CS 4072 - Topics in CS
Process Mining

Lecture # 18

April 26, 2022

Spring 2022

FAST - NUCES, CFD Campus

Dr. Rabia Maqsood

rabia.maqsood@nu.edu.pk

Today's Topics

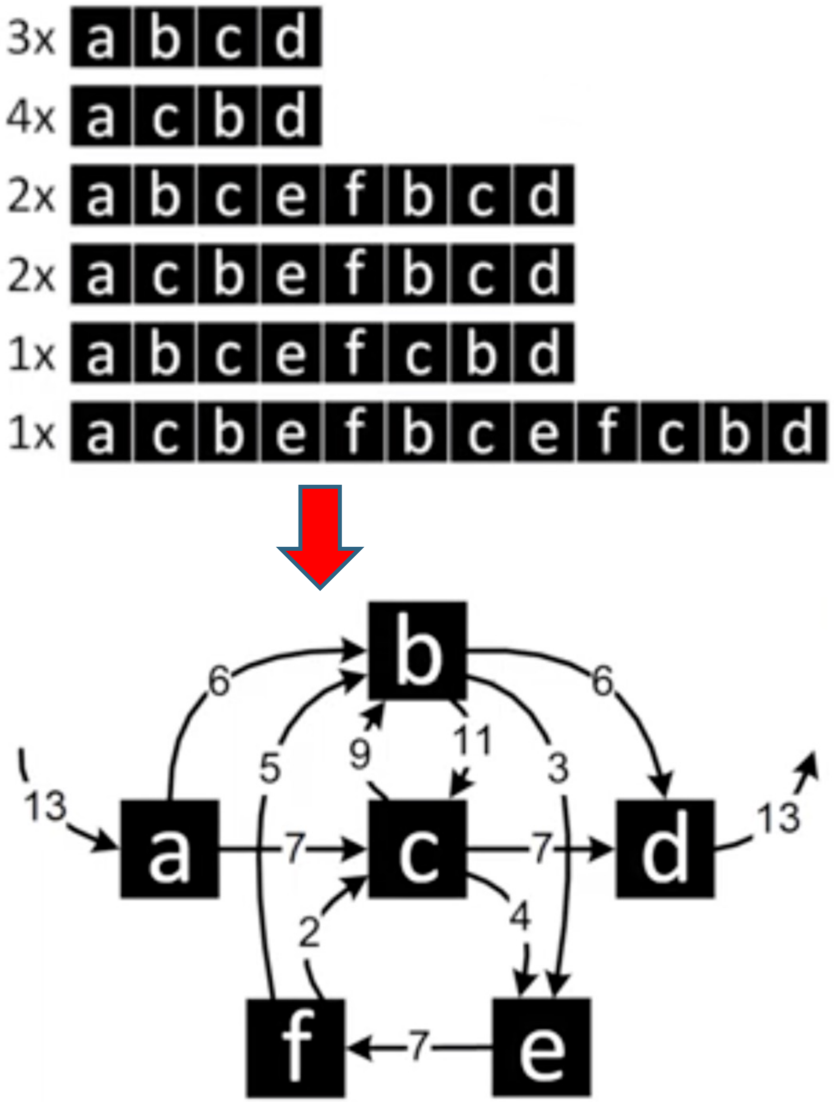
- Inductive Mining Algorithm (continued)

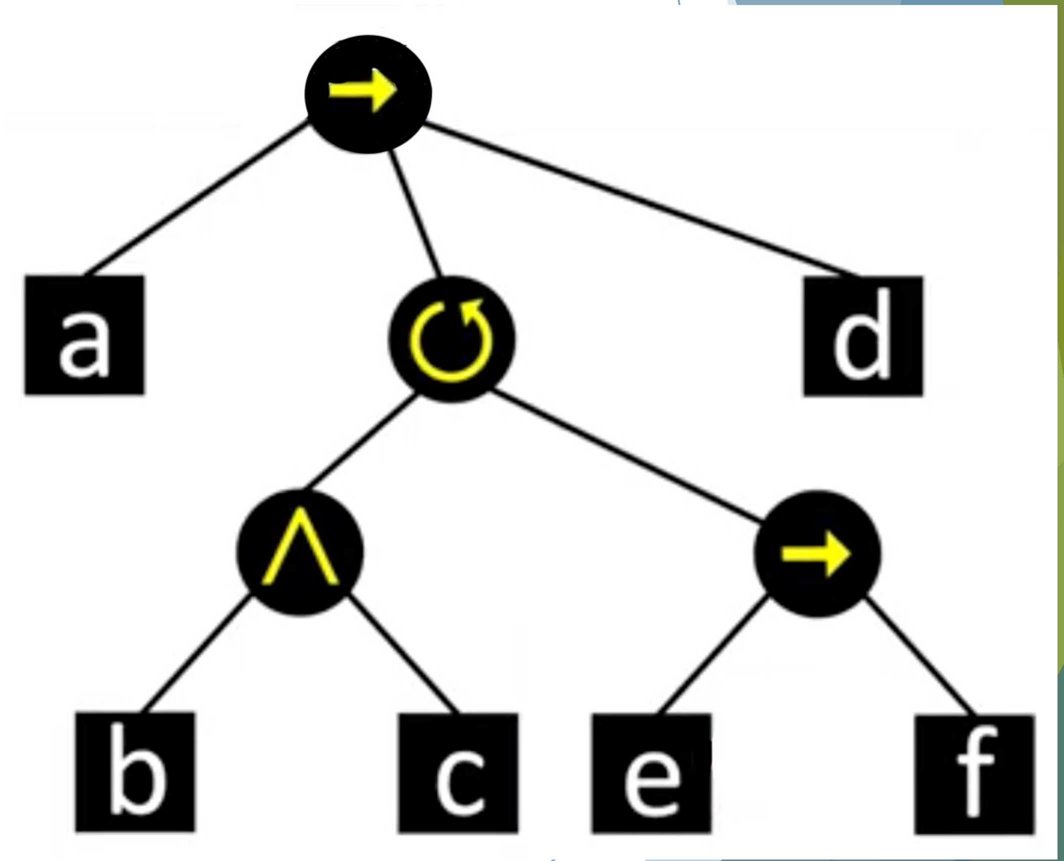
Inductive Miner Algorithm

- Basic idea:
  1. Construct a directly-follows graph based on an event log
  2. Detect patterns in the directly-followed graph
     - Identify an appropriate cut that represents one of the four possible operator nodes in the process tree
  3. Divide the event log based on the operator identified in the Step 2
  4. Repeat Steps 2 & 3 until a sub-event log cannot be divided further

The IM algorithm iteratively splits the initial event log into smaller *sublogs*.

Inductive Miner Algorithm





Directly-follows Graph

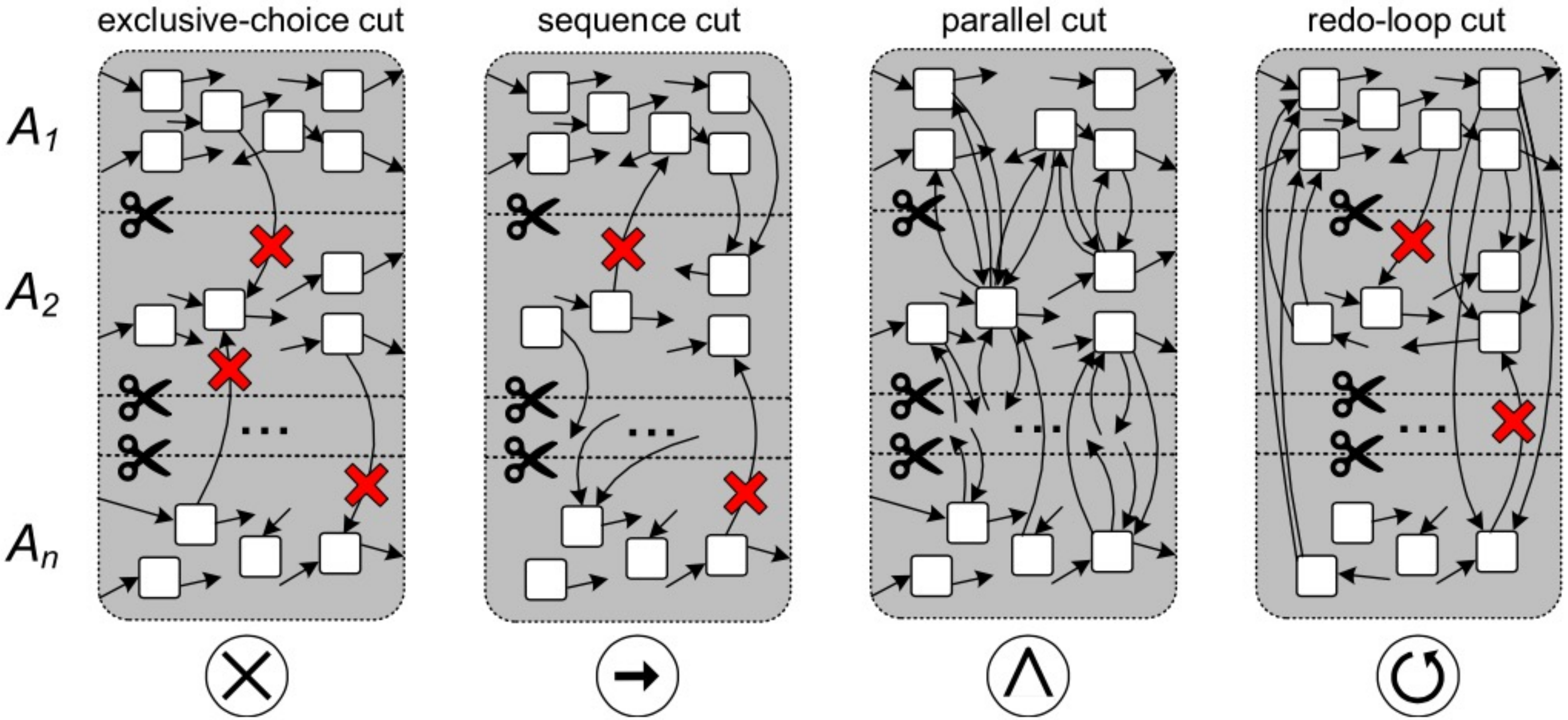
Definition 7.5 (Directly-follows graph) Let L be an event log, i.e., $L \in \mathbb{B}(\mathscr{A}^*)$. The *directly-follows graph* of L is $G(L) = (A_L, \mapsto_L, A_L^{start}, A_L^{end})$ with:

- $A_L = \{a \in \sigma \mid \sigma \in L\}$ is the set of activities in L,
- $\mapsto_L = \{(a, b) \in A \times A \mid a >_L b\}$ is the directly follows relation,[3]
- $A_L^{start} = \{a \in A \mid \exists_{\sigma \in L} a = \mathit{first}(\sigma)\}$ is the set of start activities, and
- $A_L^{end} = \{a \in A \mid \exists_{\sigma \in L} a = \mathit{last}(\sigma)\}$ is the set of end activities.

[3] $a >_L b$ if and only if there is a trace $\sigma = \langle t_1, t_2, t_3, \ldots, t_n\rangle$ and $i \in \{1, \ldots, n-1\}$ such that $\sigma \in L$ and $t_i = a$ and $t_{i+1} = b$ (see Definition 6.3).

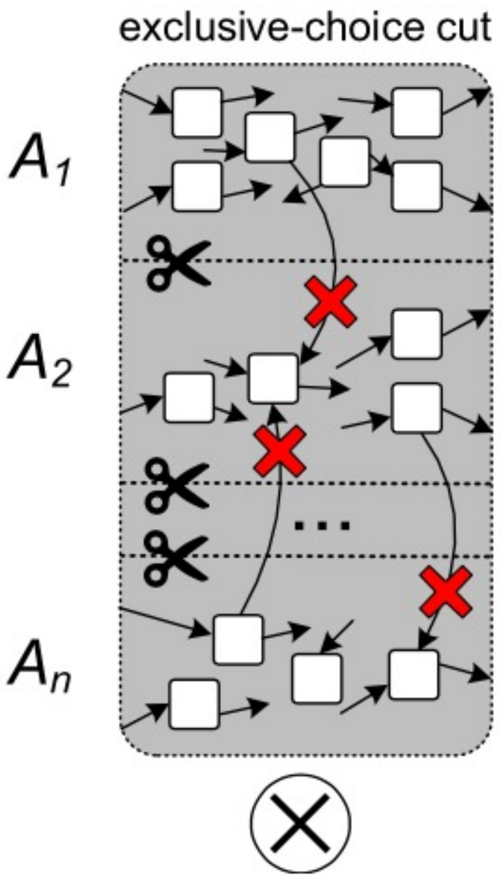
$\mapsto_L^+$ is the transitive closure of $\mapsto_L$. $a \mapsto_L^+ b$ if there is a non-empty *path* from a to b in $G(L)$, i.e., there exists a sequence of activities $a_1, a_2, \ldots, a_k$ such that $k \geq 2$, $a_1 = a$ and $a_k = b$ and $a_i \mapsto_L a_{i+1}$ for $i \in \{1, \ldots, k-1\}$. $a \not\mapsto_L^+ b$ if there is no path from a to b in the directly-follows graph.

Four types of cuts



Exclusive-choice cut

If there are two disjoint subsets of activities, then there should be **no directly-follows relation** between their activities.

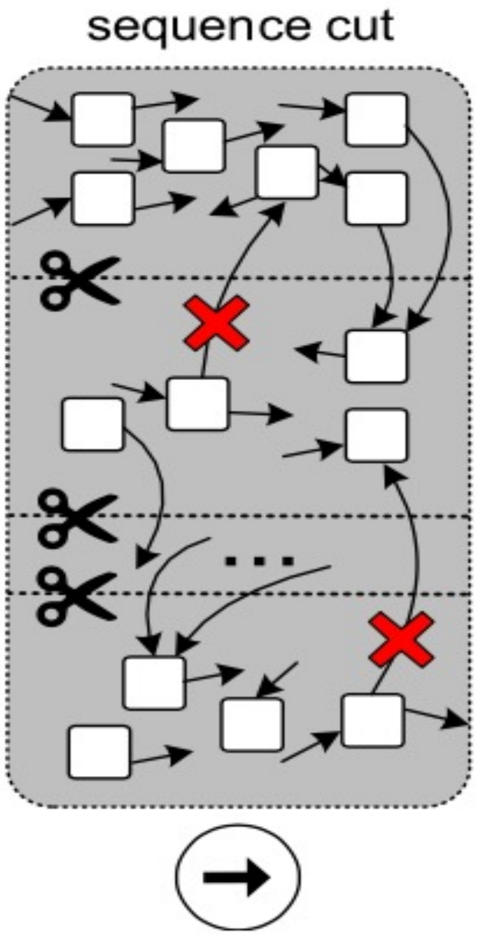


An *exclusive-choice cut* of $G(L)$ is a cut $(\times, A_1, A_2, \ldots, A_n)$ such that

– $\forall_{i,j \in \{1,\ldots,n\}} \forall_{a \in A_i} \forall_{b \in A_j}\ i \neq j \Rightarrow a \not\mapsto_L b$.

Sequence cut

Partitions the directly-follows graph into parts where arcs are going in one direction

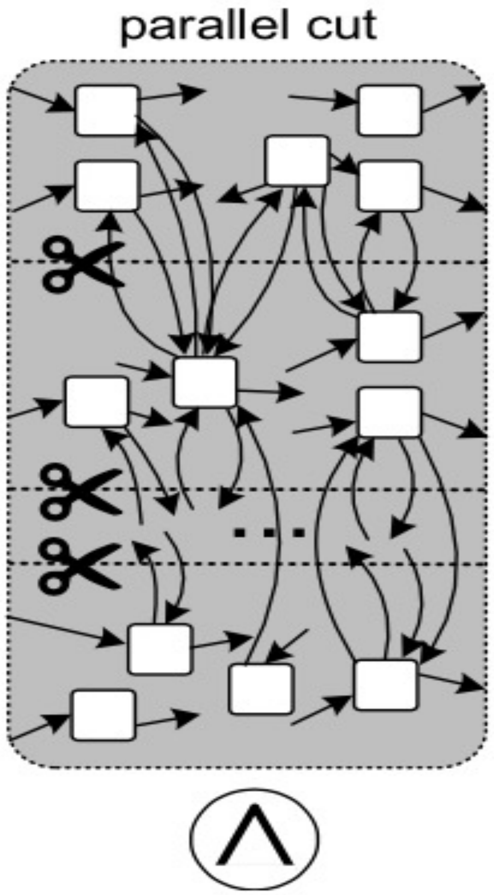


A *sequence cut* of $G(L)$ is a cut $(\rightarrow, A_1, A_2, \ldots, A_n)$ such that

– $\forall_{i,j \in \{1,\ldots,n\}} \forall_{a \in A_i} \forall_{b \in A_j}\ i < j \Rightarrow (a \mapsto_L^+ b \wedge b \not\mapsto_L^+ a)$.

Parallel cut

Any activity in one subset should be followed by any activity in the second subset (and vice-versa), then we can split the two subsets.
Also, all the subsets should have **start** and **end** activities.



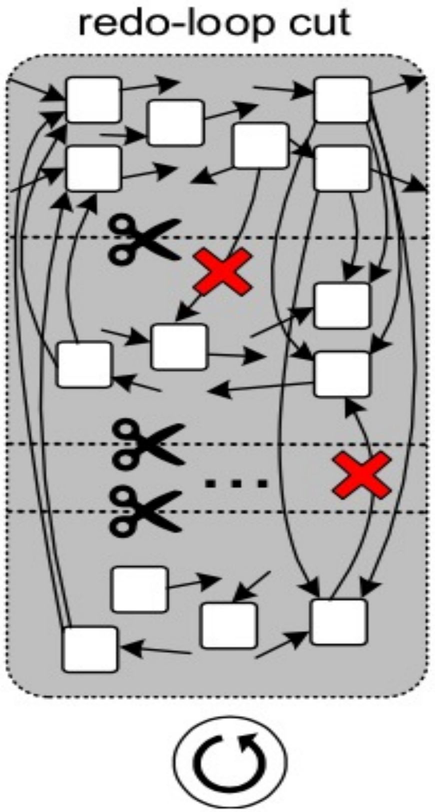
A *parallel cut* of $G(L)$ is a cut $(\wedge, A_1, A_2, \ldots, A_n)$ such that

- $\forall_{i \in \{1,\ldots,n\}}\ A_i \cap A_L^{start} \neq \emptyset\ \wedge\ A_i \cap A_L^{end} \neq \emptyset$ and
- $\forall_{i,j \in \{1,\ldots,n\}} \forall_{a \in A_i} \forall_{b \in A_j}\ i \neq j \Rightarrow a \mapsto_L b$.

Loop cut

We need **do** and **redo** parts:

- Everything should **begin** and **end** in do-part
- From all the end activities, we should be able to **move to redo-part** & we should be able to **move to the start activities in do-part** from the redo-part



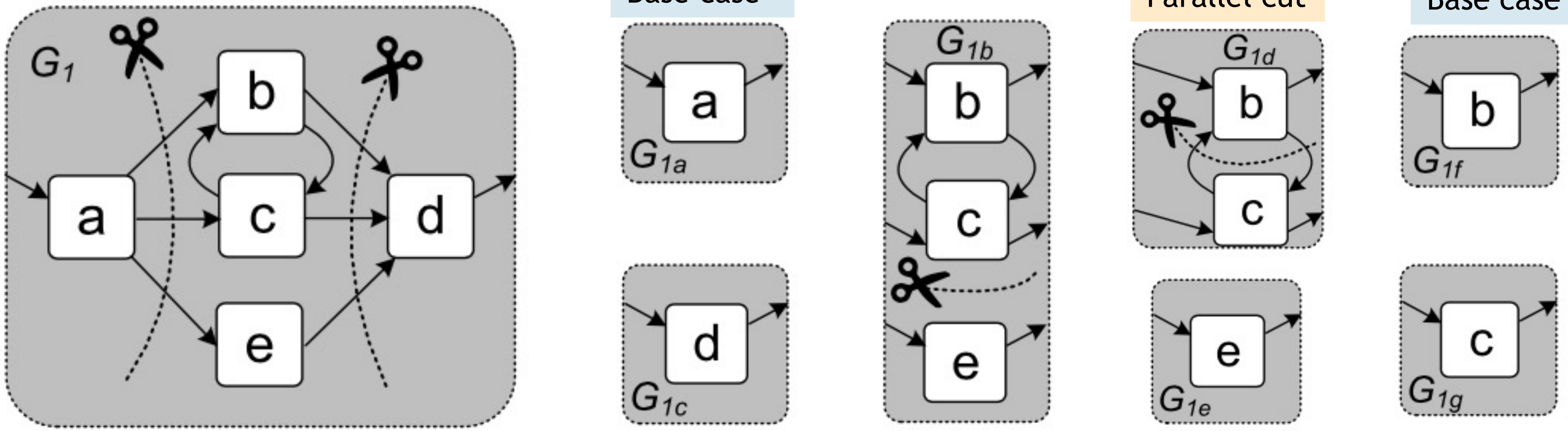
A *redo-loop cut* of $G(L)$ is a cut $(\circlearrowleft, A_1, A_2, \ldots, A_n)$ such that

- $n \geq 2$,
- $A_L^{start} \cup A_L^{end} \subseteq A_1$,
- $\{a \in A_1 \mid \exists_{i \in \{2,\ldots,n\}} \exists_{b \in A_i}\ a \mapsto_L b\} \subseteq A_L^{end}$,
- $\{a \in A_1 \mid \exists_{i \in \{2,\ldots,n\}} \exists_{b \in A_i}\ b \mapsto_L a\} \subseteq A_L^{start}$,
- $\forall_{i,j \in \{2,\ldots,n\}} \forall_{a \in A_i} \forall_{b \in A_j}\ i \neq j \Rightarrow a \not\mapsto_L b$,
- $\forall_{i \in \{2,\ldots,n\}} \forall_{b \in A_i} \exists_{a \in A_L^{end}}\ a \mapsto_L b \Rightarrow \forall_{a' \in A_L^{end}}\ a' \mapsto_L b$, and
- $\forall_{i \in \{2,\ldots,n\}} \forall_{b \in A_i} \exists_{a \in A_L^{start}}\ b \mapsto_L a \Rightarrow \forall_{a' \in A_L^{start}}\ b \mapsto_L a'$.

Solution – 1

- Run the inductive miner algorithm on the following event log and construct the resultant process tree.

$L_1 = [\langle a, b, c, d\rangle^3, \langle a, c, b, d\rangle^2, \langle a, e, d\rangle]$

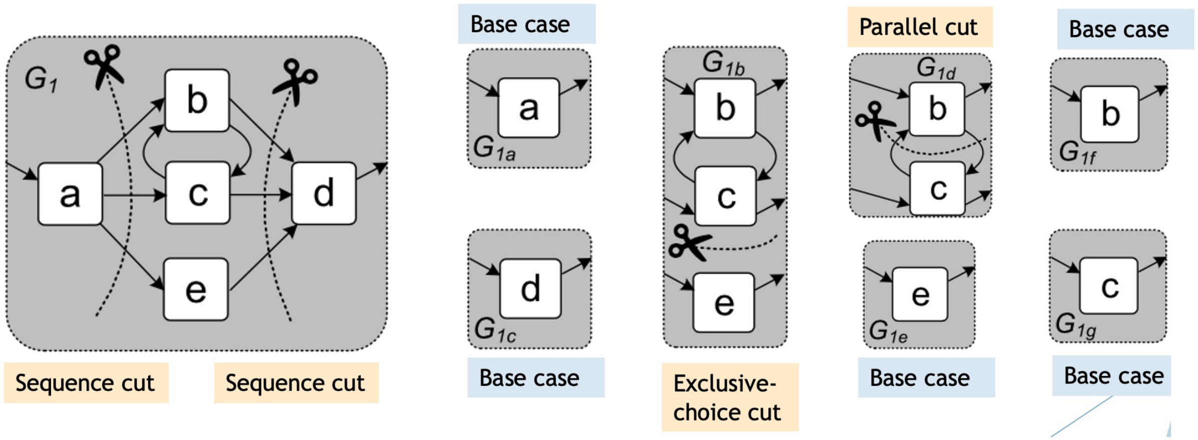


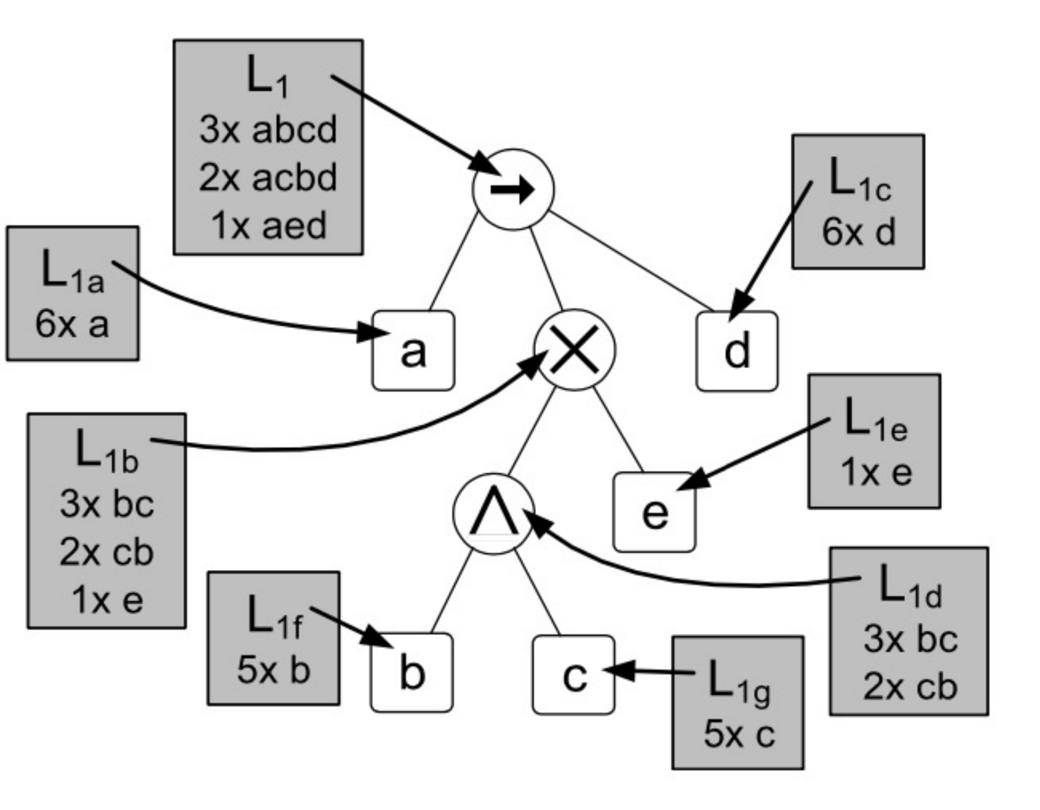
Sequence cut | Sequence cut | Base case | Base case | Exclusive-choice cut | Parallel cut | Base case | Base case | Base case

Solution – 1 (continued)

- Run the inductive miner algorithm on the following event log and construct the resultant process tree.

$L_1 = [\langle a, b, c, d\rangle^3, \langle a, c, b, d\rangle^2, \langle a, e, d\rangle]$



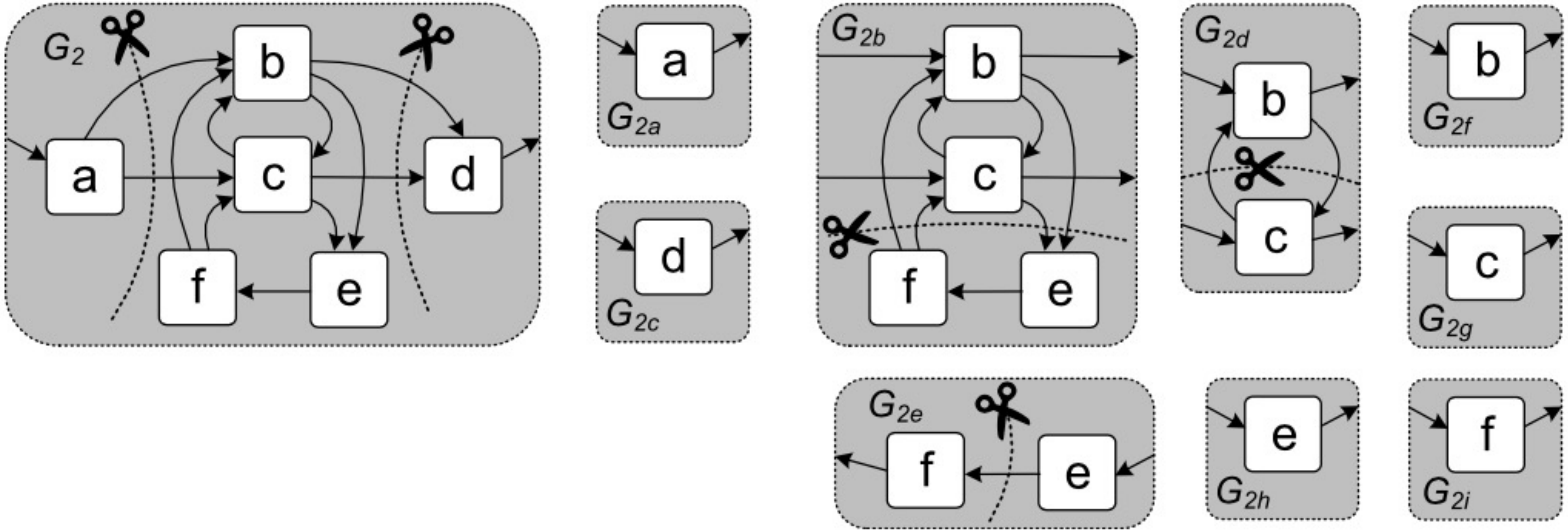


Solution – 3

- Run the inductive miner algorithm on the following event log and construct the resultant process tree.

$L_2 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^4, \langle a,b,c,e,f,b,c,d\rangle^2, \langle a,c,b,e,f,b,c,d\rangle^2,$

$\langle a,b,c,e,f,c,b,d\rangle, \langle a,c,b,e,f,b,c,e,f,c,b,d\rangle]$

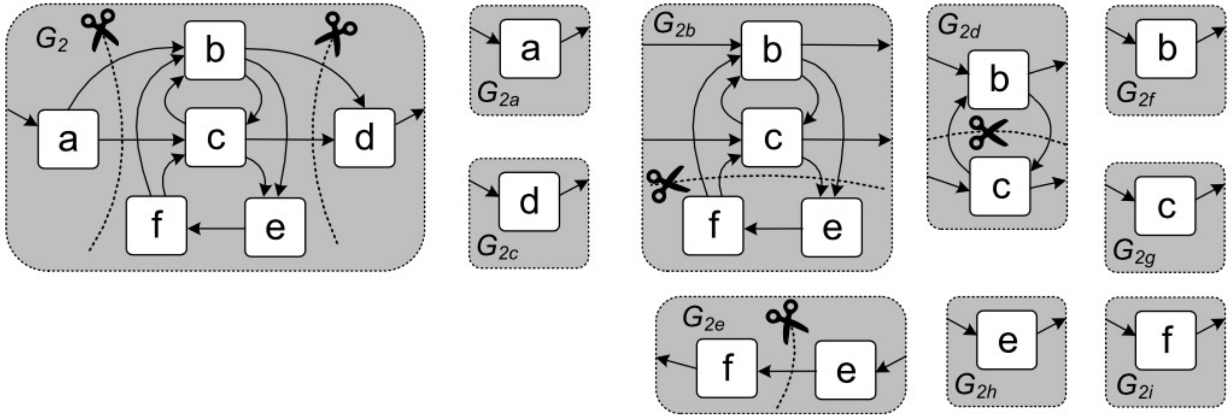


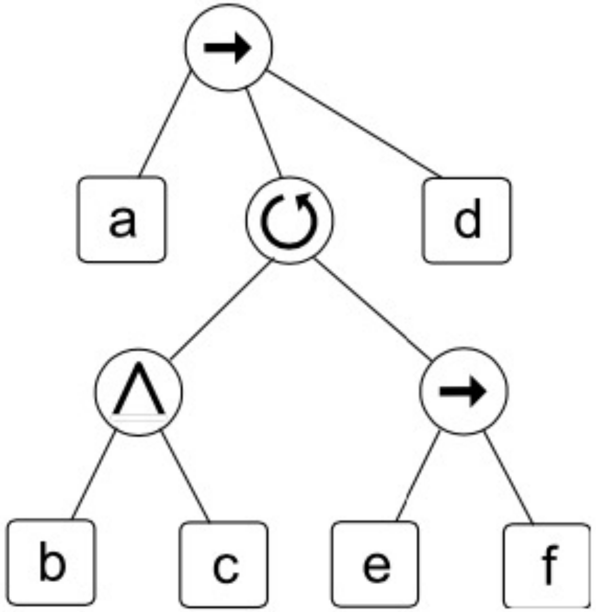
Solution – 3 (continued)

- Run the inductive miner algorithm on the following event log and construct the resultant process tree.

$L_2 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^4, \langle a,b,c,e,f,b,c,d\rangle^2, \langle a,c,b,e,f,b,c,d\rangle^2,$

$\langle a,b,c,e,f,c,b,d\rangle, \langle a,c,b,e,f,b,c,e,f,c,b,d\rangle]$



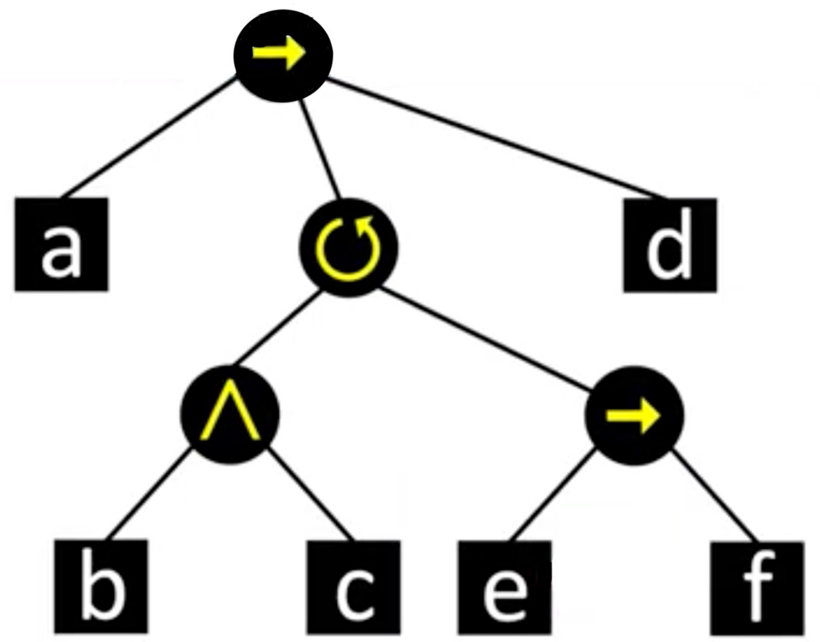


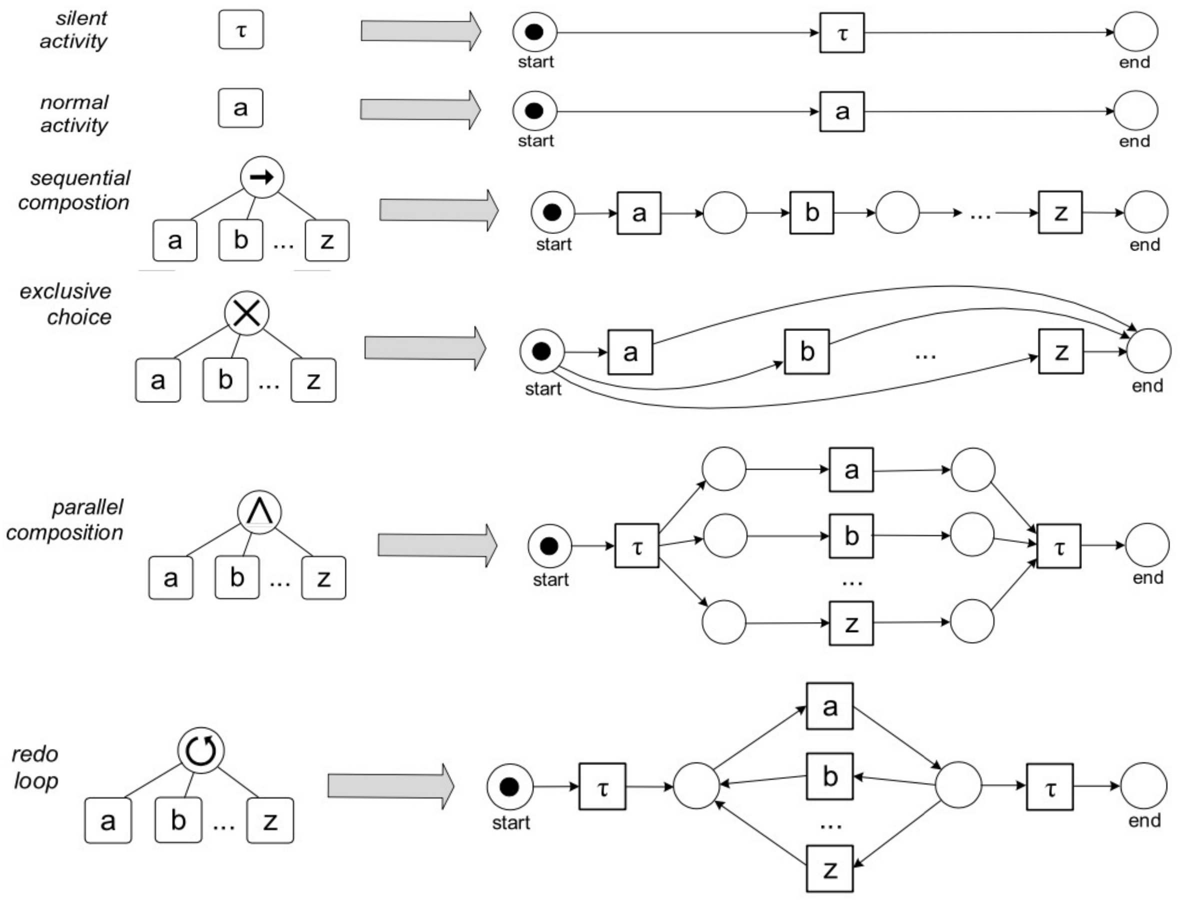
Further Readings

- Practice more problems (solutions for event log L_3 to L_{11} is available in the book, solve the problems yourself).
- Read Section 7.5.2 yourself.
- For more variants of IM, read Section 7.5.3 (optional).

Homework

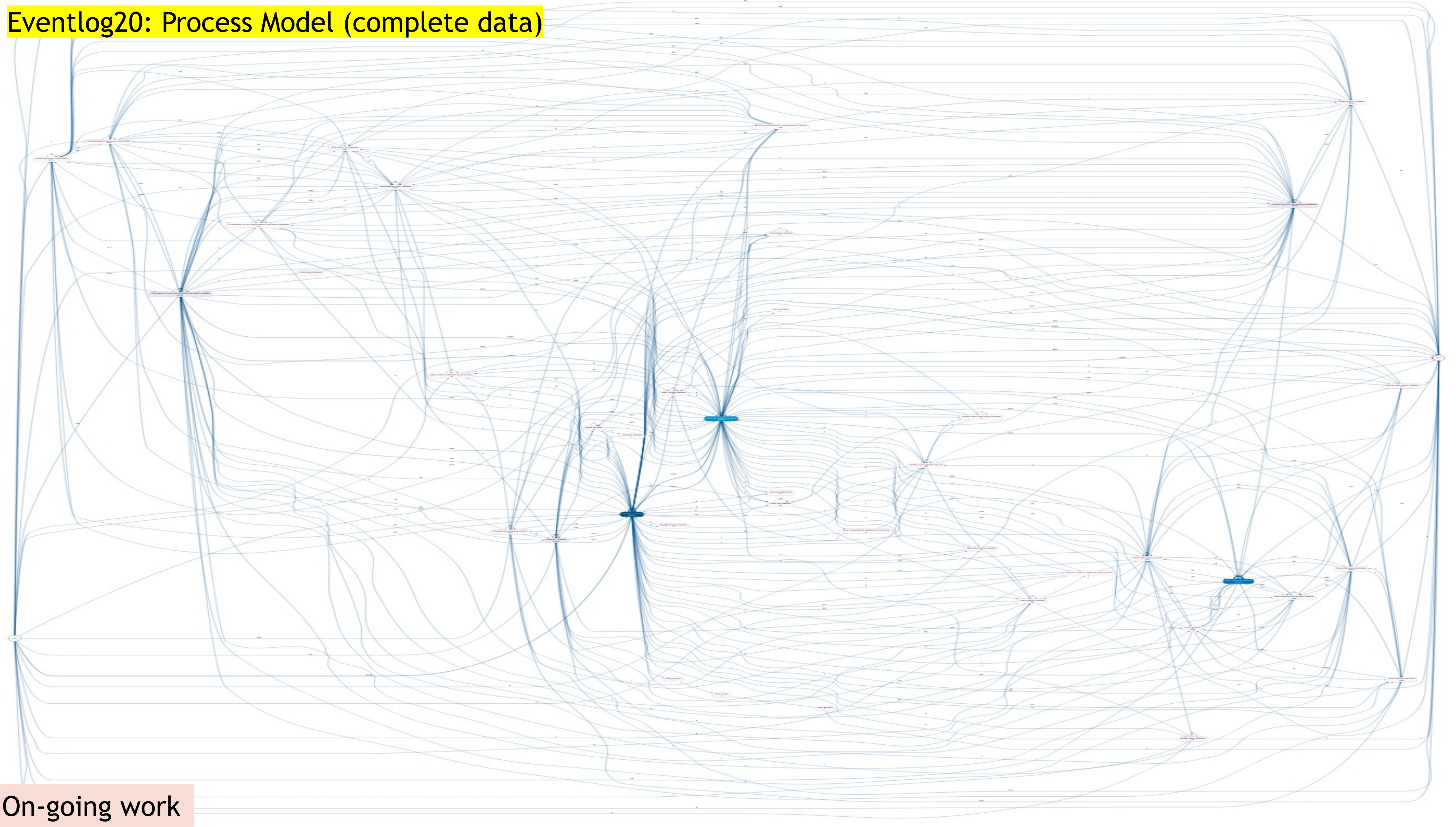
- Convert this process tree into an equivalent WF-net.





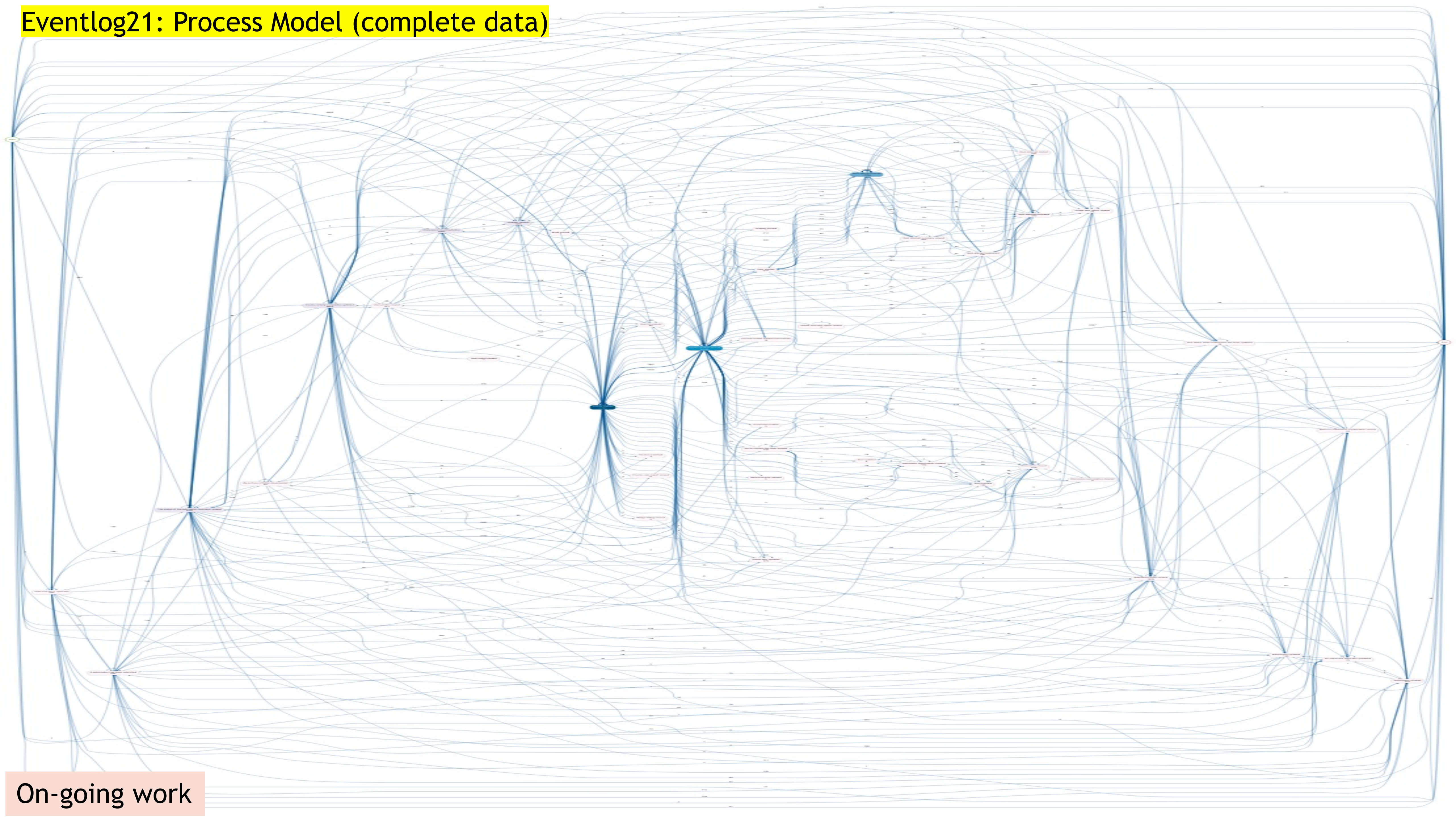
- Real process(es) can be more complex

Eventlog20: Process Model (complete data)



On-going work

Eventlog21: Process Model (complete data)



On-going work

Reading Material

- Chapter 7: Aalst
- Online resources:
  - Introduction to Process Mining with ProM (https://www.futurelearn.com/courses/process-mining)
  - Process Mining: Data science in Action (https://www.coursera.org/learn/process-mining)