

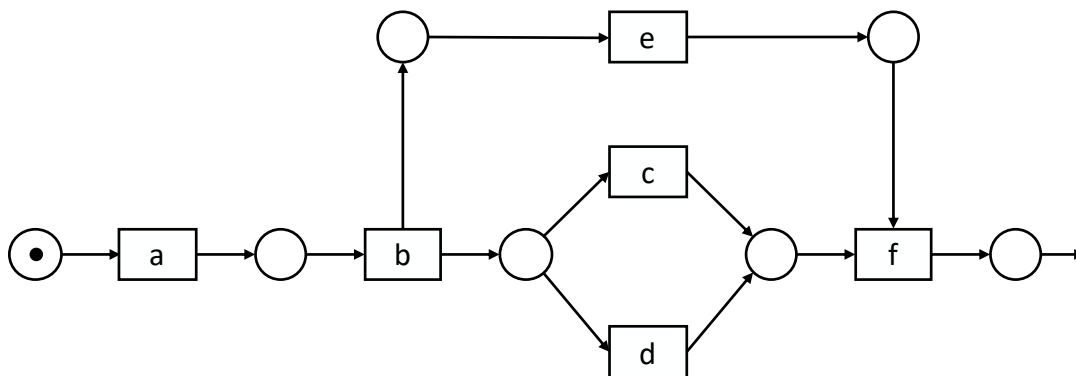
# Advanced Process Mining

Summer term 2020

## Exercise sheet 3

DFG Limitations • DFG Soundness • DFG to PN

### Exercise 1: Limitations of the Directly-Follows Graph



#	Trace
100	abcef
90	abecf
87	abdef
85	abedf
3	abef
1	abcdef

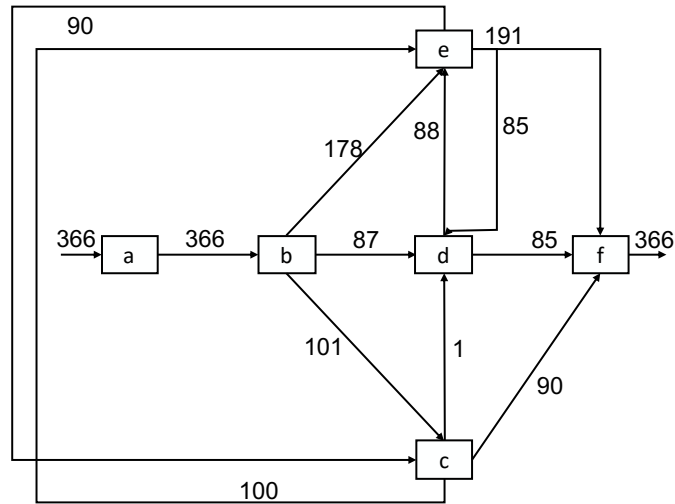
- Draw a Directly-Follows Graph for the event log above.
- Compare the Directly-Follows Graph from a) with the Petri net above. Does the DFG allow for behaviour that is not allowed according to the Petri net?
- Explain the thresholds  $\tau_{var}$ ,  $\tau_{act}$  and  $\tau_{df}$ .

**Solution**

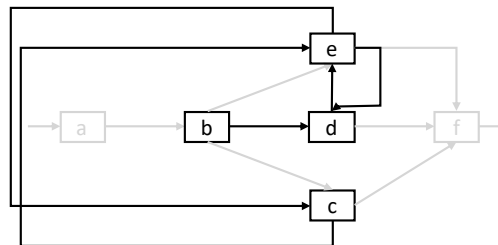
a) The Directly-Follows Matrix looks as follows:

$>_L$	a	b	c	d	e	f	E
S	366						
a		366					
b			101	87	178		
c				1	100	90	
d					88	85	
e			90	85		191	
f							366

The corresponding Directly-Follows Graph will look like this:



b) The Directly-Follows Graph allows for a loop involving activities c, d and e. The Petri net from which the event log was created does not permit any loops involving activities c and d. According to the Petri net there is an exclusive choice between c and d.



- c)  $\tau_{var}$  Defines the thresholds for the minimal number of traces for each variant included (based on  $\#_L(\sigma)$ ).
- $\tau_{act}$  Defines the minimal number of events for each activity included (based on  $\#_L(a)$ ).
- $\tau_{df}$  Defines the minimal number of direct successions for each relation included (based on  $\#_L(a, b)$ ).

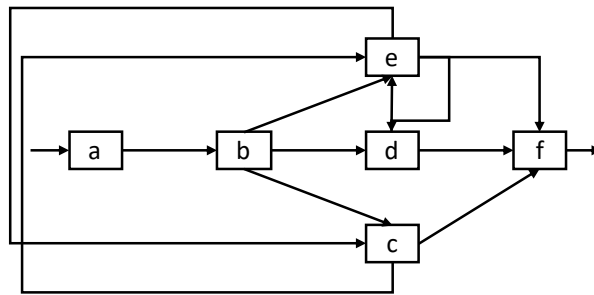
## Exercise 2: Soundness in Directly-Follows Graphs

- Define soundness in DFG in a non-formal way.
- How can a sound DFG be turned into an unsound DFG?

### Solution

- A DFG with  $N$  (the set of nodes) and  $E$  (the set of edges) is sound if every node  $x \in N$  is on a path from start to end.
- To reduce the complexity of a DFG, edges might be filtered out, possibly resulting in deadlocks and an unsound Graph.

## Exercise 3: Transformation from DFG to PN



Transform the DFG above into a Petri net. Model each transition as a combination of start and end event.

### Solution

The transformed Petri net will look like this:

