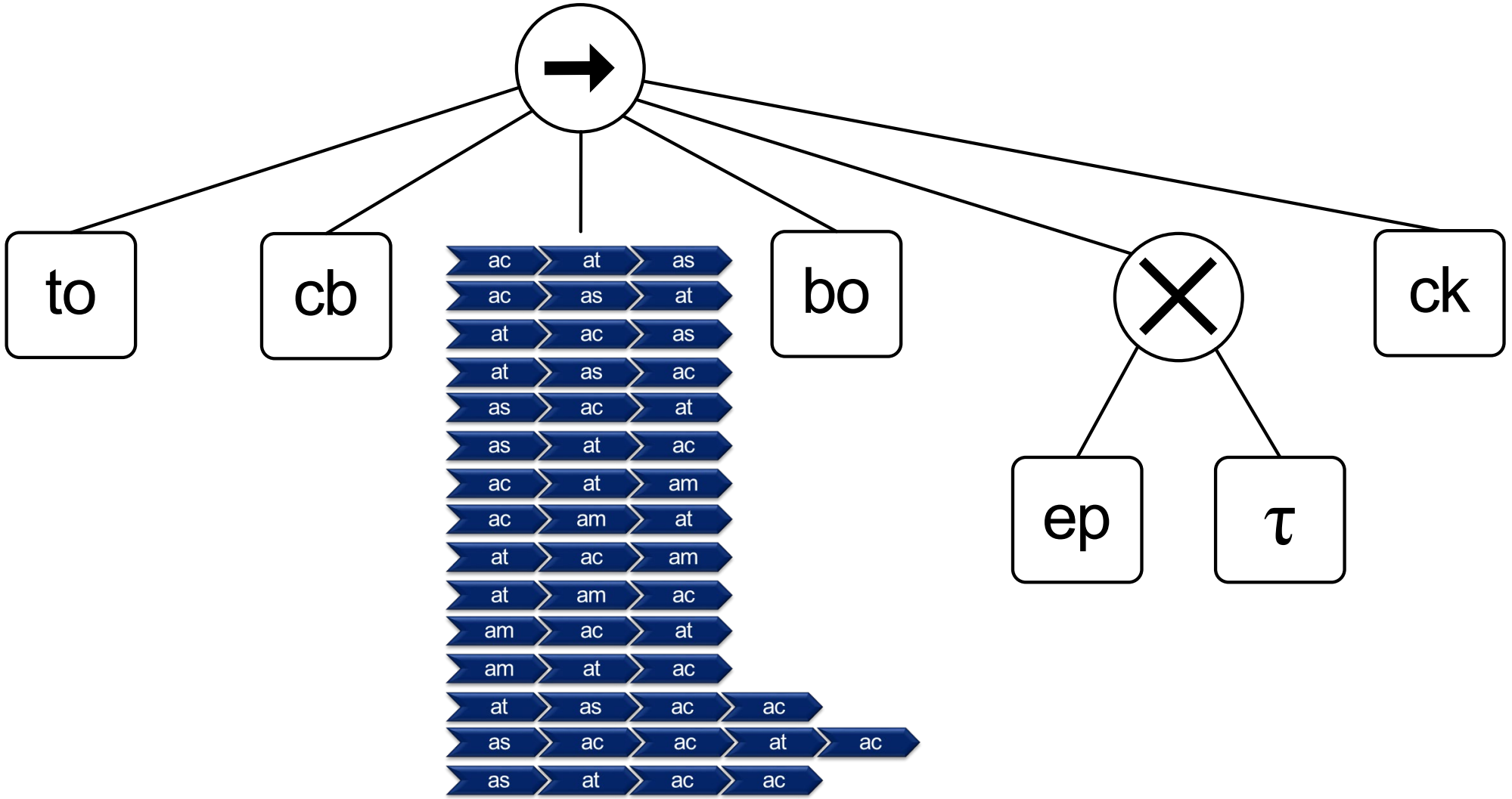
Two sub logs remain. Any log with just one unique activity can be handled easily.



Eat pizza (ep) can be skipped: Use choice node and “silent activity”

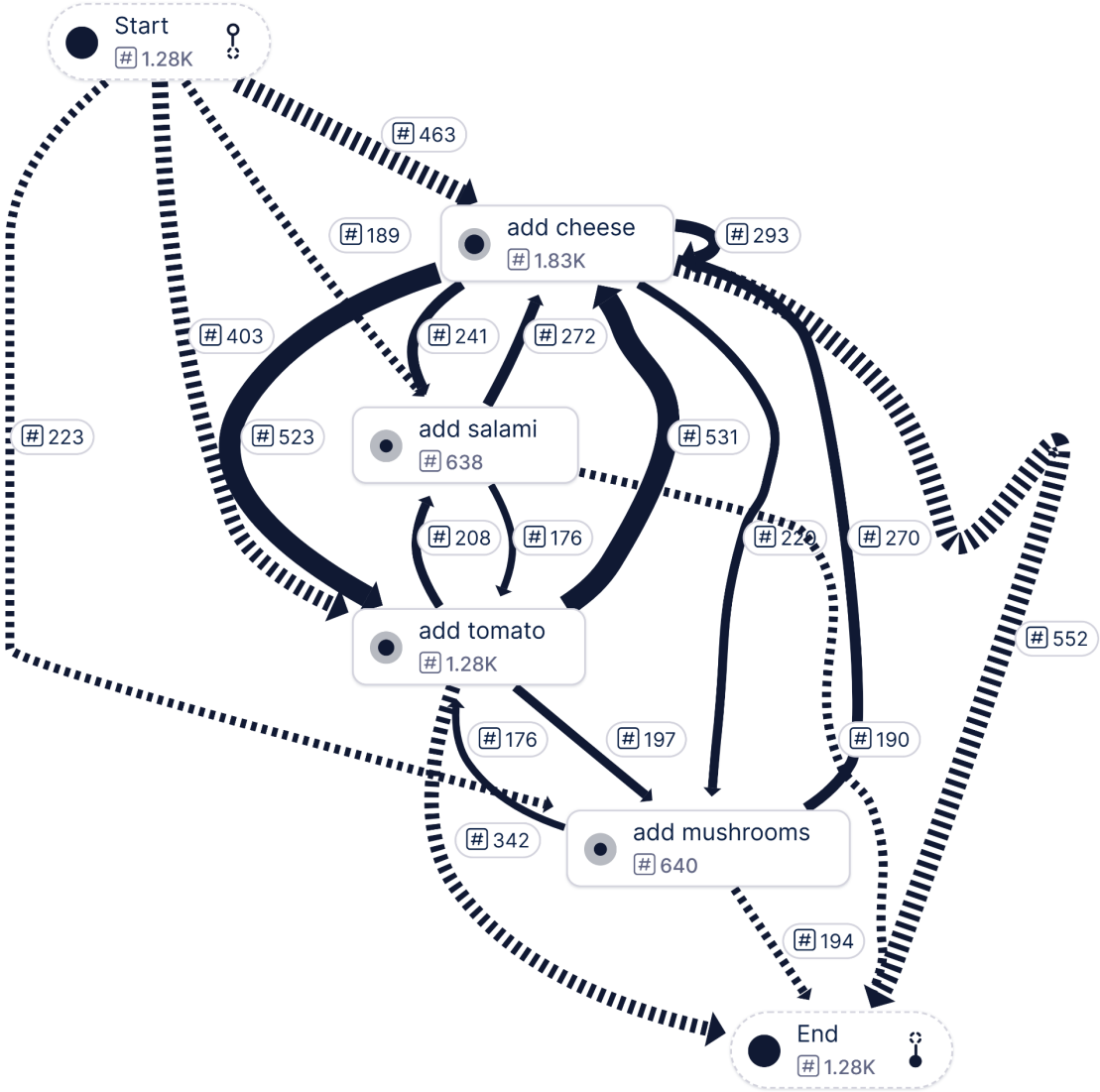


Only the blue event log remains



Continue with the blue event log: Repeat the process and create a DFG

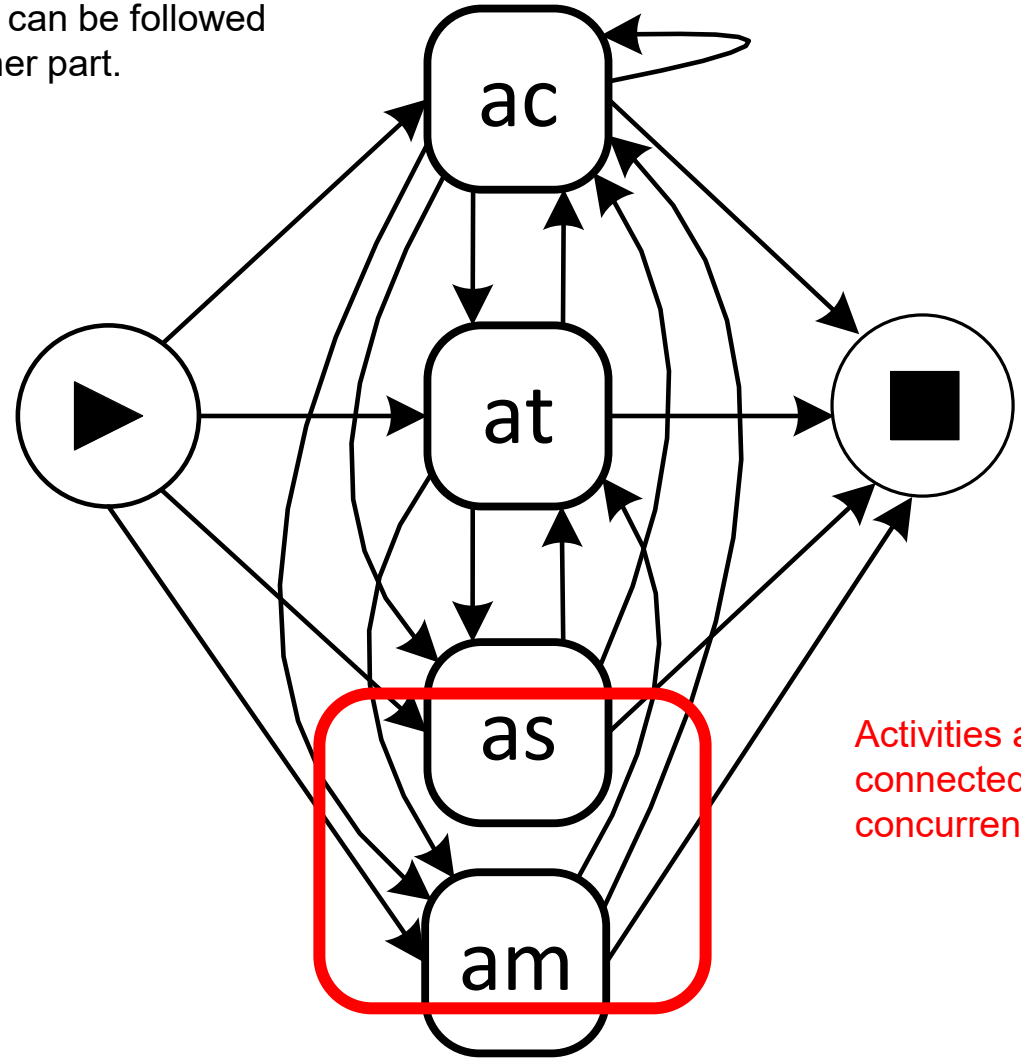
ac at as
ac as at
at ac as
at as ac
as ac at
as at ac
ac at am
ac am at
at ac am
at am ac
am ac at
am at ac
at as ac ac
as ac ac at ac
as at ac ac



Continue with the blue event log: Repeat the process

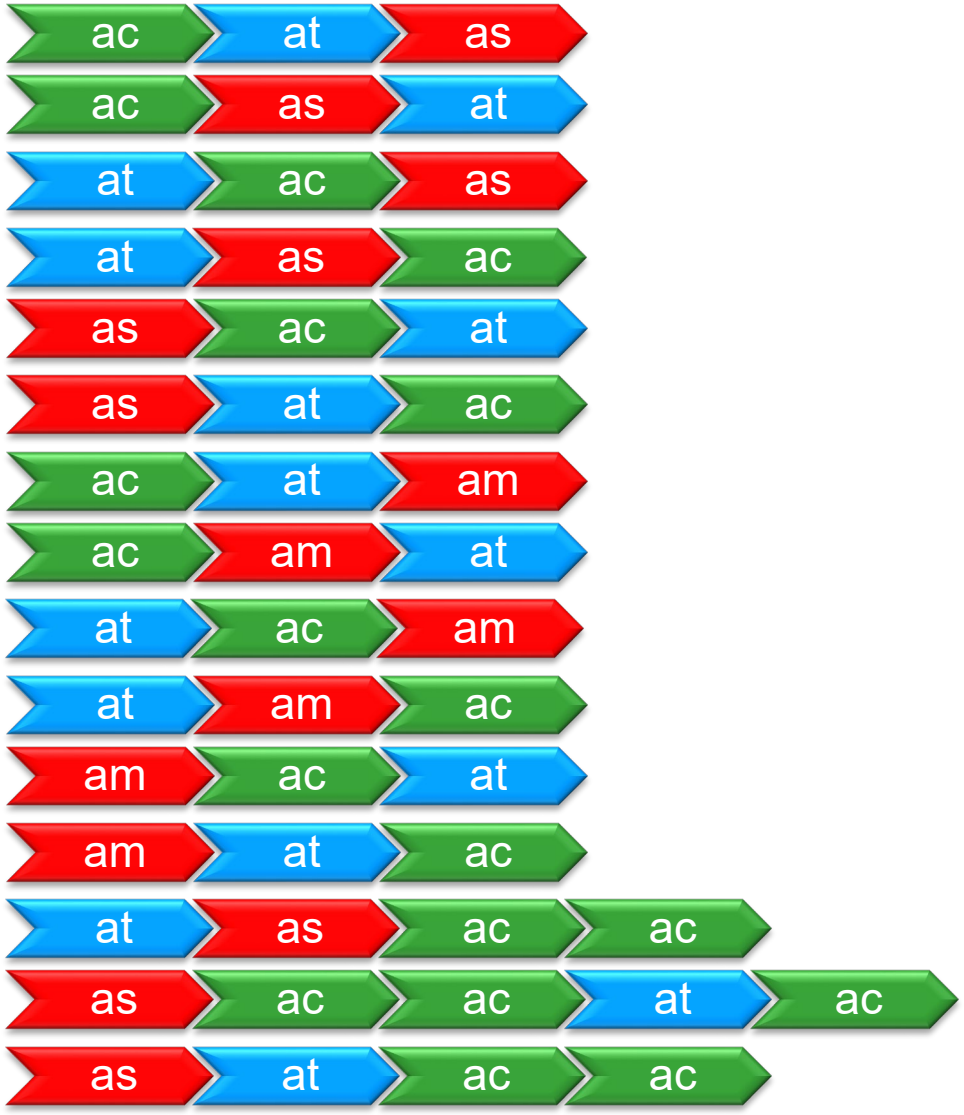
ac at as
ac as at
at ac as
at as ac
as ac at
as at ac
ac at am
ac am at
at ac am
at am ac
am ac at
am at ac
at as ac ac
as ac ac at ac
as at ac ac

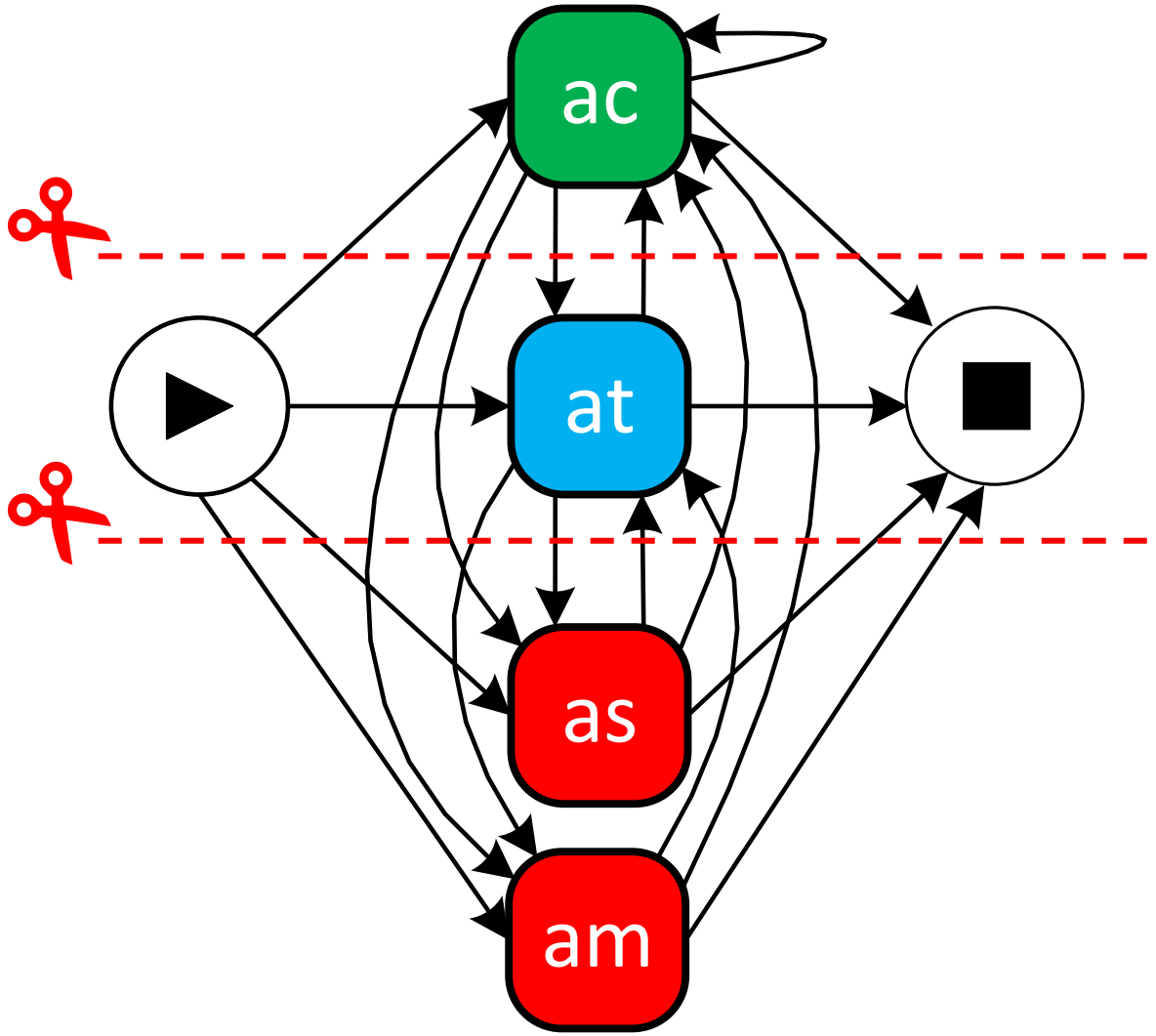
There is a parallel cut when the DFG can be split into concurrent parts where any activity in one part can be followed by any activity in another part.



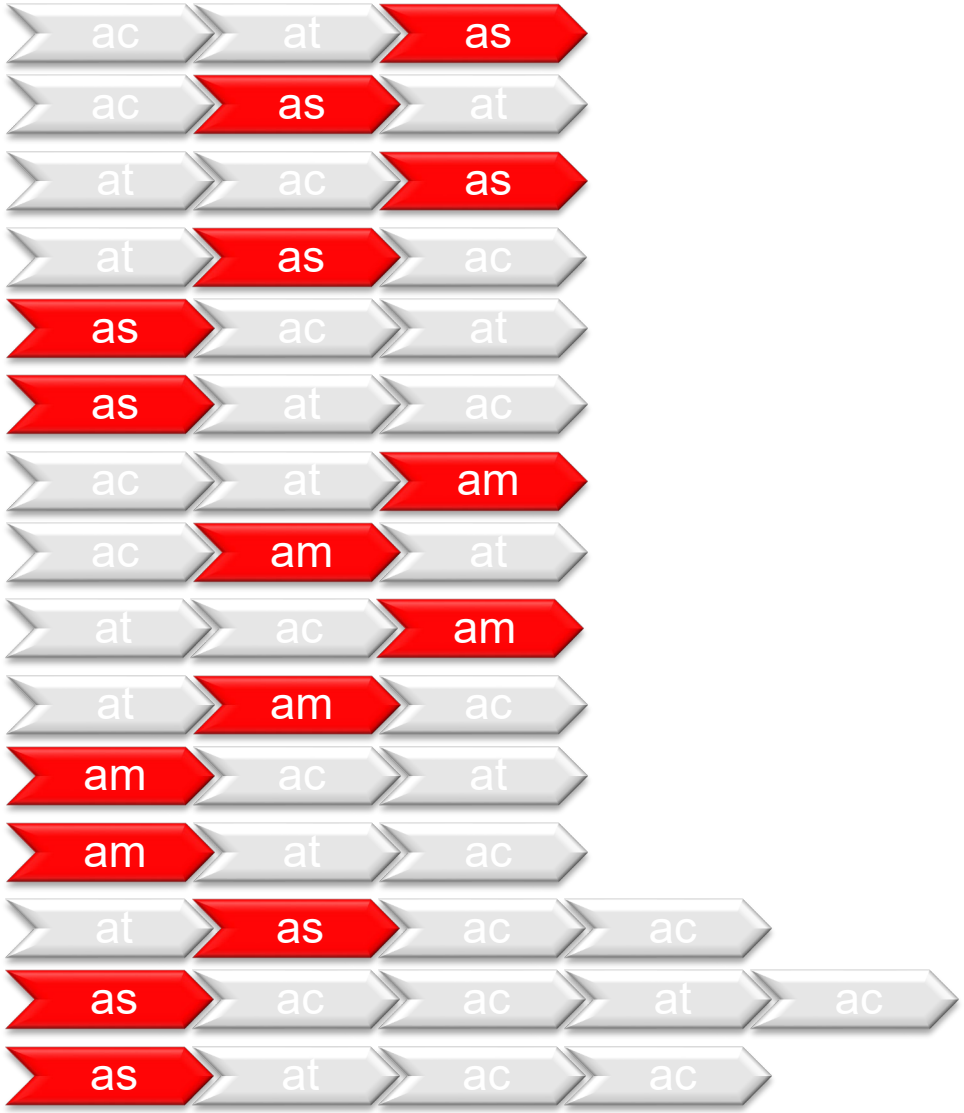
Activities as and am are not connected, i.e., not concurrent

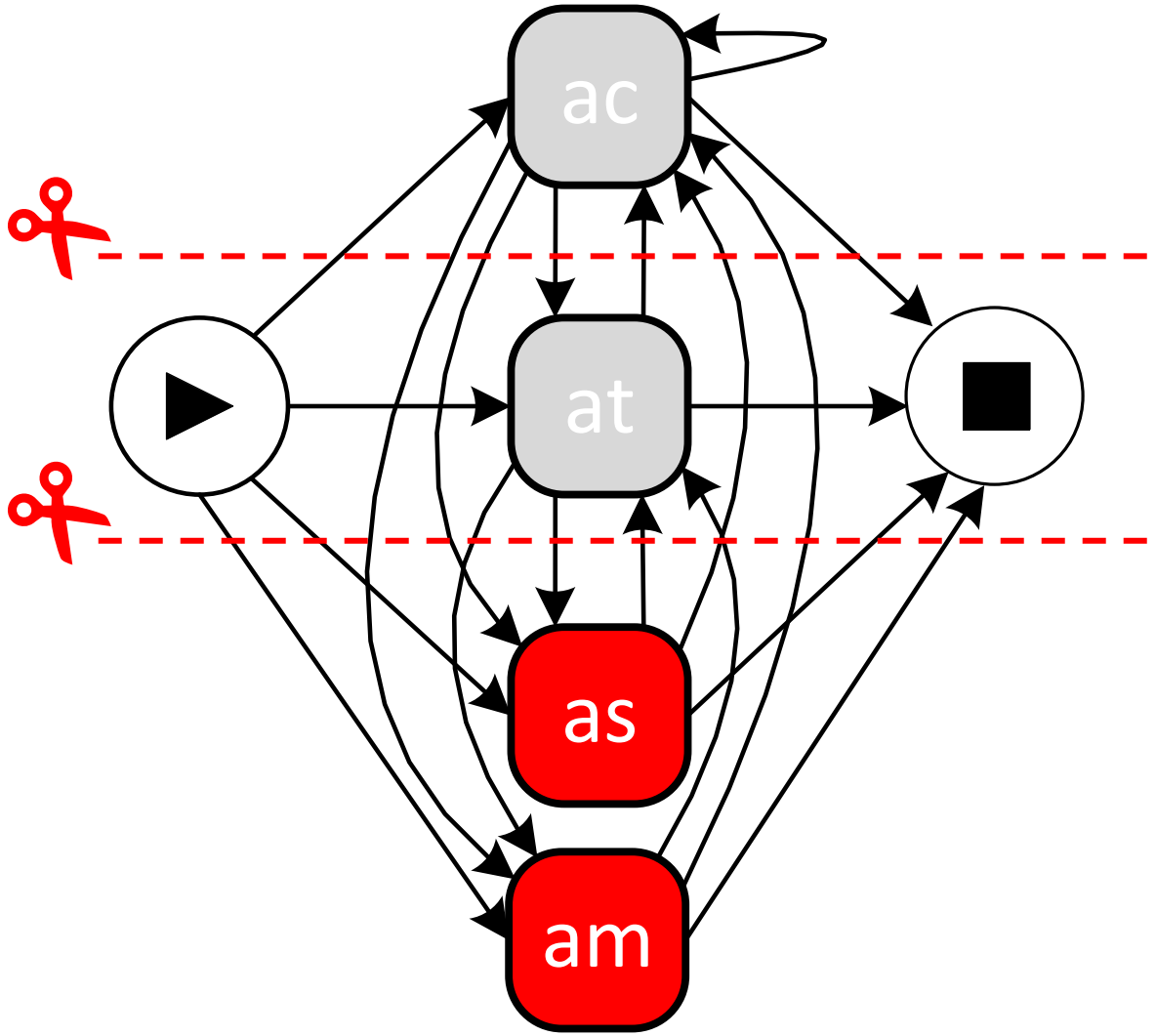
Apply a parallel cut resulting in three activity groups



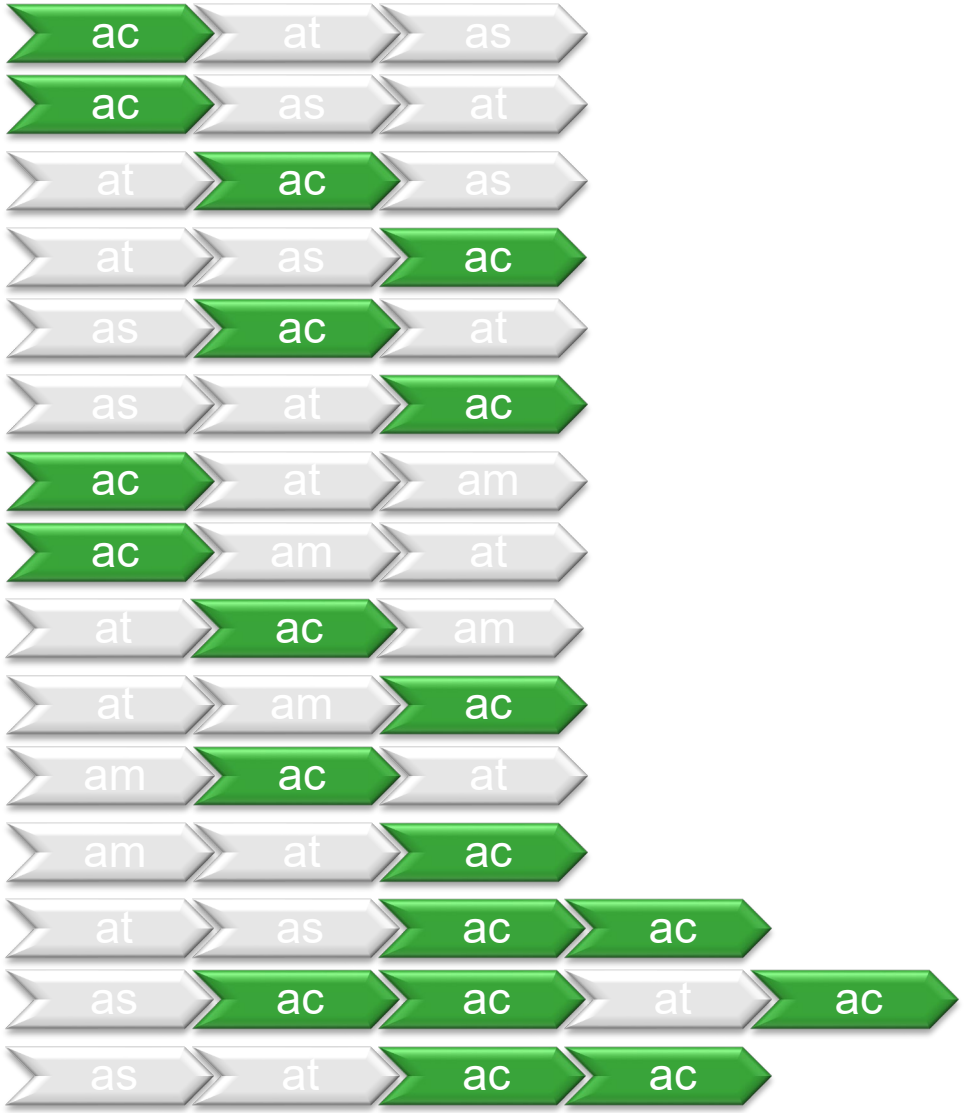


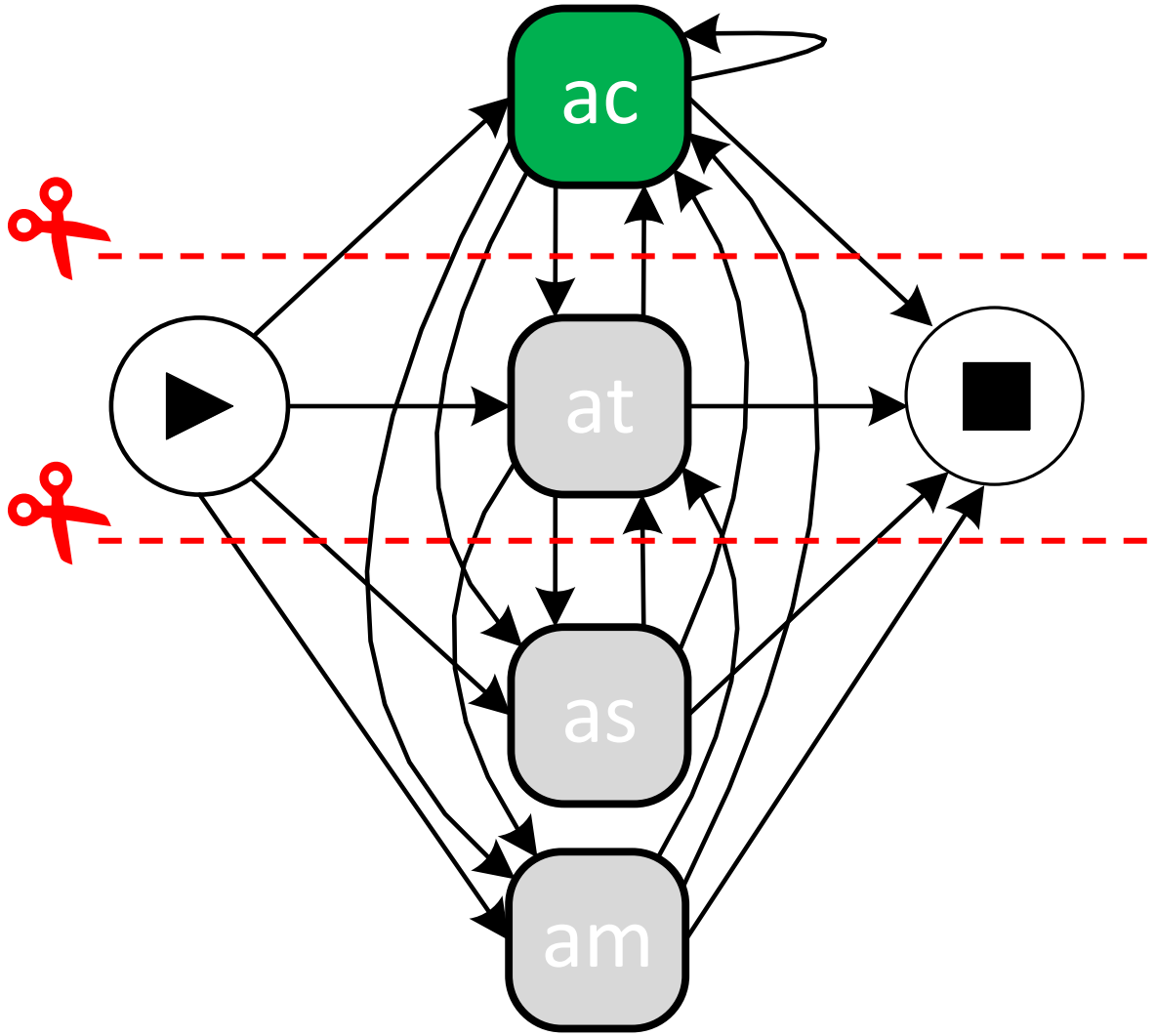
Create three new event logs (1/3)



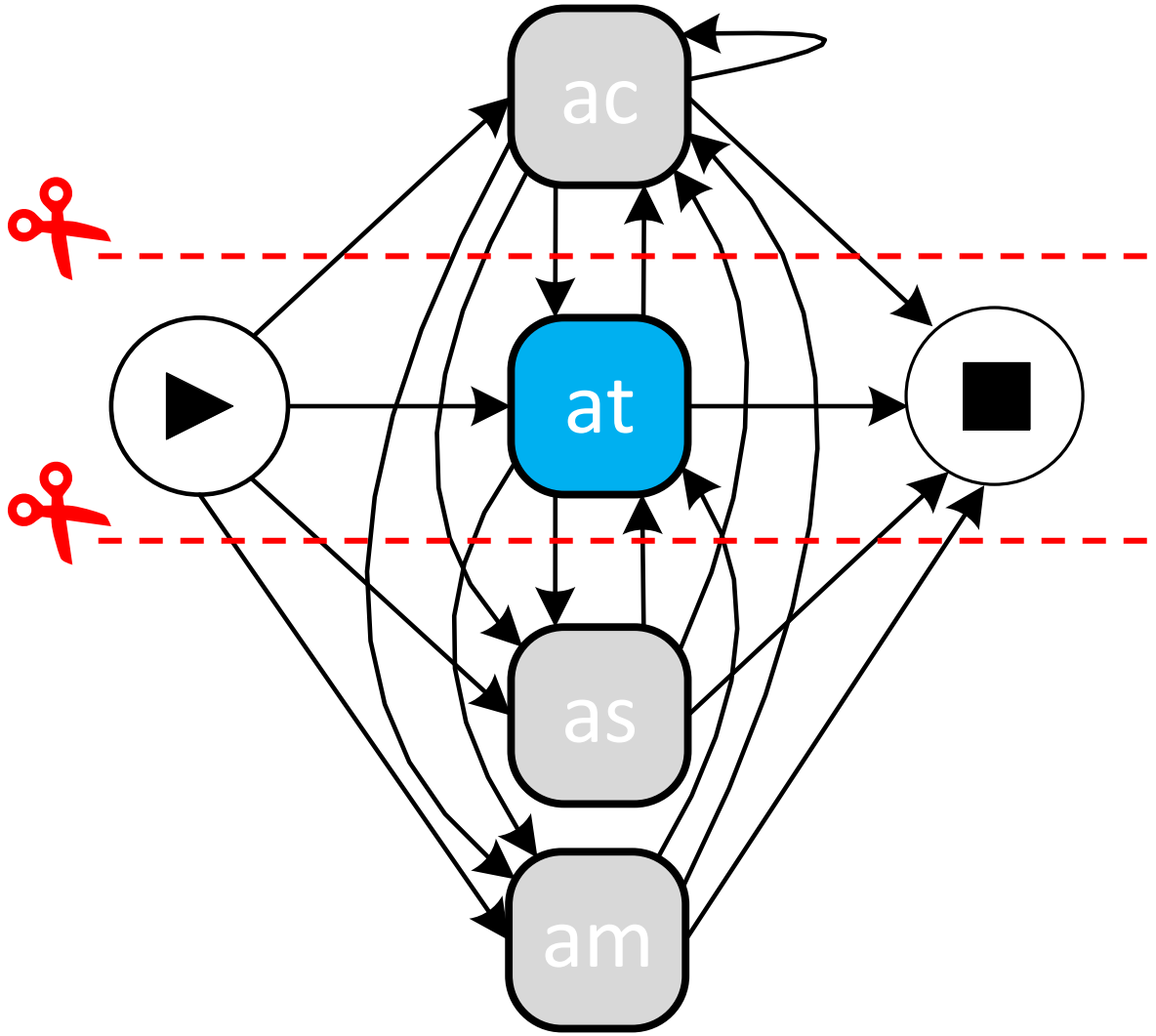


Create three new event logs (2/3)

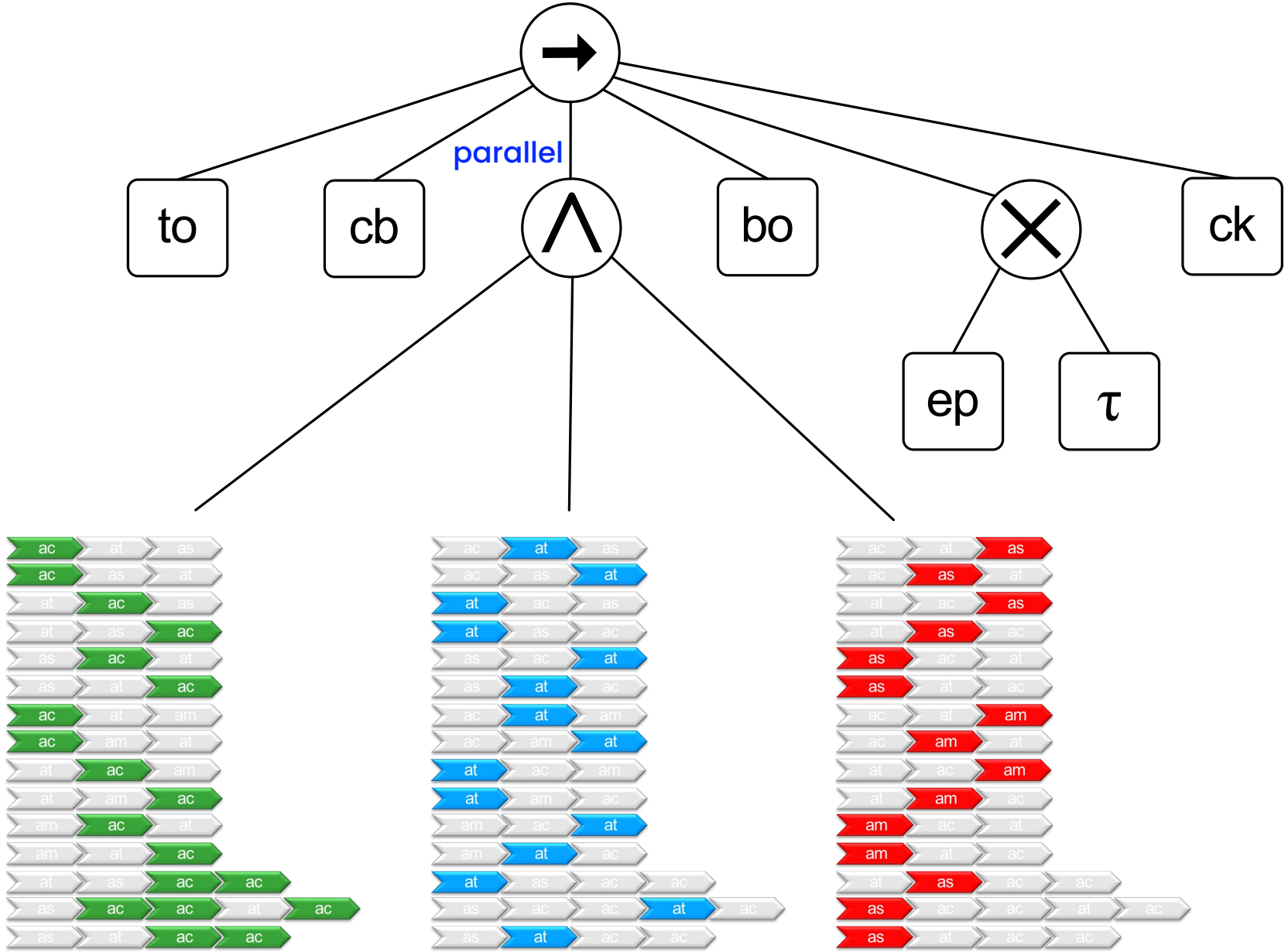




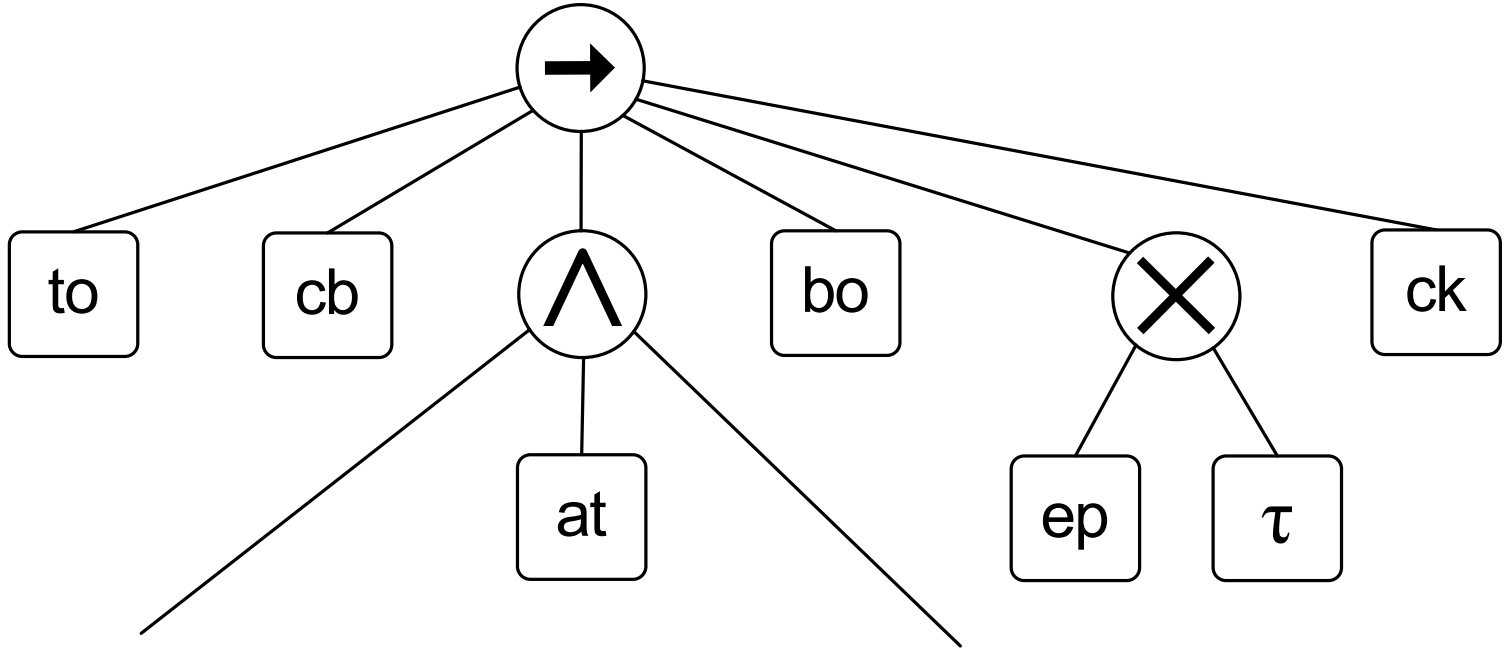
Create three new event logs (3/3)

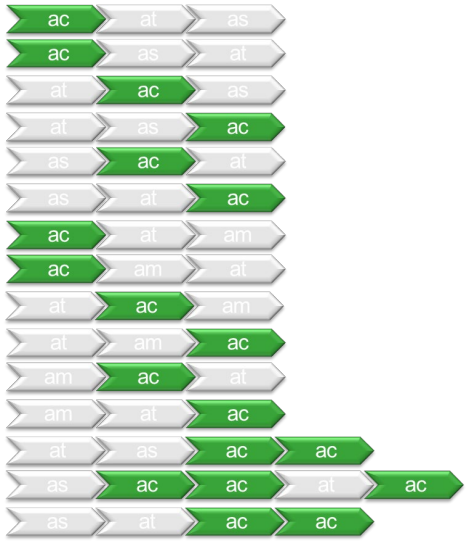


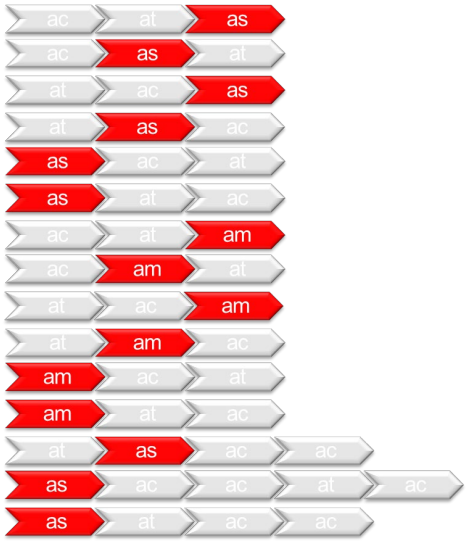
Add the parallel node and continue with the three event logs



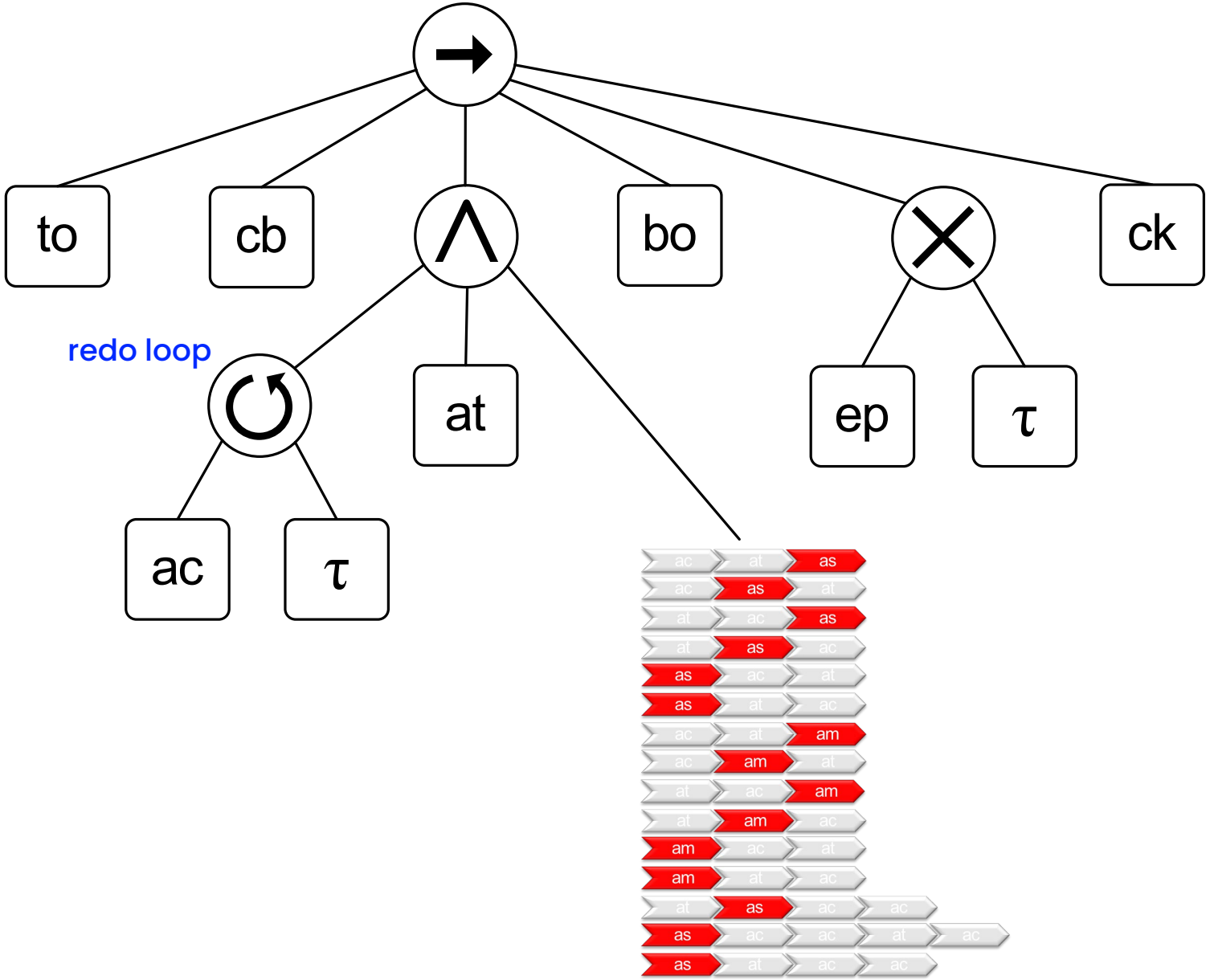
Add tomato (at) happens precisely once



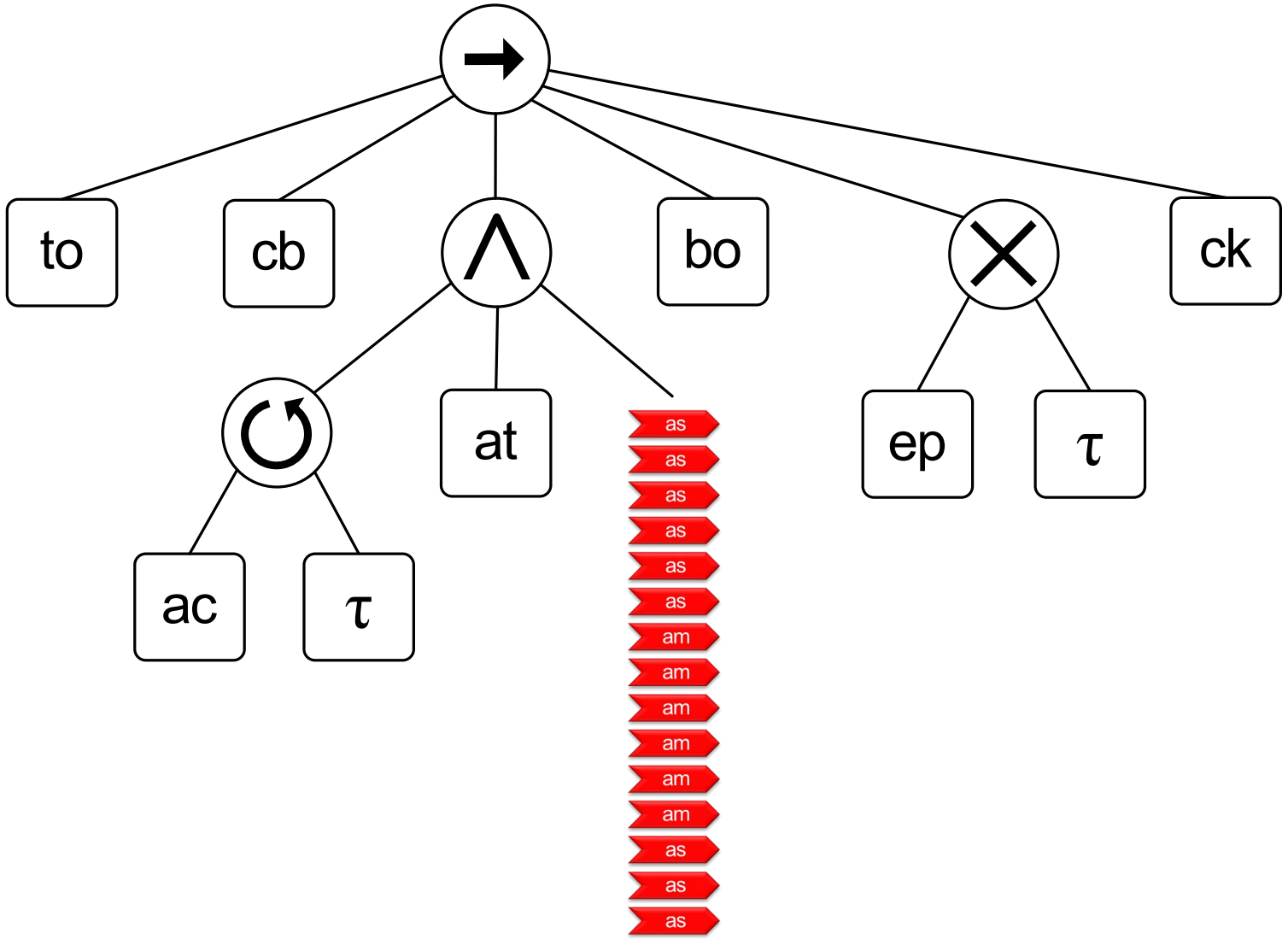




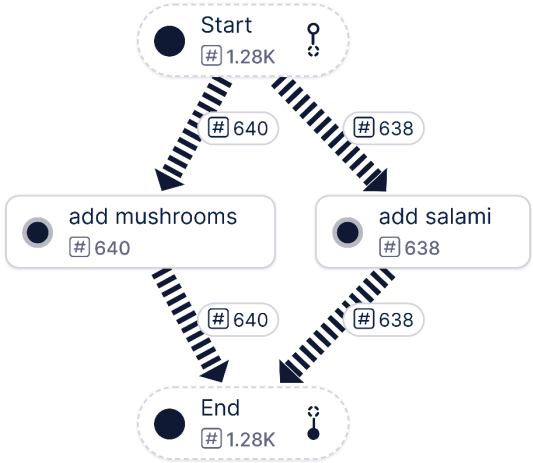
Add cheese (ac) happens at least one time

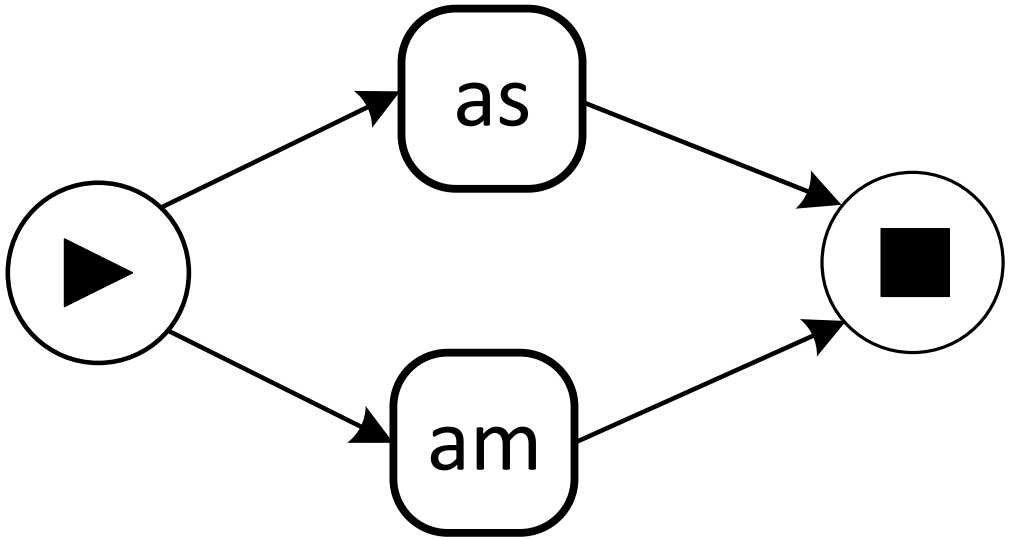


Only the red event log remains

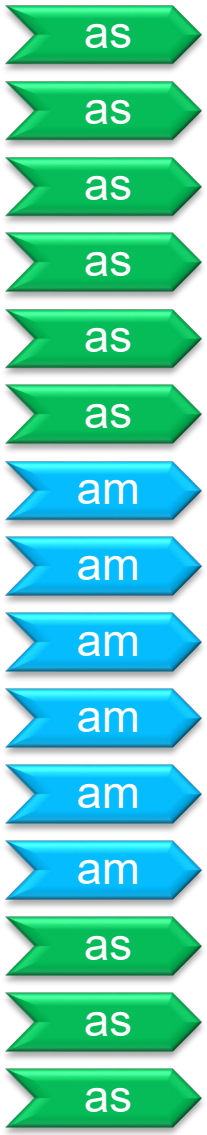


Repeat the process and create a DFG

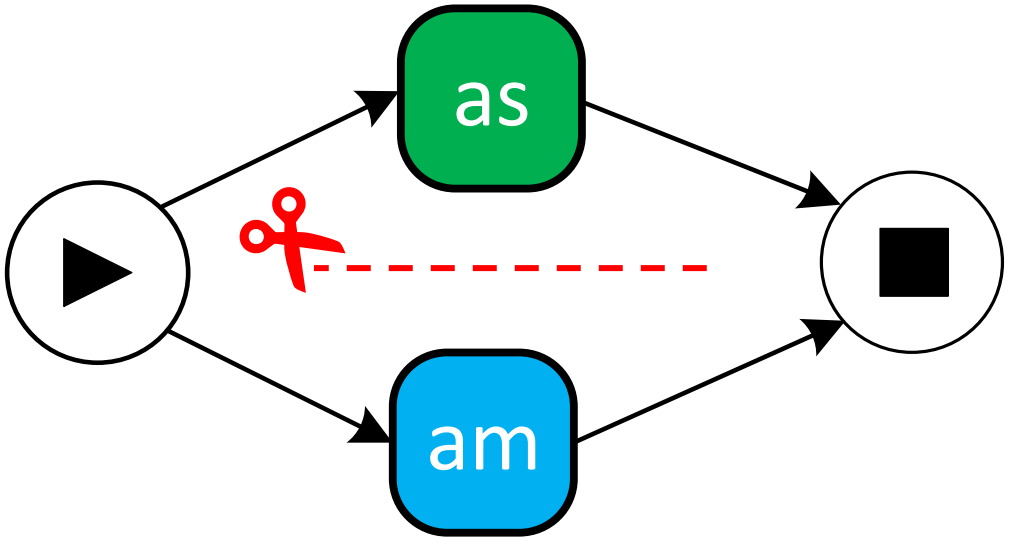




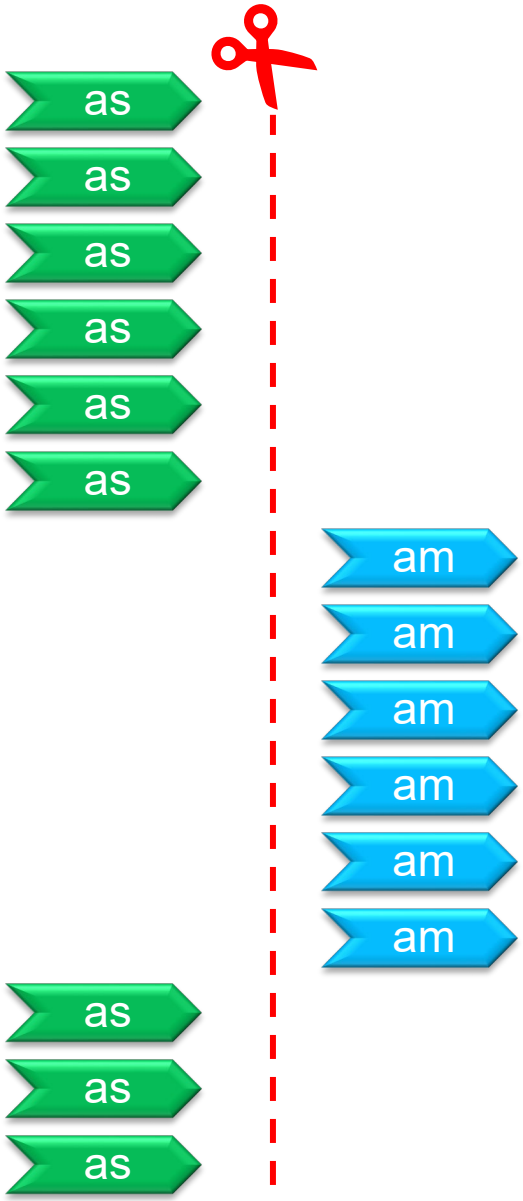
We find an exclusive-choice cut

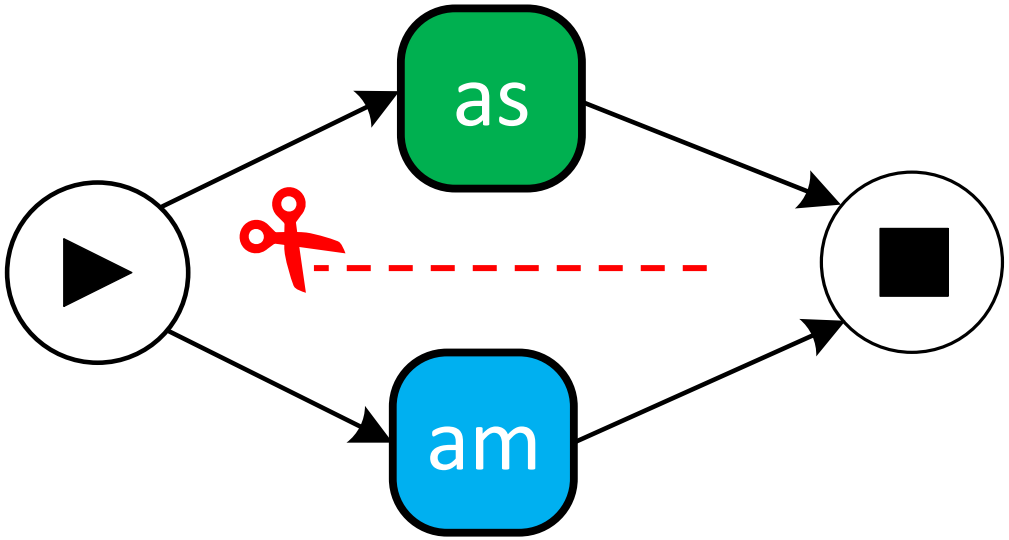


There is an exclusive-choice cut when the DFG can be split into disconnected parts after leaving out the artificial start and end.

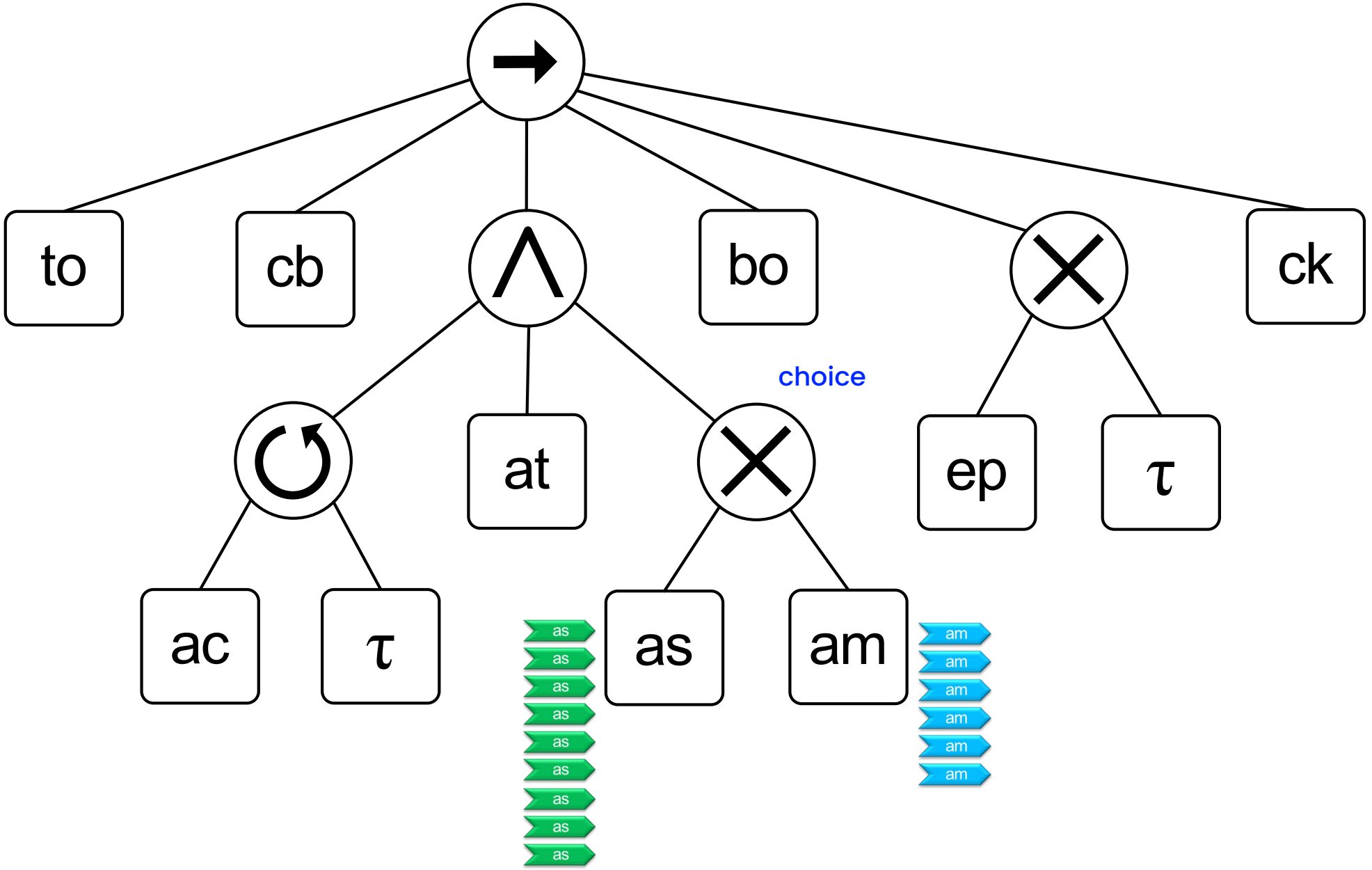


Create two new event logs

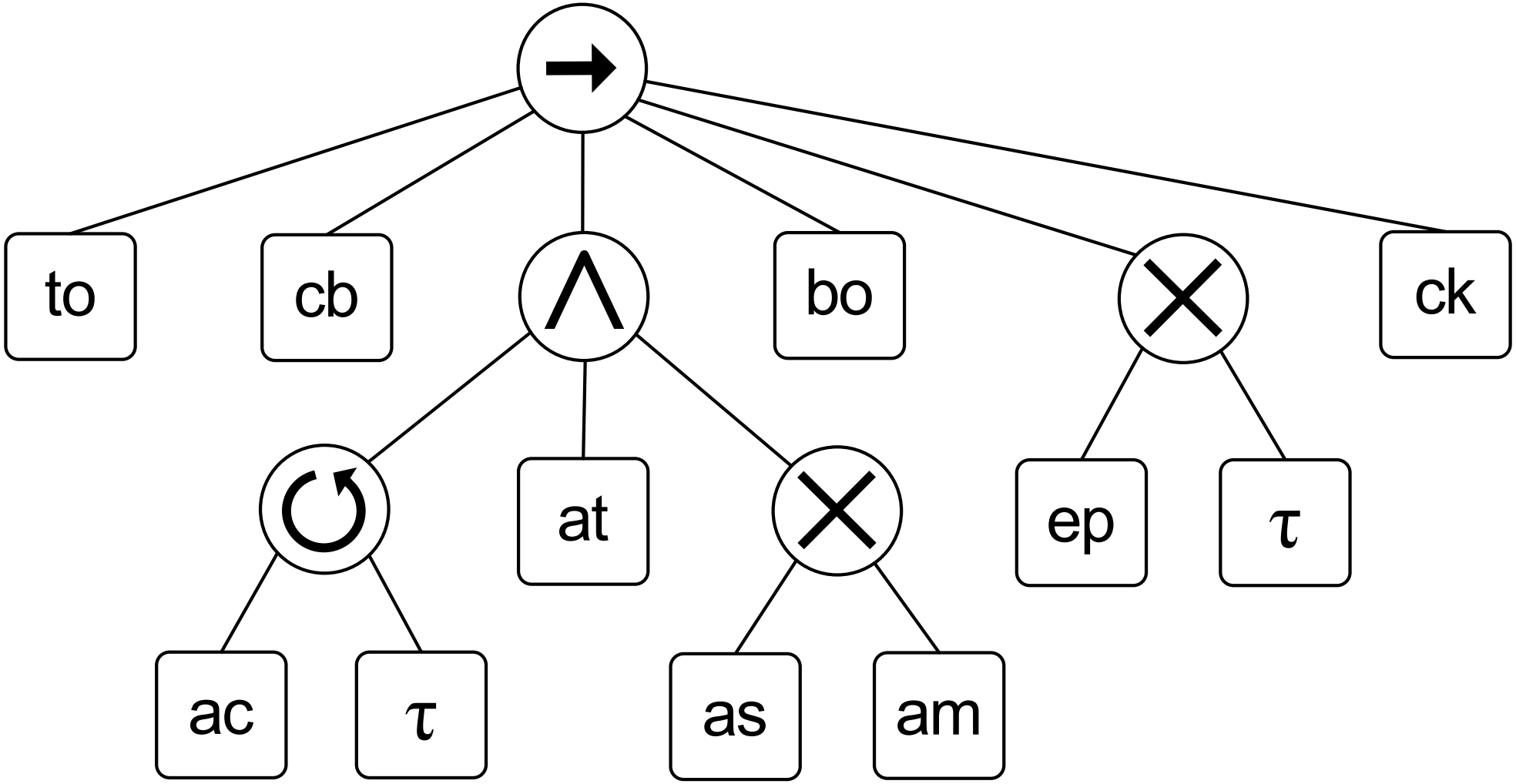




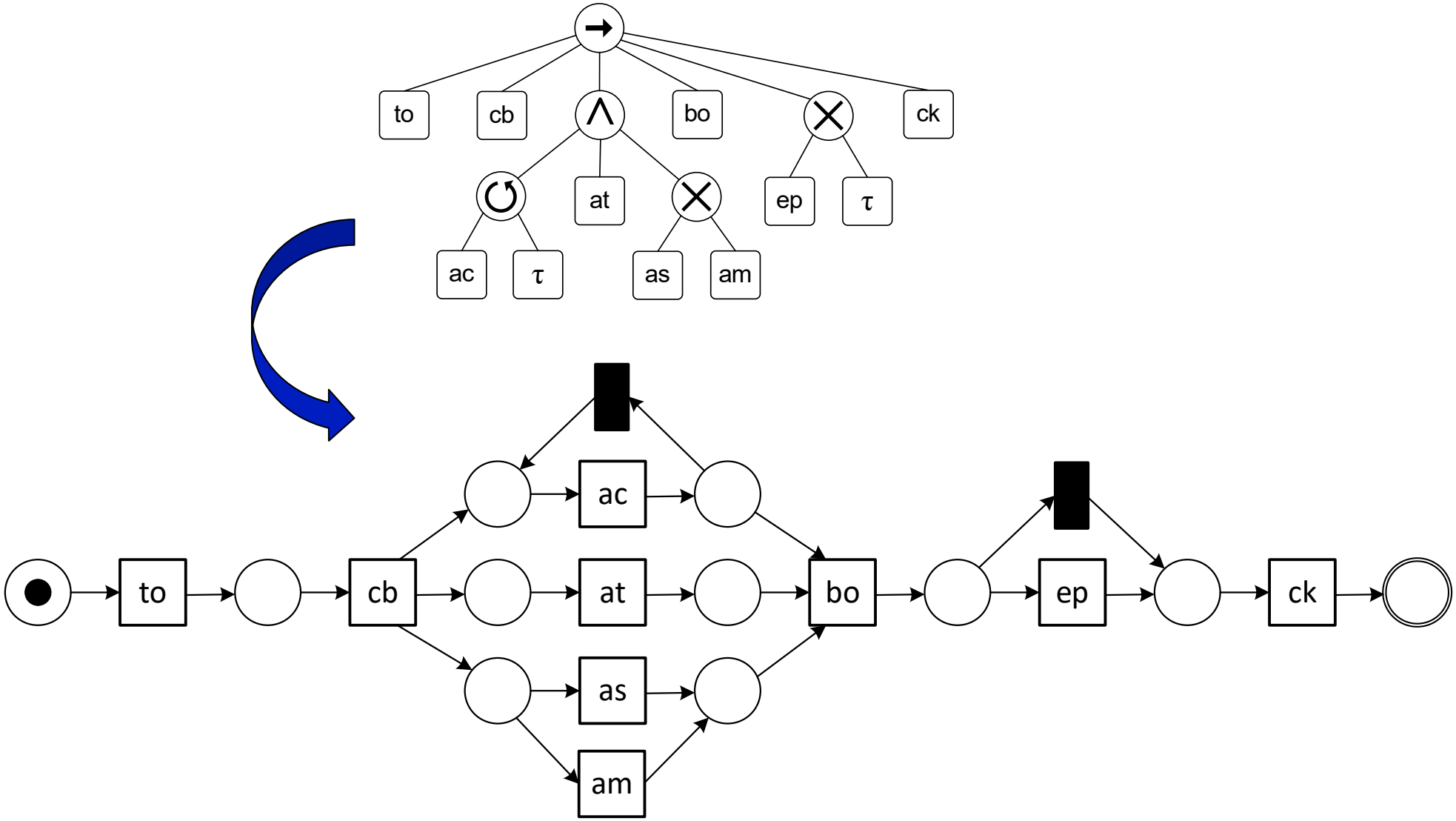
Add the choice node to choose between add salami and add mushroom



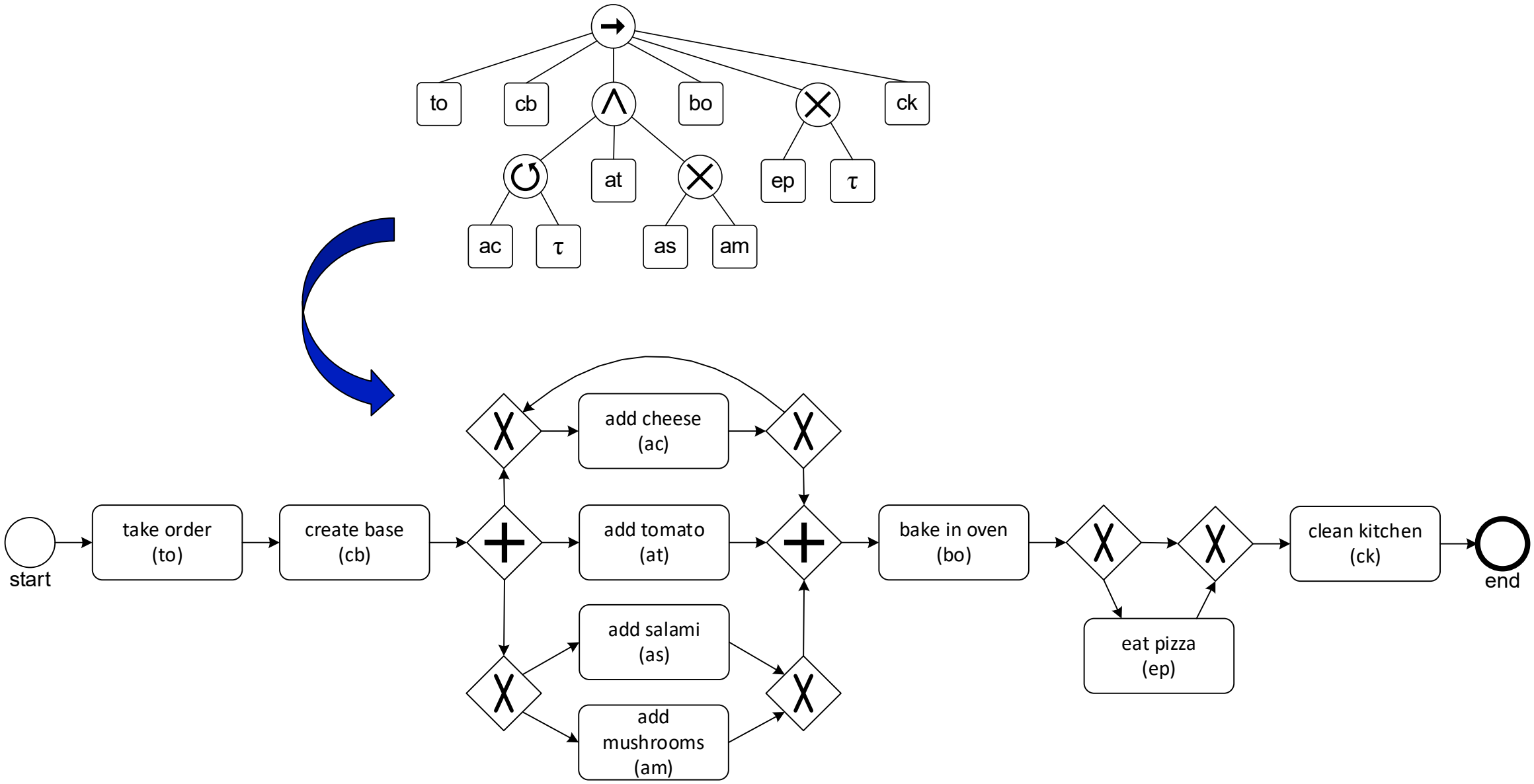
The process tree returned by the Inductive Mining algorithm



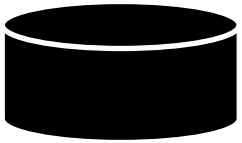
Can be visualized as a Petri net

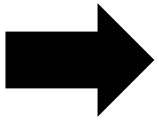


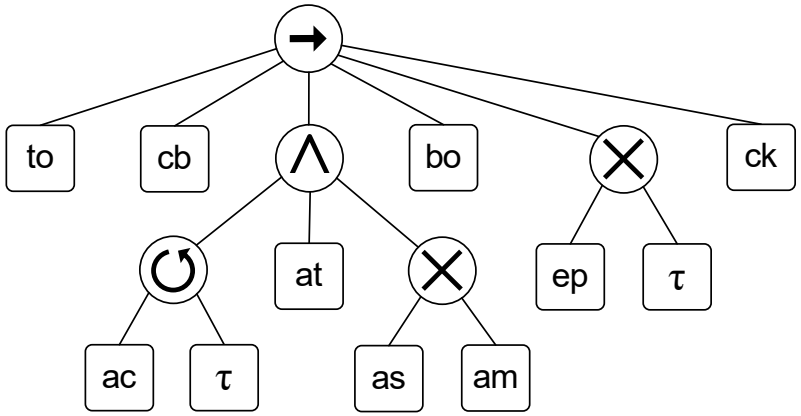
Can be visualized as a BPMN model

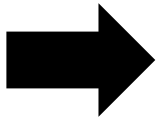


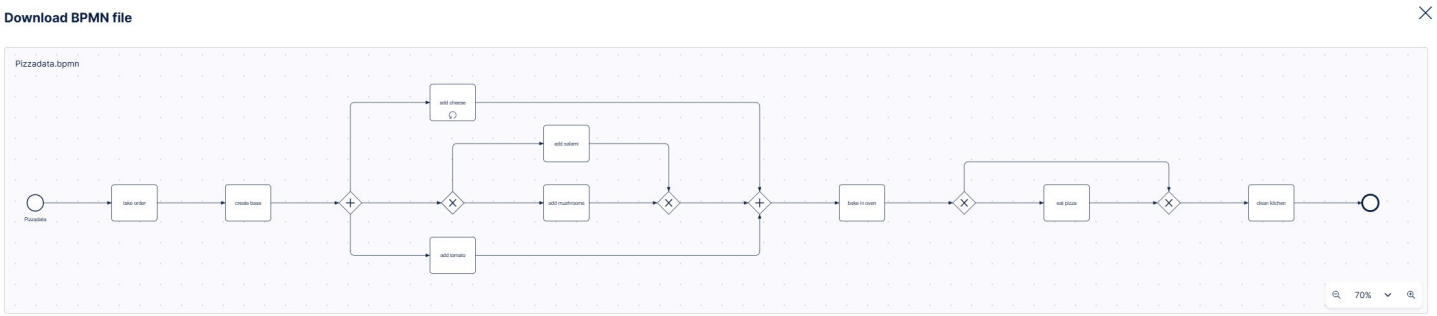
Supported by the Celonis Process Adherence Manager

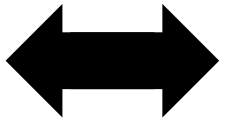


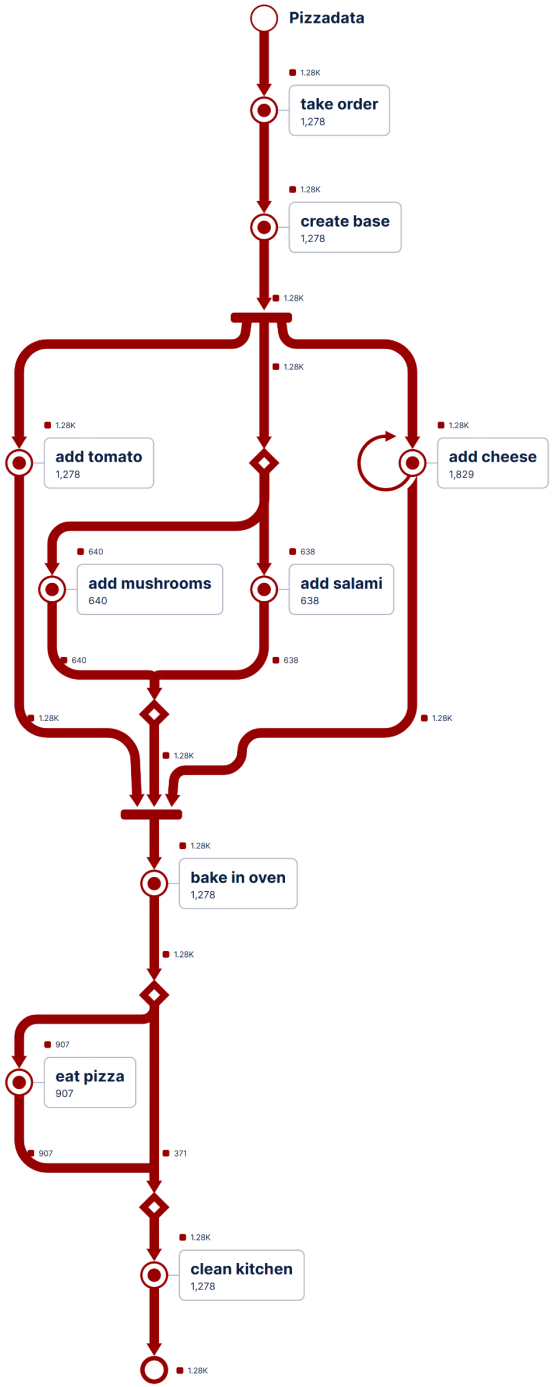




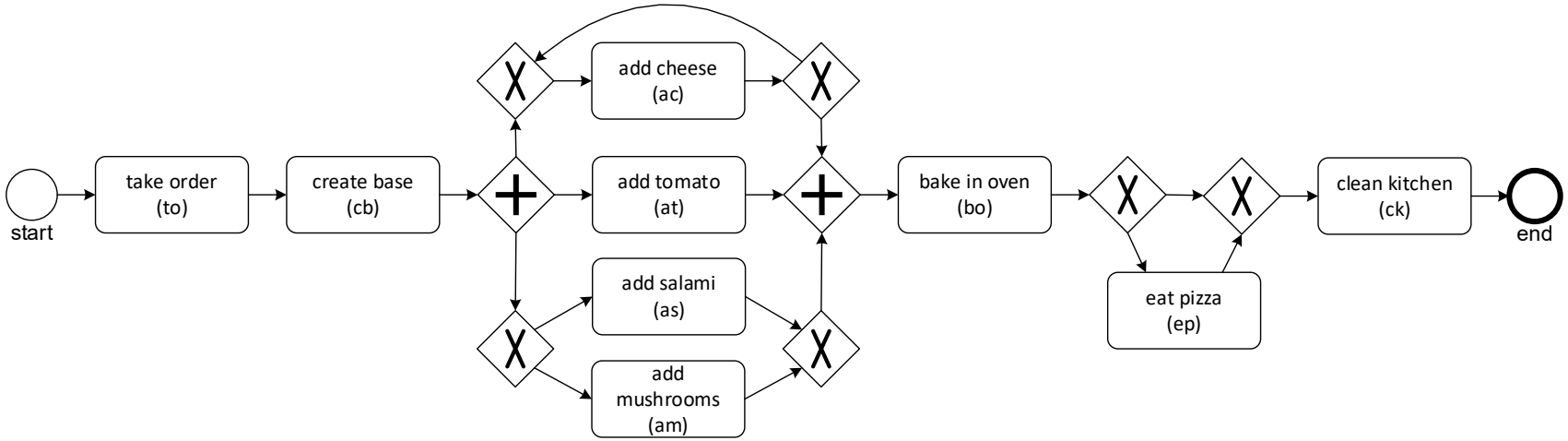


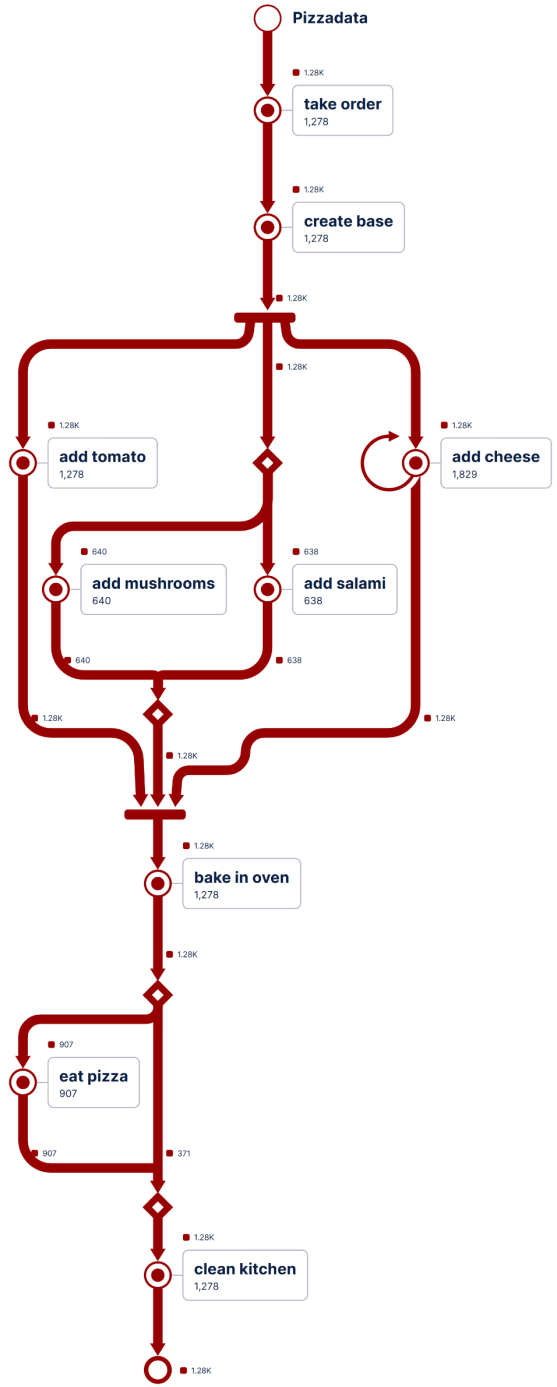






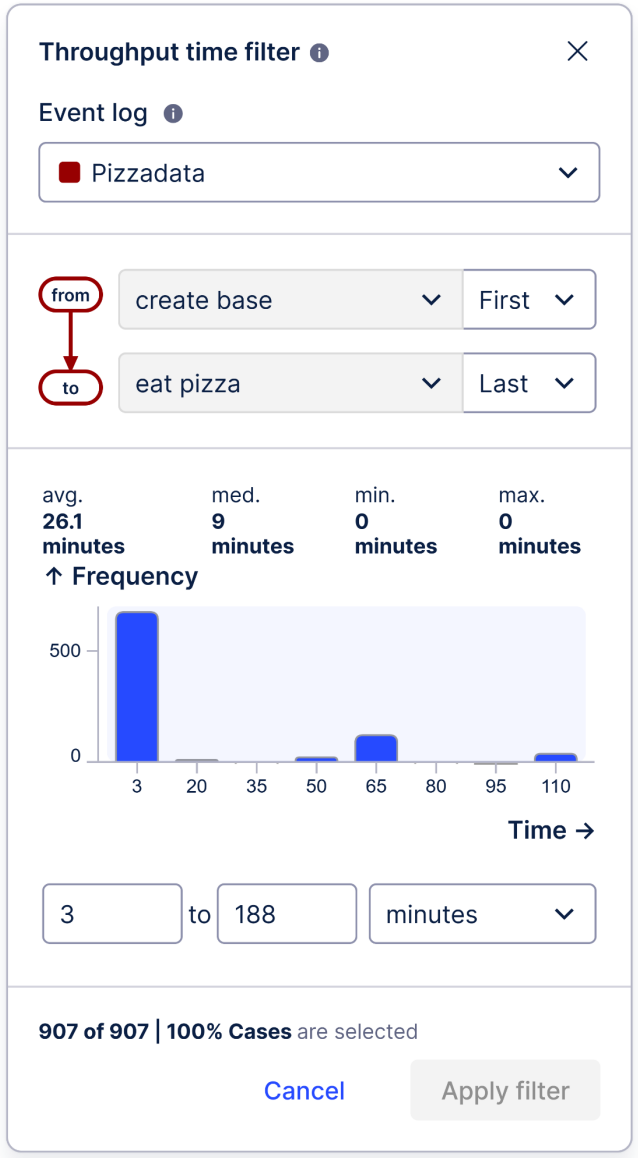
Comes with strong guarantees

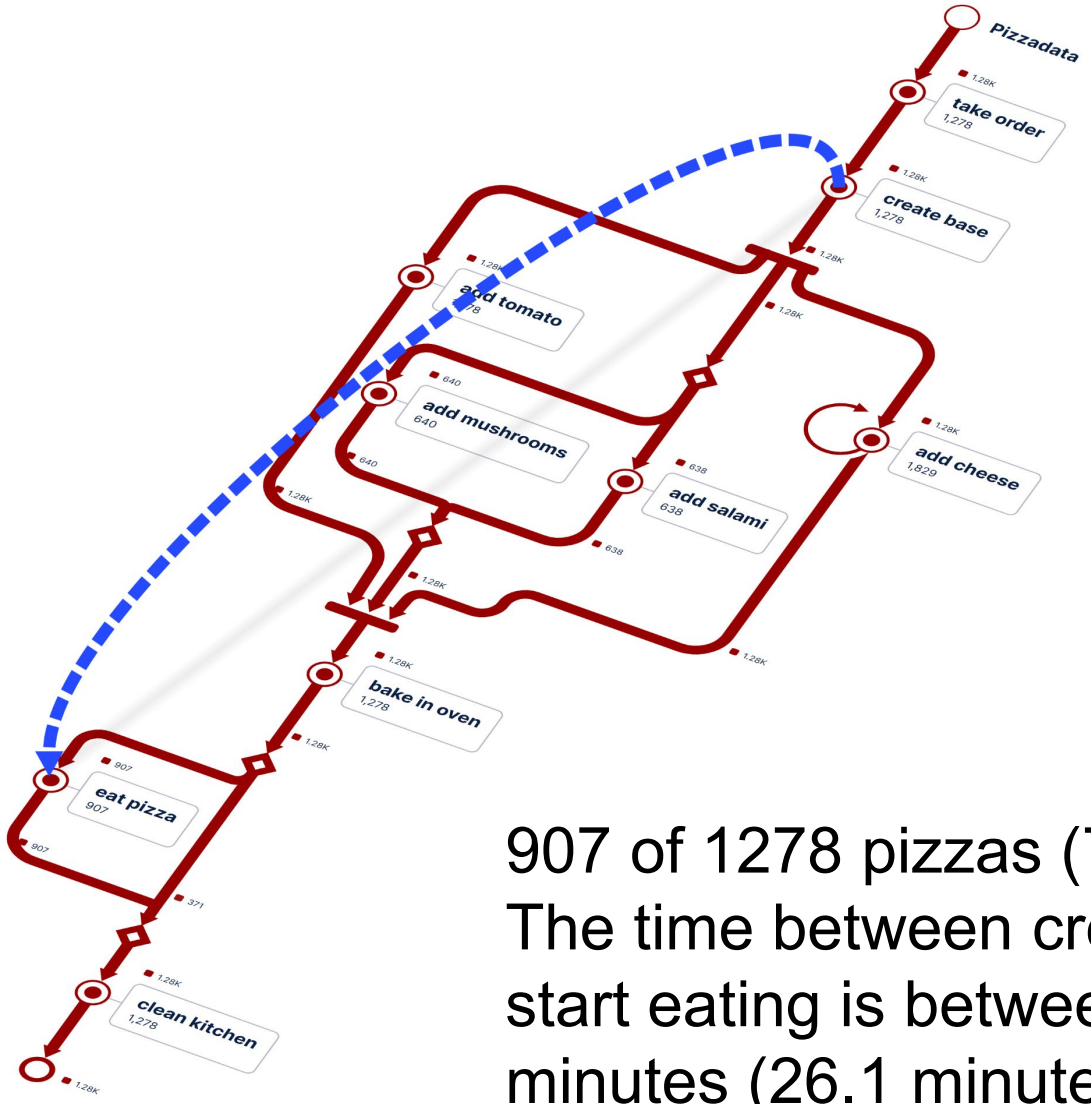




The (BPMN) model is sound and can replay the whole event log. Also, the model is much more precise than the DFG.

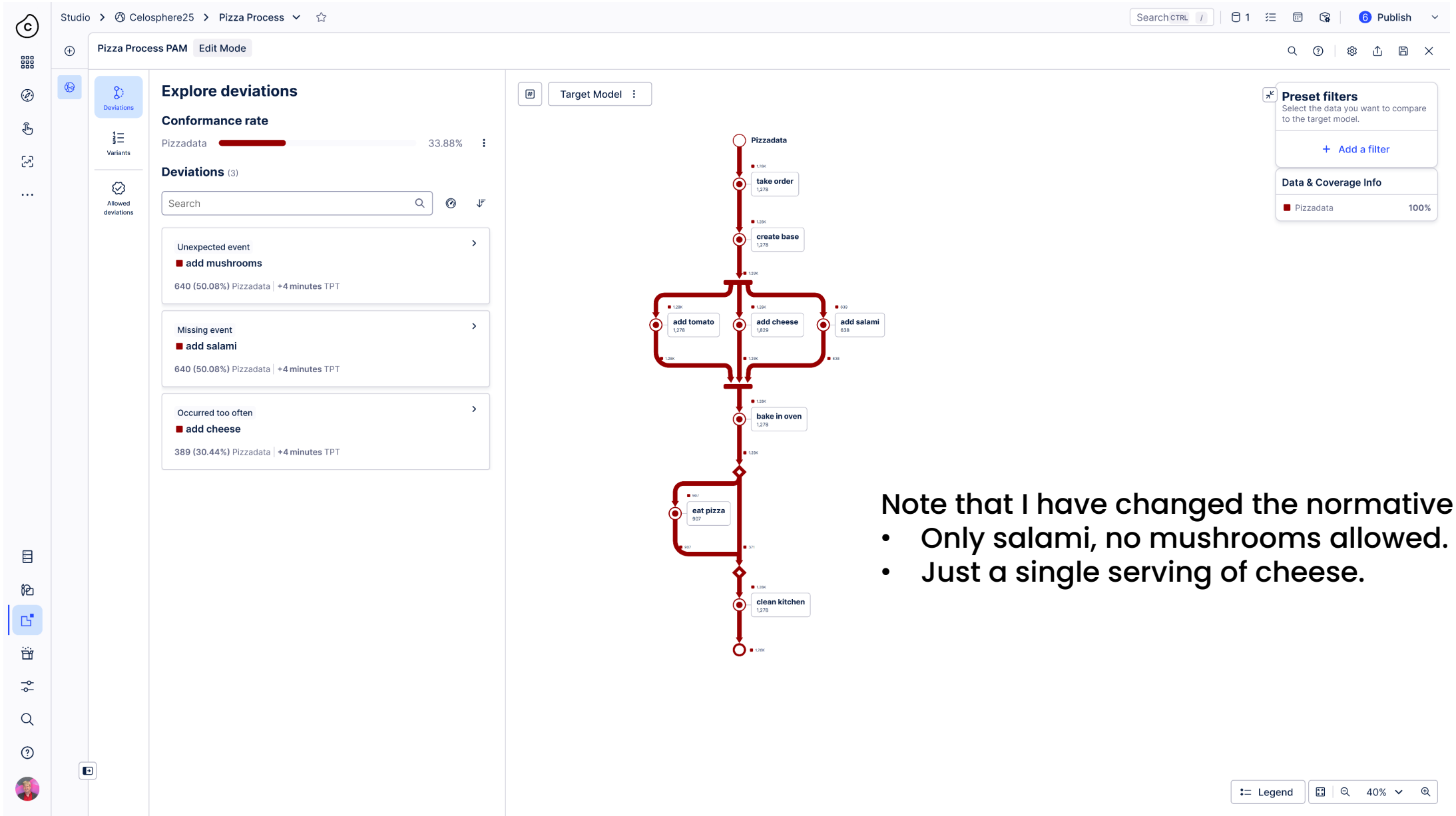
Performance analysis using the Celonis Process Adherence Manager





907 of 1278 pizzas (71%) are eaten. The time between create base and start eating is between 3 and 188 minutes (26.1 minutes on average)

Conformance checking using the Celonis Process Adherence Manager



Note that I have changed the normative model:

- Only salami, no mushrooms allowed.
- Just a single serving of cheese.

Conformance checking using the Celonis Process Adherence Manager

Conformance rate

Pizzadata 33.88%

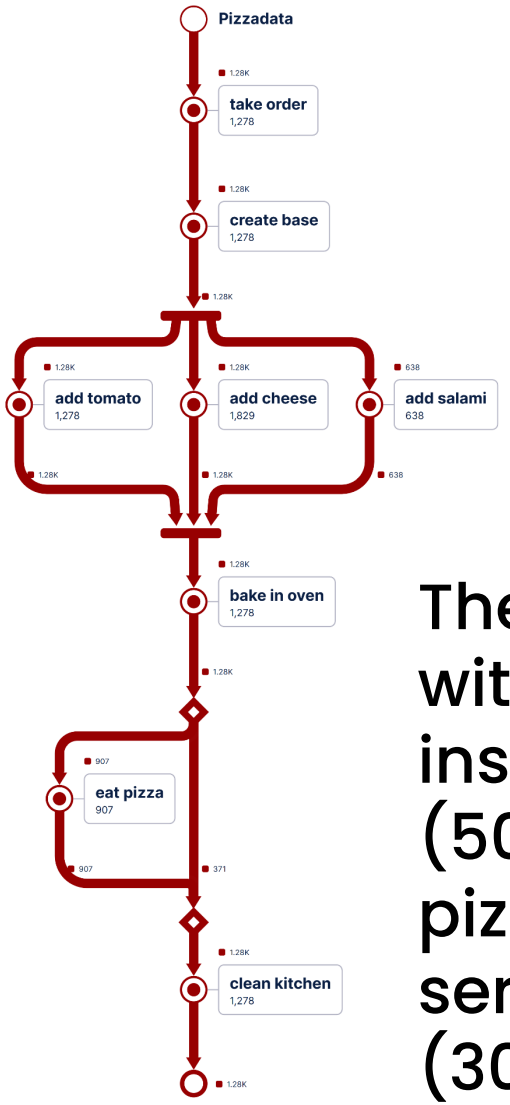
Deviations (3)

Search

Unexpected event
add mushrooms
640 (50.08%) Pizzadata | +4 minutes TPT

Missing event
add salami
640 (50.08%) Pizzadata | +4 minutes TPT

Occurred too often
add cheese
389 (30.44%) Pizzadata | +4 minutes TPT



There were 640 pizzas with mushrooms instead of salami (50.08%) and 389 pizzas with multiple servings of cheese (30.44%).

Deviations
Variants
Allowed deviations

Explore deviations

Conformance rate

Pizzadata 0%

Deviations (3)

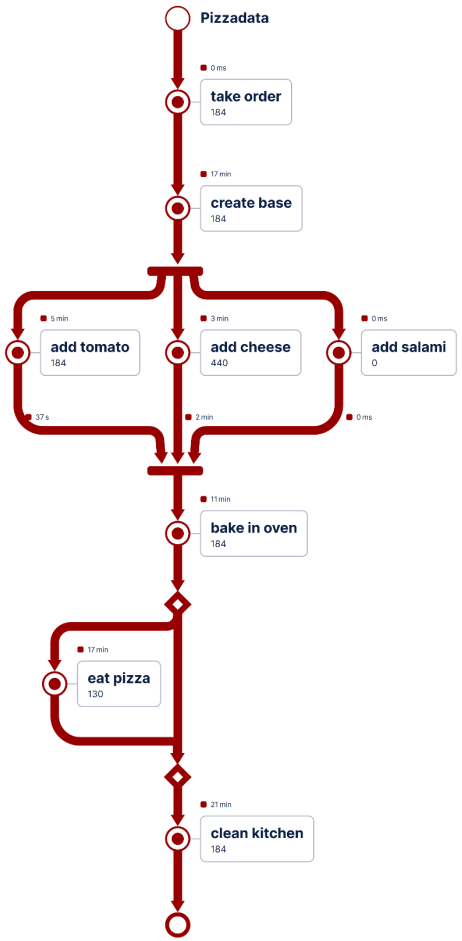
Search

Occurred too often
add cheese
184 (100%) Pizzadata | **0 milliseconds** TPT

Unexpected event
add mushrooms
184 (100%) Pizzadata | **0 milliseconds** TPT

Missing event
add salami
184 (100%) Pizzadata | **0 milliseconds** TPT

184 cases (14.4%) have all three deviations



Filters

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

Pizzadata
add salami is missing

Pizzadata
add cheese occurred too often

Pizzadata
add mushrooms occurred unexpectedly

+ Add a filter

Data & Coverage Info

Pizzadata 14.40%

Legend
40%

Studio > Celosphere25 > Pizza Process

Search CTRL /

Publish

Pizza Process PAM

Deviations

Variants

Allowed deviations

Explore deviating variants

Explore variants that include the detected deviations from your target process model.

Deviations occur in 55 variants (showing 55 of 55)

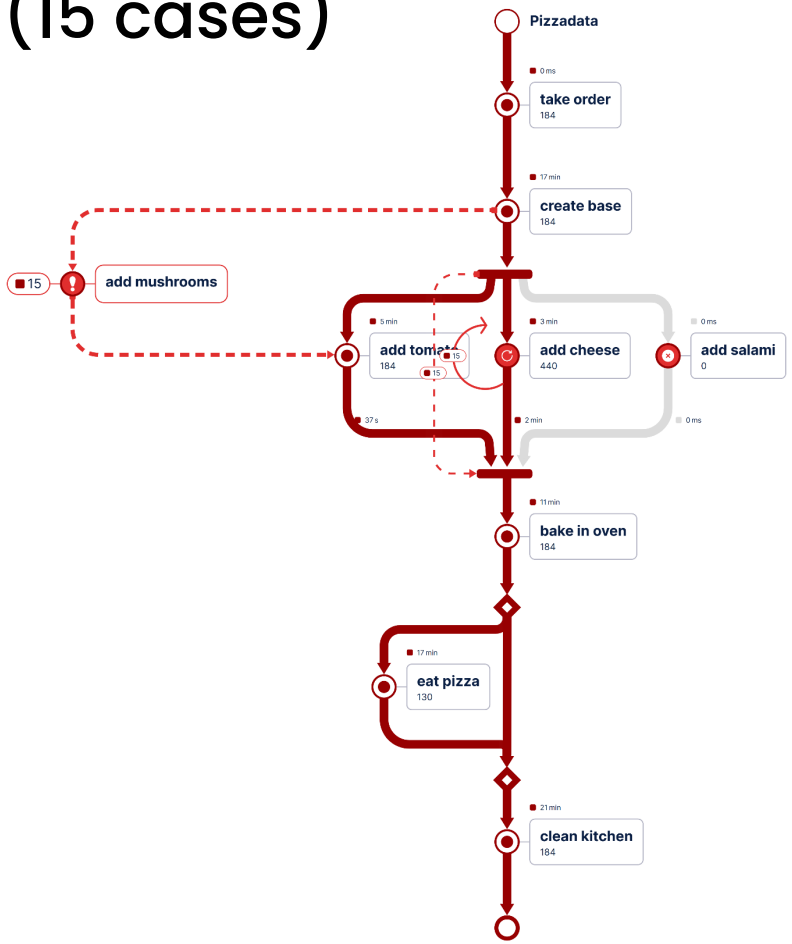
Pizzadata | Any deviation type | Any event type

Cases	Coverage	Avg TPT
15	8.15%	1 hours
15	8.15%	2 hours
11	5.98%	1 hours
11	5.98%	42 minutes
8	4.35%	1 hours
8	4.35%	1 hours
8	4.35%	1 hours
7	3.80%	48 minutes

Variant detail

Event sequence	Deviations	Time delta
take order		0
create base		14 minutes
add mushrooms	Occurred unexpectedly	1 minutes
add tomato		18 seconds
add cheese		22 seconds
add cheese	Occurred too often	43 seconds
add salami	Missing event	-
bake in oven		1 minutes
eat pizza		20 minutes
clean kitchen		22 minutes

A variant with all three problems (15 cases)



to cb am at ac ac bo ep ck

Filters

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

- Pizzadata — add salami is missing
- Pizzadata — add cheese occurred too often
- Pizzadata — add mushrooms occurred unexpectedly

+ Add a filter

Data & Coverage Info

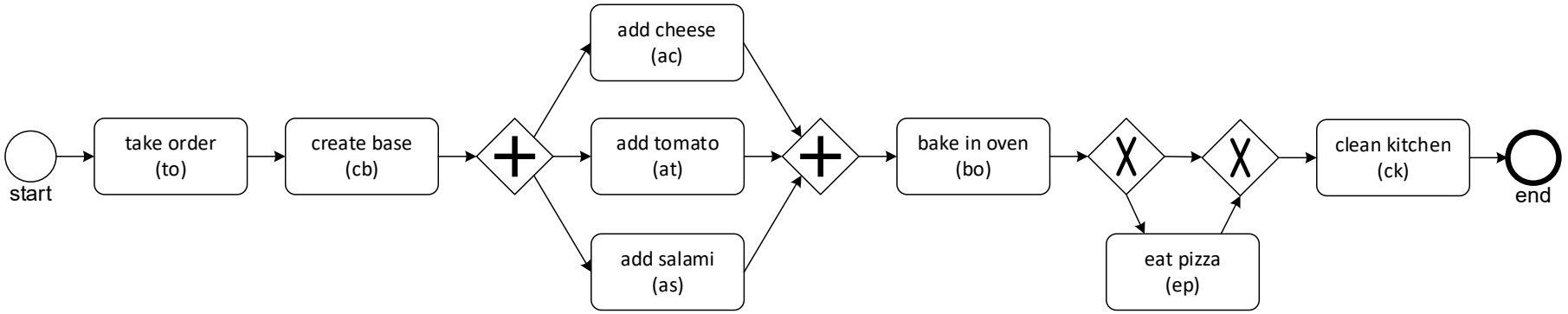
Pizzadata 14.40%

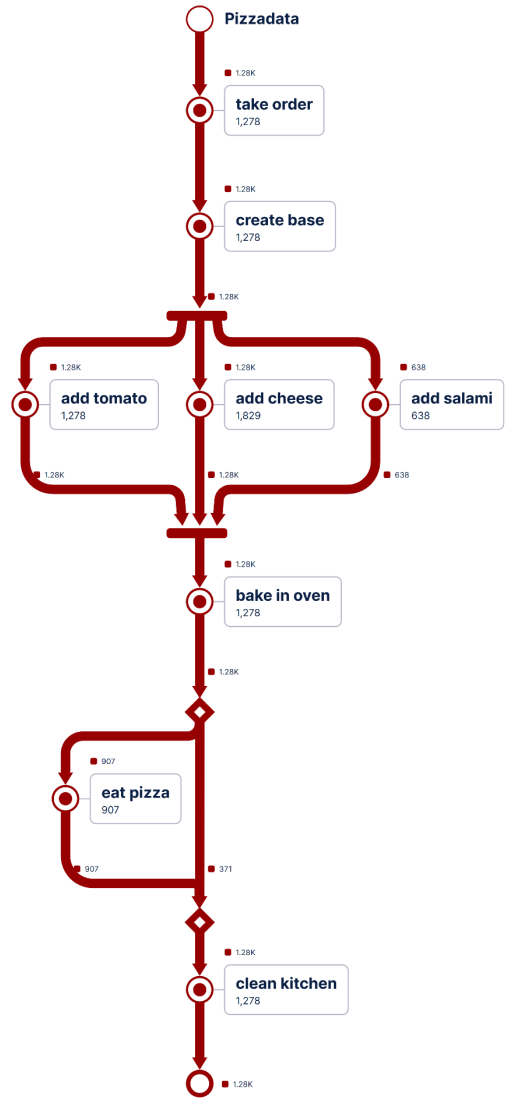
Legend | 40%

How does this work?

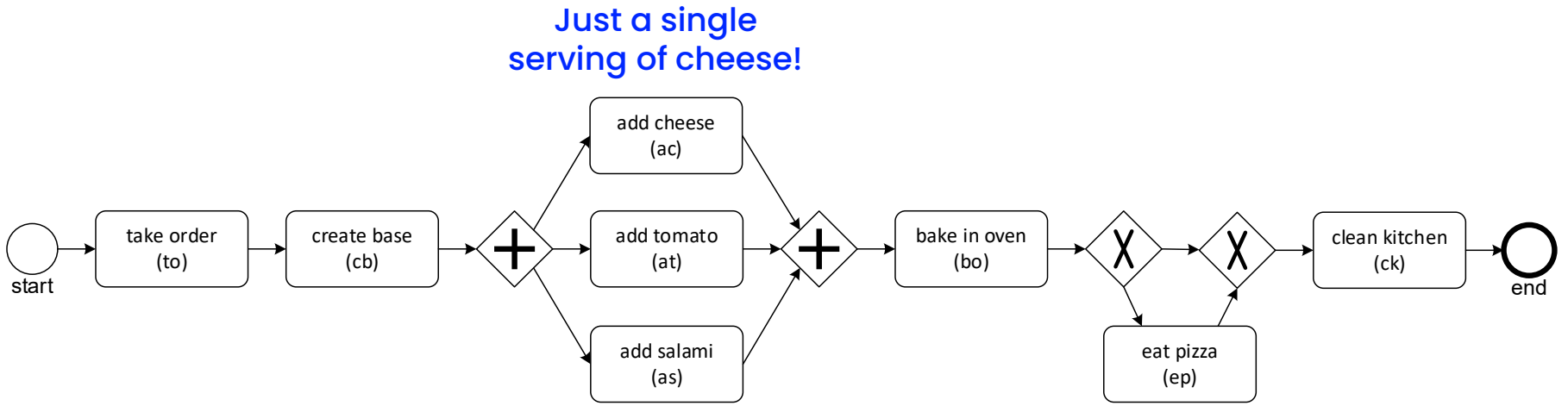
Alignments!

For conformance checking we need a process model and event data



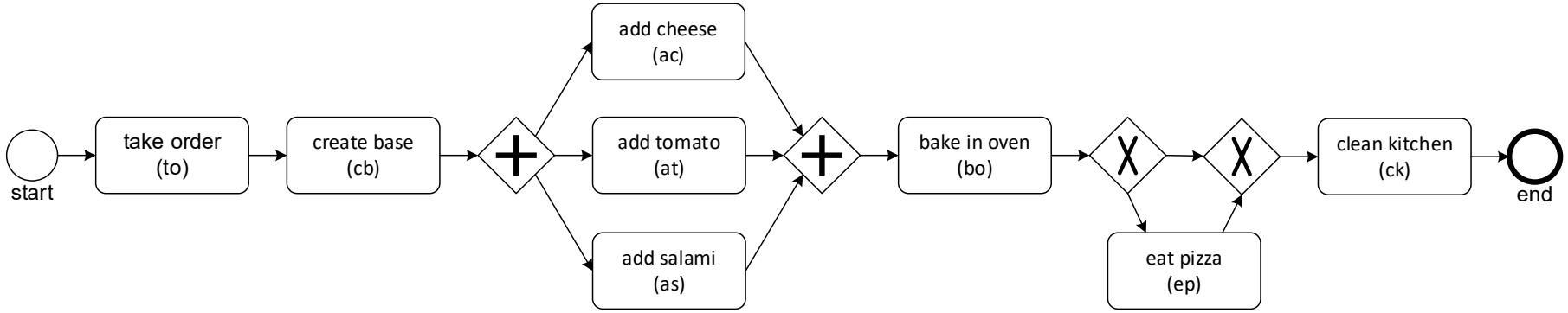


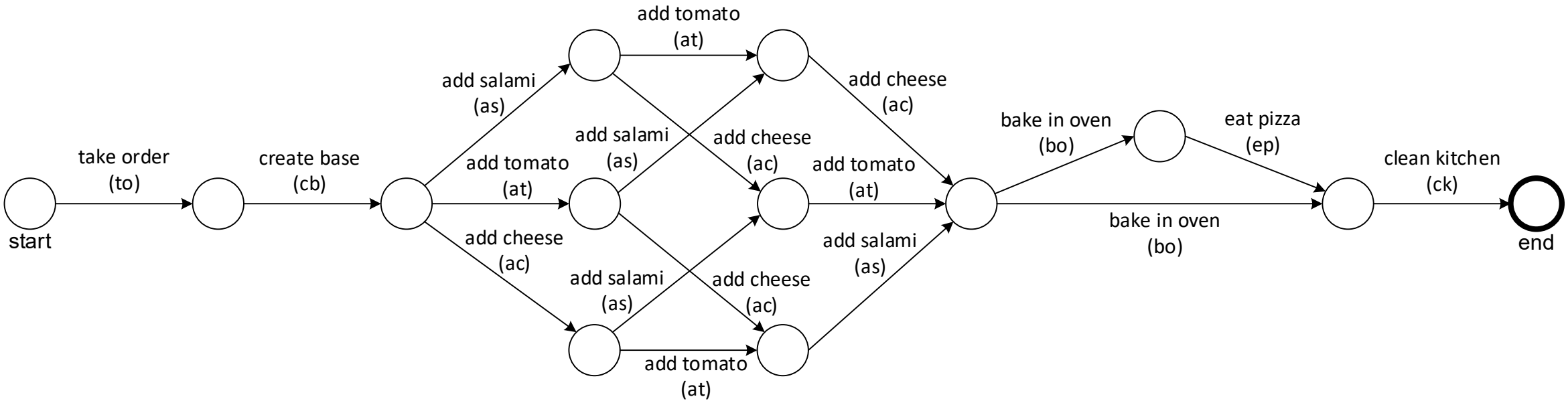
For conformance checking we need a process model and event data



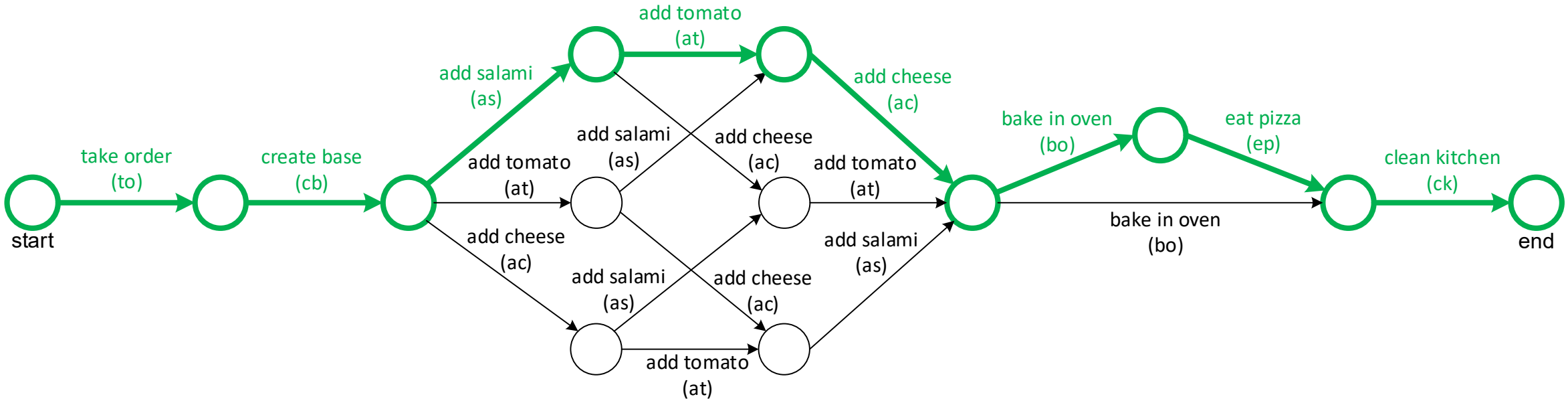
A variant appearing 15 times in the original event data.

The state space of the BPMN model (normally much larger and with loops)

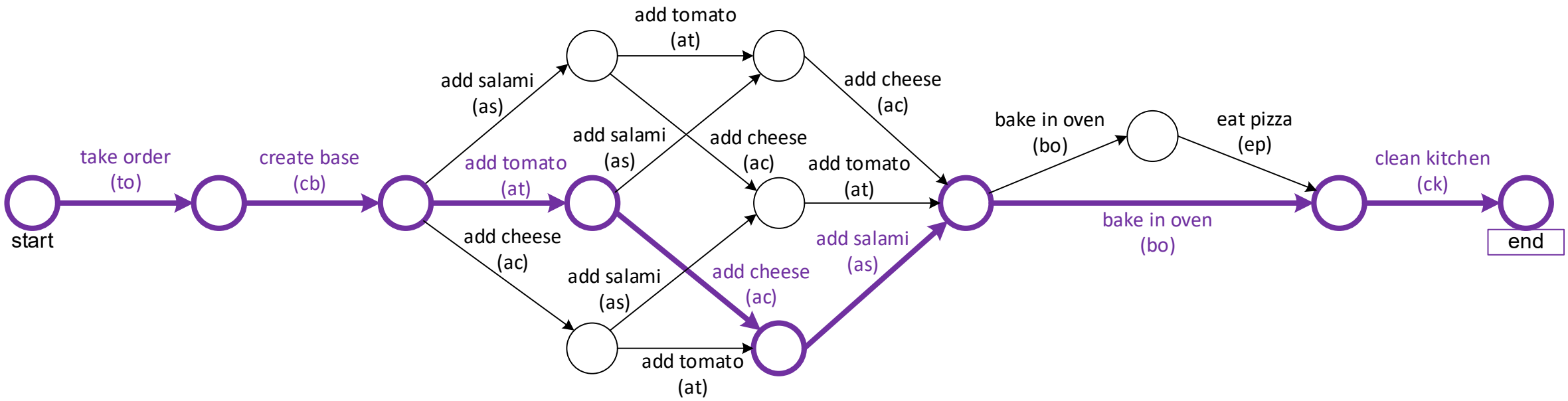




Behavior of BPMN model is any path from start to end

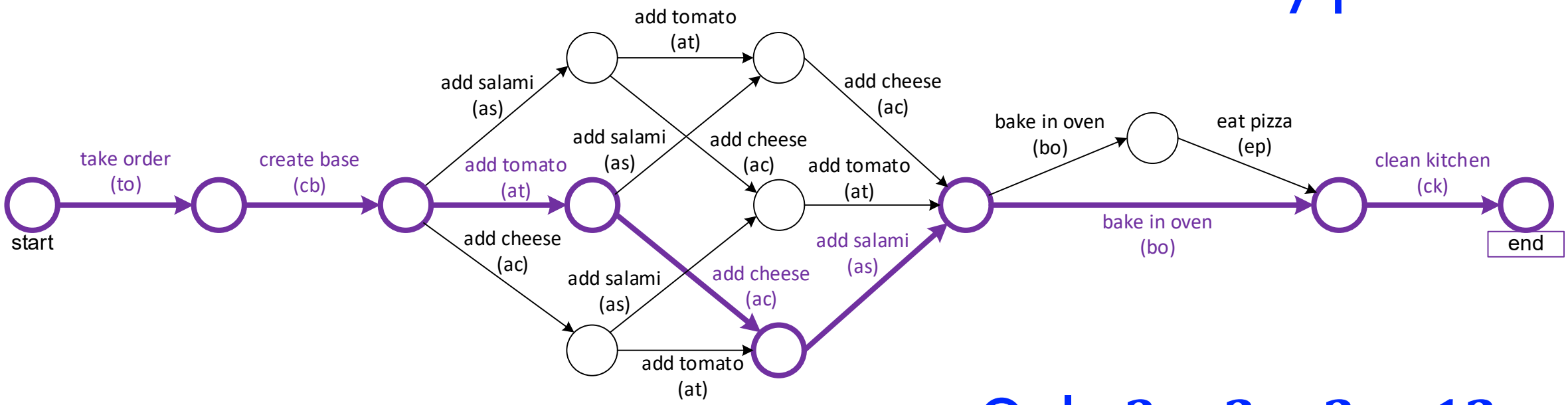


Another path



Another path

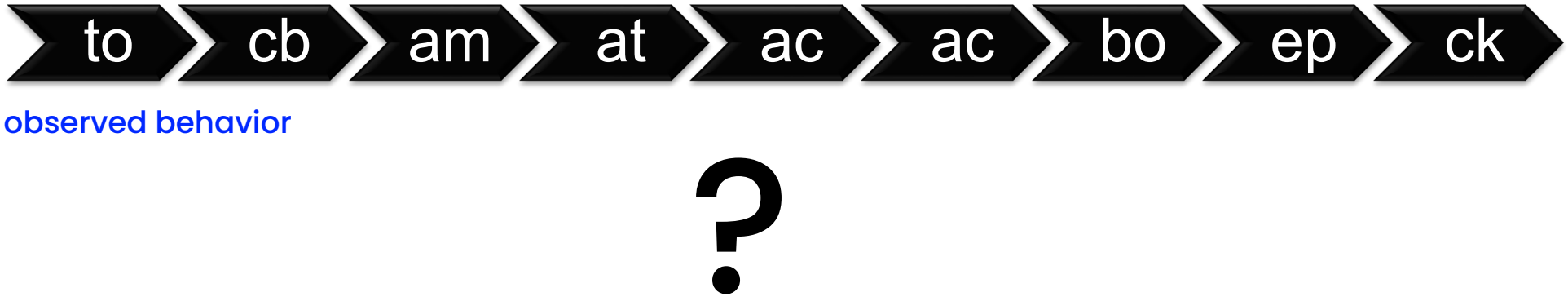
How many paths?



Only $3 \times 2 \times 2 = 12$

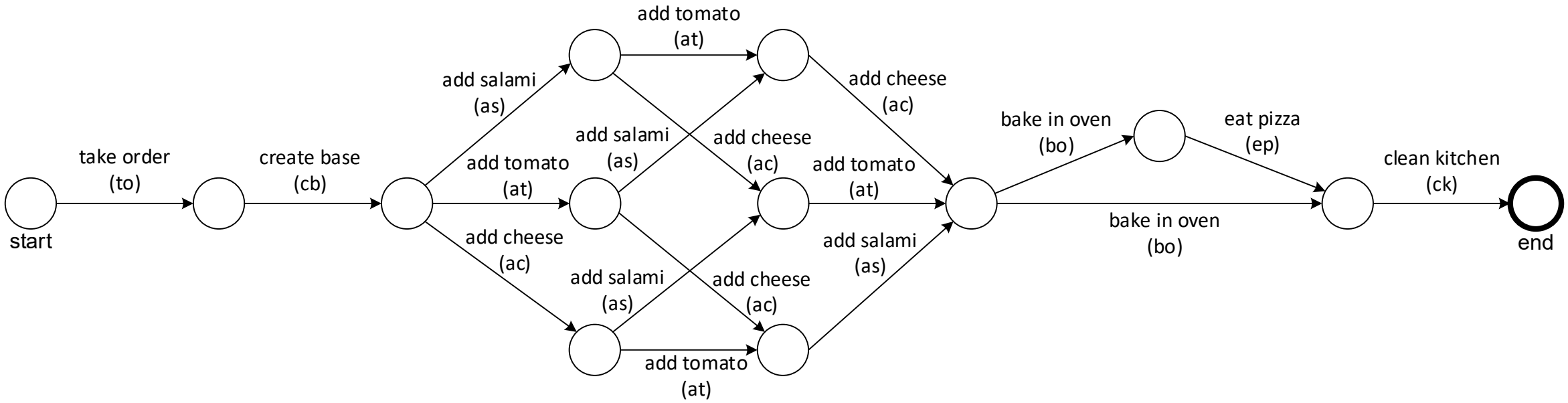
to cb at ac as bo ck

Find a path closest to the observed sequence



observed behavior

?

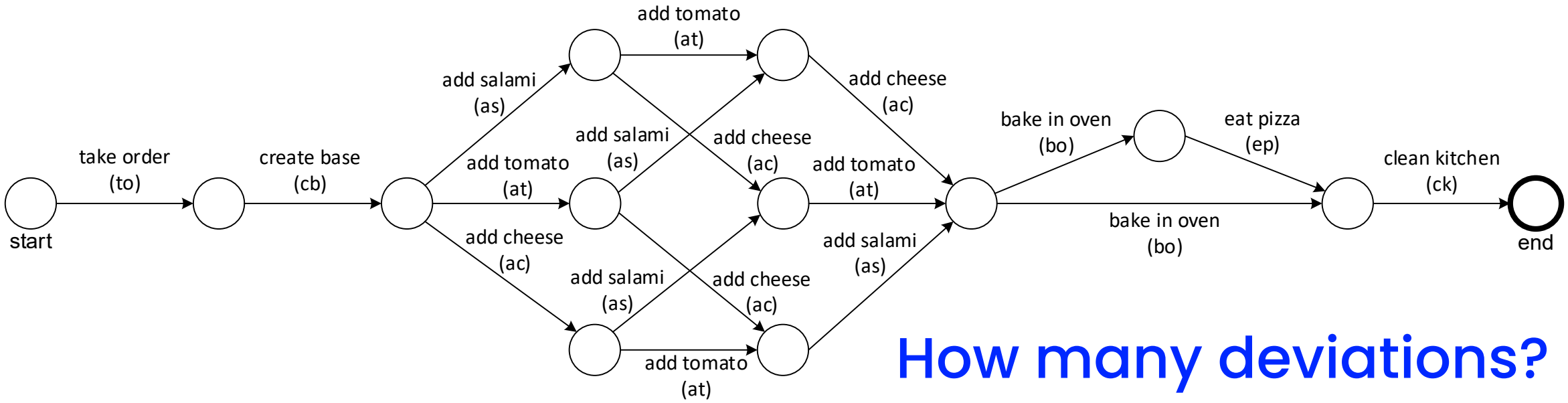


Find a path closest to the observed sequence

observed behavior

?

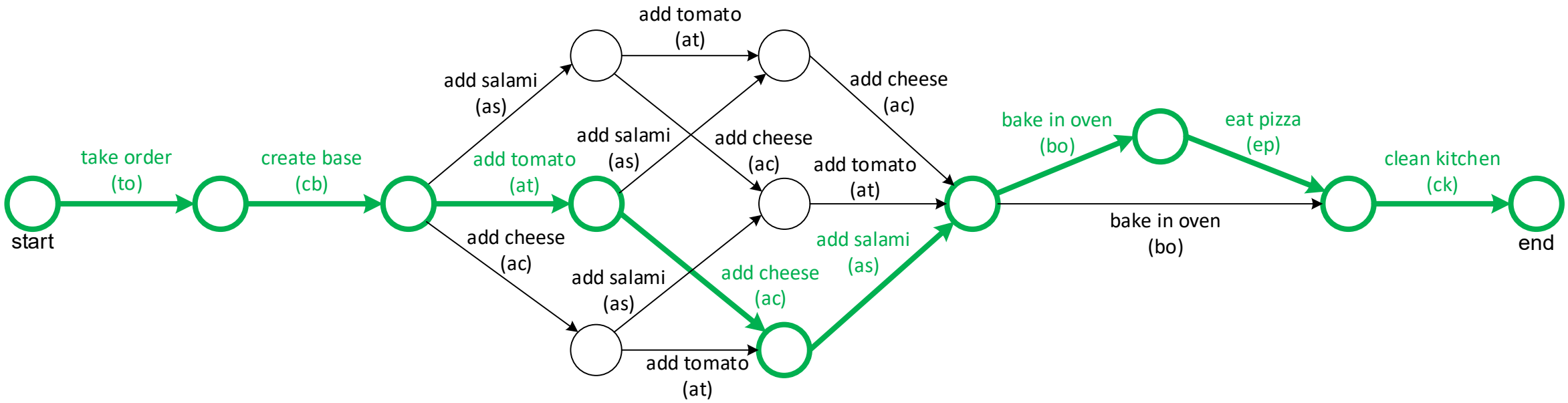
What is the path closest to the observed sequence?



How many deviations?

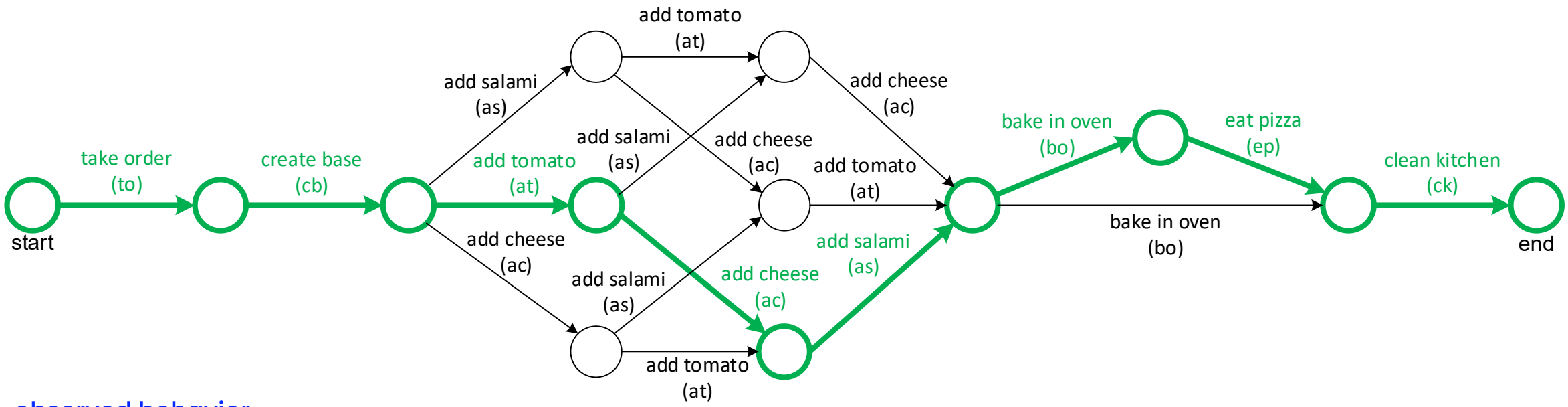
Find a path closest to the observed sequence

observed behavior



modeled behavior

This is a path closest to the observed sequence (three deviations)

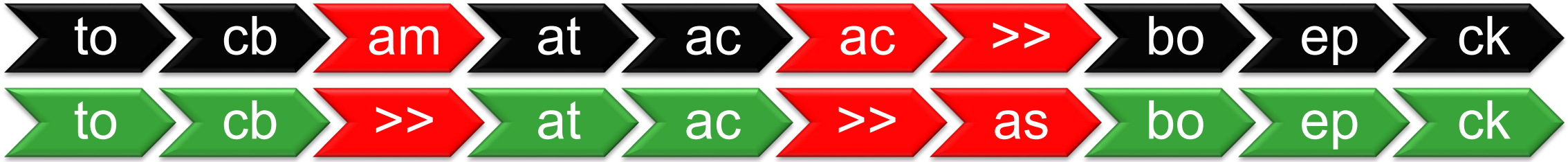


observed behavior

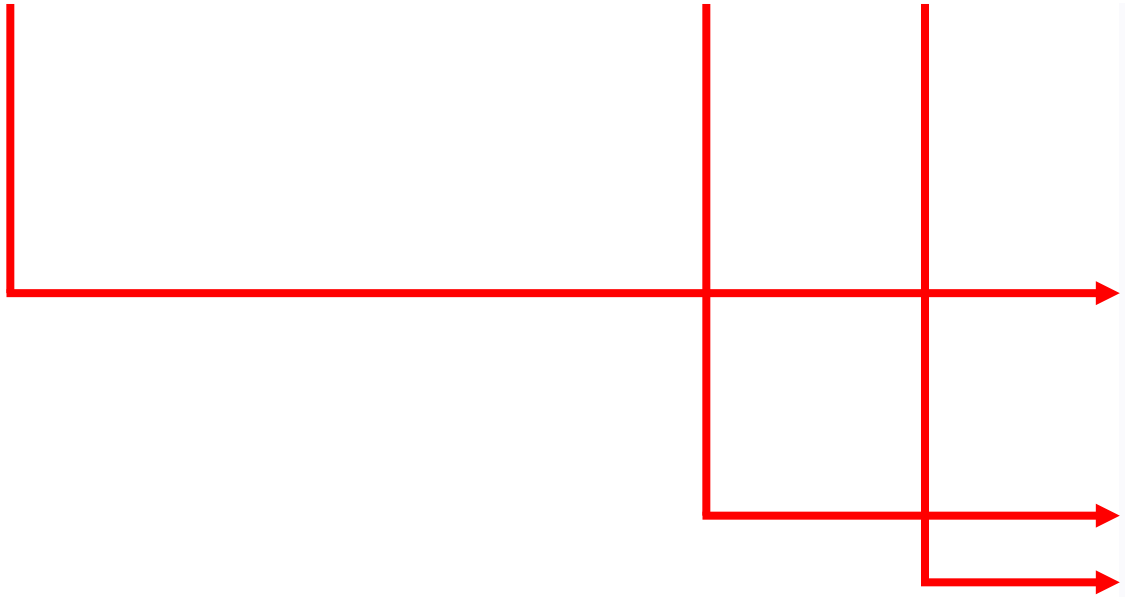
modeled behavior

Three deviations!

observed behavior (15 cases)



modeled behavior

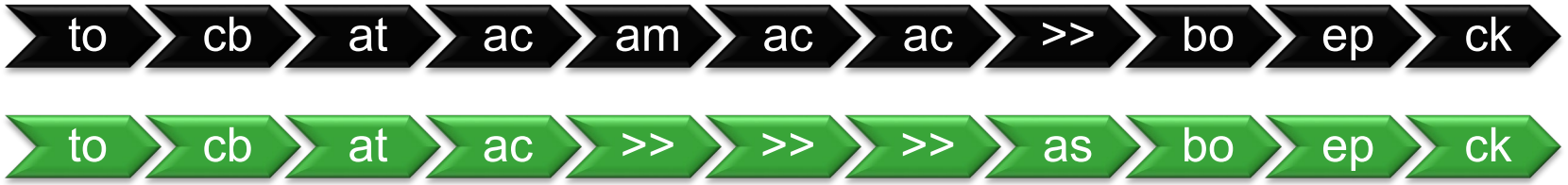


Variant detail

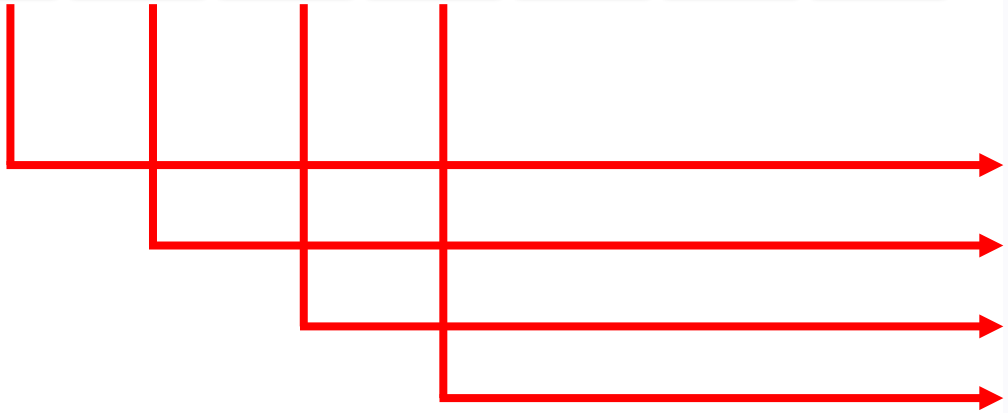
Event sequence	Deviations	Time delta
take order		0
create base		14 minutes
add mushrooms	Occurred unexpectedly	1 minutes
add tomato		18 seconds
add cheese		22 seconds
add cheese	Occurred too often	43 seconds
add salami	Missing event	-
bake in oven		1 minutes
eat pizza		20 minutes
clean kitchen		22 minutes

Another example with four deviations

observed behavior (one case)



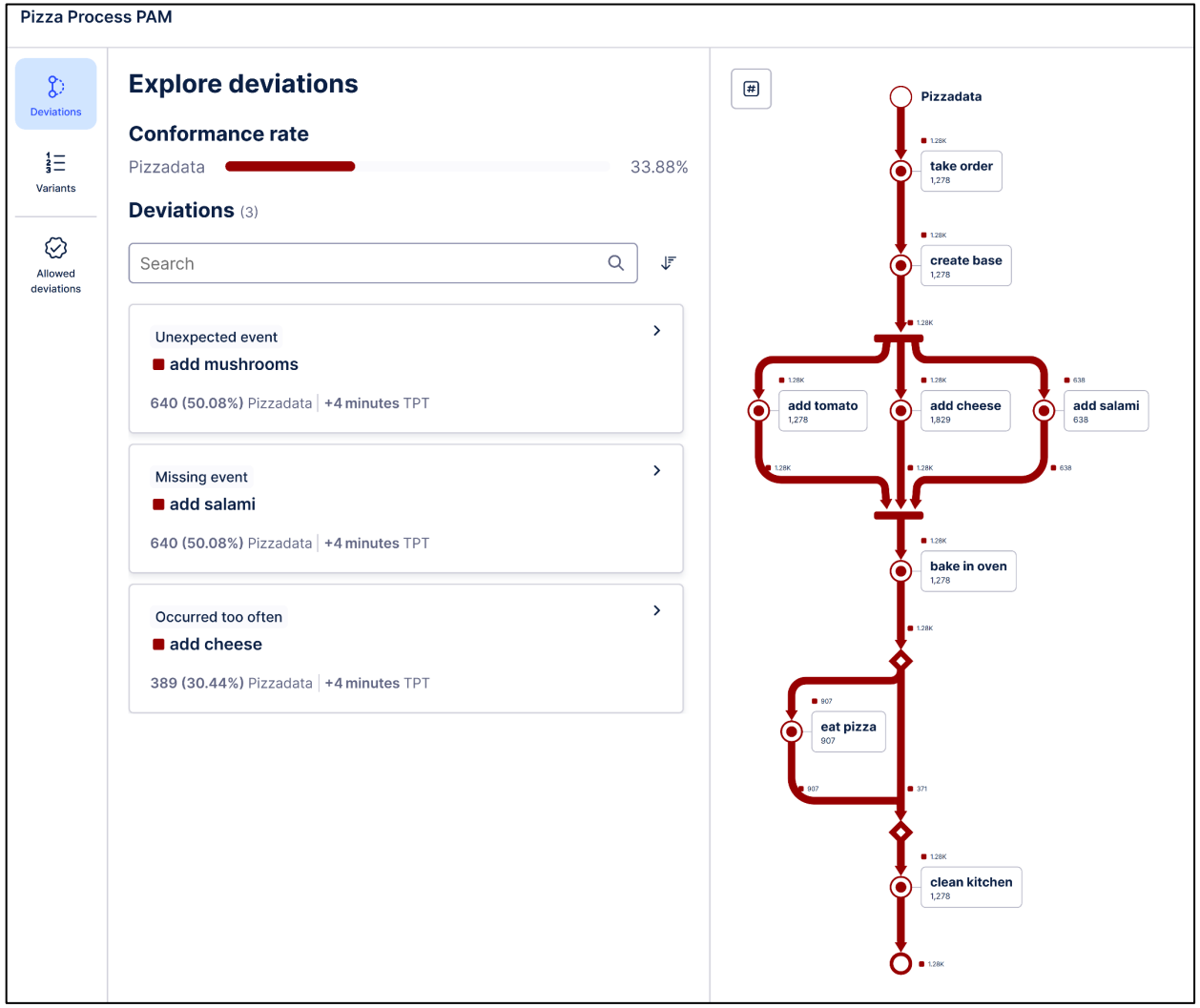
modeled behavior



Variant detail

Event sequence	Deviations	Time delta
take order		0
create base		8 minutes
add tomato		1 hours
add cheese		21 seconds
add mushrooms	Occurred unexpectedly	15 seconds
add cheese	Occurred too often	26 seconds
add cheese	Occurred too often	38 seconds
add salami	Missing event	-
bake in oven		2 minutes
eat pizza		7 minutes
clean kitchen		8 minutes

Notoriously difficult problem: Requires state-of-the-art techniques



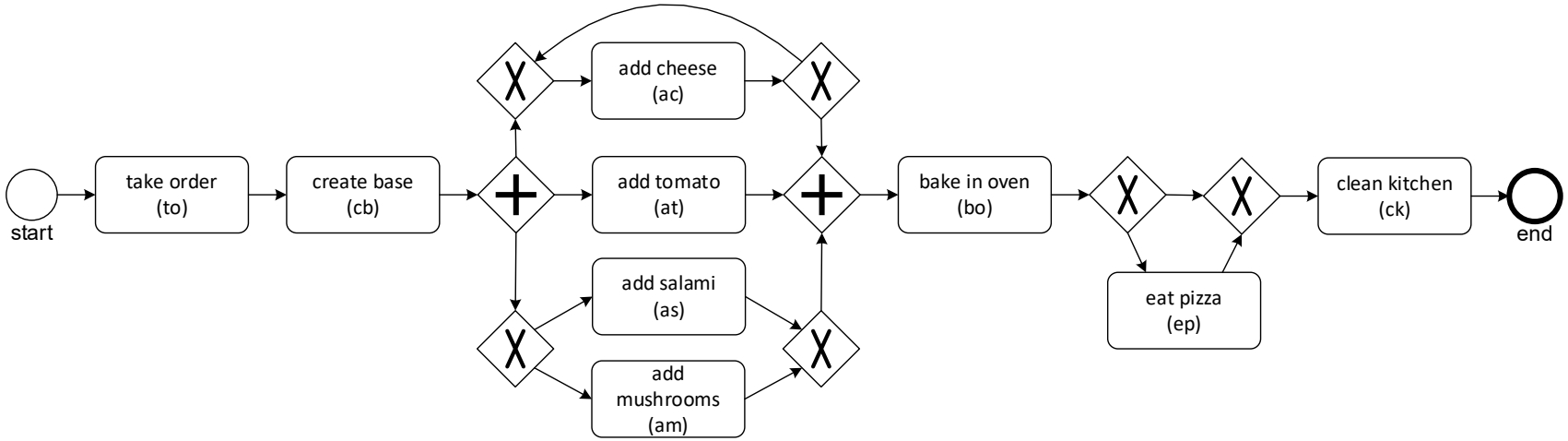
- Alignments are the “gold standard” in conformance checking.
- Requires solving an optimization problem for each case.
- Also needed for performance analysis.
- Made scalable using state-of-the-art techniques.

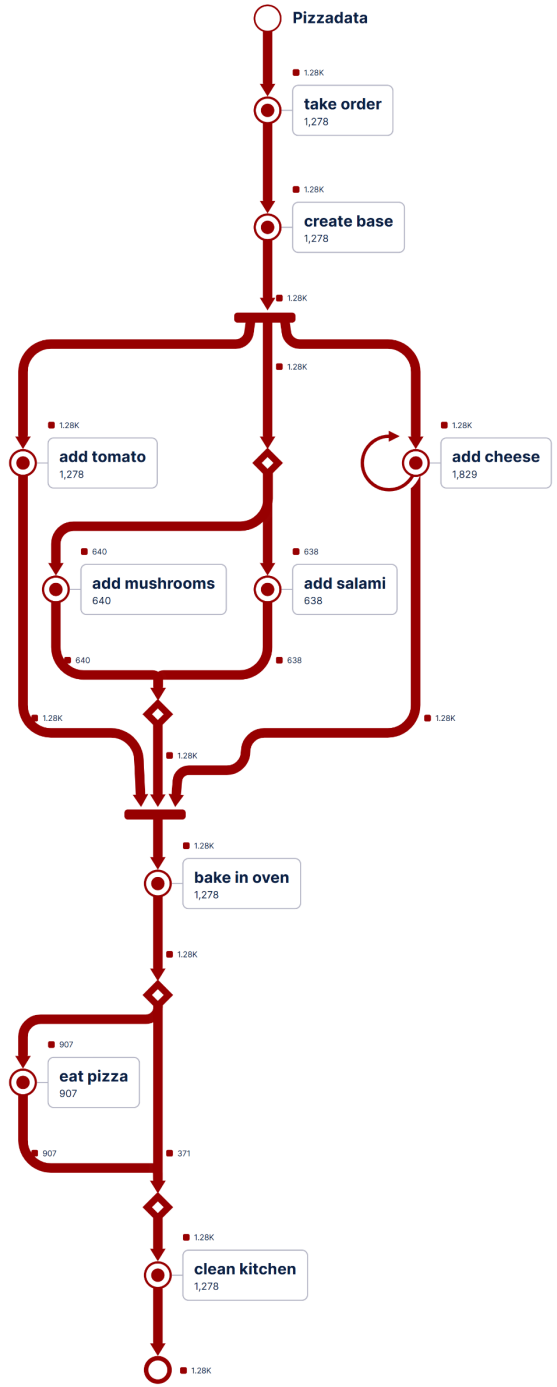
Real world Spaghetti is colored

Object-Centric Process Mining


•Variability
•Intra-object concurrency
•Inter-object concurrency

Taking a step back.

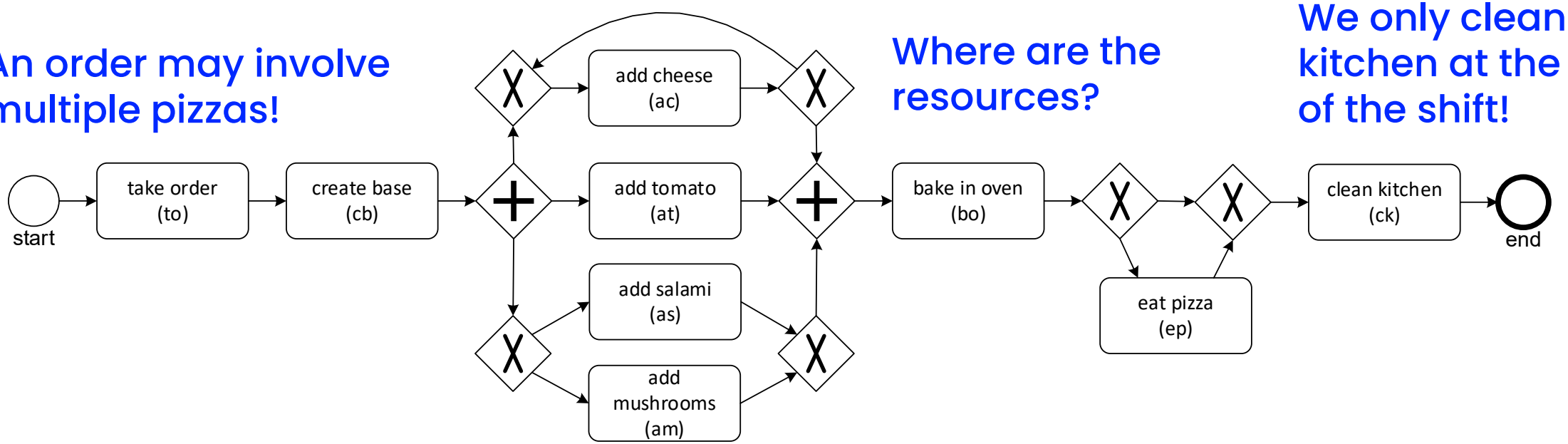




The model correctly describes the case-centric event data used as input, but what would the actual process look like?

Taking a step back.

An order may involve multiple pizzas!



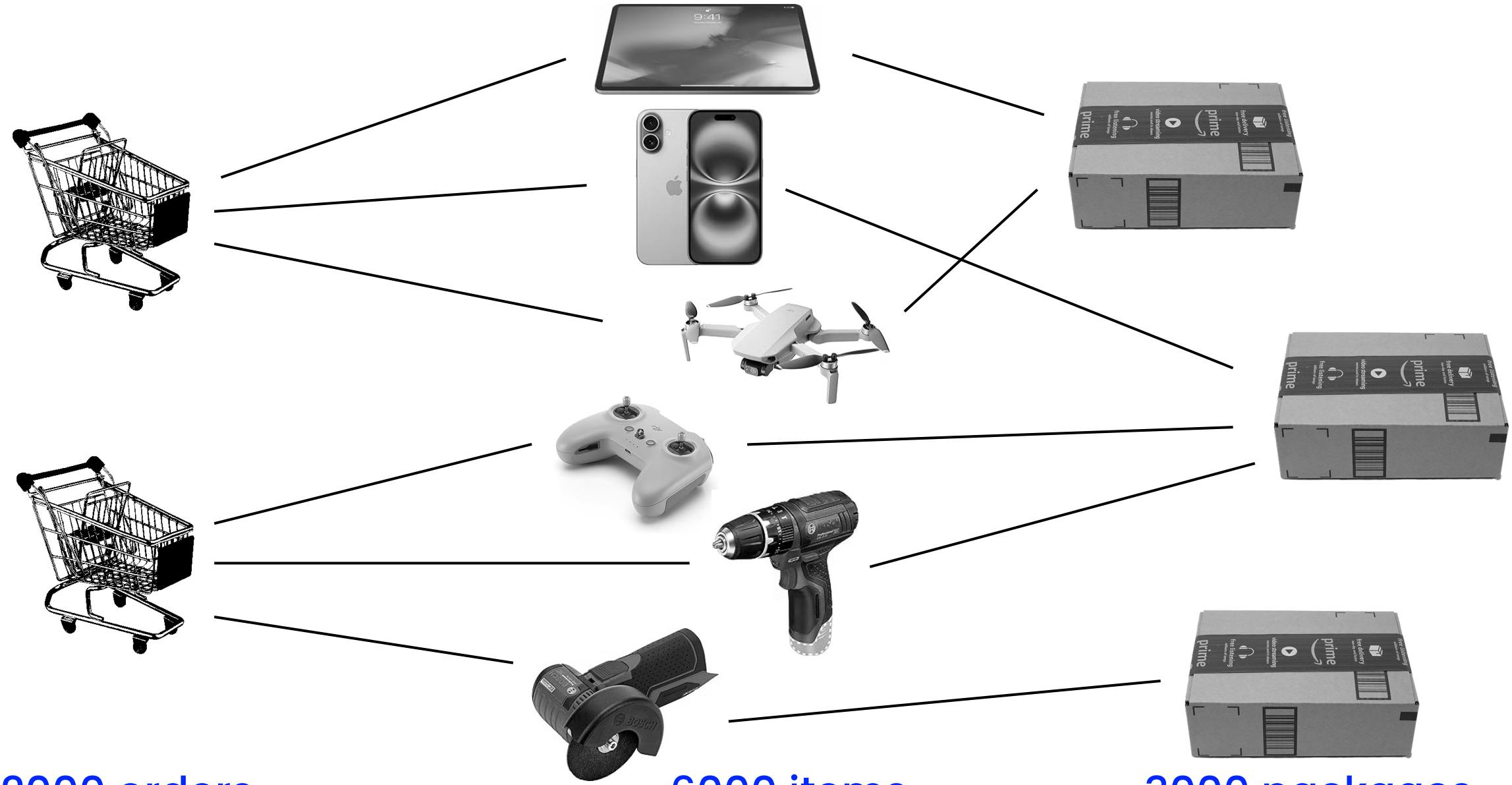
Where are the resources?

We only clean the kitchen at the end of the shift!

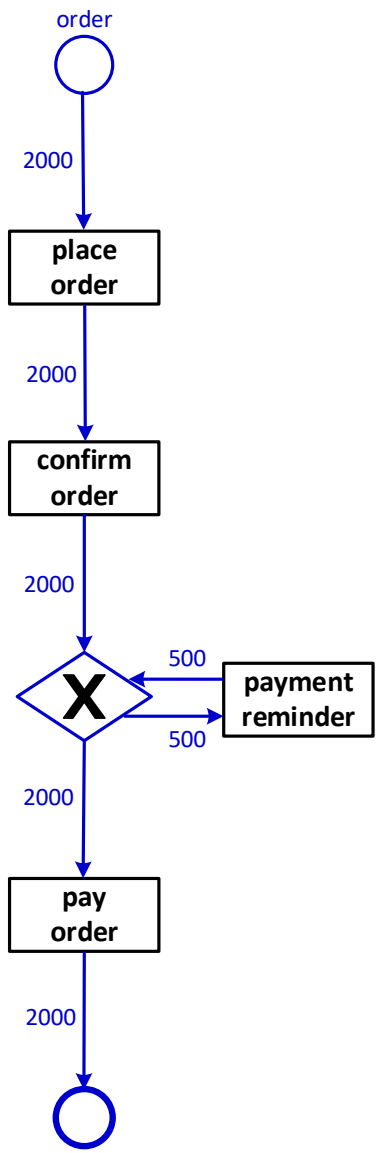
We need to buy the ingredients before!

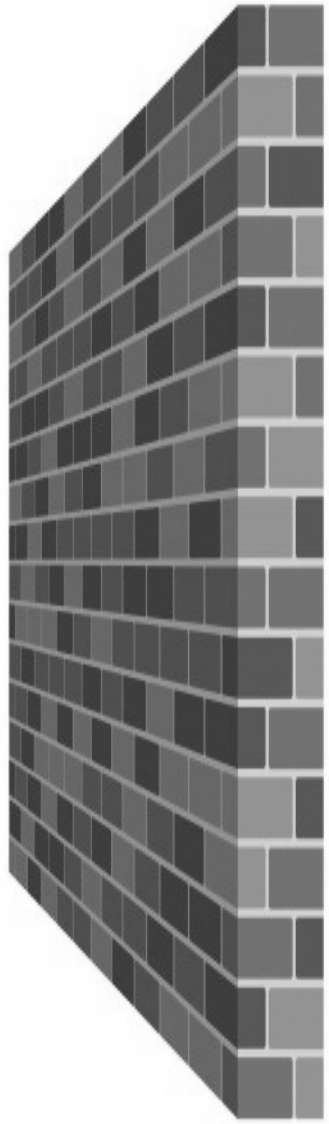
Shortages influence the process.

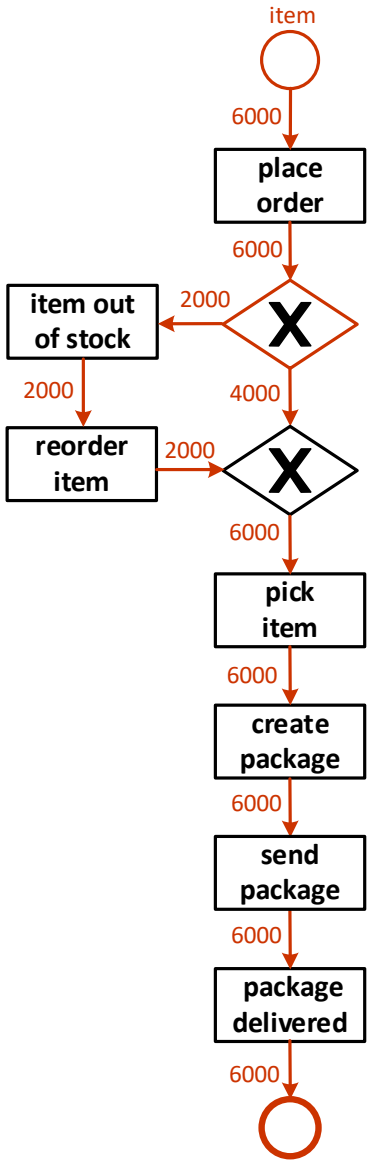
Minimal example with three types of objects: orders, items, and packages

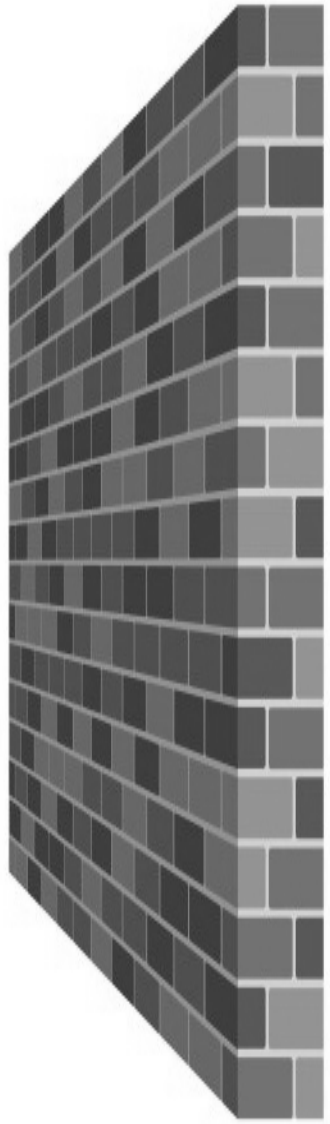


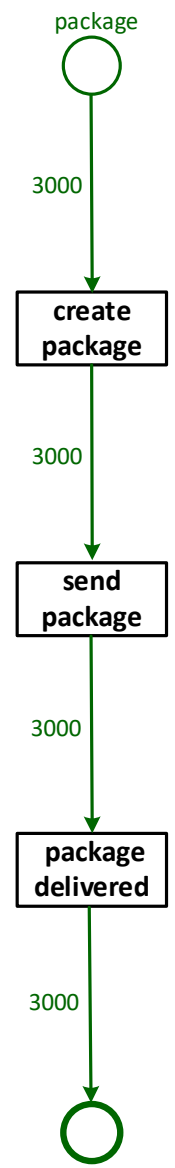
We cannot see the process by looking at disconnected object types



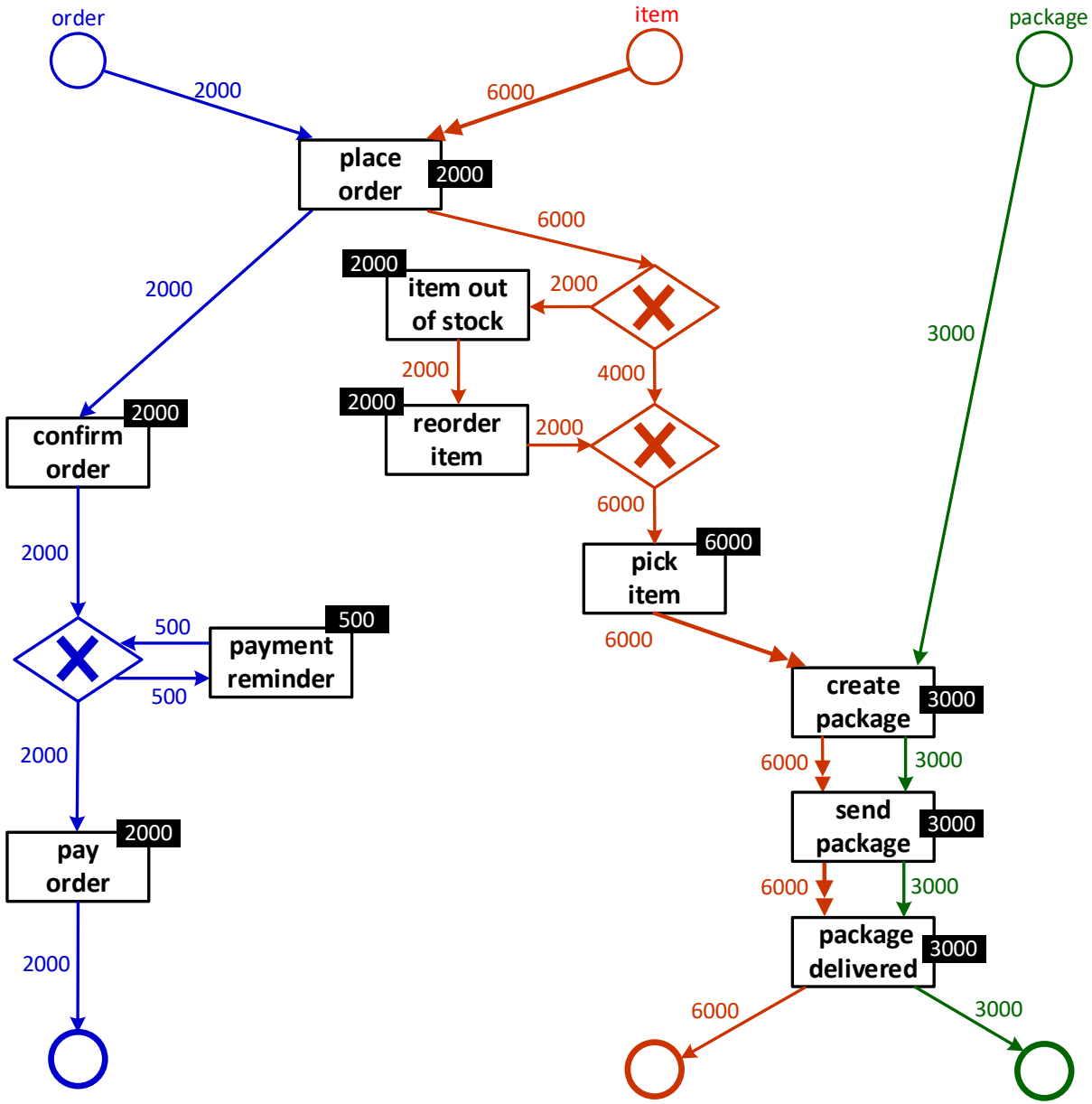








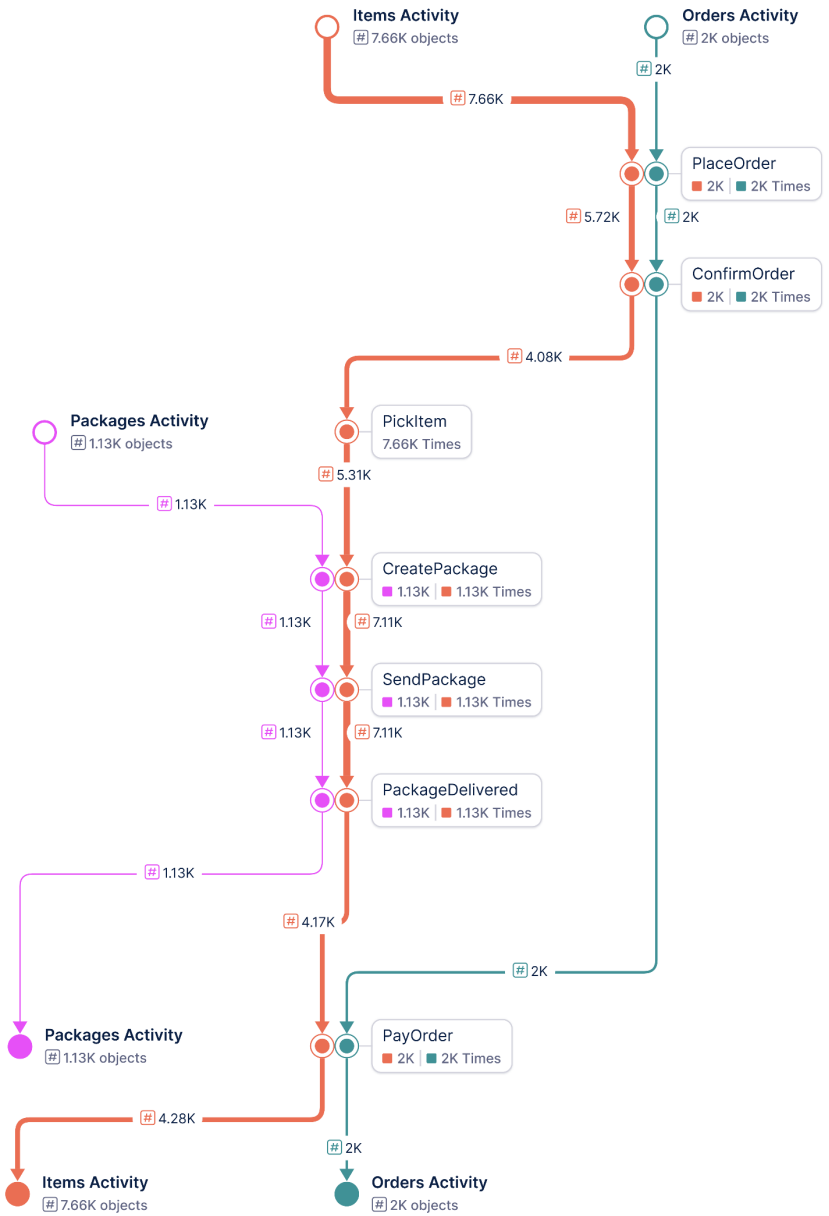
Discovered Object-Centric Process Model



Order Management View

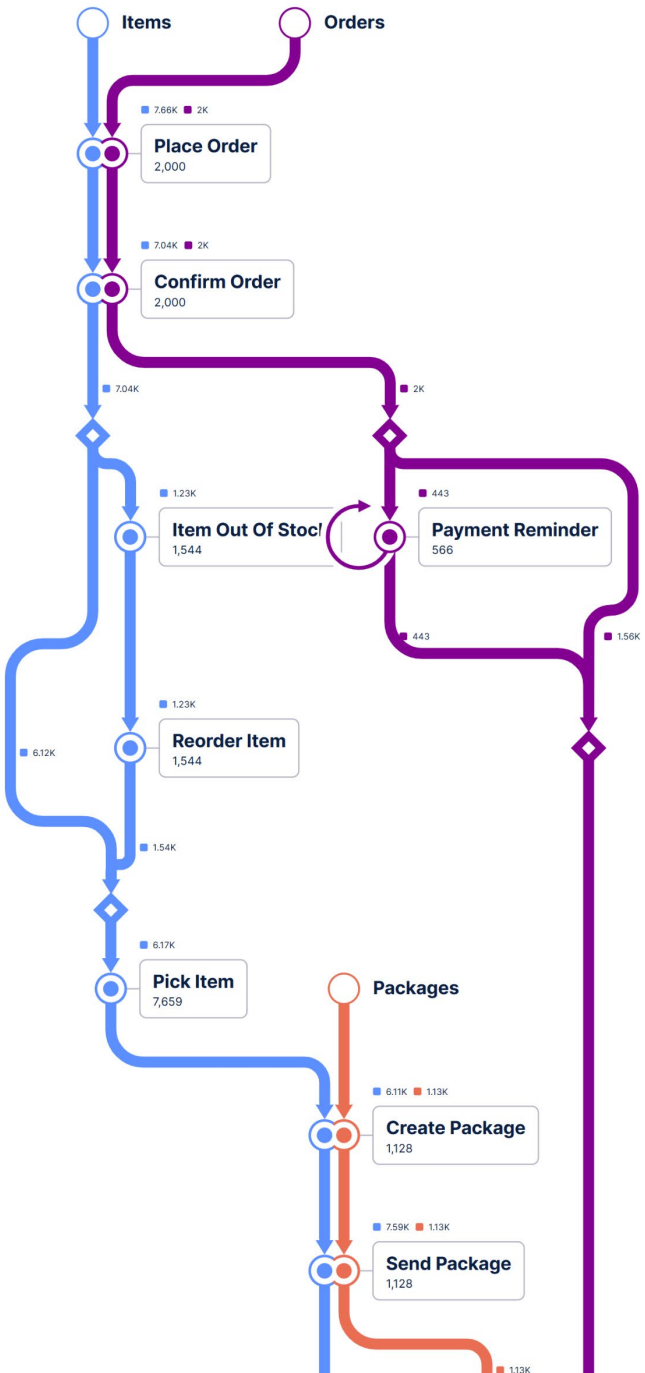
Events 7 of 11

Process Explorer
(based a more realistic event data)



DD

Legend
92%



Throughput time filter

from: Orders — Place Order — First

to: Packages — Package Delivered — Last

avg. **14.6 days** | med. **11 days** | min. **0 days** | max. **0 days**

↑ Frequency

500

0

1 9 14 19 24 29 34 39

Time →

1 to 87 days

2K of 2K | 100% are selected

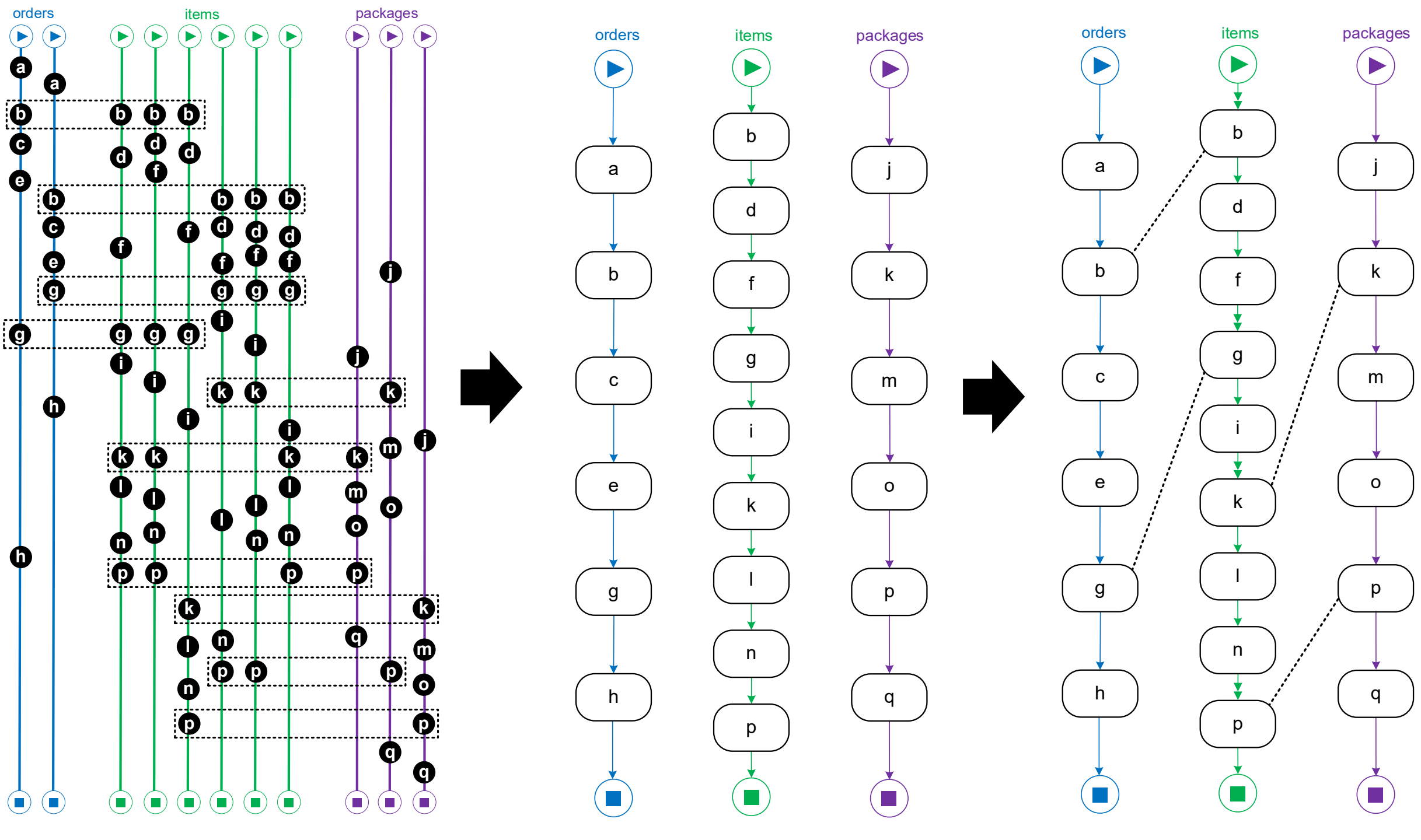
Cancel | Apply filter

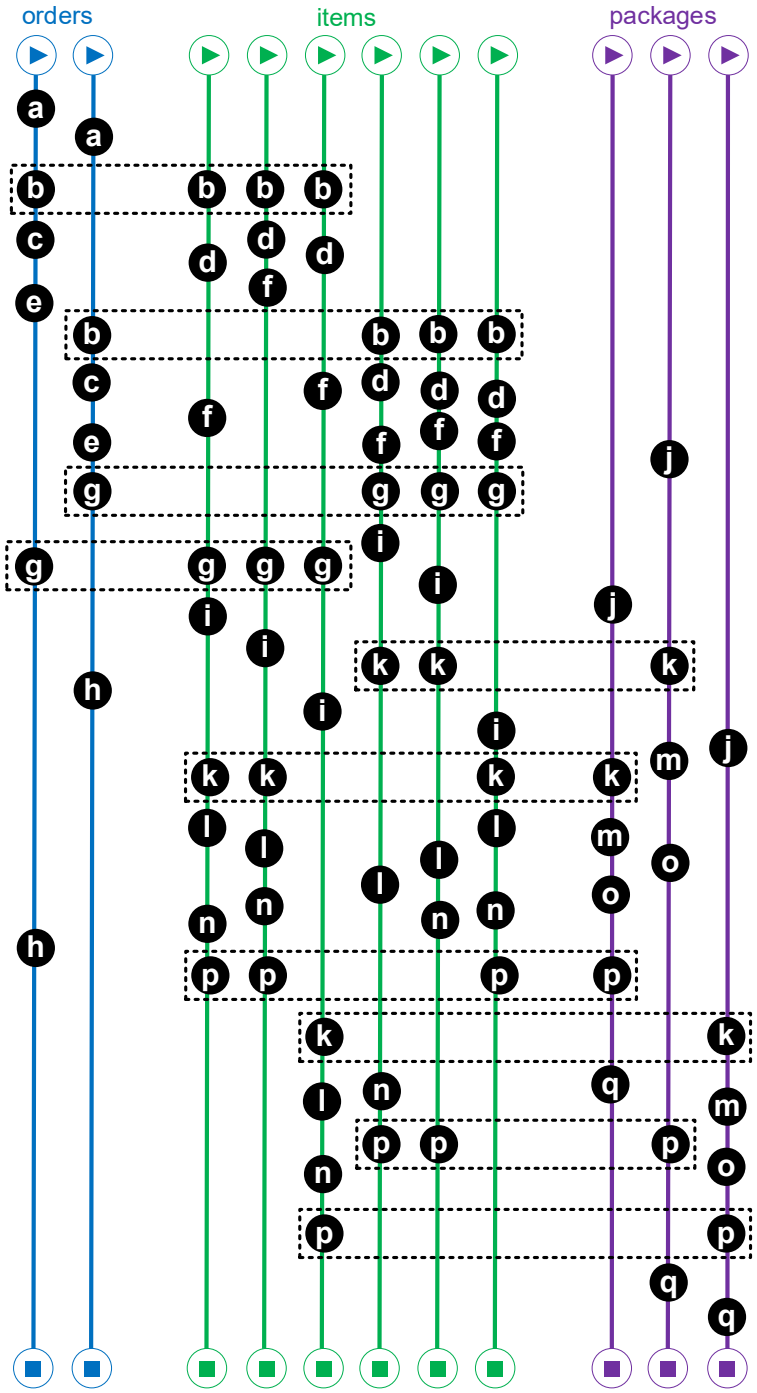
Process Adherence Manager

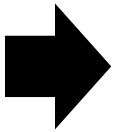
If it gets too complicated, you are probably doing something wrong …

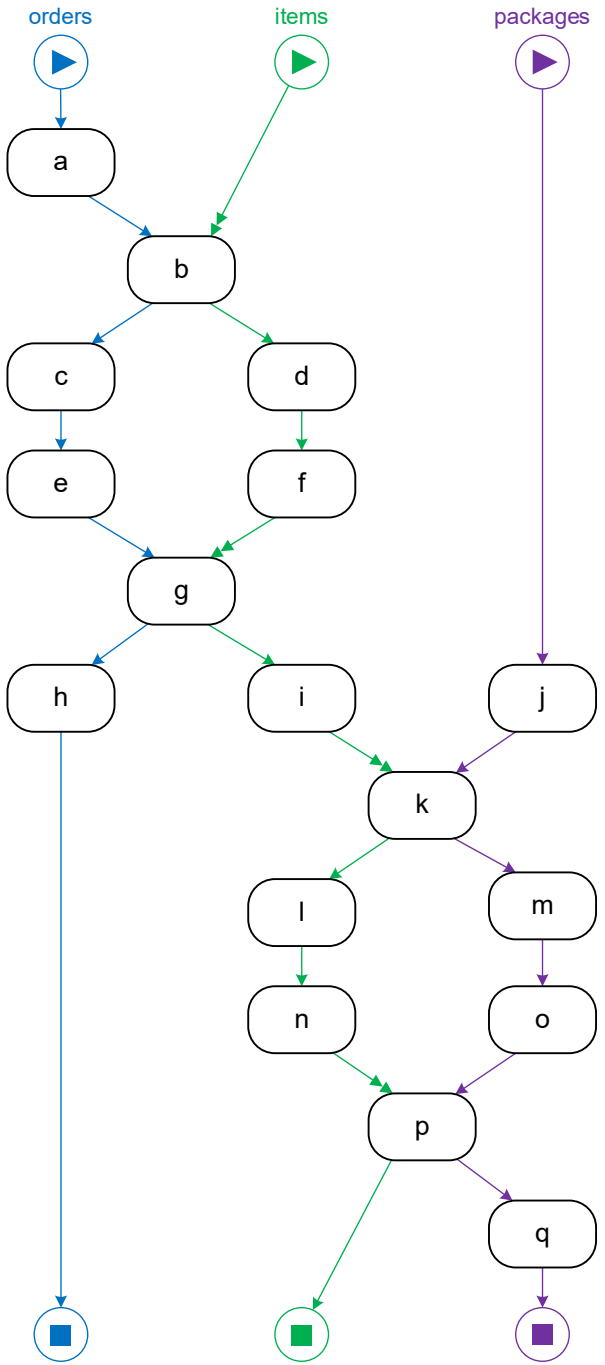
Drop your case-centric bias!

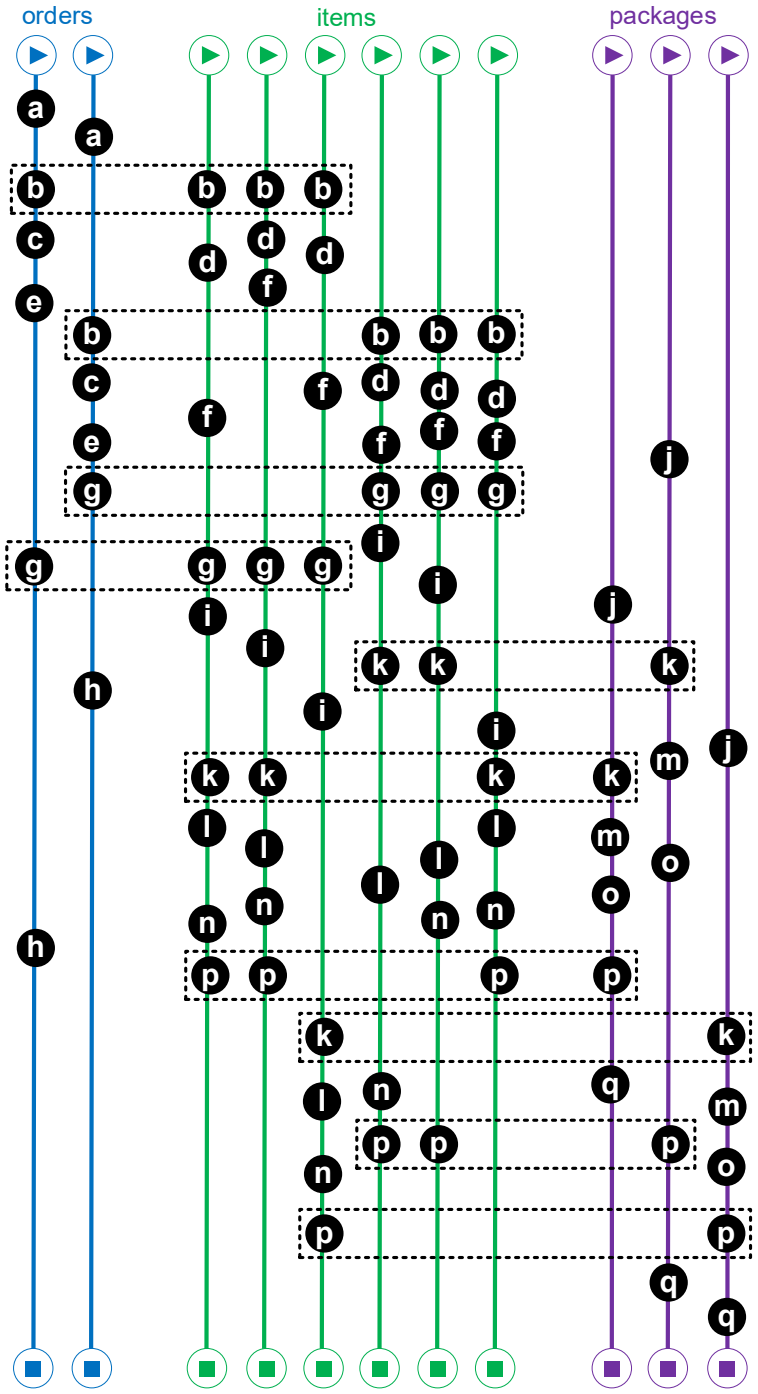
Exhibit #1


orders
items
packages
orders
items
packages
orders
items
packages

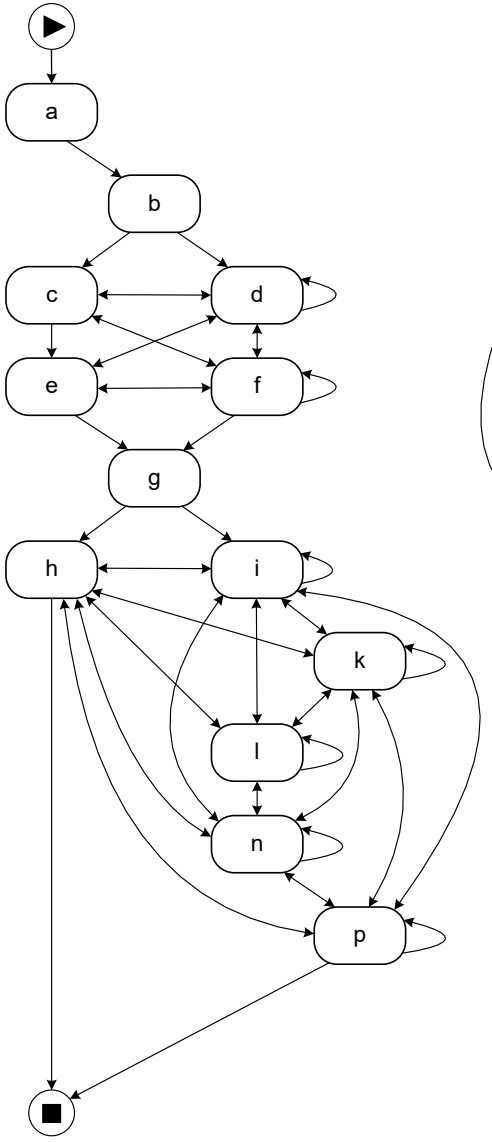

orders
items
packages



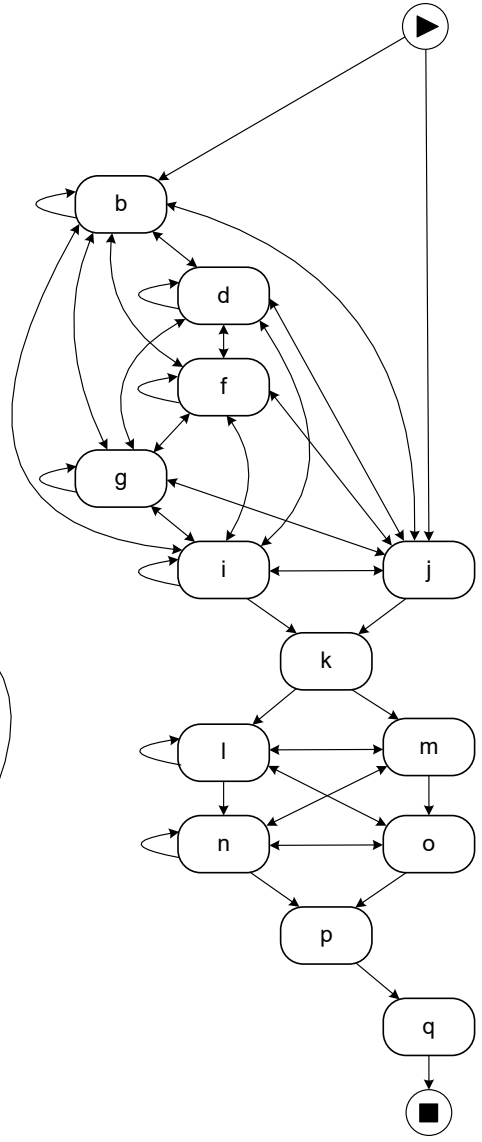

orders
items
packages
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q



orders+items



packages+items



items+orders+packages

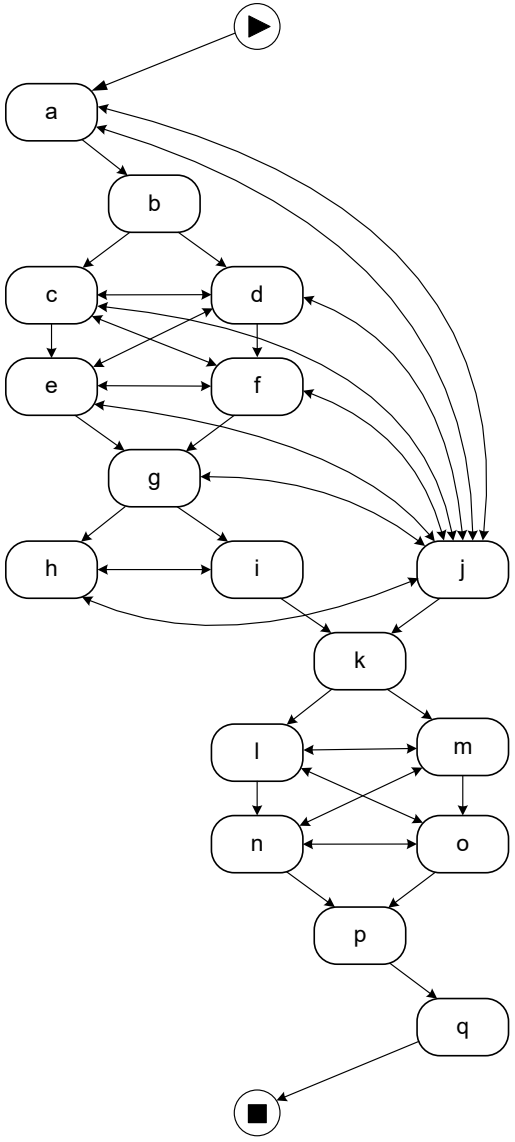
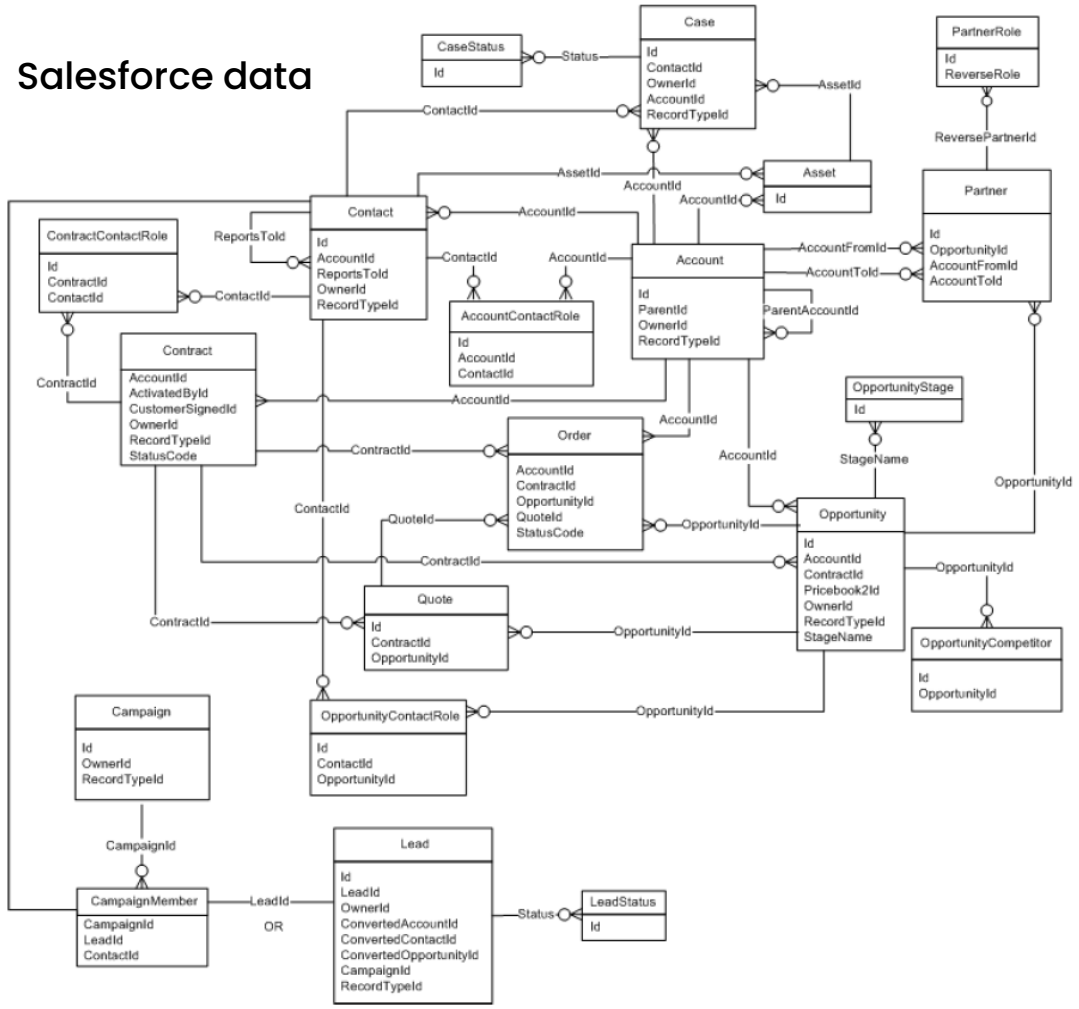
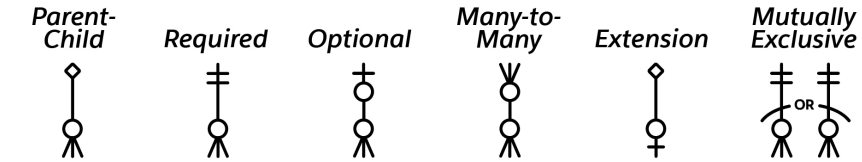


Exhibit #2

Salesforce data



Rarely one-to-one



https://developer.salesforce.com/docs/platform/data-models/

Main benefits of using Object-Centric Process Mining

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)

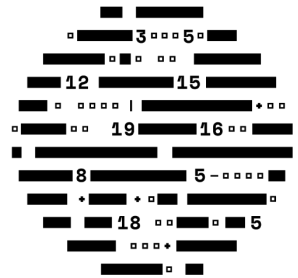
03

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.

To conclude

Process Mining Demystified: How Does it Really Work?



Celosphere 25

Thank you

prof.dr.ir. Wil van der Aalst
www.celonis.com/celosphere