C A U

Christian-Albrechts-Universität zu Kiel

Advanced Process Mining

Sommer term 2020

Exercise sheet 2

4D of Quality • DFG • Heuristic Miner

Four Dimensions of Quality

Christian-Albrechts-Universität zu Kiel

What are the four dimensions of Quality?

Recall • Fitness

Precision

Generalisation

Simplicity • Complexity

Four Dimensions of Quality
Recall • Fitness

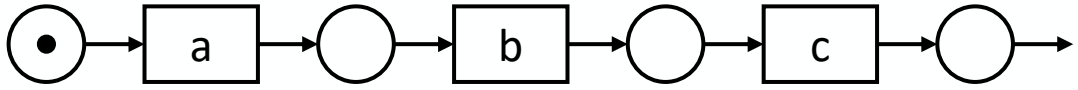
Christian-Albrechts-Universität zu Kiel

What is measured by recall?

- Behaviour recorded in the event log reproducible by the model
- The behaviour seen in the event log should be allowed by the discovered model

Does this example have good recall?

- The example has a bad fitness
- Behaviour recorded in the event log is not reproducible by the model



Looking at the extremes:

- Fitness = 1
  - Every trace in the event log is represented by the process model
- Fitness = 0
  - None of the recorded traces can be found in the process model

#	Trace
20	abc
20	ac
10	acb

Four Dimensions of Quality
Precision

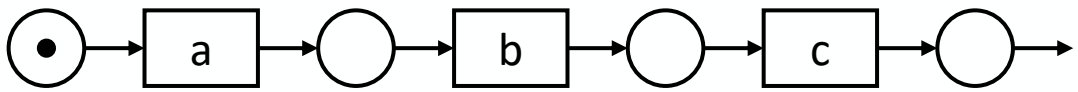
Christian-Albrechts-Universität zu Kiel

What is measured by precision?

- Behaviour producible by the model observed in the event log
- The discovered model should not permit behaviour unrelated to what has been recorded in the event log

Does this example have good precision?

- The example has a an excellent precision
- All processes producible by the process model $(< a, b, c >)$ are also recorded in the event log
- The discovered model does not permit behaviour that has not been recorded in the event log



#	Trace
20	abc
20	ac
10	acb

Looking at the extremes:

- Precision = 1
  - For every possible producible trace by the process model, exists an entry in the event log
- Precision = 0
  - None of the producible traces by the process model, exist in the event log

Four Dimensions of Quality
Generalisation

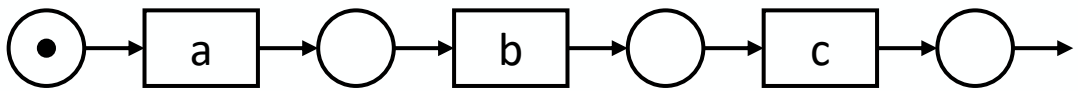
Christian-Albrechts-Universität zu Kiel

What is measured by generalisation?

- The example behaviour seen in the event log should be generalised in the discovered process model
- It assesses the extent to which the process model will be able to reproduce unseen behaviour of the process

Does this example generalise?

- No, the process model can produce only exactly one trace.



#	Trace
20	abc
20	ac
10	acb

Looking at the extremes:

- Generalisation = 1
  - The process model generalises very well
  - Unseen behaviours can be reproduced by the model
- Generalisation = 0
  - No generalisation
  - No unseen behaviours can be reproduced by the model

Four Dimensions of Quality

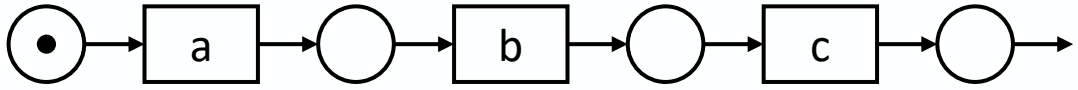
Simplicity • Complexity

What is measured by simplicity?

- The best model is the simplest model that can explain the behaviour seen in the log
- The discovered model should be as simple as possible
- The complexity of the model could be defined by the number of nodes and arcs in the underlying graph

Is this example simple?

- Yes, the process model is very simple



#	Trace
20	abc
20	ac
10	acb

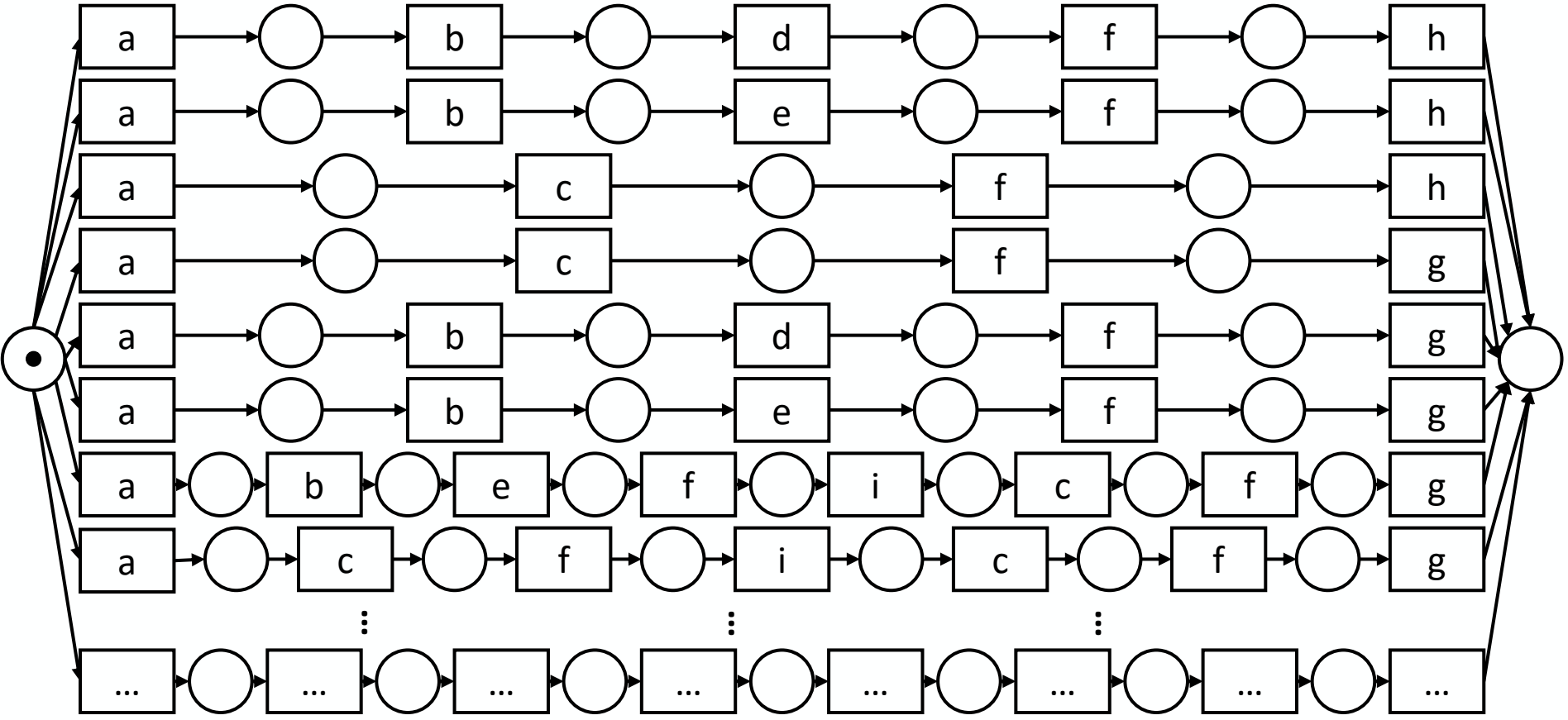
Four Dimensions of Quality

Exercise 1a

Christian-Albrechts-Universität zu Kiel

Design a process model for the event log with the following properties:

Fitness:	high	Precision:	high
Generalisation:	low	Simplicity:	low



#	Trace
342	abdfh
200	abefh
101	acfh
62	acfg
55	abdfg
17	abefg
16	abeficfg
13	acficfg
8	abefibdfibcfh
7	acficficficficfg

Four Dimensions of Quality

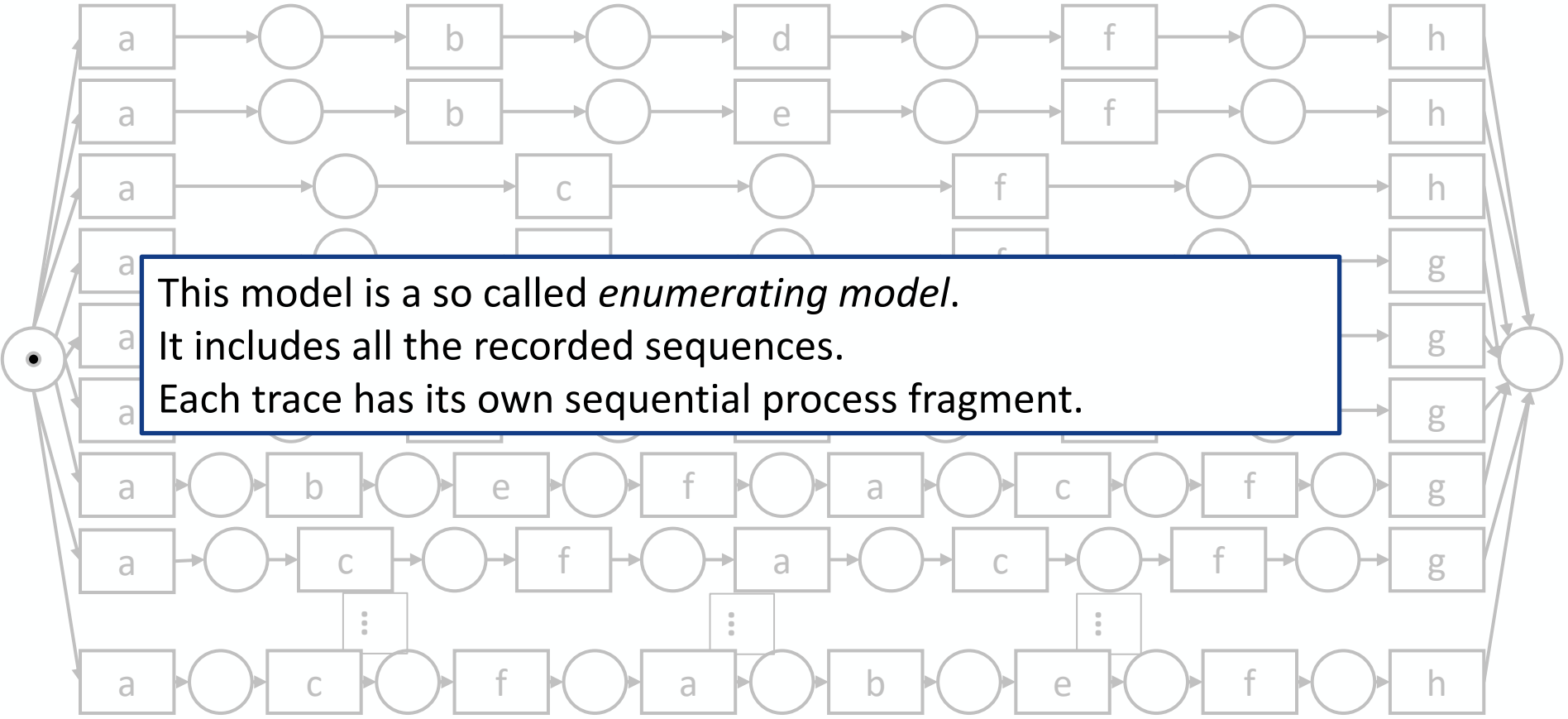
Exercise 1a

Christian-Albrechts-Universität zu Kiel

Design a process model for the event log with the following properties:

Fitness:	high	Precision:	high
Generalisation:	low	Simplicity:	low

#	Trace
342	abdfh
200	abefh
101	acfh
62	acfg
55	abdfg
17	abefg
16	abeficfg
13	acficfg
8	abefibdfibcfh
7	acficficficficfg



This model is a so called *enumerating model*.
It includes all the recorded sequences.
Each trace has its own sequential process fragment.

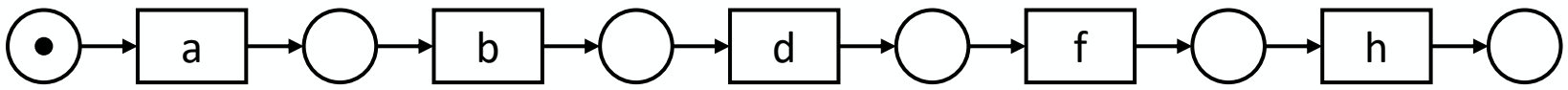
Four Dimensions of Quality

Exercise 1b

Christian-Albrechts-Universität zu Kiel

Design a process model for the event log with the following properties:

Fitness:	low	Precision:	high
Generalisation:	low	Simplicity:	high



This model only covers the most frequent trace. None of the other recorded traces are represented by this model.

#	Trace
342	abdfh
200	abefh
101	acfh
62	acfg
55	abdfg
17	abefg
16	abeficfg
13	acficfg
8	abefibdfibcfh
7	acficficficficfg

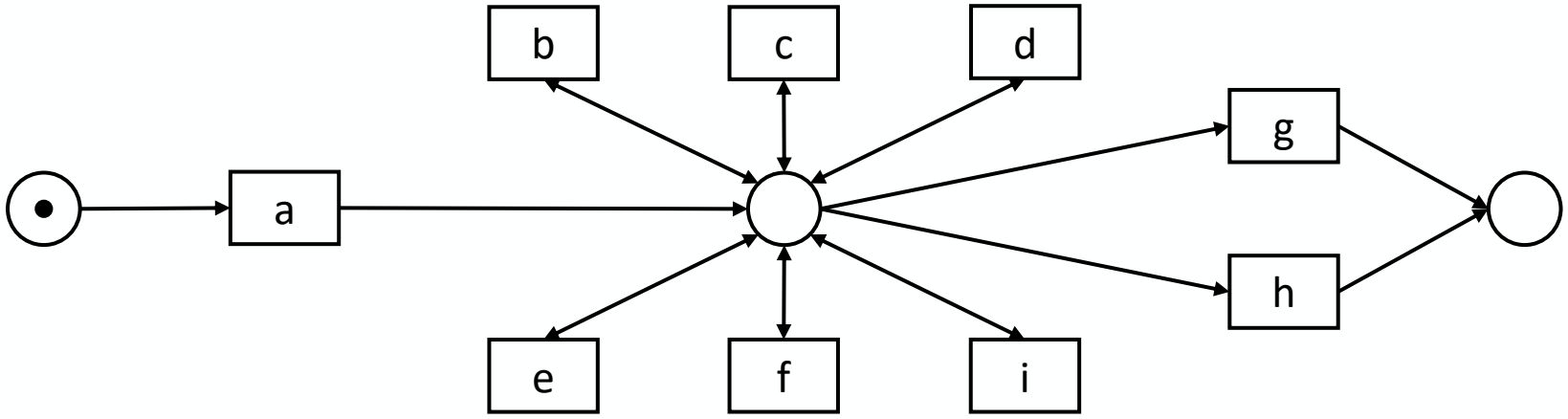
Four Dimensions of Quality

Exercise 1c

Christian-Albrechts-Universität zu Kiel

Design a process model for the event log with the following properties:

Fitness:	high	Precision:	low
Generalisation:	high	Simplicity:	high



#	Trace
342	abdfh
200	abefh
101	acfh
62	acfg
55	abdfg
17	abefg
16	abeficfg
13	acficfg
8	abefibdfibcfh
7	acficficficficfg

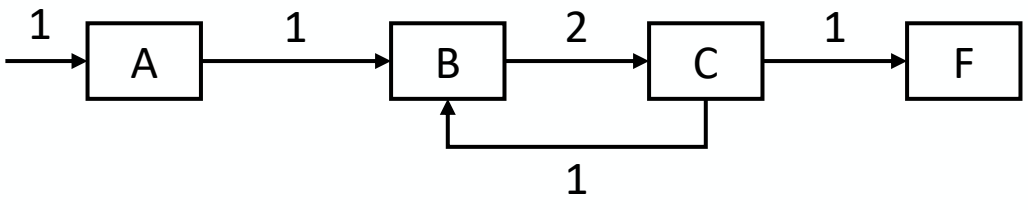
The fitness is perfect. since all recorded traces can be replayed by the model. However the precision is very bad. because this model allows much more behaviour than was recorded.
The model shown above is a variant of the so called *flower model*.

Directly-Follows Graph

- Graphical representation of the recorded traces in the event log
- Depicts the directly follows relations between activities
- A Directly-Follows graph is not a very precise model, since it allows for more behaviour than was recorded in the log
- Pre-processing (filtering) of the event log might lead to better results

Draw a Directly-Follows graph for this sequence of activities:

ABCBCF



Directly-Follows Graph

Exercise 2a

Christian-Albrechts-Universität zu Kiel

What appropriate filtering criteria can be applied on this log?

- Remove all events that have more than 5 activities
- Remove all events that do not start with activity ‘b’
- Remove all traces that occurred less than 50 times

#	Trace
342	abdfh
200	abefh
101	acfh
62	acfg
55	abdfg
17	abefg
16	abeficfg
13	acficfg
8	abefibdfibcfh
7	acficficficficfg
1	bcdh

Directly-Follows Graph

Exercise 2b

Christian-Albrechts-Universität zu Kiel

Given the event log. create a directly follows graph for traces observed:

#	Trace
342	abdfh
200	abefh
101	acfh
62	acfg
55	abdfg
17	abefg
16	abeficfg
13	acficfg
8	abefibdfibcfh
7	acficficficficfg
1	bcdh

1. Create a table with pairs of consecutive activities and their quantity

$>_L$	a	b	c	d	e	f	g	h	i
a		638	183						
b			9	405	241				
c				1		248			
d						405		1	
e						241			
f							170	651	73
g									
h									
i		16	57						

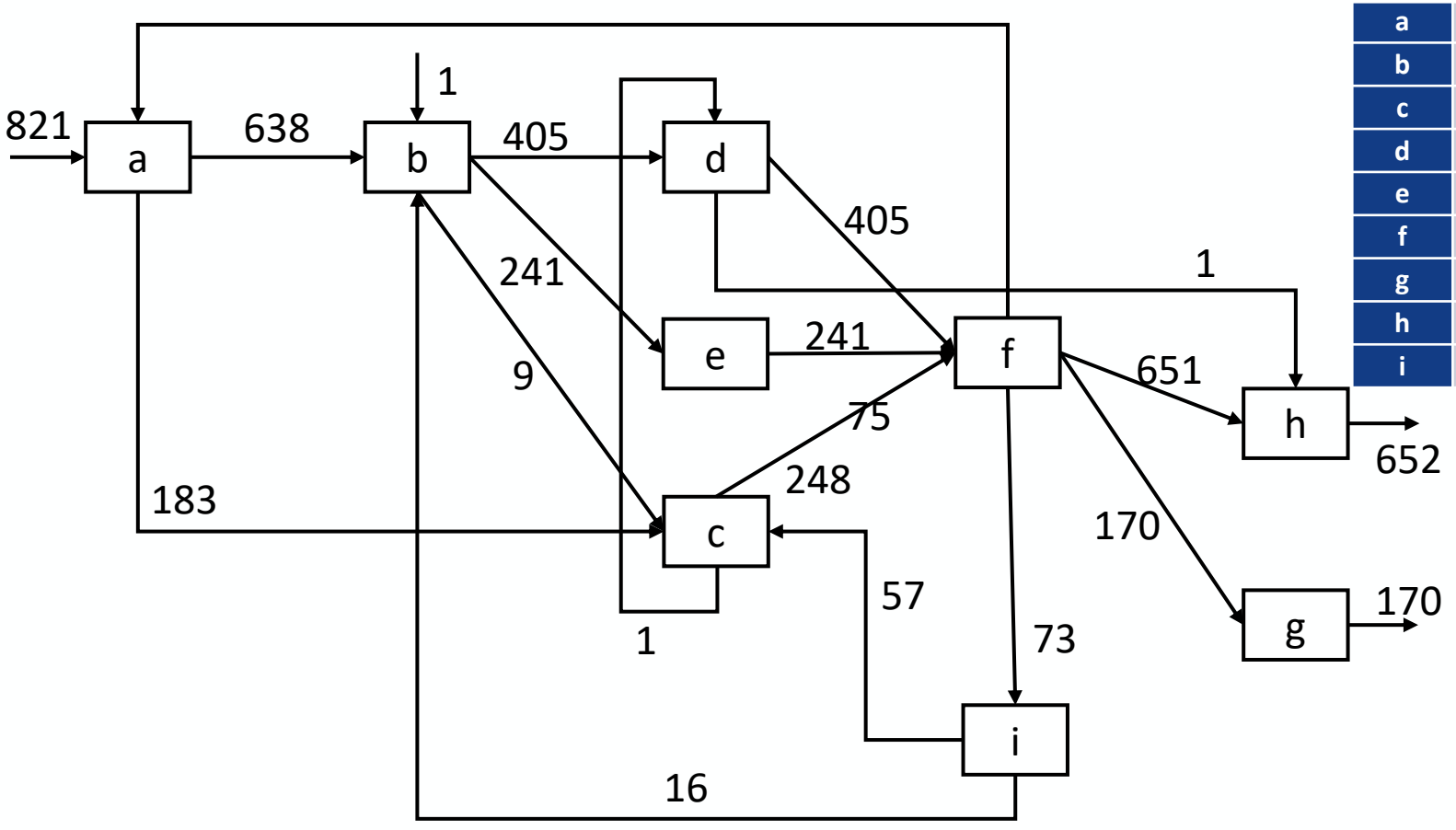
Directly-Follows Graph

Exercise 2b

Christian-Albrechts-Universität zu Kiel

2. Draw the Directly-Follows Graph from the collected information in the table:

	a	b	c	d	e	f	g	h	i
a		638	183						
b			9	405	241				
c				1		248			
d						405		1	
e						241			
f							170	651	73
g									
h									
i		16	57						



Heuristic Miner

- More frequent sequences of events have a higher influence on the discovery of the process model
- Can discover loops and skipping activities

1. Construct Directly-Follows Matrix

2. Construct dependency matrix

How to calculate the dependency relation between two activities:

$$a \Rightarrow_L b = \begin{cases} \dfrac{|a >_L b| - |b >_L a|}{|a >_L b| + |b >_L a| + 1} & , if\ a \neq b \\ \dfrac{|a >_L a|}{|a >_L a| + 1} & , else \end{cases}$$

If $a \Rightarrow_L b$ is close to 1, then a strong positive dependency exists.
a is often the cause of b

If $a \Rightarrow_L b$ is close to -1, then a strong negative dependency exists.
b is often the cause of a

If $a \Rightarrow_L b$ is close to 0, then a and b may be parallel

3. Filter & Define threshold

4. Construct Petri net

Heuristic Miner
Exercise 3

Christian-Albrechts-Universität zu Kiel

Discover a process model from the given event log. Use the Heuristic Miner.

1. Construct Directly-Follows Matrix
2. Construct dependency matrix
3. Filter & Define threshold
4. Construct Petri net

#	Trace
342	abdfh
200	abefh
101	acfh
62	acfg
55	abdfg
17	abefg
16	abeficfg
13	acficfg
8	abefibdfibcfh
7	acficficficficfg
1	bcdh

Heuristic Miner
Exercise 3

Christian-Albrechts-Universität zu Kiel

1. Construct the Directly-Follows Matrix

$>_L$	a	b	c	d	e	f	g	h	i
a		638	183						
b			9	405	241				
c				1		248			
d						405		1	
e						241			
f							170	651	73
g									
h									
i		16	57						

#	Trace
342	abdfh
200	abefh
101	acfh
62	acfg
55	abdfg
17	abefg
16	abeficfg
13	acficfg
8	abefibdfibcfh
7	acficficficficfg
1	bcdh

Heuristic Miner
Exercise 3

Christian-Albrechts-Universität zu Kiel

2. Construct the Dependency Matrix

$$a \Rightarrow_L b = \begin{cases} \dfrac{|a >_L b| - |b >_L a|}{|a >_L b| + |b >_L a| + 1} & ,if\ a \neq b \\ \dfrac{|a >_L a|}{|a >_L a| + 1} & ,else \end{cases}$$

$\Rightarrow_L$	a	b	c	d	e	f	g	h	i
a		0.998	0.995						
b	-0.998		0.900	0.998	0.996				-0.941
c	-0.995	-0.900		0.500		0.996			-0.983
d		-0.998	-0.500			0.998		0.500	
e		-0.996				0.996			
f			-0.996	-0.998	-0.996		0.994	0.998	0.986
g						-0.994			
h				-0.500		-0.998			
i		0.941	0.983			-0.986			

$>_L$	a	b	c	d	e	f	g	h	i
a		638	183						
b			9	405	241				
c				1		248			
d						405		1	
e						241			
f							170	651	73
g									
h									
i		16	57						

Heuristic Miner
Exercise 3

3. Define a threshold and filter the dependency matrix:

$$|\Rightarrow_L| > \mathbf{0.95}$$

$\Rightarrow_L$	a	b	c	d	e	f	g	h	i
a		0.998	0.995						
b	-0.998		0.900	0.998	0.996				-0.941
c	-0.995	-0.900		0.500		0.996			-0.983
d		-0.998	-0.500			0.998		0.500	
e		-0.996				0.996			
f			-0.996	-0.998	-0.996		0.994	0.998	0.986
g						-0.994			
h				-0.500		-0.998			
i		0.941	0.983			-0.986			

Heuristic Miner
Exercise 3

Christian-Albrechts-Universität zu Kiel

4. Construct the Petri net:

$$|\Rightarrow_L| > \mathbf{0.95}$$

$\Rightarrow_L$	a	b	c	d	e	f	g	h	i
a		0.998	0.995						
b	-0.998		0.900	0.998	0.996				-0.941
c	-0.995	-0.900		0.500		0.996			-0.983
d		-0.998	-0.500			0.998		0.500	
e		-0.996				0.996			
f			-0.996	-0.998	-0.996		0.994	0.998	0.986
g						-0.994			
h				-0.500		-0.998			
i		0.941	0.983			-0.986