Database Process Mining

Alifah Syamsiyah
Boudewijn F. van Dongen
Wil M.P. van der Aalst

AIS Meeting – 21.09.17

Event Data

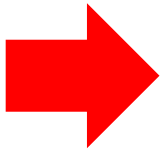
Process Mining

Discovery Results

Provide **instant discovery**

Enable **discovery based on time window**

Store **richer information for alignment**



Instant Discovery

Discovery based on Time Window

Alignment

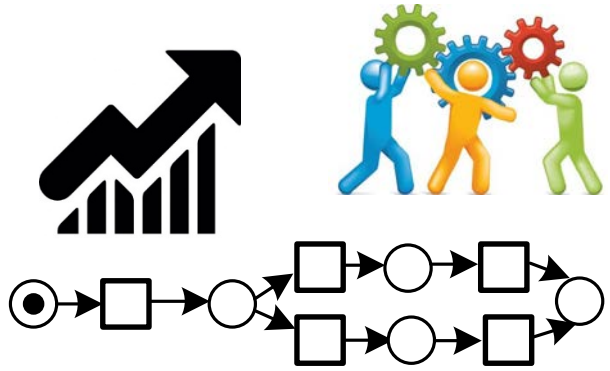
Experimental Results

Conclusion

Loading

Abstraction

Mining



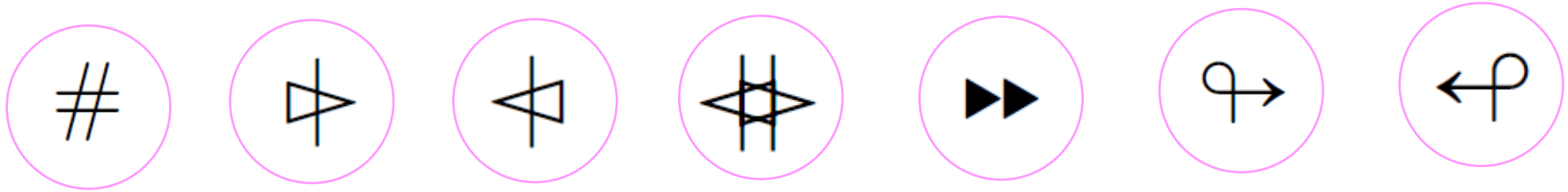
Intermediate Structure

DFR

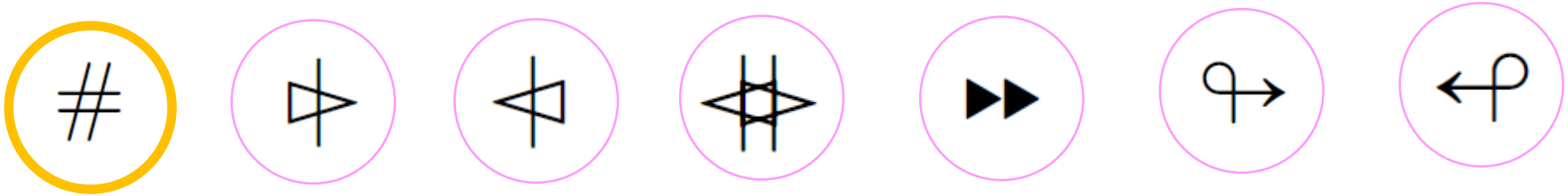
Declare relations

Prefix automaton

Incrementally update the intermediate structure



What are these relations?
How to incrementally updates these relations?

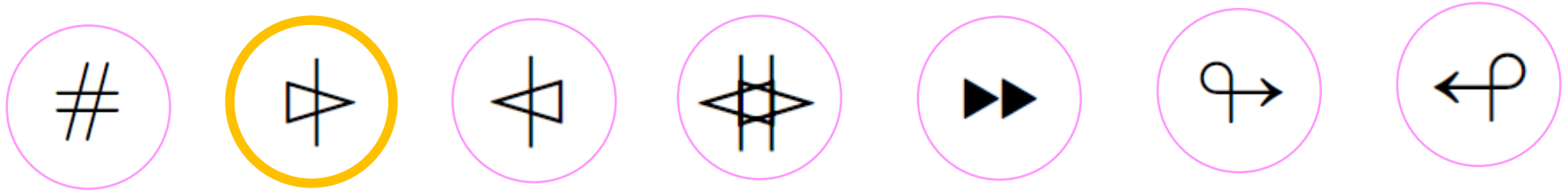


Occurrence $\#(a)$
Counts the occurrences of activity *a*

L = {<a, b, a, a, c, a>,
<a, b, a, a, c, a, d>}

$\#$ (a) = 8

Σ = {a, b, c, d}



No Following $\not\triangleright(a,b)$

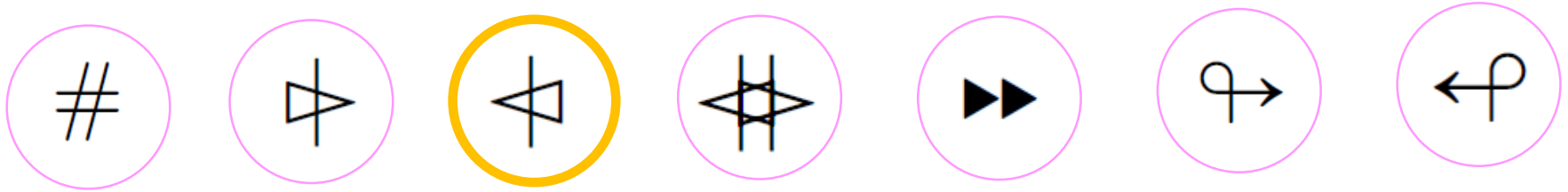
Counts the occurrences of activity *a* with no following *b*

L = {<a, b, a, a, c, a>,
<a, b, a, a, c, a, d>}

$\not\triangleright$ (a,d) = 4

$\not\triangleright$ (a,c) = 2

Σ = {a, b, c, d}



No Preceding $\not\triangleleft(a,b)$

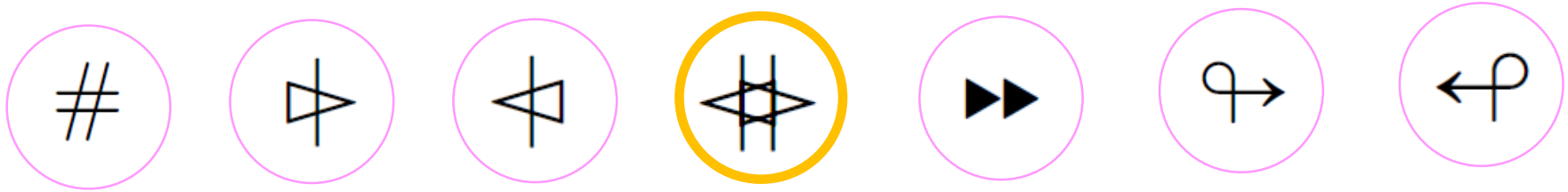
Counts the occurrences of activity *a* with no preceding *b*

L = {<a, b, a, a, c, a>,
<a, b, a, a, c, a, d>}

$\not\triangleleft$(a,b) = 2

$\not\triangleleft$(a,c) = 6

Σ = {a, b, c, d}



No Co-Occurring (*a*,*b*)

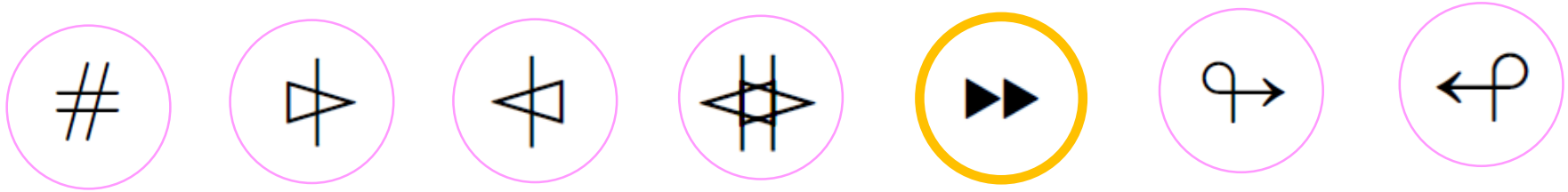
Counts the occurrences of activity *a* with no co-occurring *b*

L = {<a, b, a, a, c, a>,
<a, b, a, a, c, a, d>}

(a,c) = 0

(a,d) = 4

Σ = {a, b, c, d}



Directly Follows Relation ▸▸ (a,b)

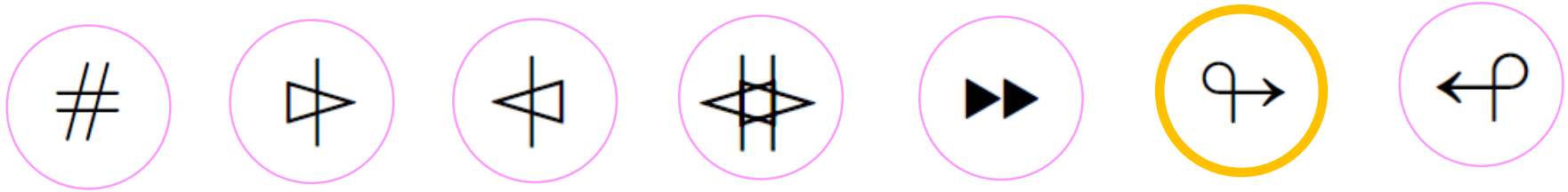
Counts the number of times activity *a* is directly followed by activity *b*

L = {<a, b, a, a, c, a>,
<a, b, a, a, c, a, d>}

Σ = {a, b, c, d}

▸▸ (a,b) = 2

▸▸ (a,c) = 2



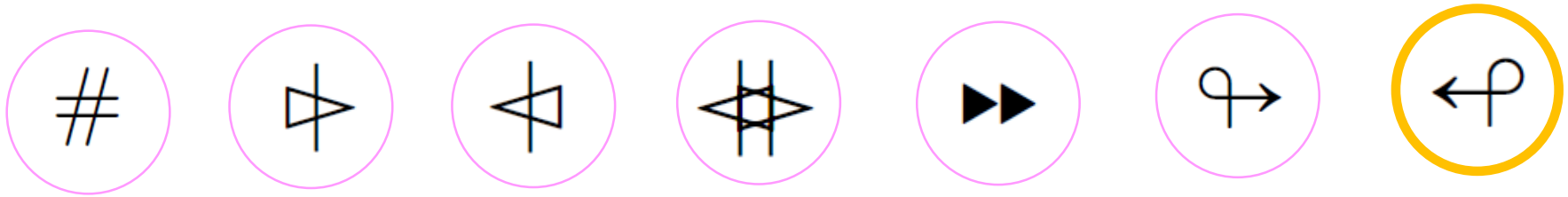
Repetition $\curvearrowright(a,b)$

Counts how many times after an occurrence of *a*, *a* repeats until the first occurrence of *b*. If no *b* occurs after *a*, then the repetitions after *a* are not counted.

L = {<a, b, a, a, c, a>,
<a, b, a, a, c, a, d>}

Σ = {a, b, c, d}

$\curvearrowright(a,c) = 4$
$\curvearrowright(a,b) = 0$
$\curvearrowright(a,d) = 3$



Repetition (Backwards) $\overleftarrow{\rho}(a,b)$

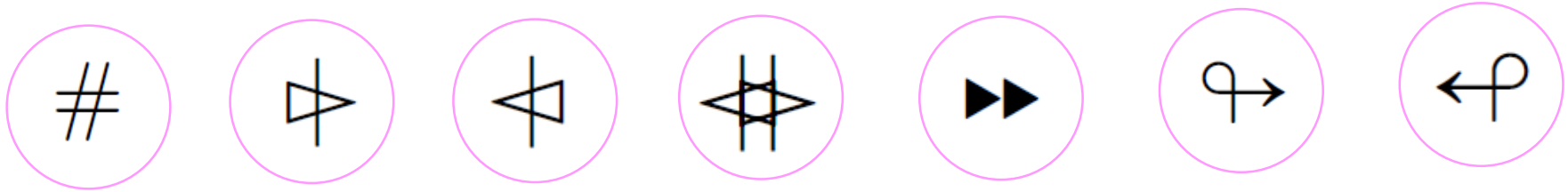
Is similar to $\rho\rightarrow$ but reading the trace backwards

L = {<a, b, a, a, c, a>,
<a, b, a, a, c, a, d>}

$\overleftarrow{\rho}(a,c) = 0$

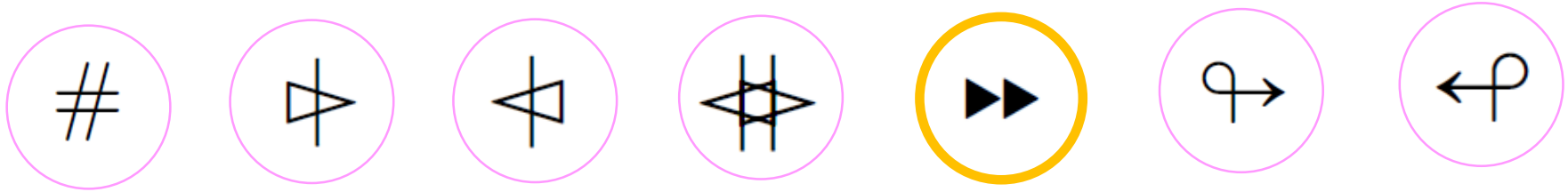
$\overleftarrow{\rho}(a,b) = 4$

Σ = {a, b, c, d}



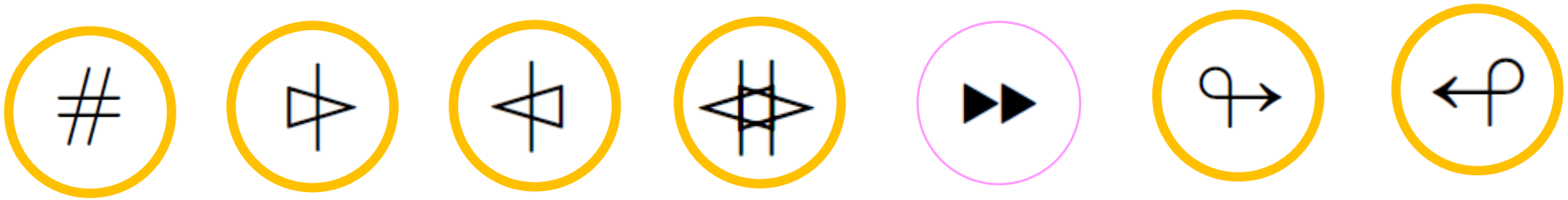
What are these relations?

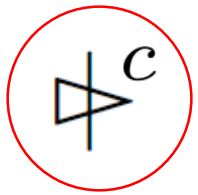
How to incrementally updates these relations?



L = { <a, b, a, a, c, a > }

Activity 1	Activity 2	Frequency
a	b	1
b	a	1
a	a	1
a	c	1
c	a	1

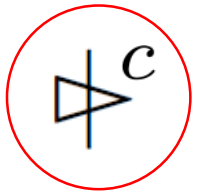




Controller function  (a,b)

If $a \neq b$: counts the occurrences of activity a with no following b in trace t.

If $a = b$: counts the occurrence of a in trace t.



Σ = {a, b, c, d}

For a new activity x:
- ***add** 1 to the row of x*
- ***reset** the column of x except (x,x)*

	a	b	c	d
a	I I I I	I I I	I I I	I I I I
b	I	I	I	I
c	I	I	I	I
d	I	I	I	I

L = { <a, b, a, a, c, a, d> }

Occurrence

(a) = 4 # (c) = 1

(b) = 1 # (d) = 1

$\not\vdash^{c}$

	a	b	c	d
a	I I I I	I I I	I	
b		I		
c		I	I	
d	I	I	I	I

L = { <a, b, a, a, c, a, d> }

No Following

$\not\triangleright$ (a,b) = 2

$\not\triangleright$ (a,c) = 0

$\not\triangleright^{c}$

	a	b	c	d
a	I I I I	I I I	I	
b		I		
c		I	I	
d	I	I	I	I

L = { <a, b, a, a, c, a, d> }

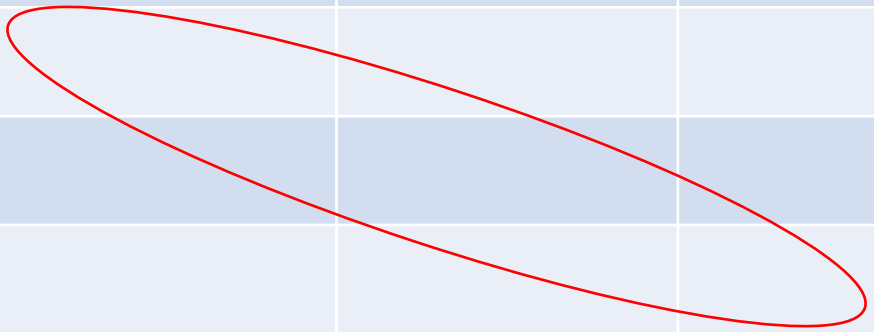
No Preceding

$\nvdash_{L'}$ (a,b) = 1 $\nvdash_{L'}$ (a,d) = 1

$\nvdash_{L'}$ (a,c) = 1

$\nvdash^{c}$

	a	b	c	d
a	l	l	l	l
b				
c				
d				



L = { <a, > }

No Co-Occurrence

$\nmid_{L'}$ (a,b) = 0 $\nmid_{L'}$ (a,d) = 1

$\nmid_{L'}$ (a,c) = 1

$\nmid^{c}$

	a	b	c	d
a	I	I	I	I
b	I	I	I	I
c				
d				

L = { <a, b, > }

Repetition

$\not\rightarrow_{L'}$(a,c) = 2

3 occurrence of a with no following c

=

After an occurrence of a, a repeats twice until the first occurrence of c

$\not\triangleright^{c}$

	a	b	c	d
a	I I I	I I	I I I	I I I
b		I	I	I
c	I	I	I	I
d				

L = { <a, b, a, a, c, > }

Repetition (Backwards)

$\overleftarrow{\rho}_{L'}$ (a,b) = 2

First occurrence

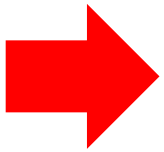
1st repetition

2nd repetition

$\not\vdash^{c}$

	a	b	c	d
a	I I I I	I I I	I I I	I I I I
b	I	I	I	I
c	I	I	I	I
d				

L = { <a, b, a, a, c, a, > }



Instant Discovery

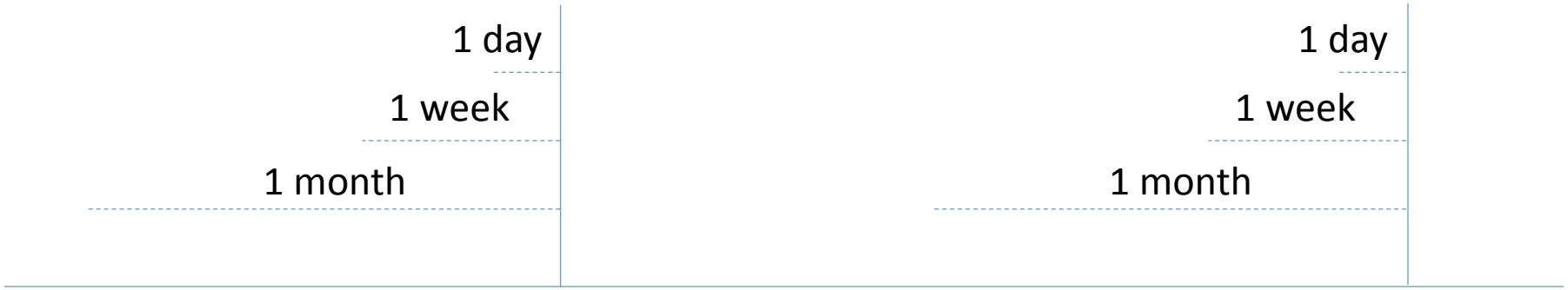
Discovery based on Time Window

Alignment

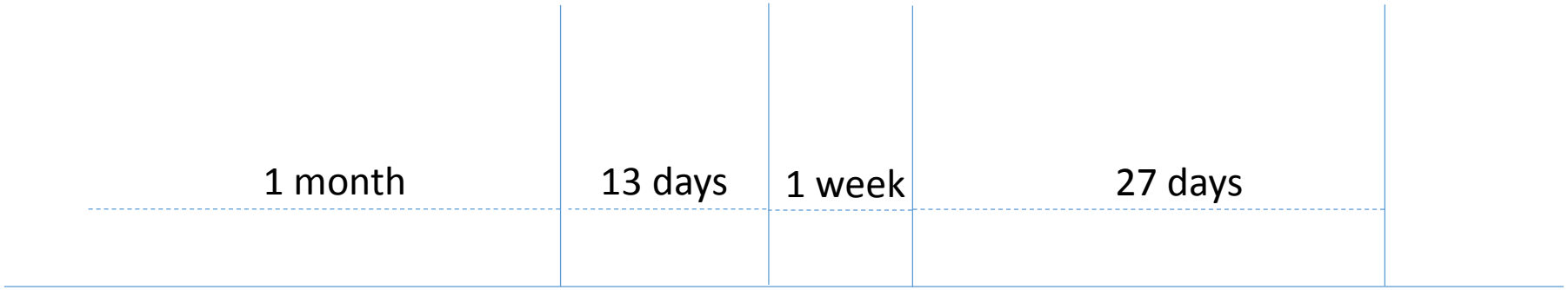
Experimental Results

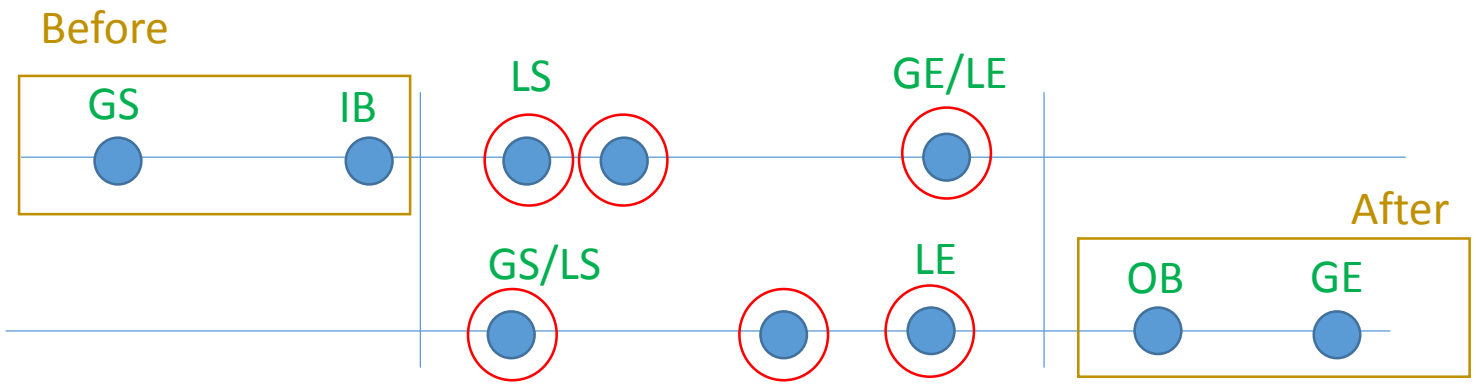
Conclusion

Sliding Window



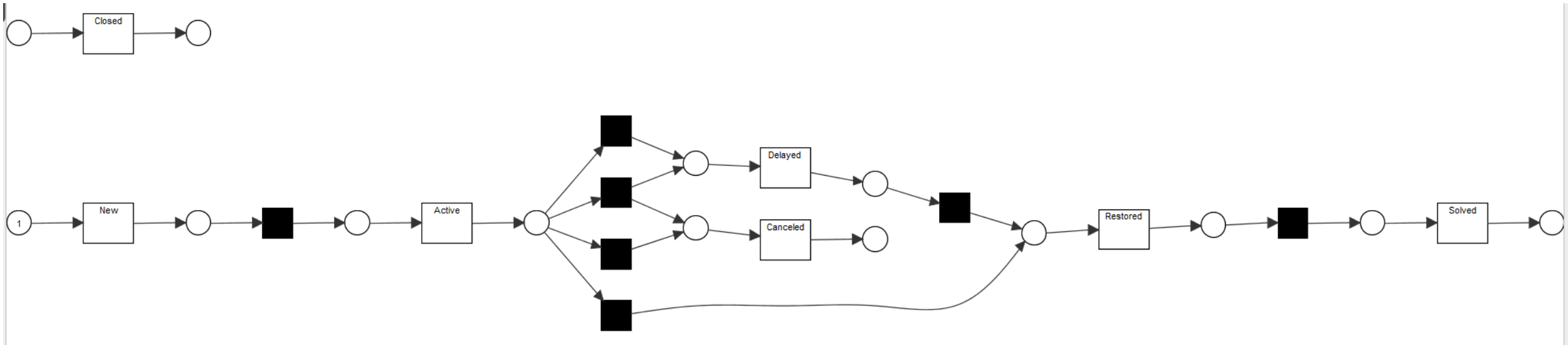
Chained Window





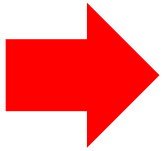
O = event always included
IB = input boundary events
OB = output boundary events
GS = global start events
GE = global end events
LS = local start events
LE = local end events

Telefonica Process Model of the last 11 days



Instant Discovery

Discovery based on Time Window



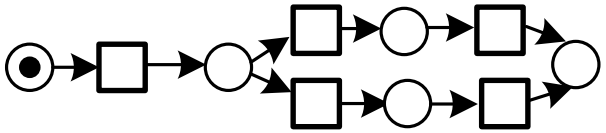
Alignment

Experimental Results

Conclusion

Types of Alignment

	Optimal Alignment	Prefix Alignment	Partial Alignment
Initial Marking	Fixed for all traces	Fixed for all traces	Different from one trace to the others
Final Marking	Fixed for all traces	Different from one trace to the others	Different from one trace to the others



a	b c d	e
a b	d	f
	a d	c e

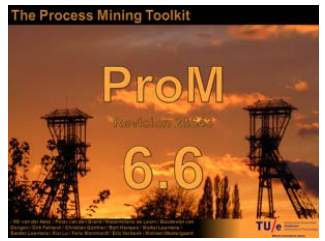
Motivation

- The existing technique is not able to give reliable fitness value for partial traces. Incomplete traces, for example, are treated as deviations.
- In a steady state process, we want to get stable fitness values in any window time
- In a big log, we want to have fitness value without taking into account all data
- We want to get an alarm as soon as a deviation happens (don’t need to wait until a process finishes)

Store events, marking,
and last sequence

Sequence of events
Last marking

Current marking

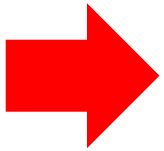


Calculate fitness

Instant Discovery

Discovery based on Time Window

Alignment



Experimental Results

Conclusion

Experimental Setup

BPI Challenge 2017 (31509 traces, 1202267 events)

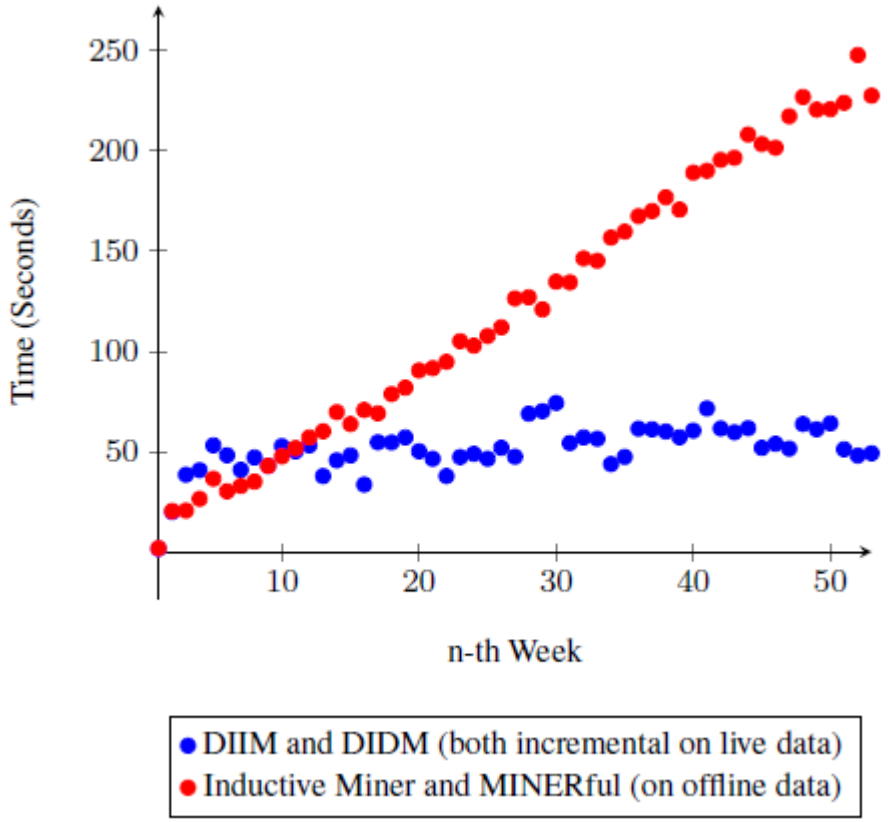
Weekly report on the last working day (every Friday)

In each report, both procedural and declarative models are mined

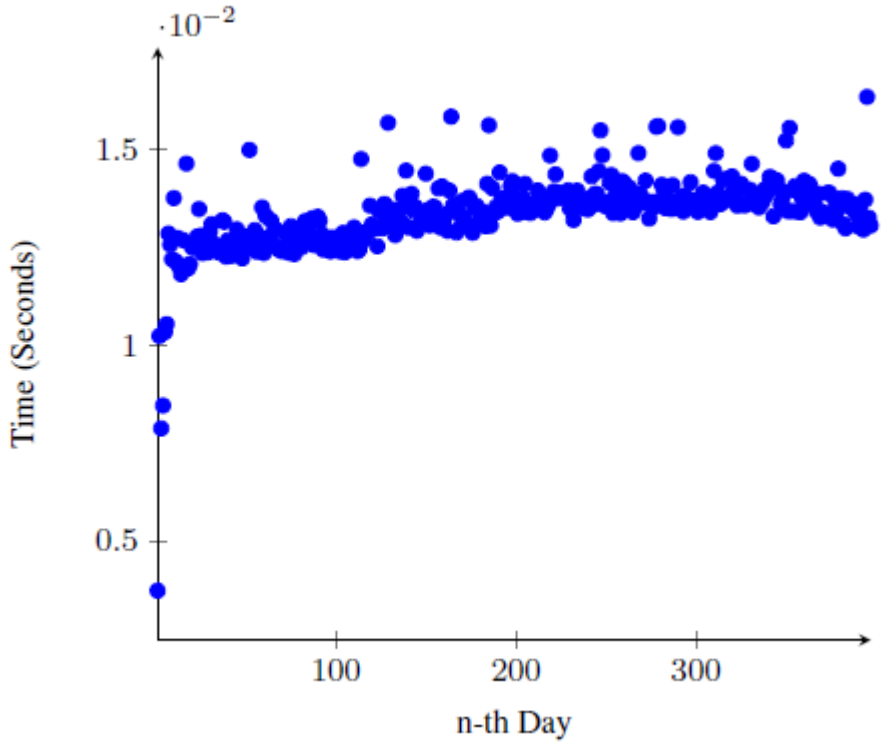
Implemented proposed approach:

- DIIM (Database-Incremental Inductive Miner)
- DIDM (Database-Incremental Declare Miner)

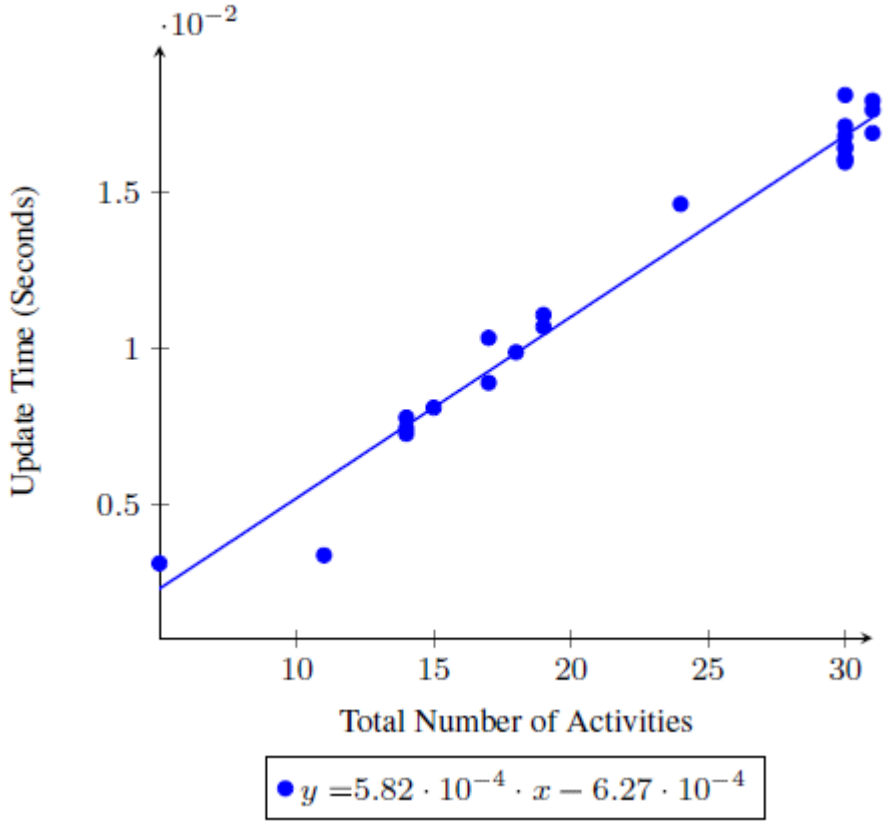
Recurrent Process Discovery Using
DIIM and DIDM vs Traditional Inductive Miner and MINERful



Average Update Per Event



The Influences of Number of Activities to Update Time



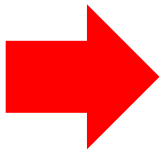
Instant Discovery

Discovery based on Time Window

Alignment

Experimental Results

Conclusion



- We can reduce analysis time by storing and computing intermediate structures in a persistent storage (e.g. database) and in an incremental way.
- We showed that our incremental implementations is constant for updating per event, while the retrieval and mining times are independent of the size of the underlying data.
- The use of database in process mining enables discovery based on time window and stores richer information for alignment.

Future Works

- Updating the intermediate structures in batches of events
- Keep the intermediate structures live under removal of events.