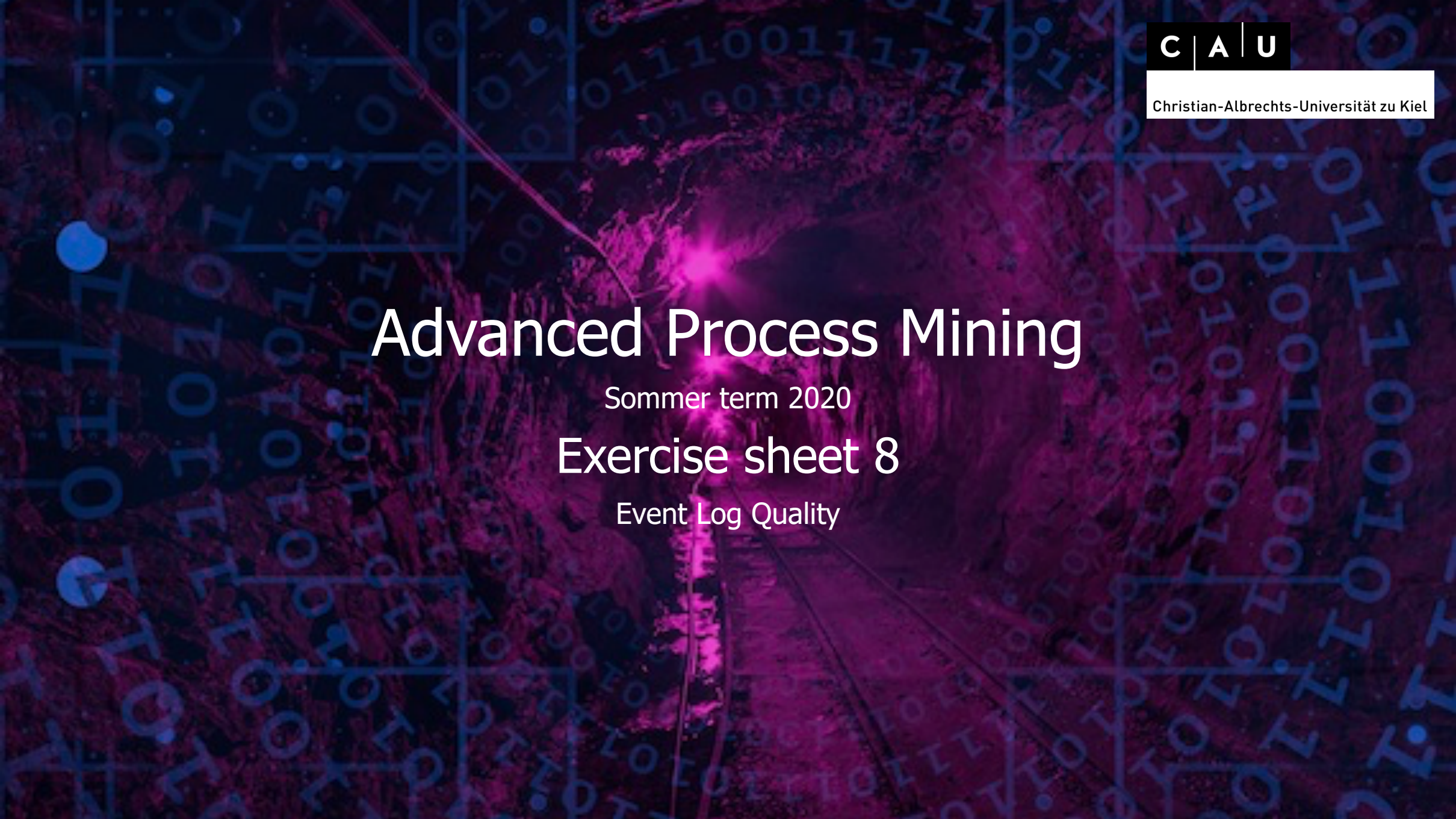


C A U

Christian-Albrechts-Universität zu Kiel

Advanced Process Mining

Sommer term 2020

Exercise sheet 8

Event Log Quality

Noise vs. Outlier

What is Noise?

Christian-Albrechts-Universität zu Kiel

Wrongly recorded data that does not necessarily hint toward an anomaly in the process execution.

Noise vs. Outlier
What are Outliers?

Outliers are anomalies recorded in the process.

Outlier behaviour diverges from the expected process.

Noise vs. Outlier

Exercise 1

Christian-Albrechts-Universität zu Kiel

Explain the difference between noise and outlier using an example.

Outlier

Customer getting billed before placing the actual order.

Interesting anomaly

Noise

Logging system does not account for DST and messes up the time on the day of the time change.

Issue with the correctness of data

Types of Outliers
Exercise 2a

Explain briefly the key aspects of a contextual outlier.

A contextual outlier is an observation that is not to be expected in the underlying context.

Types of Outliers
Exercise 2b

Christian-Albrechts-Universität zu Kiel

Design a (short) event log that contains a contextual outlier.
Explain why the outlier is contextual.

Case ID	Timestamp	Activity
1	01.06.20	Print US shipping label
1	01.06.20	File tax declaration
1	01.06.20	Load package on truck
2	05.06.20	Print EU shipping label
2	05.06.20	Load package on truck
3	06.06.20	Print US shipping label
3	06.06.20	File tax declaration
3	06.06.20	Load package on air-plane

In the following event log, case ID 1 presents a contextual outlier.
If the package is determined to go to the US, road-transportation is not a valid mode of transport.

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Inspect the event log and identify possible event log imperfections.
If possible correct them.

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	01.06.20 - 00:44:53	ship package
2	06.02.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	order received
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	06.02.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 08:33:31	receive order
3	02.06.20 - 09:17:13	take goods out of storage
3	02.06.20 - 12:12:19	place goods in parcel
3	02.06.20 - 12:12:43	weigh package
3	02.06.20 - 12:13:13	seal package
3	02.06.20 - 12:13:32	hand over package
3	02.06.20 - 13:41:32	ship package

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	01.06.20 - 00:44:53	ship package
2	06.02.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	order received
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 09:17:13	take goods out of storage
3	02.06.20 - 12:12:19	place goods in parcel
3	02.06.20 - 12:12:43	weigh package
3	02.06.20 - 12:13:13	seal package
3	02.06.20 - 12:13:32	hand over package
3	02.06.20 - 13:41:32	ship package

Incorrect data: timestamp

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	**01.06.20 - 23:17:19**	**package goods**
1	**01.06.20 - 00:44:53**	**ship package**
2	06.02.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	order received
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 09:17:13	take goods out of storage
3	02.06.20 - 12:12:19	place goods in parcel
3	02.06.20 - 12:12:43	weigh package
3	02.06.20 - 12:13:13	seal package
3	02.06.20 - 12:13:32	hand over package
3	02.06.20 - 13:41:32	ship package

Incorrect data: timestamp

In case 1 a wrong timestamp is used. The shipping of the package takes place hours before the order was received.

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	**01.06.20 - 23:17:19**	**package goods**
1	**02.06.20 - 00:44:53**	**ship package**
2	06.02.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	order received
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 09:17:13	take goods out of storage
3	02.06.20 - 12:12:19	place goods in parcel
3	02.06.20 - 12:12:43	weigh package
3	02.06.20 - 12:13:13	seal package
3	02.06.20 - 12:13:32	hand over package
3	02.06.20 - 13:41:32	ship package

Incorrect data: timestamp

In case 1 a wrong timestamp is used. The shipping of the package takes place hours before the order was received.

The correct time stamp for the *ship package* activity should be 02.06.20 - 00:44:53.

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	02.06.20 - 00:44:53	ship package
2	06.02.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	order received
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 09:17:13	take goods out of storage
3	02.06.20 - 12:12:19	place goods in parcel
3	02.06.20 - 12:12:43	weigh package
3	02.06.20 - 12:13:13	seal package
3	02.06.20 - 12:13:32	hand over package
3	02.06.20 - 13:41:32	ship package

Imprecise data: timestamp

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	02.06.20 - 00:44:53	ship package
2	**06.02.20 - 06:20:01**	**decline order**
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	order received
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 09:17:13	take goods out of storage
3	02.06.20 - 12:12:19	place goods in parcel
3	02.06.20 - 12:12:43	weigh package
3	02.06.20 - 12:13:13	seal package
3	02.06.20 - 12:13:32	hand over package
3	02.06.20 - 13:41:32	ship package

Imprecise data: timestamp

The rejection of the order in case 2 according to the event log was executed four months before the customer has placed the order.
Possibly the month and date got mixed up.

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	02.06.20 - 00:44:53	ship package
2	**02.06.20 - 06:20:01**	**decline order**
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	order received
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 09:17:13	take goods out of storage
3	02.06.20 - 12:12:19	place goods in parcel
3	02.06.20 - 12:12:43	weigh package
3	02.06.20 - 12:13:13	seal package
3	02.06.20 - 12:13:32	hand over package
3	02.06.20 - 13:41:32	ship package

Imprecise data: timestamp

The rejection of the order in case 2 according to the event log was executed four months before the customer has placed the order.
Possibly the month and date got mixed up.

The correct time stamp should be 02.06.20 - 06:20:01

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	02.06.20 - 00:44:53	ship package
2	02.06.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	order received
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 09:17:13	take goods out of storage
3	02.06.20 - 12:12:19	place goods in parcel
3	02.06.20 - 12:12:43	weigh package
3	02.06.20 - 12:13:13	seal package
3	02.06.20 - 12:13:32	hand over package
3	02.06.20 - 13:41:32	ship package

Irrelevant data: event

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	02.06.20 - 00:44:53	ship package
2	02.06.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	order received
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	**02.06.20 - 09:17:13**	**take goods out of storage**
3	**02.06.20 - 12:12:19**	**place goods in parcel**
3	**02.06.20 - 12:12:43**	**weigh package**
3	**02.06.20 - 12:13:13**	**seal package**
3	**02.06.20 - 12:13:32**	**hand over package**
3	02.06.20 - 13:41:32	ship package

Irrelevant data: event

The packaging activity in case 3 is logged in much deeper detail than the other cases. In this case the events
place goods in parcel,
weigh package,
seal package,
hand over package
represent distinct process steps in a level too granular.

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	02.06.20 - 00:44:53	ship package
2	02.06.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	order received
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	**02.06.20 - 12:13:32**	**package goods**
3	02.06.20 - 13:41:32	ship package

Irrelevant data: event

The packaging activity in case 3 is logged in much deeper detail than the other cases. In this case the events
place goods in parcel,
weigh package,
seal package,
hand over package
represent distinct process steps in a level too granular.

They can be summarised into the single process step:
package goods

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	02.06.20 - 00:44:53	ship package
2	02.06.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	order received
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 12:13:32	package goods
3	02.06.20 - 13:41:32	ship package

Imprecise data: event attributes

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	02.06.20 - 00:44:53	ship package
2	02.06.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	**02.06.20 - 06:50:42**	**order received**
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 12:13:32	package goods
3	02.06.20 - 13:41:32	ship package

Imprecise data: event attributes

In case 4 the activity label
order received
is used instead of
receive order.
The semantic meaning of the activity is the same, yet the names are distinct.

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	02.06.20 - 00:44:53	ship package
2	02.06.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	**02.06.20 - 06:50:42**	**recieve order**
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 12:13:32	package goods
3	02.06.20 - 13:41:32	ship package

Imprecise data: event attributes

In case 4 the activity label
order received
is used instead of
receive order.
The semantic meaning of the activity is the same, yet the names are distinct.

The different descriptions should be unified.

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	01.06.20 - 18:33:32	recieve order
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	02.06.20 - 00:44:53	ship package
2	02.06.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	recieve order
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 12:13:32	package goods
3	02.06.20 - 13:41:32	ship package

Incorrect data: activity name

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	**01.06.20 - 18:33:32**	**recieve order**
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	02.06.20 - 00:44:53	ship package
2	02.06.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	recieve order
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 12:13:32	package goods
3	02.06.20 - 13:41:32	ship package

Incorrect data: activity name

The first activity of case 2
recieve order
is written incorrectly.
This activity therefore does not accurately reflect the process step that generated the log entry.

Event Log Imperfection

Exercise 3a

Christian-Albrechts-Universität zu Kiel

Case ID	Timestamp	Activity
1	01.06.20 - 18:00:42	receive order
1	01.06.20 - 18:20:01	locate appropriate warehouse
2	**01.06.20 - 18:33:32**	**receive order**
1	01.06.20 - 19:17:17	take goods out of storage
1	01.06.20 - 23:17:19	package goods
1	02.06.20 - 00:44:53	ship package
2	02.06.20 - 06:20:01	decline order
3	02.06.20 - 06:21:00	receive order
2	02.06.20 - 06:25:11	inform customer
4	02.06.20 - 06:50:42	recieve order
3	02.06.20 - 07:00:02	locate appropriate warehouse
4	02.06.20 - 07:25:01	decline order
4	02.06.20 - 08:24:18	inform customer
3	02.06.20 - 12:13:32	package goods
3	02.06.20 - 13:41:32	ship package

Incorrect data: activity name

The first activity of case 2
recieve order
is written incorrectly.
This activity therefore does not accurately reflect the process step that generated the log entry.

The correct spelling is *receive order*.

Event Log Imperfection

Exercise 3b

What might be wrong when assuming that all event log imperfections have been caused by logging errors?

If all anomalies or imperfections are assumed to be logging errors that need correction flaws in the process can be overseen and remain unnoticed.

Eventually-Follows Graph

Exercise 4a

Draw both a directly-follows graph and an eventually-follows graph based on the event log L.

L = $[\langle ABCDEDEF \rangle, \langle ACDBEF \rangle, \langle ABCEDEDF \rangle, \langle AFCDBE \rangle]$

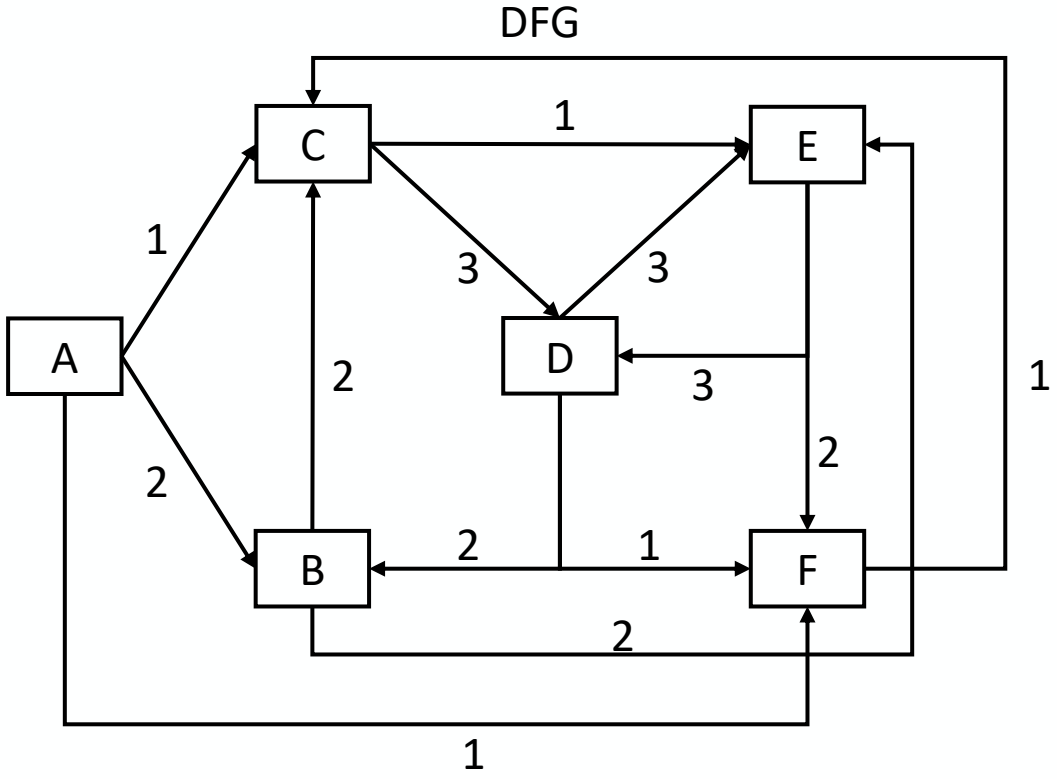
Eventually-Follows Graph

Exercise 4a

Christian-Albrechts-Universität zu Kiel

Draw both a directly-follows graph and an eventually-follows graph based on the event log L.

L = $[\langle ABCDEDEF\rangle, \langle ACDBEF\rangle, \langle ABCEDEDF\rangle, \langle AFCDBE\rangle]$



Eventually-Follows Graph

Exercise 4a

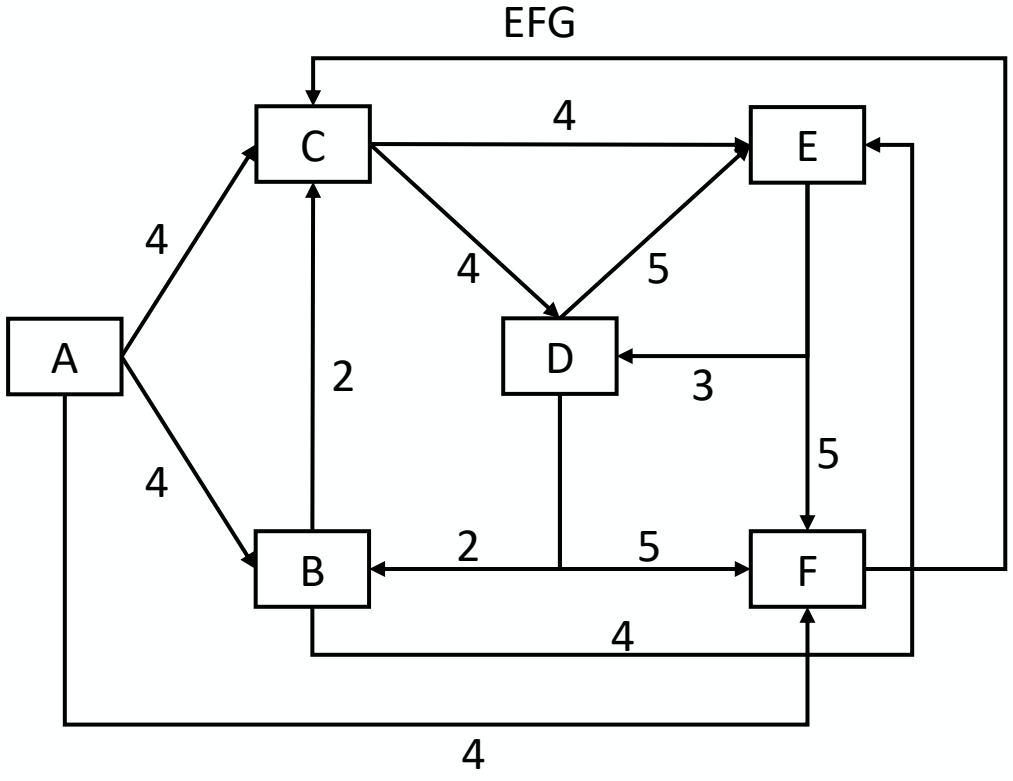
Christian-Albrechts-Universität zu Kiel

Draw both a directly-follows graph and an eventually-follows graph based on the event log L.

$$L = [\langle ABCDEDEF \rangle, \langle ACDBEF \rangle, \langle ABCEDEDF \rangle, \langle AFCDBE \rangle]$$

It is irrelevant how often B is followed by E. All that matters is, that it is eventually followed by it.

(Source: …)



If an activity occurs multiple times in a trace, like E, we count and sum up the eventually followed activities for every single E.

(Source: https://www.win.tue.nl/~dfahland/publications/LeemansFA_2013_bpi.pdf)

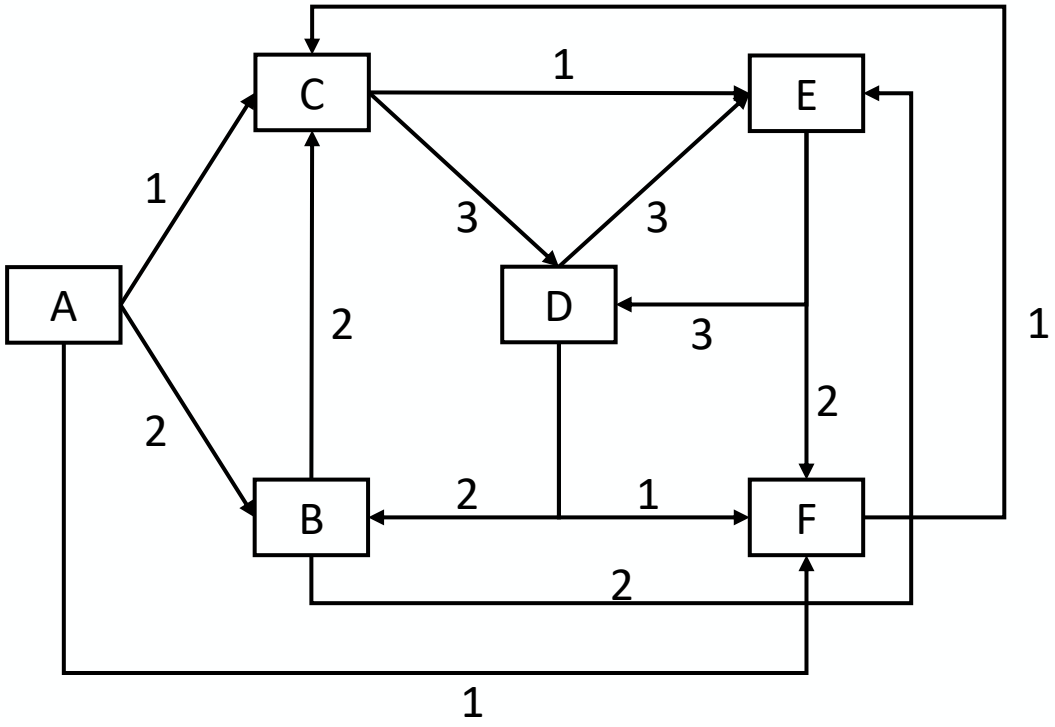
Eventually-Follows Graph

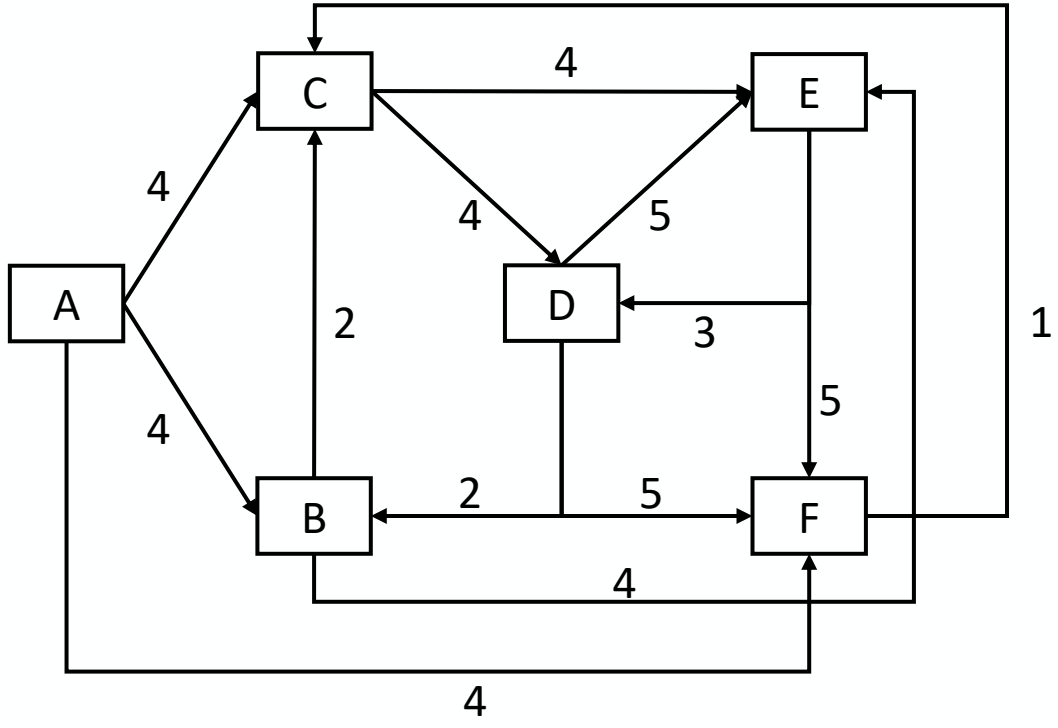
Exercise 4a

Christian-Albrechts-Universität zu Kiel

Draw both a directly-follows graph and an eventually-follows graph based on the event log L.

L = $[\langle ABCDEDEF\rangle, \langle ACDBEF\rangle, \langle ABCEDEDF\rangle, \langle AFCDBE\rangle]$





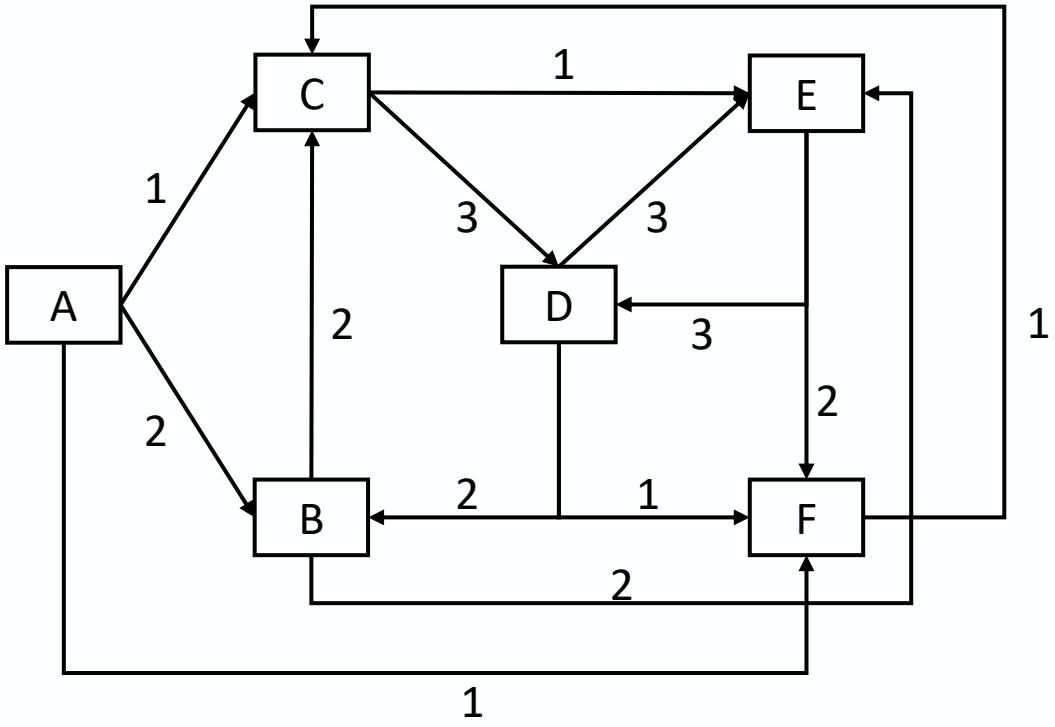
Eventually-Follows Graph

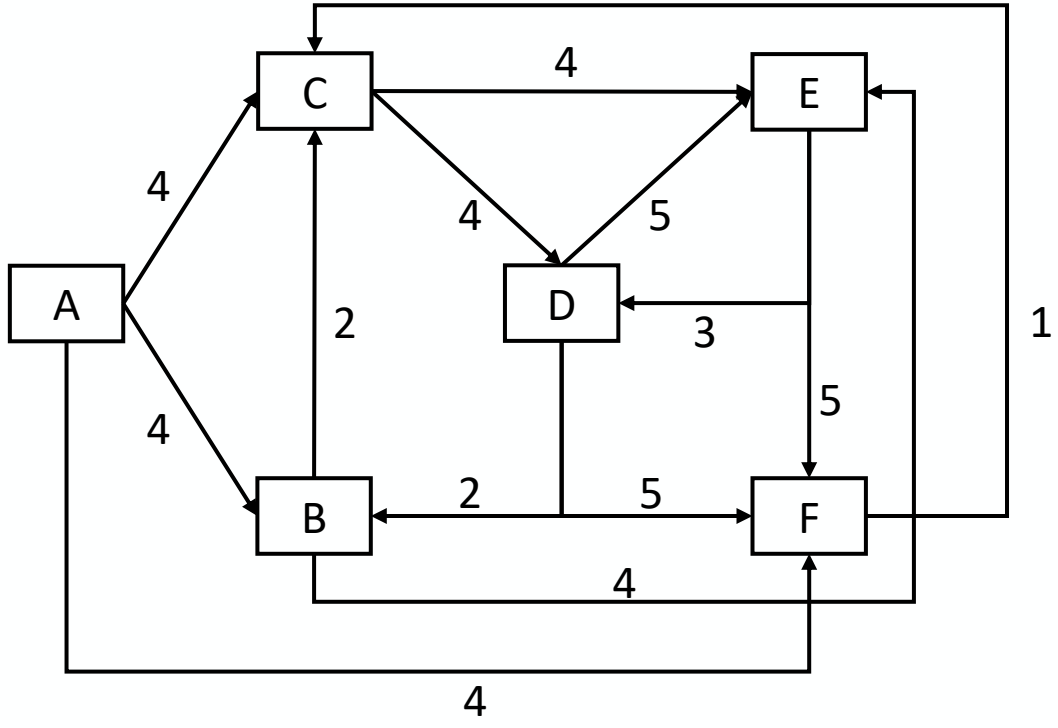
Exercise 4a

Christian-Albrechts-Universität zu Kiel

Try to identify outliers in both graphs.

L = $[\langle ABCDEDEF \rangle, \langle ACDBEF \rangle, \langle ABCEDEDF \rangle, \langle AFCDBE \rangle]$





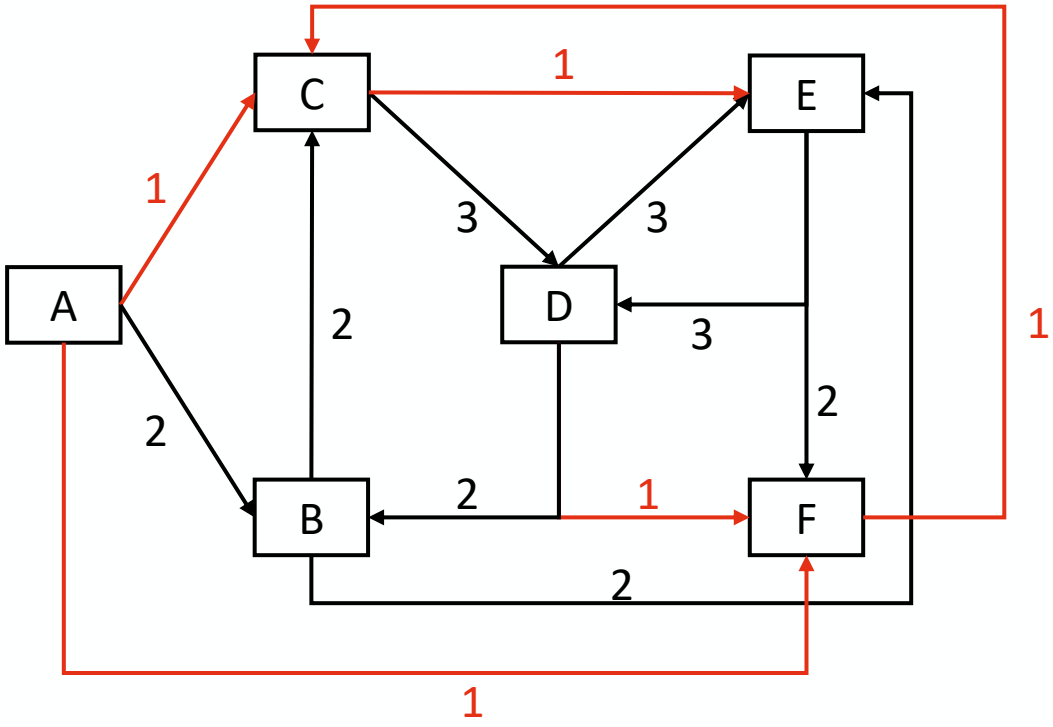
Eventually-Follows Graph

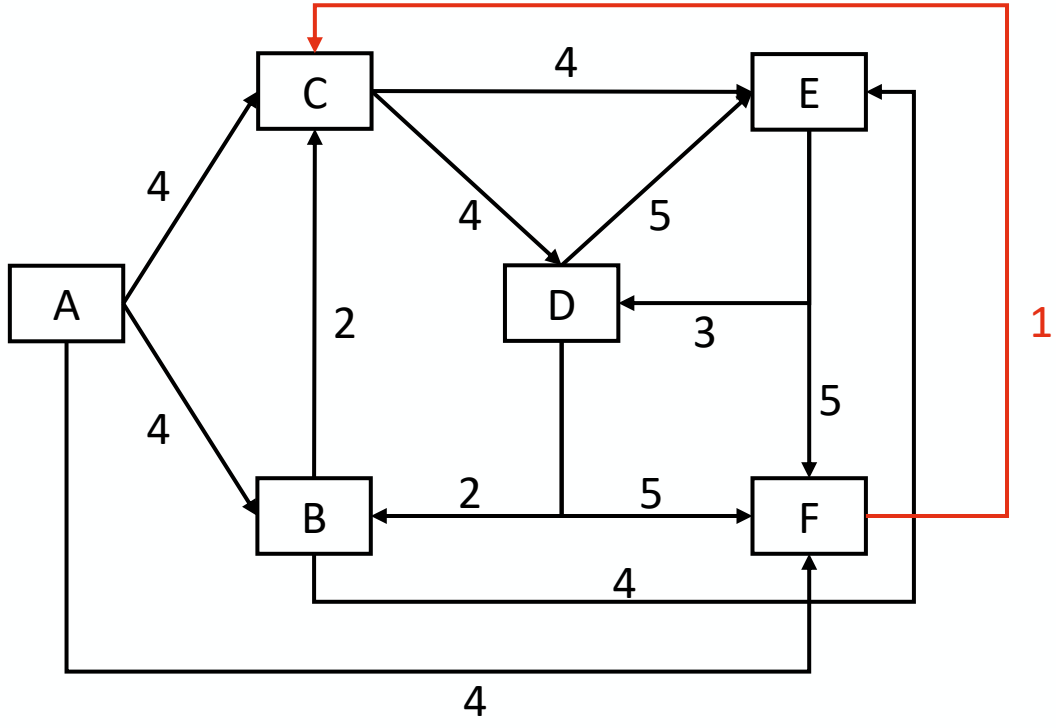
Exercise 4a

Christian-Albrechts-Universität zu Kiel

Try to identify outliers in both graphs.

$$L = [\langle ABCDEDEF \rangle, \langle ACDBEF \rangle, \langle ABCEDEDF \rangle, \langle AFCDBE \rangle]$$





Eventually-Follows Graph

Exercise 4b

Explain which of the graphs is better suited to identify outliers.

The eventually-follows graph is suited better to identify outliers.
Rare occurrences of activities still affects the edge count of the EFG and increases it, but the regular more frequent behaviour is amplified by the EFG.