Christian-Albrechts-Universität zu Kiel

Process Mining

Prof. Dr. Agnes Koschmider

Vorlesungsüberblick

Christian-Albrechts-Universität zu Kiel

0	Organisation und Einführung
I	Workflow-Muster
II	Prozessmodellierung
III	Prozessanalyse
IV	Process Mining Grundlagen
V	Verfahren – Process Discovery
VI	Verfahren – Conformance Checking
VIII	Werkzeuge und aktuelle Forschungsfragen

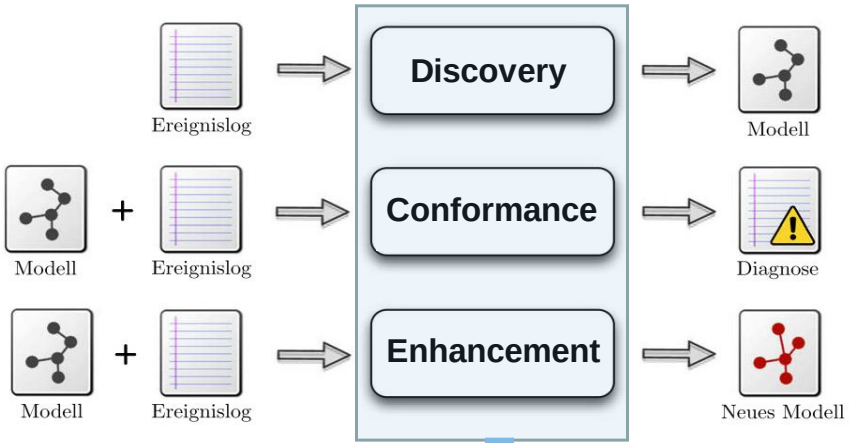
ÜBERBLICK

Überblick

Christian-Albrechts-Universität zu Kiel

Kombination aus Verfahren und Perspektiven

Verfahren



Perspektiven

- Kontrollfluss (Ablauf)
- Organisation (Rollen/Ressourcen)
- Zeit (Dauer von Aktivitäten)
- Daten (Geschäftsobjekte)

	Kontrollfluss	Organisation	Zeit	Daten
Discovery	X	X		
Conformance	X	X	X	X
Enhancement		X	X	X

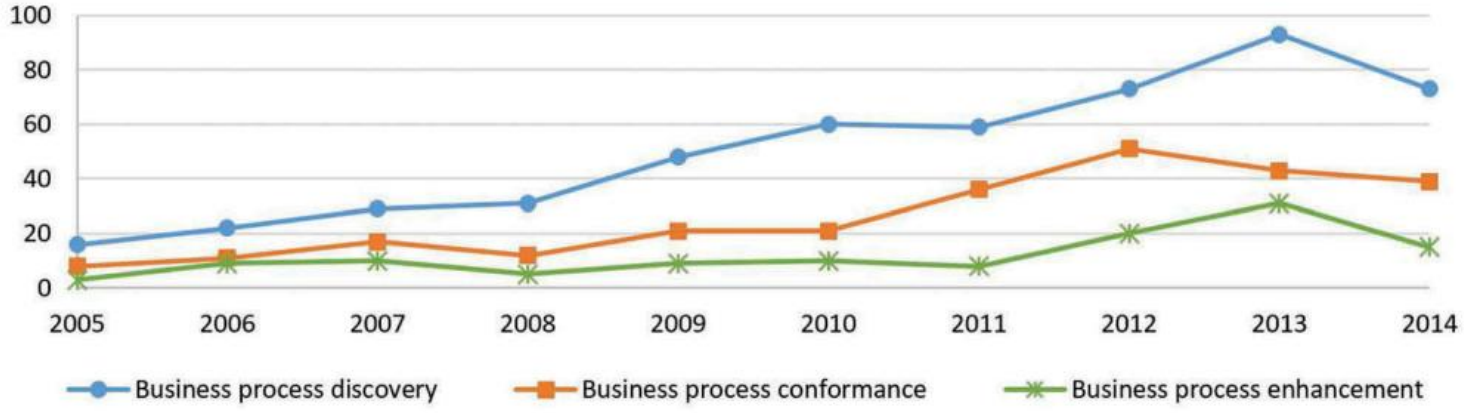
X = wird in der Vorlesung betrachtet, für die anderen Felder existieren bisweilen keine etablierten Techniken

Überblick

Christian-Albrechts-Universität zu Kiel

Kombination aus Verfahren und Perspektiven

- Aktuelle Literaturübersicht (Systematic mapping study)
- Anzahl der Arbeiten auf dem Gebiet Process Mining nach den drei Typen von Verfahren im Zeitraum 2005-2014



Quelle: Cardenas Maita, A.R., Correa Martins, L., Lopez Paz, C.R., Rafferty, L., Hung, P.C.K., Marques Peres, S., Fantinato, M.: A systematic mapping study of process Mining. Enterprise Information Systems, Vol. 12, No. 5, S. 505-549, Taylor&Francis, 2018.

Christian-Albrechts-Universität zu Kiel

PROCESS DISCOVERY

Process Discovery

Kontrollflussperspektive

- Die Anwendung von Discovery-Verfahren für die Kontrollfluss-perspektive ist das dominante Anwendungsgebiet von Process Mining

- Deutsche Übersetzungen: Prozess- oder Modell
  - Entdeckung
  - Synthese
  - Erstellung
  - Extraktion

	Kontrollfluss	Organisation	Zeit	Daten
Discovery	X	X		
Conformance	X	X	X	X
Enhancement		X	X	X

Process Discovery

Problemstellung

- Gegeben sei ein Event Log $\mathcal{L}$. Ein Process-Discovery-Algorithmus ist eine Funktion, die $\mathcal{L}$ auf ein Prozessmodell M abbildet, so dass M das Verhalten repräsentiert, das im Log $\mathcal{L}$ beobachtet werden kann.

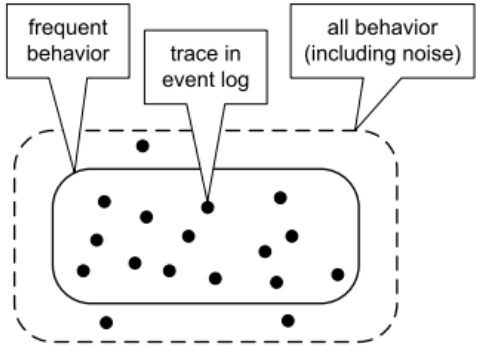
- Offen:
  - Welche Art von Prozessmodell soll für die Repräsentation gewählt werden?
    - Petri-Netze, EPK, BPMN, …
  - Welche Attribute aus dem Event Log sollen durch den Algorithmus ausgewertet werden?
    - Für den Kontrollfluss sind üblicherweise nur der Aktivitätsname eines Ereignisses, sowie der Zeitstempel und die Fallzugehörigkeit relevant

Process Discovery

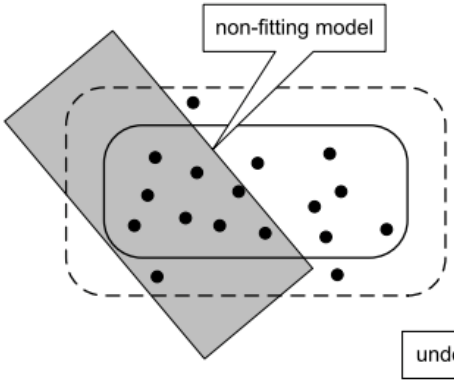
Christian-Albrechts-Universität zu Kiel

Herausforderungen für Discovery-Algorithmen

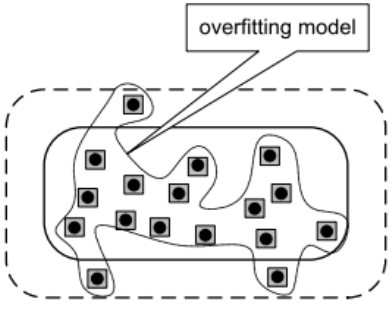
(1): Legende

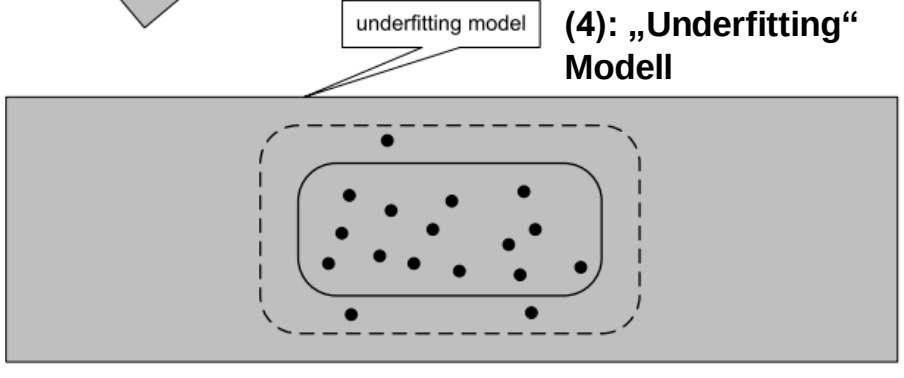


(2): Nicht passendes Modell

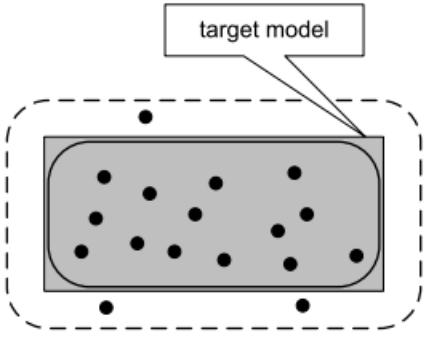


(3): „Overfitting“ Modell





(4): „Underfitting“ Modell



(5): Zielmodell (passend)

(2)-(4) sind Herausforderungen für Discovery-Algorithmen

Quelle: Van der Aalst, Wil M. P: Process Mining: Data Science in Action. Springer. S. 196, 2016.

Process Discovery

- Eines der ersten und prominentesten Verfahren für die Entdeckung eines Ablaufmodells in Form eines Prozessmodells ist der α-Algorithmus:
  - van der Aalst, Wil M. P. and Weijters, A. J. M. M. and Maruster, L. (2003). "Workflow Mining: Discovering process models from event logs", IEEE Transactions on Knowledge and Data Engineering, Vol. 16.

- Grundidee des α-Algorithmus:
  - Bestimme zunächst die Ordnungsbeziehungen zwischen den einzelnen Aktivitäten im Event Log durch die Analyse der Sequenzen
  - Leite daraus die Kausalitäten innerhalb des Prozesses ab und erstelle ein passendes Workflow-Netz

Process Discovery

- Bestimme für alle Kombinationen aus Aktivitäten die Ordnungsbeziehungen durch Analyse der Sequenzen in allen Fällen im Event Log
- Es existieren die folgenden vier möglichen Ordnungsbeziehungen zwischen zwei Aktivitäten A und B:
  - A > B (Direkte Nachfolge): gdw. in einem Fall auf A direkt B folgt.
  - A $\rightarrow$ B (Kausalität): gdw. stets nur A>B und niemals B>A beobachtet wird.
  - A || B (Nebenläufigkeit): gdw. sowohl A>B als auch B>A beobachtet wird.
  - A # B (Auswahl): gdw. weder A>B als auch B>A beobachtet wird.
- Daraus lässt sich eine sogenannte Footprint-Matrix erstellen

Process Discovery

Christian-Albrechts-Universität zu Kiel

Event Log

Fall	Aktivität
1	a
1	b
1	g
1	h
1	j
1	k
1	i
1	l
2	a
2	c
2	d
2	e
2	f
2	g
2	j
2	h
2	i
2	k
2	l

Traces

Fall 1: <*a,b,g,h,j,k,i,l*>

Fall 2: <*a,c,d,e,f,g,j,h,i,k,l*>

Direkte Nachfolge

a > b	d > e
b > g	e > f
g > h	f > g
h > j	g > j
j > k	j > h
k > i	h > i
i > l	i > k
a > c	k > l
c > d	

Footprint-Matrix

	a	b	c	d	e	f	g	h	i	j	k	l
a	#	→	→	#	#	#	#	#	#	#	#	#
b	←	#	#	#	#	#	→	#	#	#	#	#
c	←	#	#	→	#	#	#	#	#	#	#	#
d	#	#	←	#	→	#	#	#	#	#	#	#
e	#	#	#	←	#	→	#	#	#	#	#	#
f	#	#	#	#	←	#	→	#	#	#	#	#
g	#	←	#	#	#	←	#	→	#	→	#	#
h	#	#	#	#	#	#	←	#	→	||	#	#
i	#	#	#	#	#	#	#	←	#	#	||	→
j	#	#	#	#	#	#	←	||	#	#	→	#
k	#	#	#	#	#	#	#	#	||	←	#	→
l	#	#	#	#	#	#	#	#	←	#	←	#

A → B (Kausalität): gdw. stets nur A>B und niemals B>A beobachtet wird.
A || B (Nebenläufigkeit): gdw. sowohl A>B als auch B>A beobachtet wird.
A # B (Auswahl): gdw. weder A>B als auch B>A beobachtet wird.

Process Discovery

- Bestimme Start- und End-Aktivität durch Betrachten der Fälle im Event Log

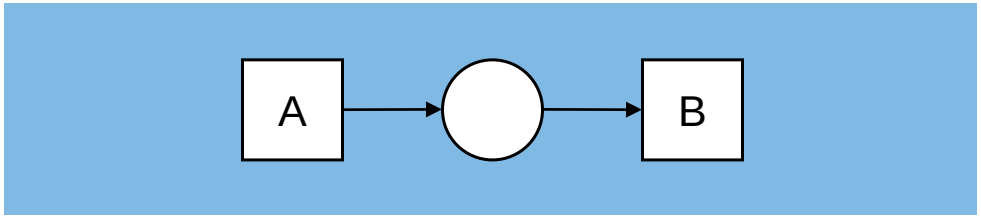
Voriges Beispiel: Fall 1: <*a*,*b*,*g*,*h*,*j*,*k*,*i*,*l*>
Fall 2: <*a*,*c*,*d*,*e*,*f*,*g*,*j*,*h*,*i*,*k*,*l*>

Start-Aktivität: *a*
End-Aktivität: *l*

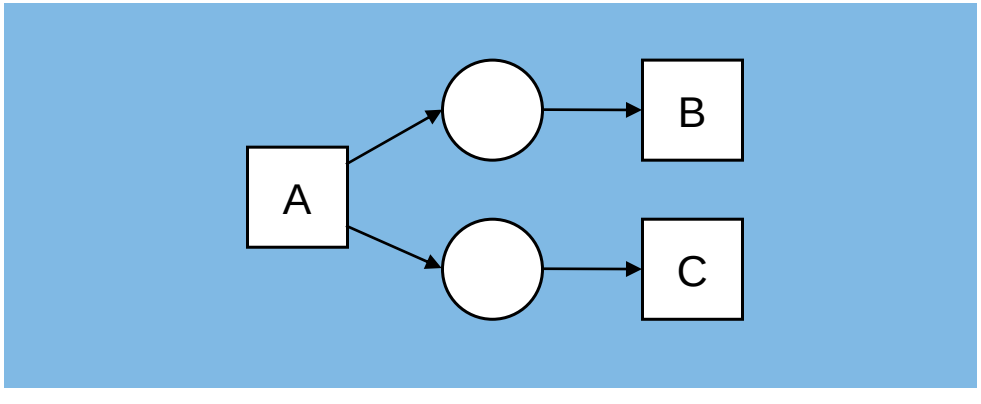
- Anhand der Informationen in der Footprint-Matrix können die folgenden Kontrollflussmuster abgeleitet werden:
  - Sequenz
  - AND-Split
  - XOR-Split
  - AND-Join
  - XOR-Join

Process Discovery

- $A \rightarrow B$: **Sequenz**



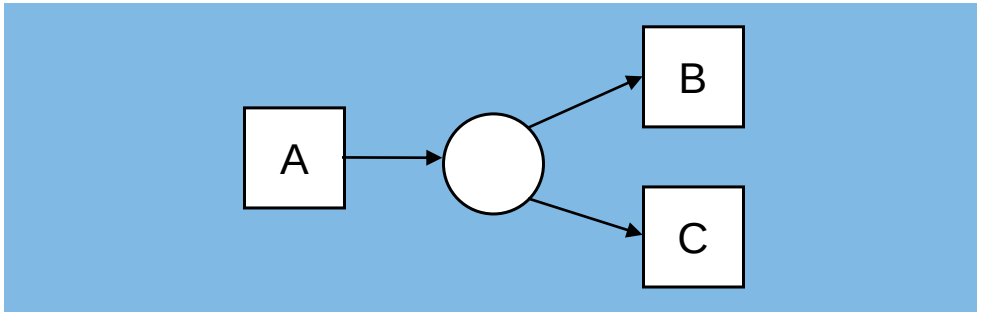
- $A \rightarrow B$, $A \rightarrow C$ und $B \| C$: **AND-Split**



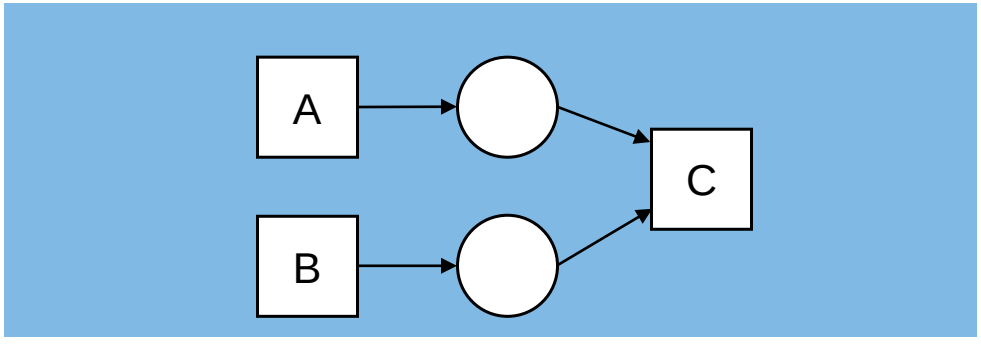
A → B (Kausalität): gdw. stets nur A>B und niemals B>A beobachtet wird.
A || B (Nebenläufigkeit): gdw. sowohl A>B als auch B>A beobachtet wird.
A # B (Auswahl): gdw. weder A>B als auch B>A beobachtet wird.

Process Discovery

- $A \rightarrow B$, $A \rightarrow C$ und $B \# C$: **XOR-Split**



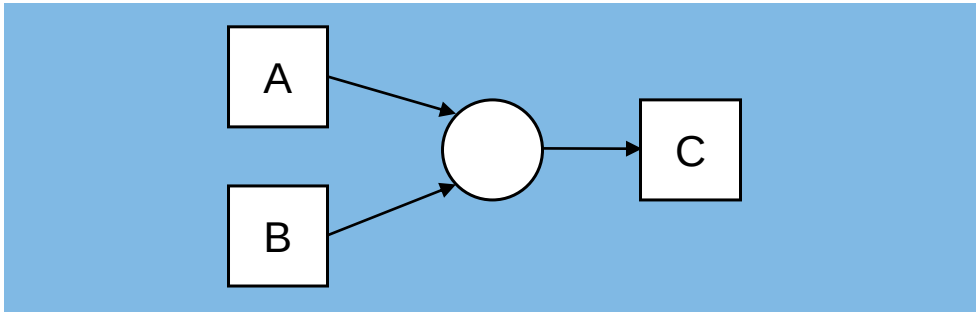
- $A \rightarrow C$, $B \rightarrow C$ und $A || B$: **AND-Join**



A → B (Kausalität): gdw. stets nur A>B und niemals B>A beobachtet wird.
A || B (Nebenläufigkeit): gdw. sowohl A>B als auch B>A beobachtet wird.
A # B (Auswahl): gdw. weder A>B als auch B>A beobachtet wird.

Process Discovery

- $A \rightarrow C$, $B \rightarrow C$ und $A \# B$: **XOR-Join**

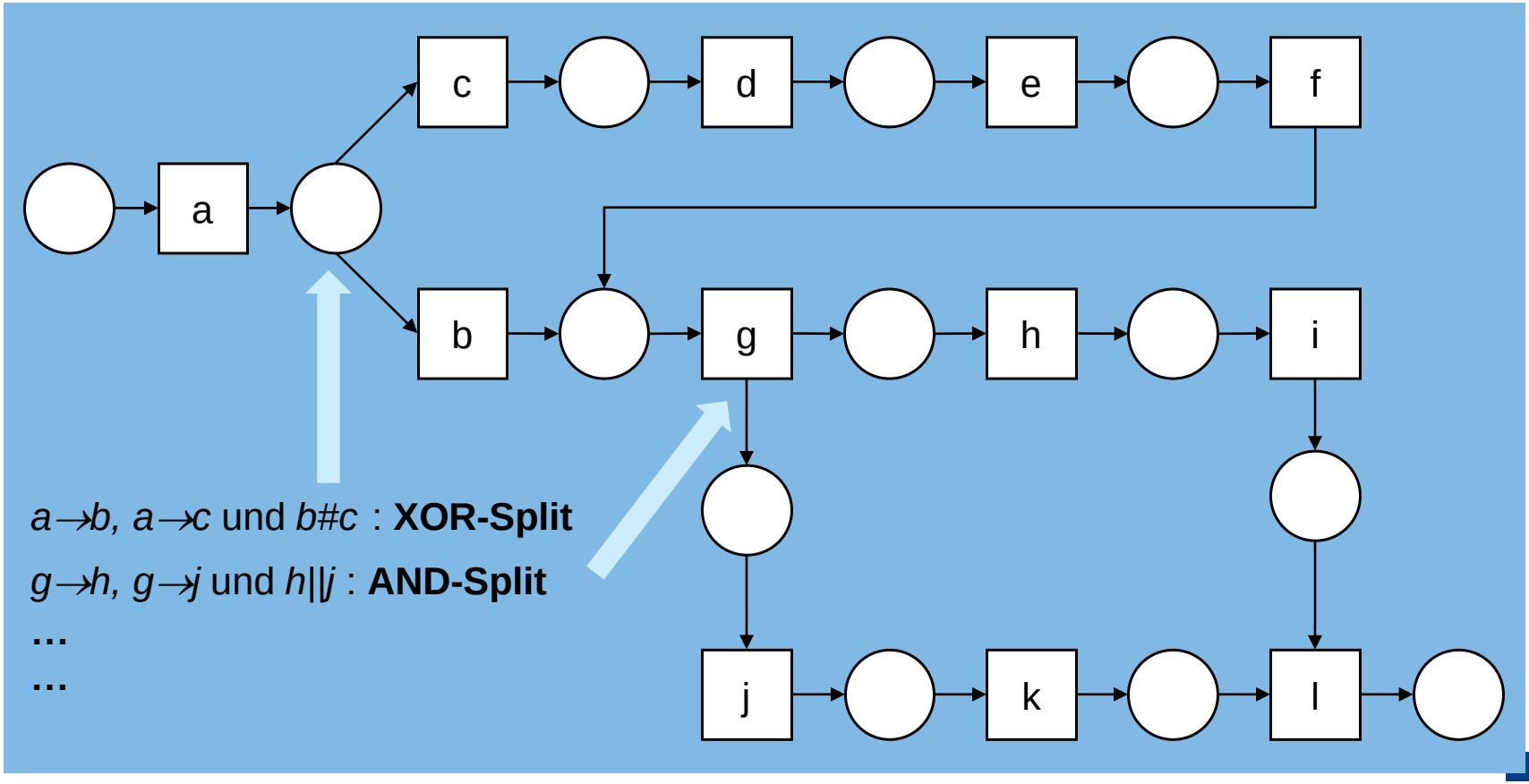


- Auf Grundlage der vorgestellten Kontrollflussmuster kann ein Workflow-Netz konstruiert werden.

A → B (Kausalität): gdw. stets nur A>B und niemals B>A beobachtet wird.
A || B (Nebenläufigkeit): gdw. sowohl A>B als auch B>A beobachtet wird.
A # B (Auswahl): gdw. weder A>B als auch B>A beobachtet wird.

Process Discovery

Christian-Albrechts-Universität zu Kiel



Process Discovery

1. Identifiziere alle Aktivitäten in dem Ereignislog (T_L)
2. Erstelle die Footprintmatrix
3. Identifiziere alle Aktivitäten, mit denen ein Fall beginnt (T_I)
4. Identifiziere alle Aktivitäten, mit denen ein Fall endet (T_O)
5. Identifiziere die potentielle Menge an Verbindungen (X_L)
   a) Füge alle Paare a → b hinzu
   b) Füge alle Tripel a → (b#c) hinzu (exklusive Auswahl)
   c) Füge alle Tripel (b#c) → d hinzu (einfache Zusammenführung)
6. Eliminiere alle überflüssigen Elemente (Y_L)
   1. Entferne a -> b und a -> c, wenn a -> (b#c) enthalten ist
   2. Entferne b -> d und c -> d, wenn (b#c) -> d enthalten ist
7. Erzeuge alle Stellen des Workflow-Netzes aus Y_L, inklusive der Start- und Endstellen (S_L)
8. Erzeuge die Flussrelation (F_L)
9. Somit ergibt sich das Workflow-Netz N = (S_L, T_L, F_L)

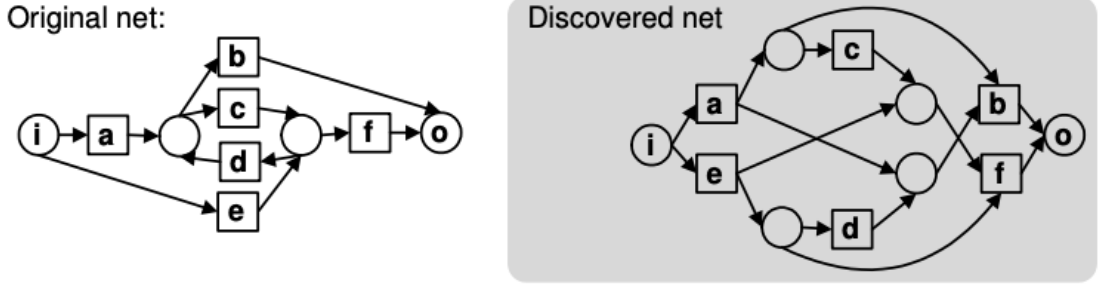
Process Discovery

- Die Konstruktion eines Workflow-Netzes gelingt unter den folgenden Voraussetzungen:
  - Die Anforderungen an ein Event Log sind erfüllt (chronologische Ordnung, Aktivitäten sind Fällen zugeordnet)
  - Jede Aktivität des Prozesses taucht in mindestens einem Fall im Event Log auf (Activity Completeness)
  - Wenn eine Aktivität B auf eine Aktivität A folgen kann, dann gibt es mindestens einen Fall in welchem wir dies beobachten können (Behavioral Completeness)
- Herausforderungen für den α-Algorithmus sind:
  - Schleifen, implizite Abhängigkeiten, duplizierte Aktivitäten
  - Kein Einbezug von Häufigkeiten im Auftreten von Ordnungsbeziehungen
  - Rauschen (Noise) im Event Log:
    - Falsch aufgezeichnete Aktivitäten
    - Abweichungen von der vorgesehenen Prozessausführung

Process Discovery

Schleifen → α+-Algorithmus

- Der α-Algorithmus kann kurze Schleifen (siehe "Original net") nicht erkennen (siehe „Discovered net")



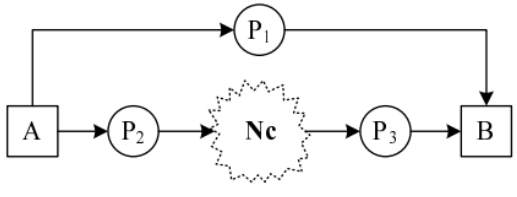
- Der α+-Algorithmus kann Prozessmodelle mit Schleifen der Länge 1 und Schleifen der Länge 2 entdecken, indem die Ordnungsbeziehungen des α-Algorithmus umdefiniert und um zwei weitere Ordnungsbeziehungen ergänzt wurden

Quelle: Alves de Medeiros, A; Dongen, B.F.; van der Aalst, W. M. P.; Weijters, A. J. M. M: Process Mining: Extending the Alpha-algorithm to mine short loops, Information Systems IE&IS, TU Eindhoven, Report, 2004.

Process Discovery

Implizite Abhängigkeiten → α++-Algorithmus

- Im unten gezeigten Modell gibt es eine implizite Abhängigkeit zwischen den Aktivitäten A und B:
  - Sobald Aktivität A ausgeführt wurde, müssen erst andere Aktivitäten N_c ausgeführt werden, bevor Aktivität B ausgeführt werden kann. Erst danach wird B ausgeführt.
  - Somit kann niemals die direkte Nachfolge von B auf A (A > B) beobachtet werden



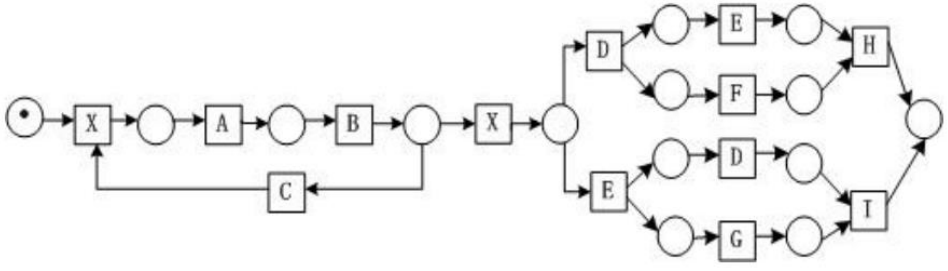
- Der α++-Algorithmus kann Prozessmodelle mit Nicht-Free-Choice-Konstrukten erstellen, indem die Ordnungsbeziehungen des α+-Algorithmus erweitert werden, um implizite Abhängigkeiten entdecken zu können

Quelle: Wen, L., Wang, J., Sun, J: Detecting Implicit Dependencies Between Tasks from Event Logs, AP Web 2006, Springer, S. 591-603.

C | A | U
Christian-Albrechts-Universität zu Kiel

Duplizierte Aktivitäten → α*-Algorithmus

- Im unten gezeigten Modell treten duplizierte Aktivitäten auf:
  - Aktivität X: 2x
  - Aktivität E: 2x
  - Aktivität D: 2x



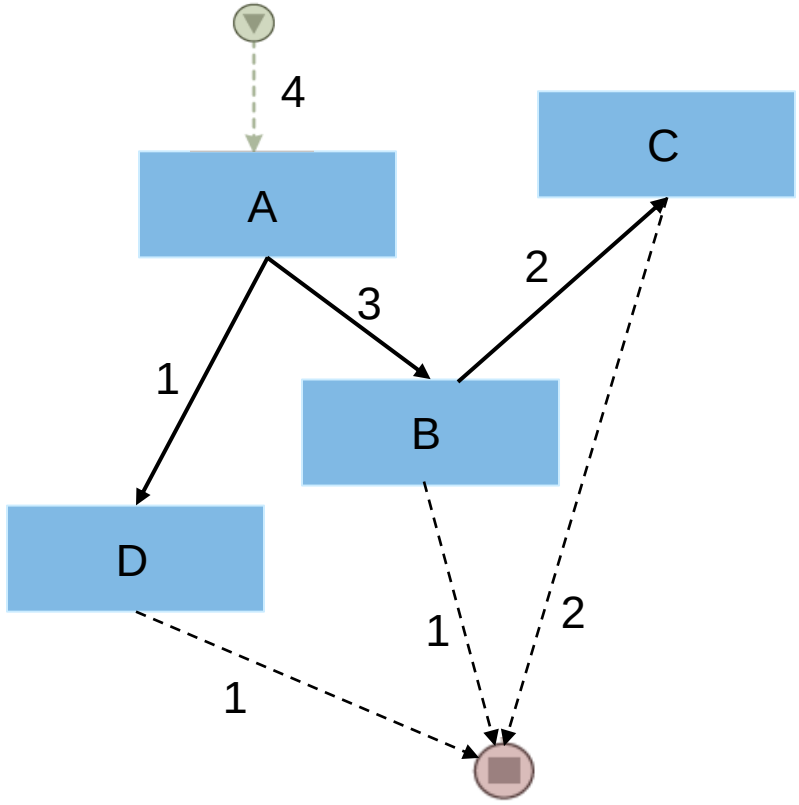
- Der α*-Algorithmus kann Prozessmodelle mit duplizierten Aktivitäten erstellen, indem duplizierte Aktivitäten zunächst durch eine Heuristik identifiziert und umbenannt werden, der klassische α-Algorithmus ausgeführt wird, und anschließend die umbenannten Aktivitäten wieder ihren ursprünglichen Namen erhalten.

Quelle: Li, J., Liu, D., Yang B.: Process Mining: Extending α-Algorithm to Mine Duplicate Tasks in Process Logs. AP-Web 2007 / WAIM 2007, Springer, S. 396-407, 2007.

Directly-Follows Graph

Christian-Albrechts-Universität zu Kiel

Relation	Häufigkeit
(-,A)	4
(A,B)	3
(A,D)	1
(B,C)	2
(A,-)	1
(B,-)	1
(C,-)	2
(D,-)	1



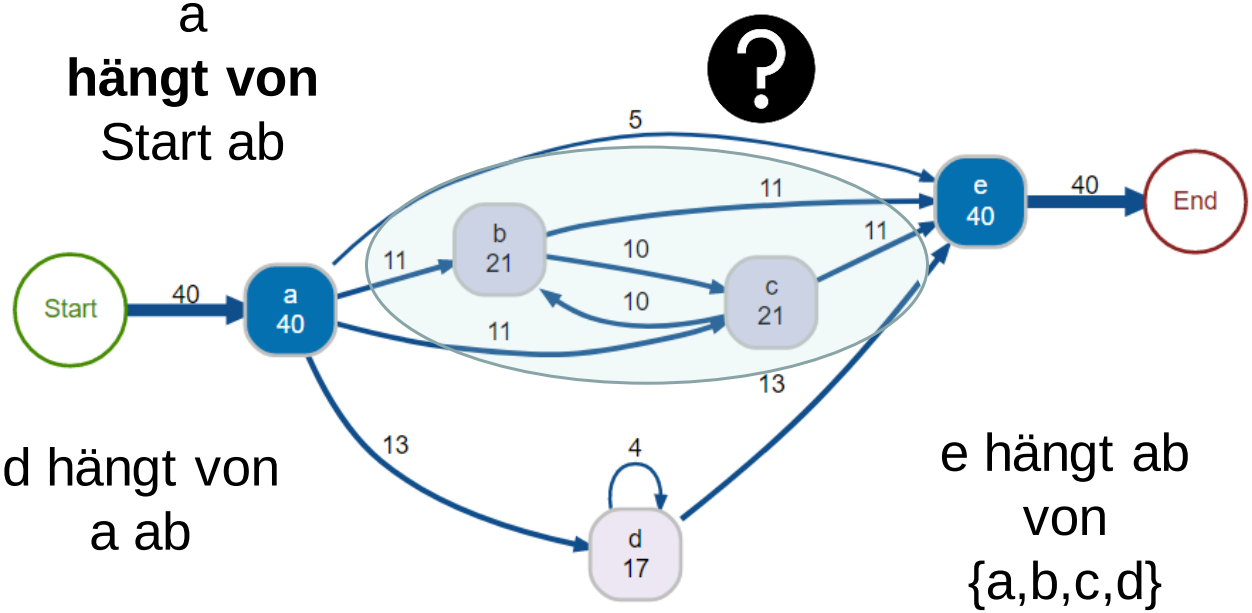
- beschreibt, welche Aktivitäten direkt aufeinander folgen und mit welchen Aktivitäten ein Trace beginnt oder endet

Heuristic Miner

Christian-Albrechts-Universität zu Kiel

- Versuche, "kausale" Abhängigkeiten zu erfassen, anstatt Vorgänger/Nachfolger Beziehungen
- konzentriert sich auf die Berechnung der Abhängigkeitshäufigkeit

a **hängt von** Start ab



d hängt von a ab

e hängt ab von {a,b,c,d}

Heuristic Miner

Christian-Albrechts-Universität zu Kiel

Abhängigkeitsmaß

$$a \Rightarrow^L b = \begin{cases} \dfrac{|a >^L b| - |b >^L a|}{|a >^L b| + |b >^L a| + 1}, & \text{for } a \neq b \\ \dfrac{|a >^L a|}{|a >^L a| + 1} & \text{otherwise} \end{cases}$$

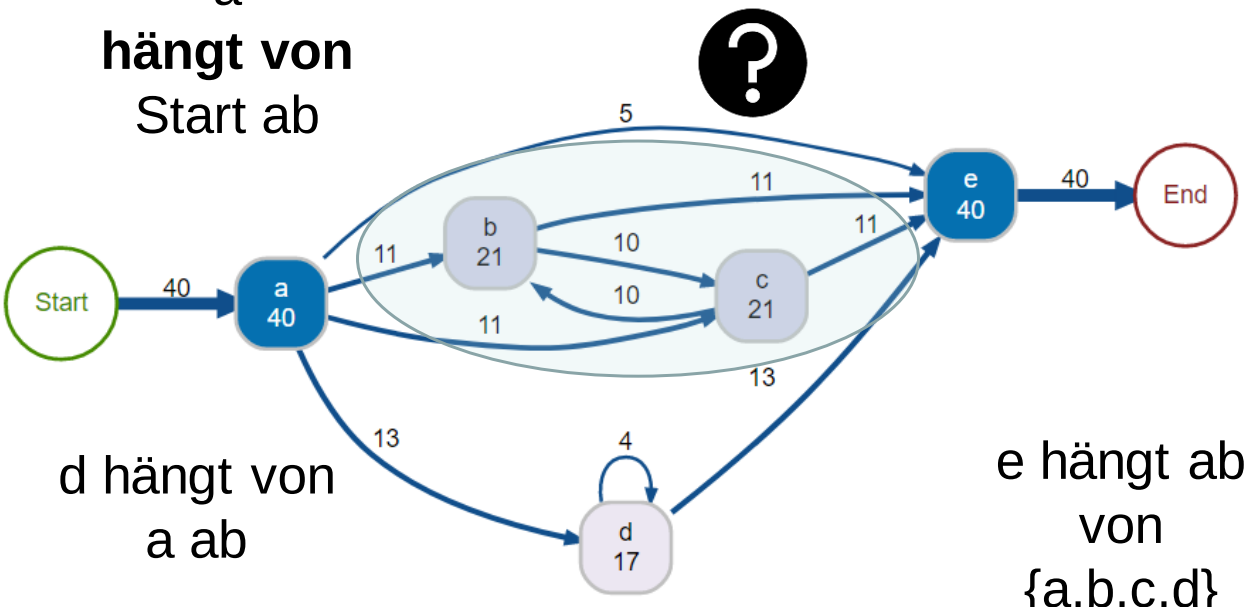
L	Ereignislog
a and **b**	Aktivitäten / Aktivitätsidentifizier
$a >^L b$	a directly-follows b
$\lvert \rvert$	Häufigkeit / Kardinalität

Anwendung des Abhängigkeitsmaß

Christian-Albrechts-Universität zu Kiel

$$a \Rightarrow^L b = \begin{cases} \dfrac{|a >^L b| - |b >^L a|}{|a >^L b| + |b >^L a| + 1}, & \text{for } a \neq b \\ \dfrac{|a >^L a|}{|a >^L a| + 1} & \text{otherwise} \end{cases}$$

a **hängt von** Start ab



d hängt von a ab

e hängt ab von {a,b,c,d}

Heuristic Miner

Christian-Albrechts-Universität zu Kiel

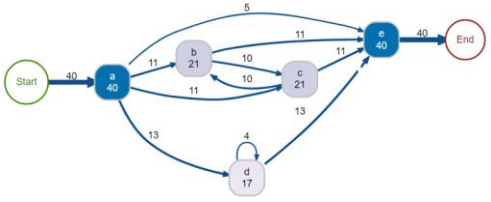
$$a \Rightarrow^L b = \begin{cases} \dfrac{|a >^L b| - |b >^L a|}{|a >^L b| + |b >^L a| + 1}, & \text{for } a \neq b \\ \dfrac{|a >^L a|}{|a >^L a| + 1} & \text{otherwise} \end{cases}$$

Intuition

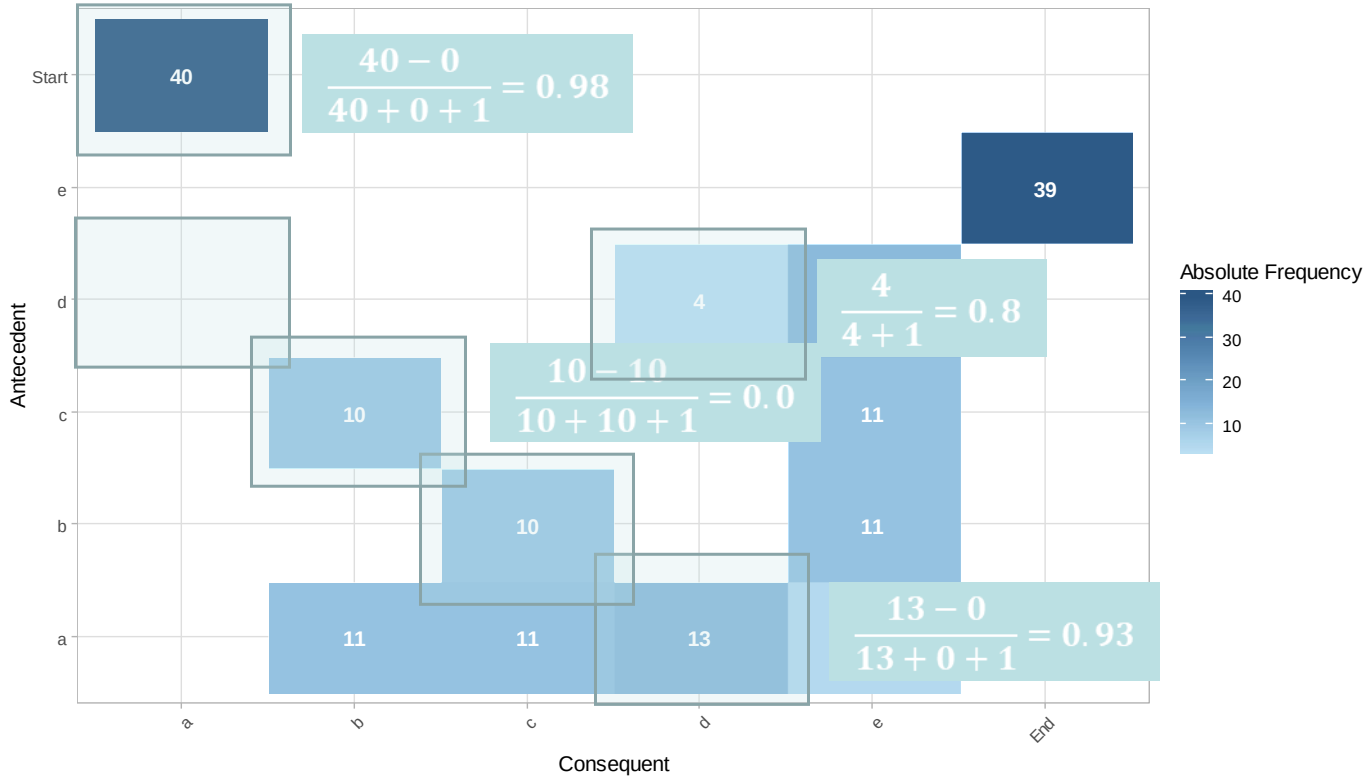
- $|a \Rightarrow^L b|$ goes towards 1 ➔ Causal
- $|a \Rightarrow^L b|$ goes towards 0 ➔ Parallel
- $|a \Rightarrow^L b|$ goes towards -1 ➔ Inverse Causal

Heuristic Miner

Christian-Albrechts-Universität zu Kiel

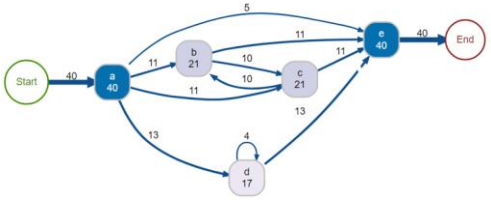


$$a \Rightarrow^{L} b = \begin{cases} \dfrac{|a >^{L} b| - |b >^{L} a|}{|a >^{L} b| + |b >^{L} a| + 1}, & \text{for } a \neq b \\ \dfrac{|a >^{L} a|}{|a >^{L} a| + 1} & \text{otherwise} \end{cases}$$

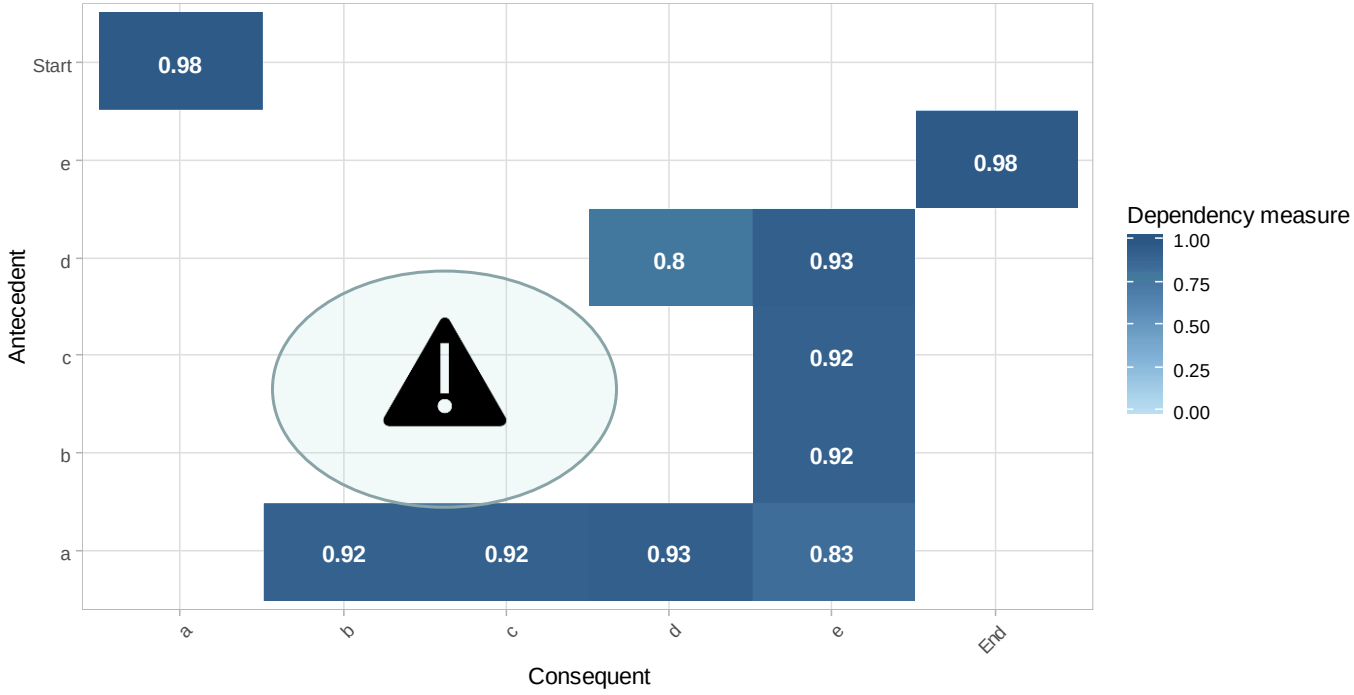


Heuristic Miner

Christian-Albrechts-Universität zu Kiel



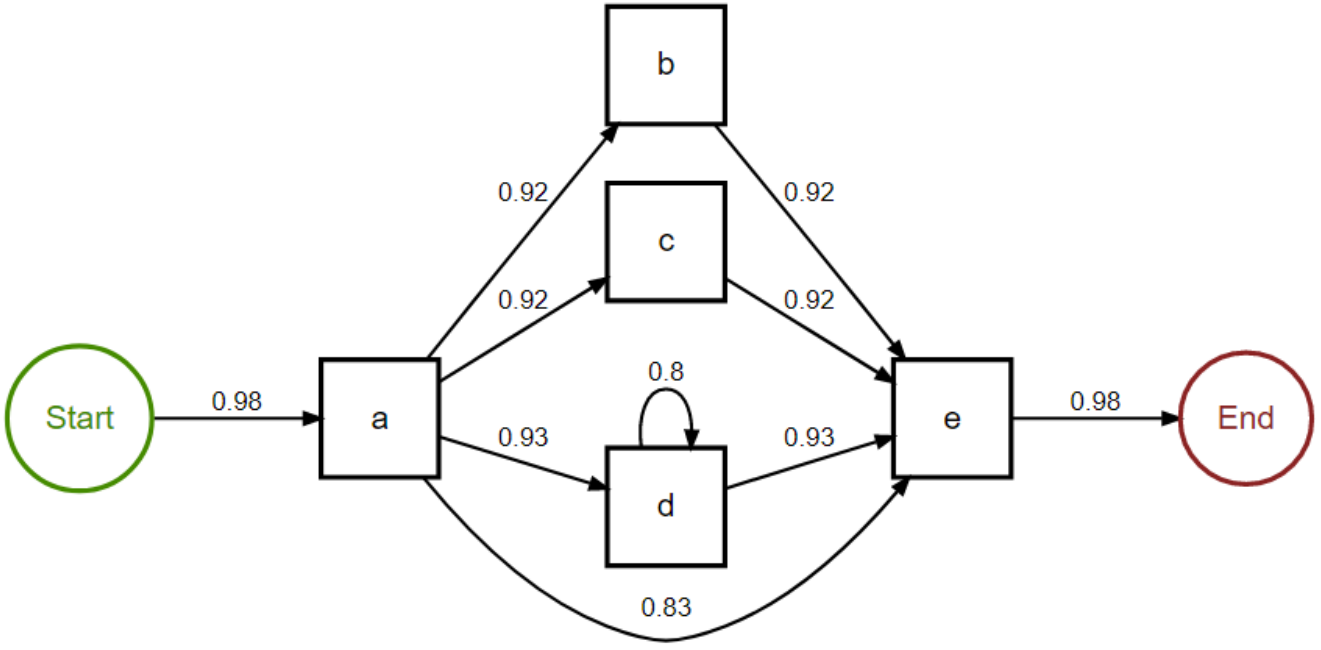
$$a \Rightarrow^{L} b = \begin{cases} \dfrac{|a >^{L} b| - |b >^{L} a|}{|a >^{L} b| + |b >^{L} a| + 1}, \text{for } a \neq b \\ \dfrac{|a >^{L} a|}{|a >^{L} a| + 1} \quad \text{otherwise} \end{cases}$$



Heuristic Miner

Christian-Albrechts-Universität zu Kiel

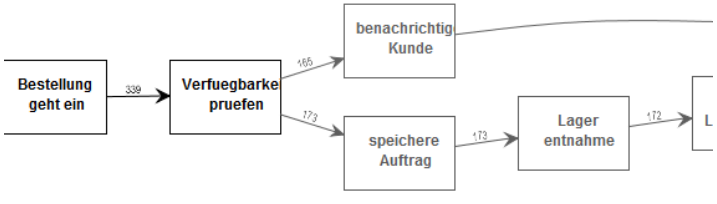
Output – Abhängigkeitsgraph



Process Discovery

Heuristic Mining

- Berücksichtigung der Häufigkeiten von Ereignissen und Sequenzen bei der Konstruktion eines Prozessmodells
  - Erstellung einer Häufigkeitentabelle
  - Ableitung der Ordnungsrelationen auf dieser Tabelle
  - Erstellung eines Prozessmodells (WF-Netz oder Causal Net)
- Daher vergleichsweise unempfindlich gegenüber Rauschen (Noise) und behavioural uncompleteness (Log muss ein Beispiel für zu findendes Verhalten aufführen)

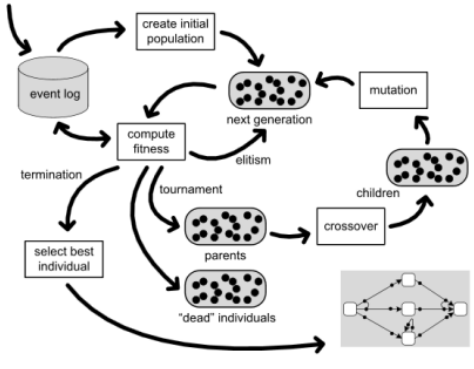


- Das abgebildete Modell ist ein Causal Net (Knoten = Aktivitäten, Kanten = kausale Abhängigkeiten), die Häufigkeit der Kanten-traversierung durch das Log ist durch die Kanteninschriften dargestellt

Process Discovery

Genetic Mining

- Zufällige Erstellung von Prozessmodellen als mögliche Kandidaten für endgültiges Modell
- Berechnung der „Fitness“ des Modells in Bezug auf das Event Log (mehr dazu im nächsten Abschnitt Conformance Checking)
- Veränderung der Lösungen durch genetische Operatoren (Crossover, Mutation) und so schrittweise Annäherung an ein Prozessmodell mit der höchsten Fitness



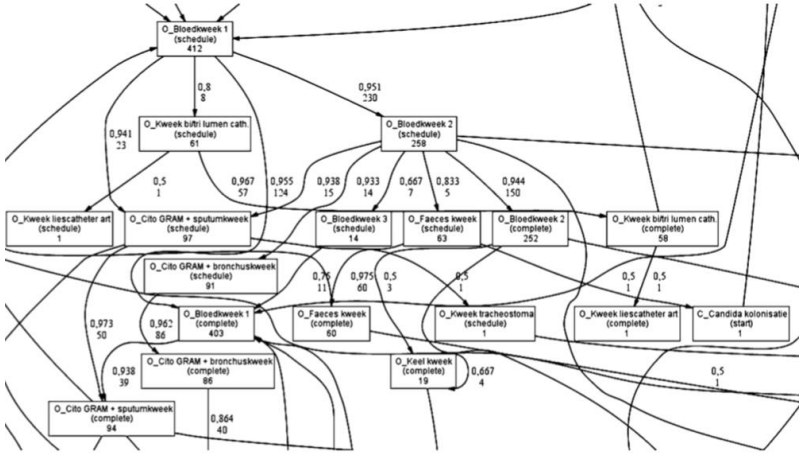
Quelle: Process Mining – Discovery, Conformance and Enhancement of Business Processes, van der Aalst, Springer 2011

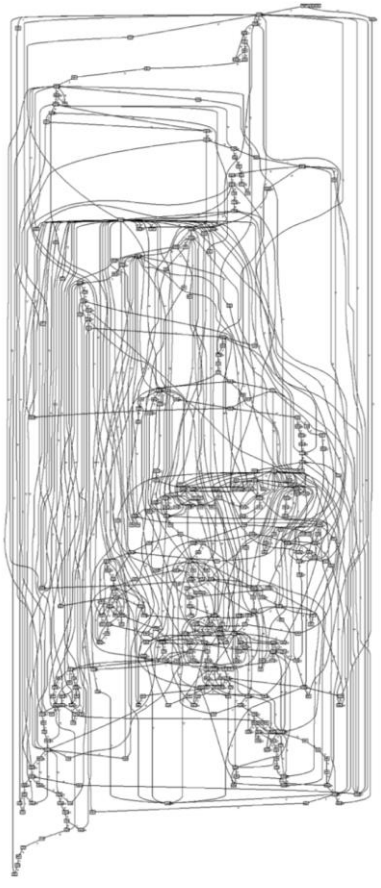
Process Discovery

Christian-Albrechts-Universität zu Kiel

Spaghetti-Modelle

- Viele Discovery-Verfahren produzieren sogenannte Spaghetti-Modelle, sehr unstrukturierte Prozesse mit vielen Pfaden und Aktivitäten
- Spaghetti-Prozesse finden sich häufig auf den Gebieten Produktentwicklung, Service, Ressource Management und Vertieb/CRM





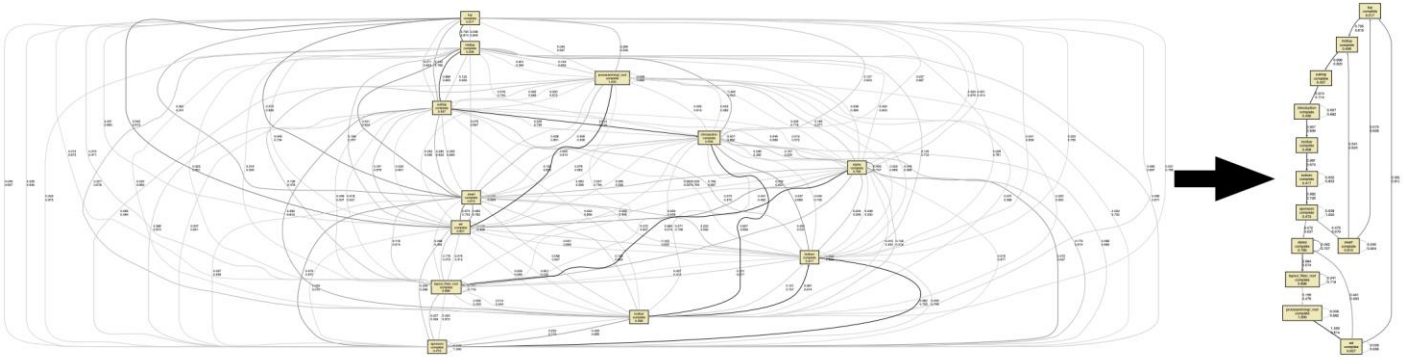
Quelle: Van der Aalst, Wil M. P: Process Mining: Data Science in Action. Springer. S. 14 und 422, 2016.

Process Discovery

Christian-Albrechts-Universität zu Kiel

Fuzzy Mining

- Spaghetti-Modelle als Discovery-Ergebnis sind für Endanwender zu unübersichtlich und schwer interpretierbar
- Das sogenannte Fuzzy Mining erlaubt über Schieberegler ein „Zoomen“ im Modell für eine bessere Übersicht der häufig genutzten Pfade und/oder der häufig auftretenden Aktivitäten
- Fuzzy Mining ist daher gut geeignet für komplexe, wenig strukturierte Prozesse



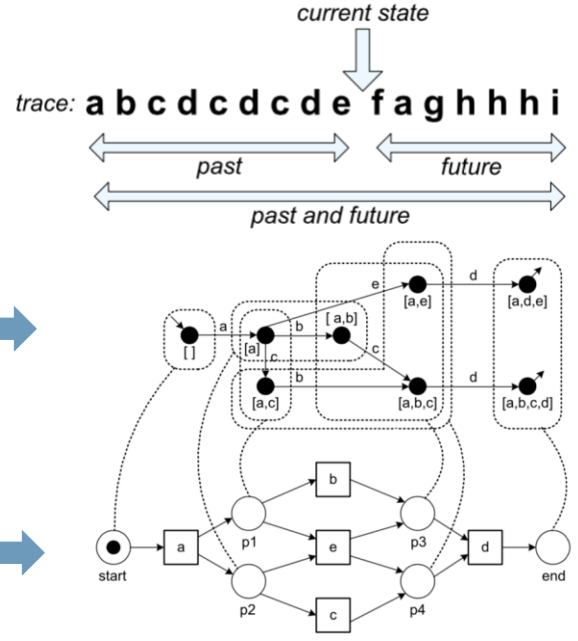
Quelle: Günther, C.W., van der Aalst, W.M.P.: Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics, BPM 2007, Springer, S. 328-343, 2007.

Process Discovery

Christian-Albrechts-Universität zu Kiel

Region based Mining (state-based)

- Der Alpha-Algorithmus basiert auf dem Erstellen einer Footprint-Matrix, d.h. auf dem Identifizieren kausaler Zusammenhänge zwischen Prozessaktivitäten
- Beim sogenannten State-based Region Mining werden stattdessen die Zustände im Prozess betrachtet (Zustand = Stelle im Petri-Netz):
  - Dabei entspricht jede Position (vor der ersten, nach der letzten und jeweils zwischen zwei Aktivitäten) im Trace einem Zustand
  - Für diese Zustände wird in einem ersten Schritt ein Transitionssystem erstellt und anschließend in einem zweiten Schritt aus diesem ein Petri-Netz konstruiert
  - Die Ergebnisse sind vergleichbar mit denen anderer Discovery-Algorithmen



Quelle: Van der Aalst, Wil M. P: Process Mining: Data Science in Action. Springer. S. 212ff, 2016.

Process Discovery

Christian-Albrechts-Universität zu Kiel

Region based Mining (language-based)

- Das Language-based Region Mining bestimmt ebenso die Stellen im Petri-Netz, allerdings nicht auf der Basis eines Transitionssystems, sondern auf Basis einer Sprache
- Dabei wird ein Petri-Netz konstruiert, welches für die gegebene Sprache (abgeleitet aus dem Event Log) das minimale Netzverhalten abbildet:
  - Kernidee: Das Einfügen von Stellen in einem Petri-Netz bestehend aus nur Aktivitäten in einem Flower-Modell (siehe rechts) beschränkt das Verhalten
  - Frage: Welche Stellen können eingefügt werden, so dass das Verhalten im Log abgebildet wird?
  - Lösung: Anwendung von Verfahren der linearen Optimierung aus dem OR

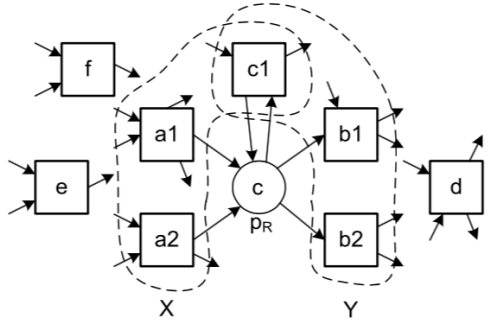
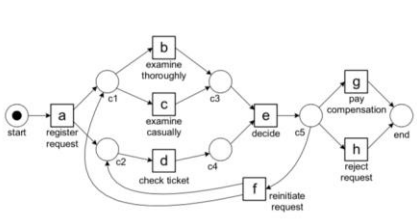


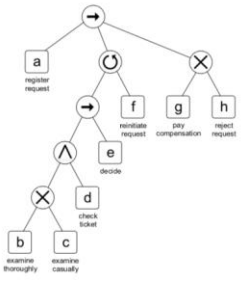
Bild: Van der Aalst, Wil M. P: Process Mining: Data Science in Action. Springer. S. 219, 2016.
Quelle: Bergenthum, R., Desel, J., Lorenz, R., Mauser, S.: Process Mining based on Regions of Languages. BPM 2007, Springer, S. 375-383, 2007.

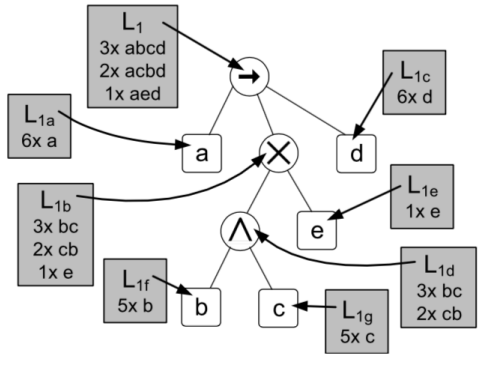
Process Discovery

Inductive Mining

- Petri-Netze, WF-Netze, BPMN-Modelle, EPKs, … können Anomalien wie Deadlocks, Livelocks usw. aufweisen
- Sogenannte Prozessbäume (process trees) erfüllen dahingegen garantiert die Soundness-Eigenschaft
- Kernidee: Wiederholtes Identifizieren des wichtigsten Trenners (“split“) im Event Log und Bestimmen des Operators, Fortfahren damit auf den dadurch entstehenden zwei Sub-Event-Logs



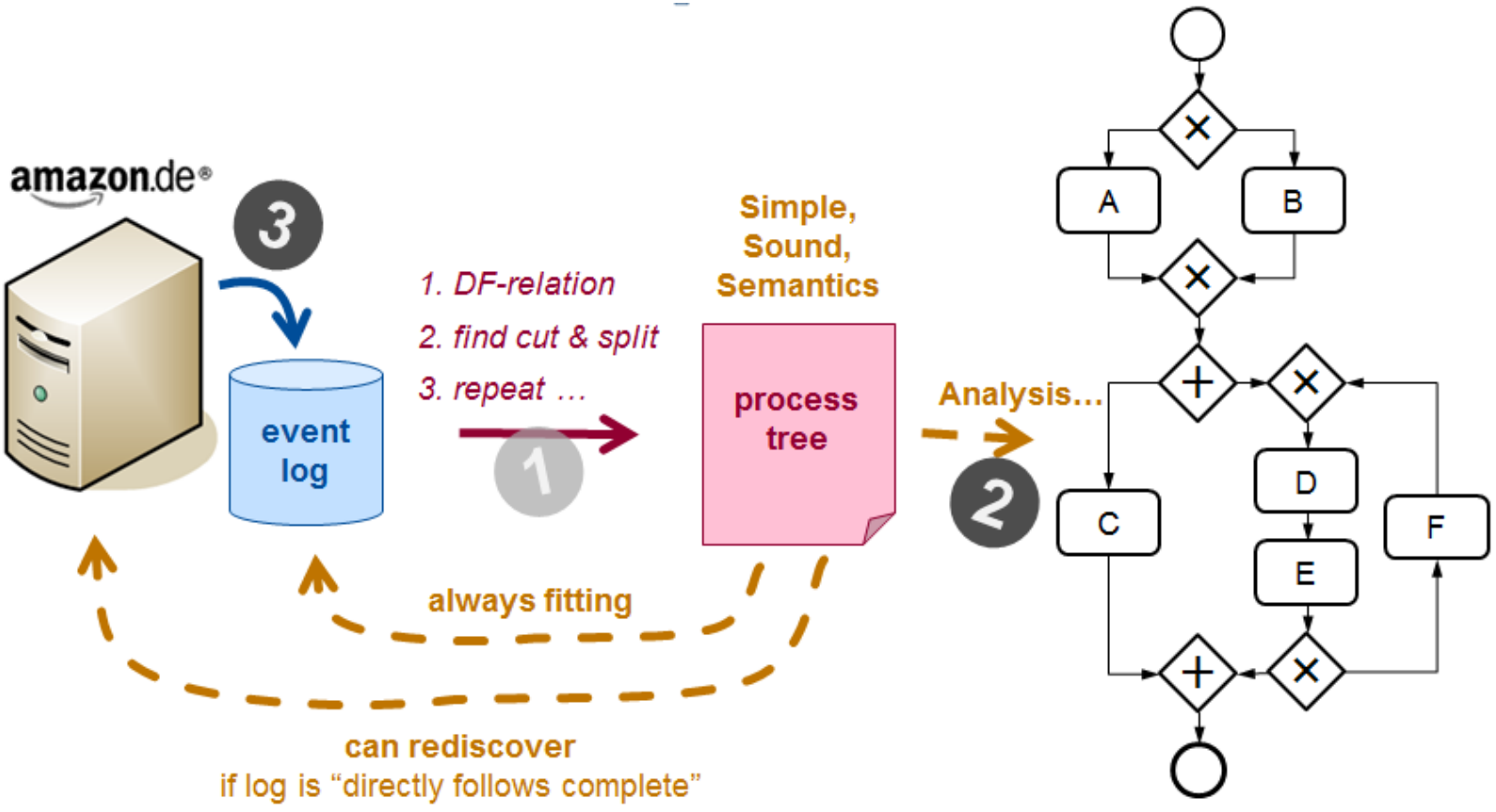




- Aus dem resultierenden Prozessbaum kann ein Petri-Netz abgeleitet werden, welches die Soundness-Eigenschaften erfüllt

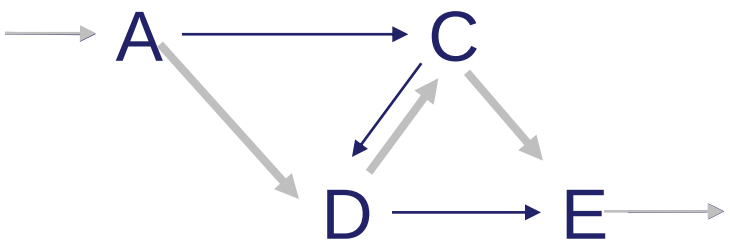
Bilder: Van der Aalst, Wil M. P: Process Mining: Data Science in Action. Springer. S. 224, S. 230, 2016.

Inductive Miner



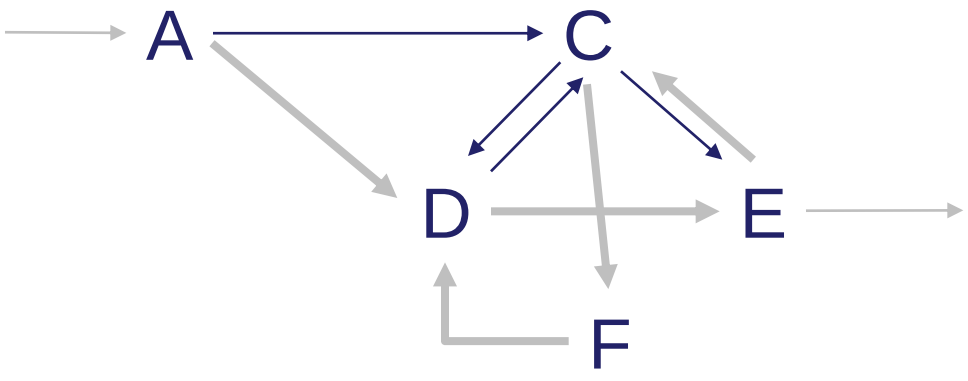
Bottom-Up Discovery: Directly-Follows Relation

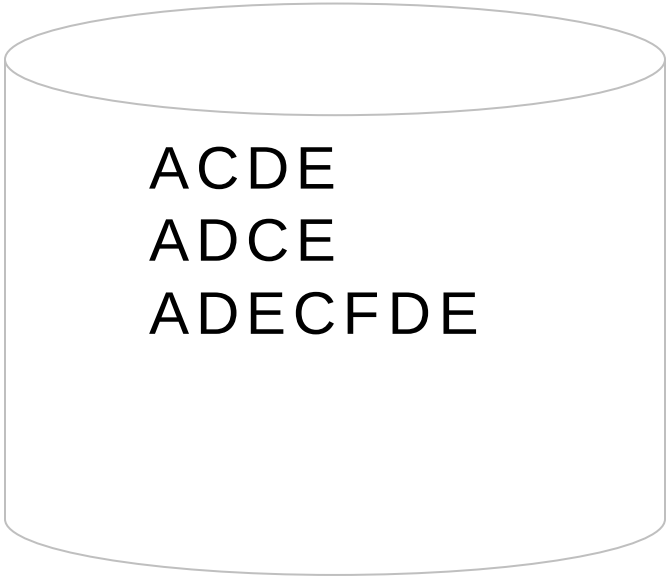
Christian-Albrechts-Universität zu Kiel



Bottom-Up Discovery: Directly-Follows Relation

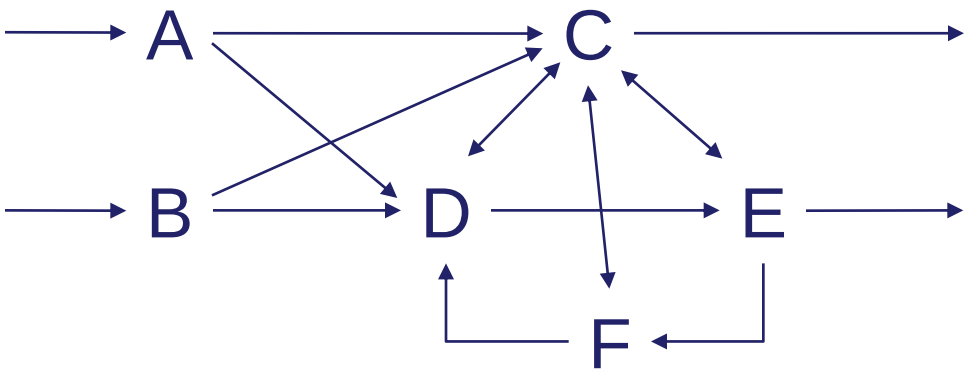
Christian-Albrechts-Universität zu Kiel

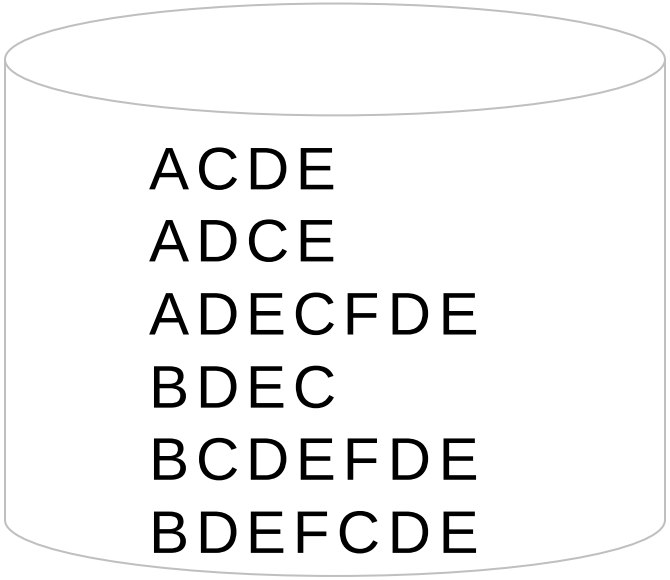




Bottom-Up Discovery: Directly-Follows Relation

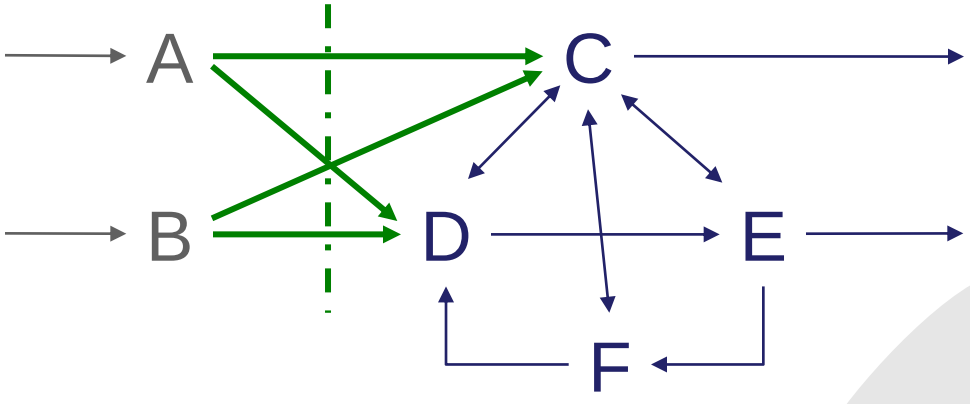
Christian-Albrechts-Universität zu Kiel

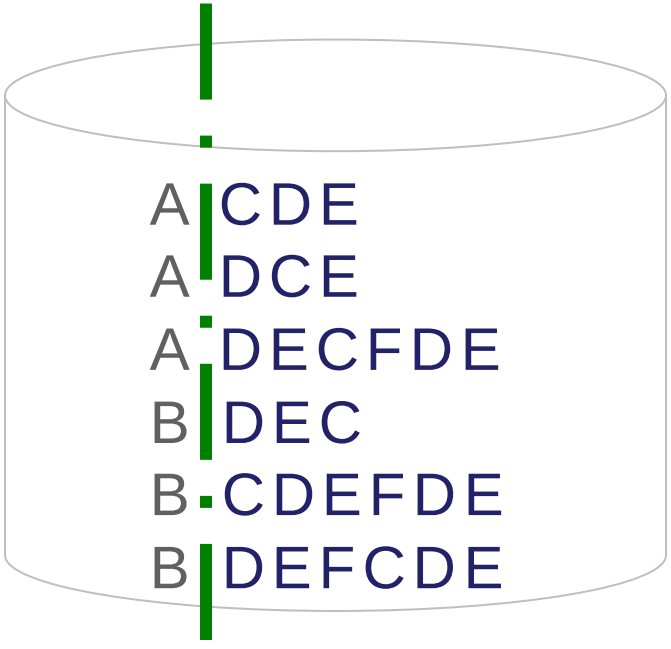


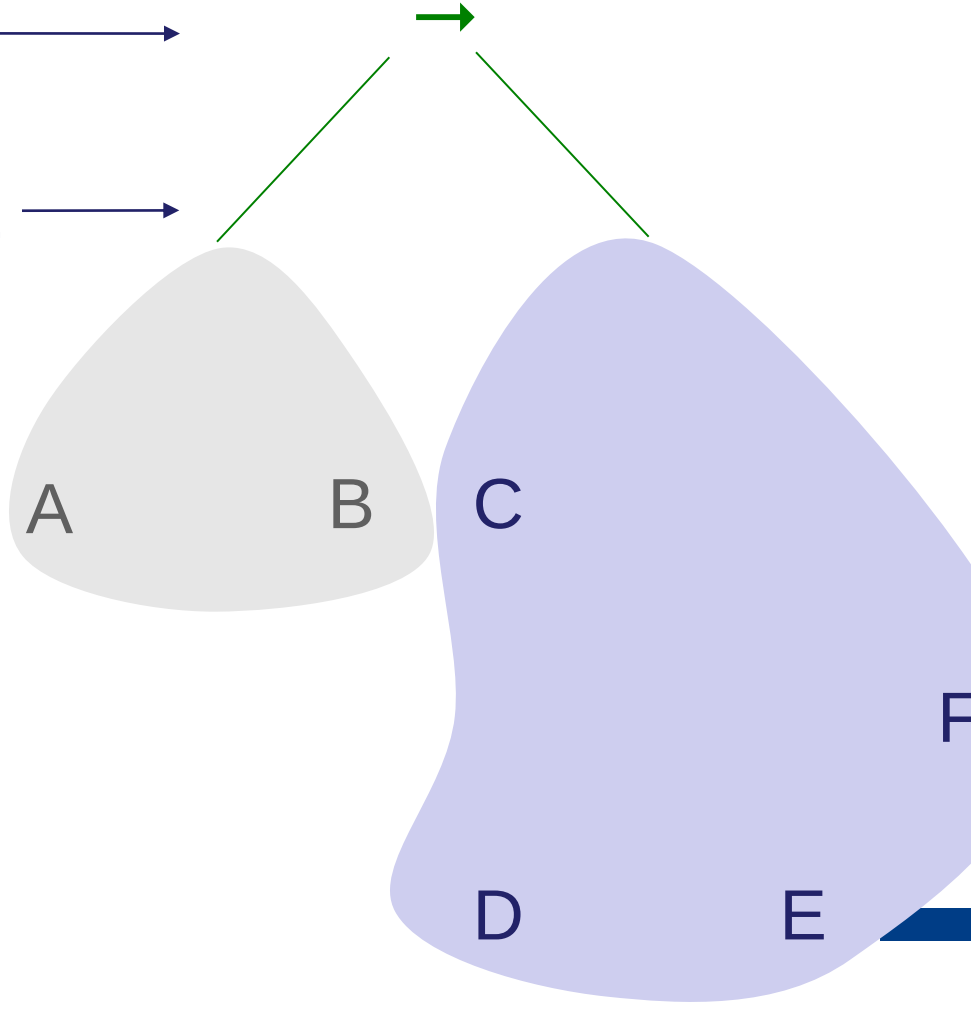


Dominant Behavioral Relation: Sequence Cut

Christian-Albrechts-Universität zu Kiel

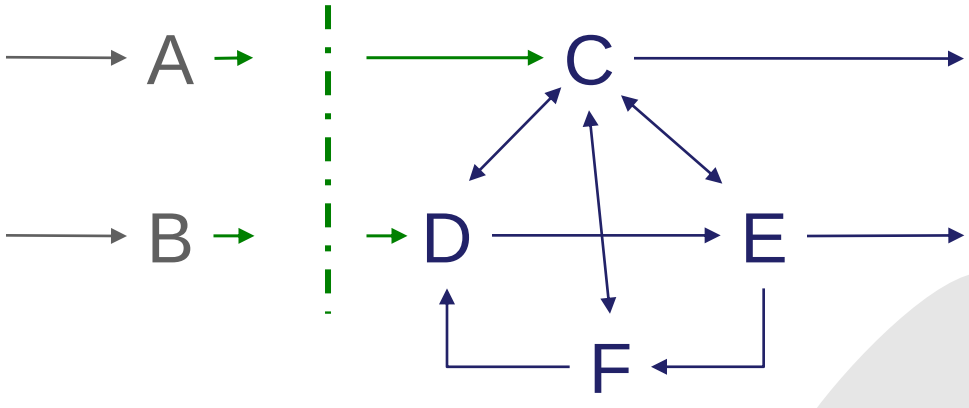


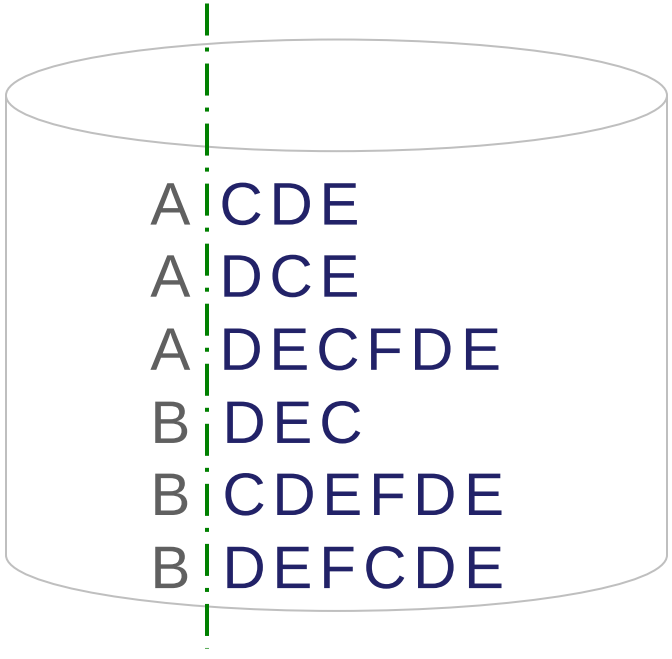


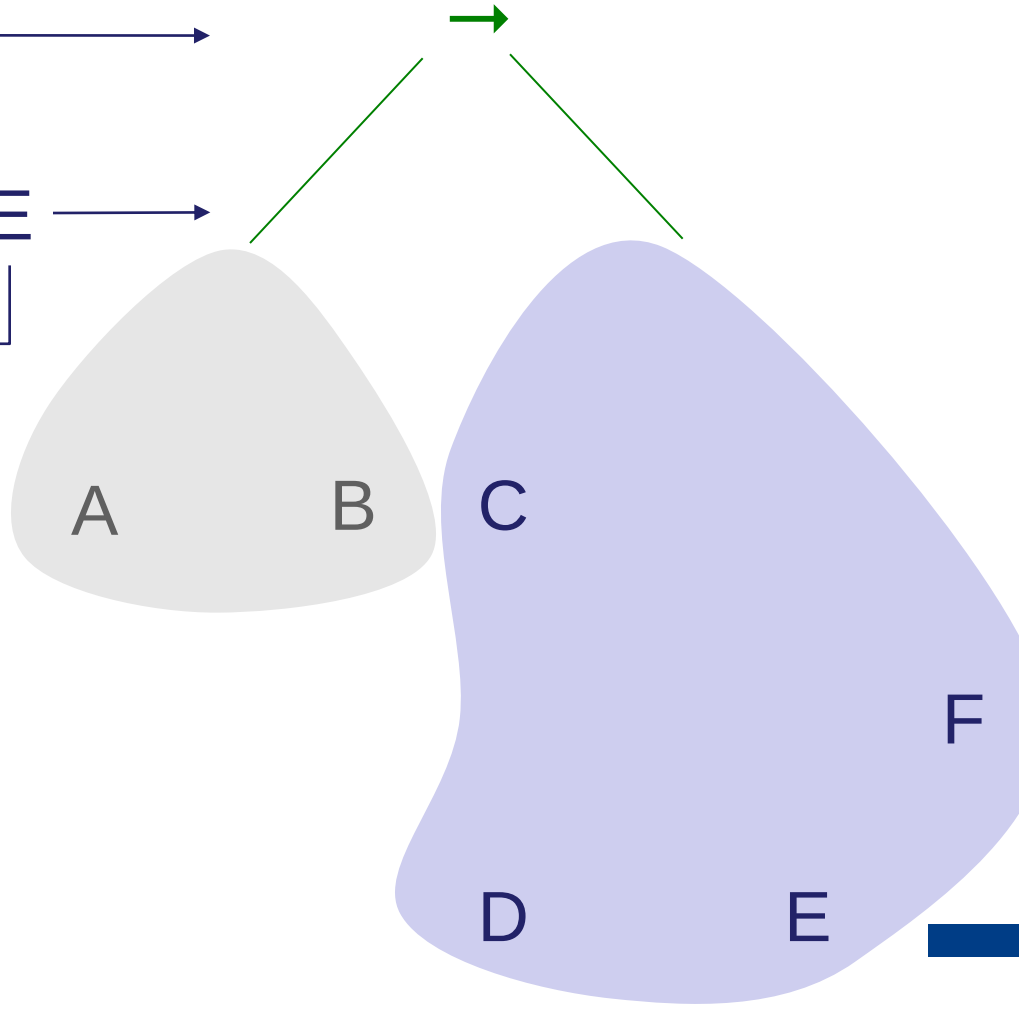


Split Along Cut & Recurse

Christian-Albrechts-Universität zu Kiel

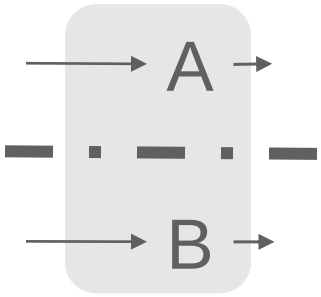


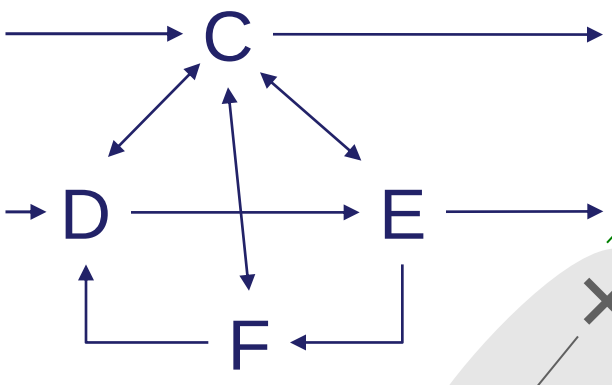


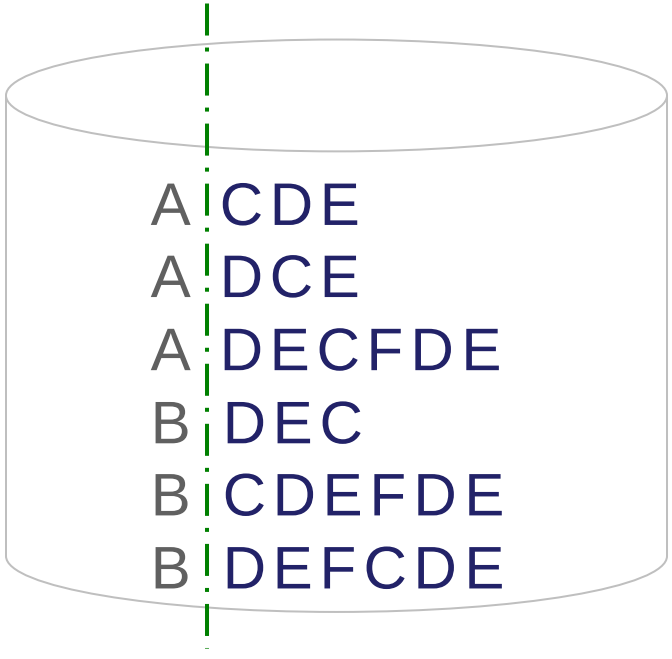


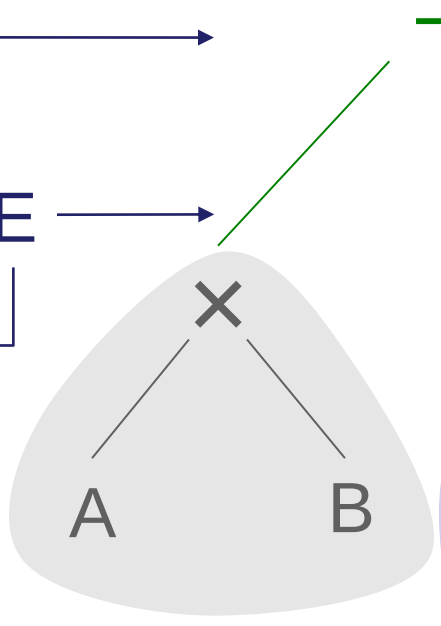
Choice Cut & Base Case

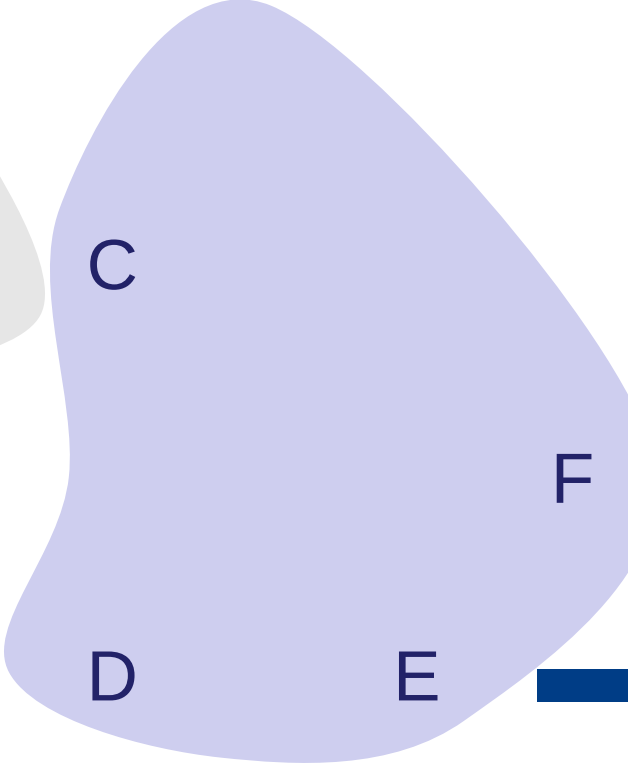
Christian-Albrechts-Universität zu Kiel





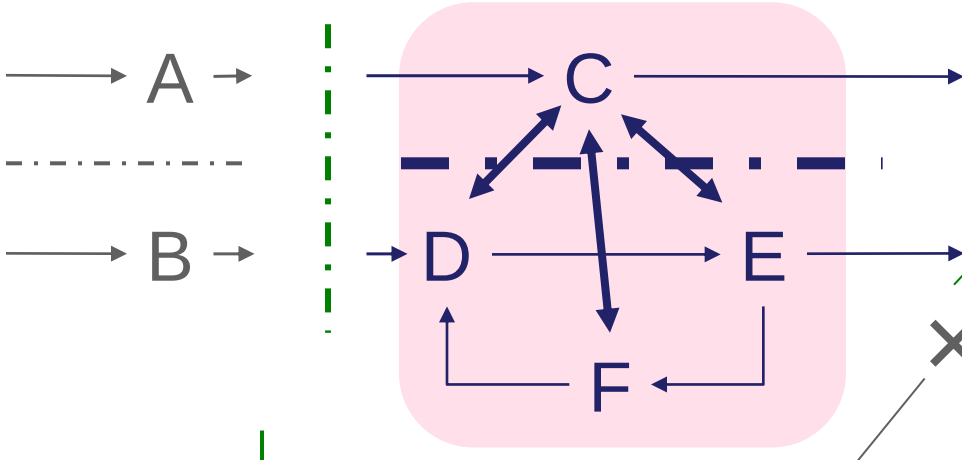


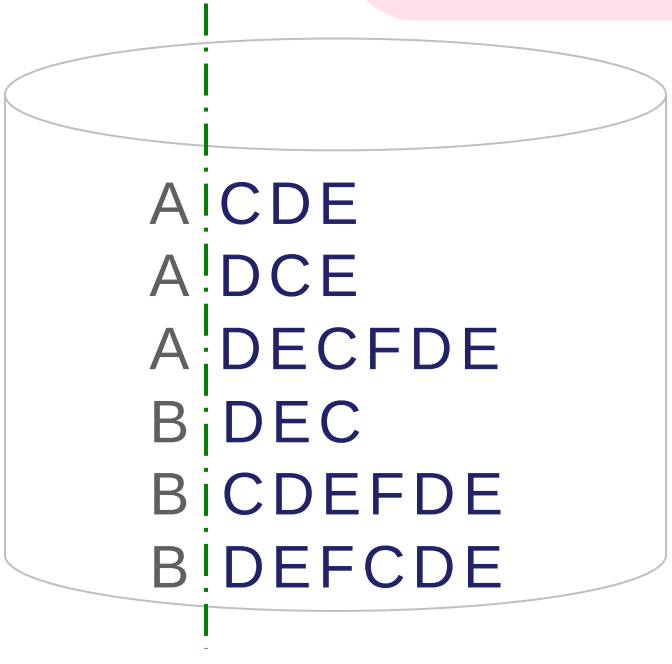


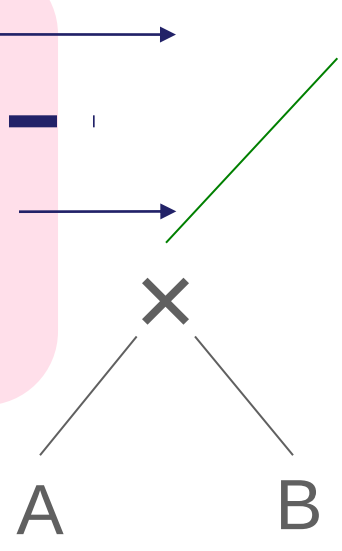


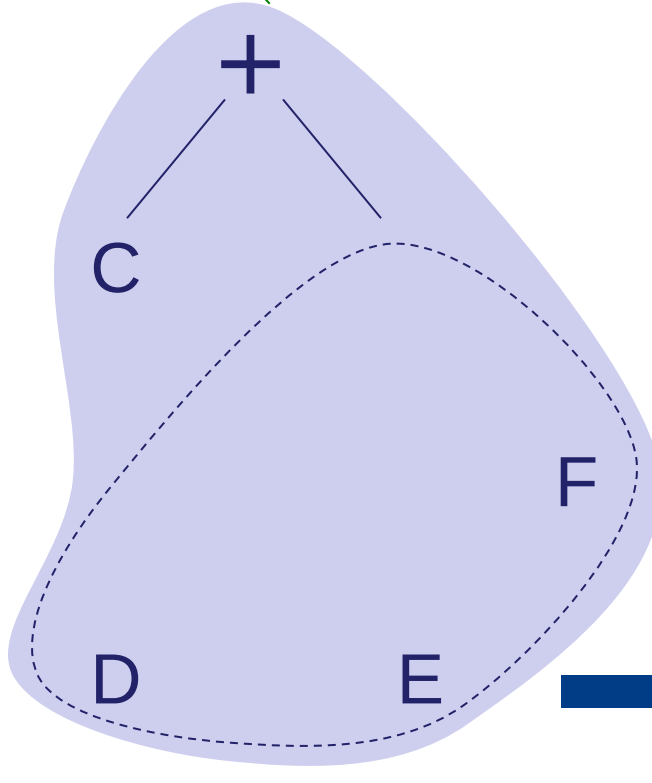
Parallel Cut

Christian-Albrechts-Universität zu Kiel



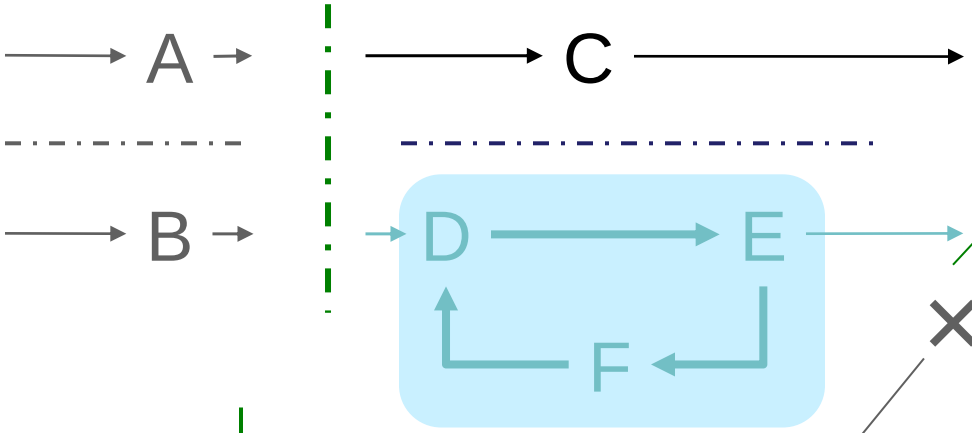


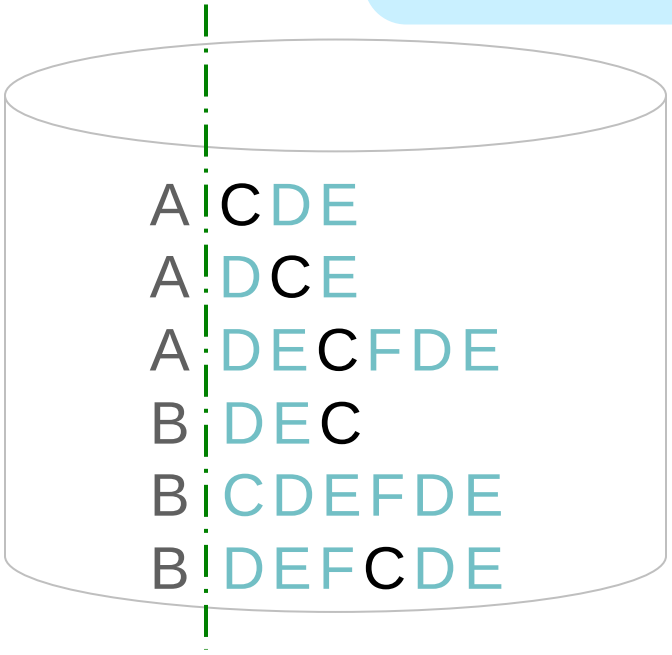


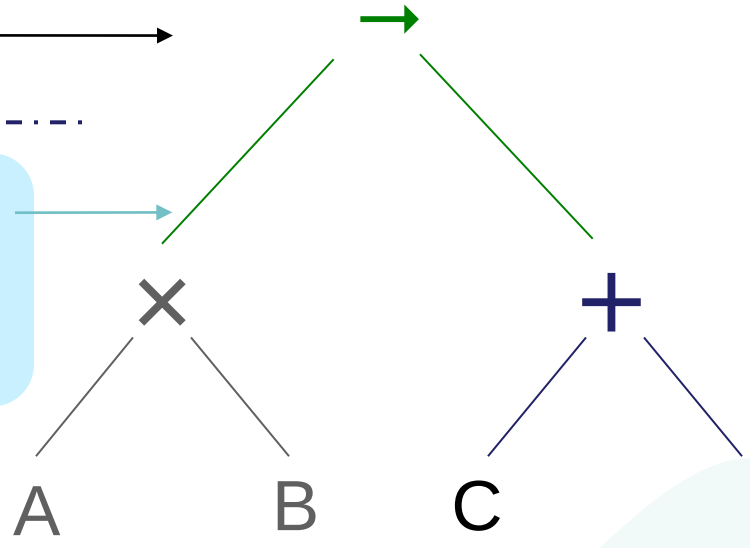


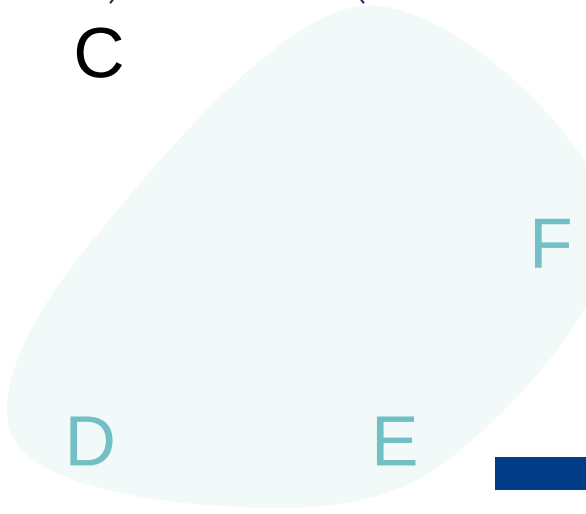
Loop Cut

Christian-Albrechts-Universität zu Kiel



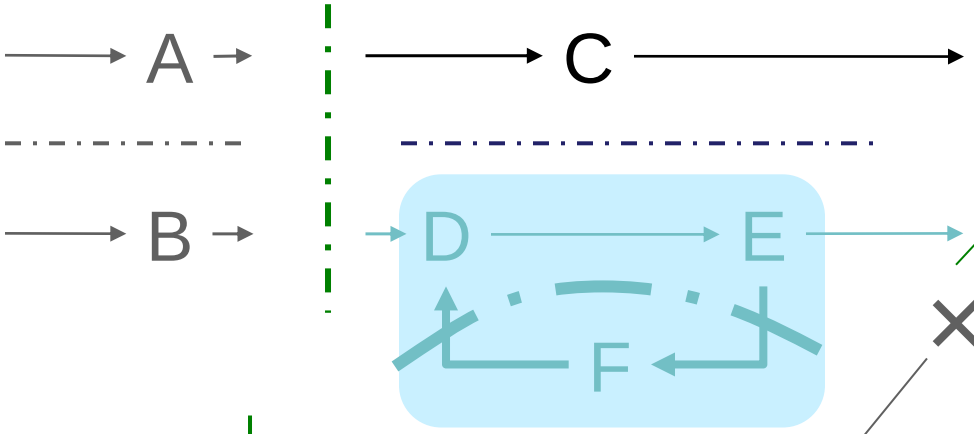


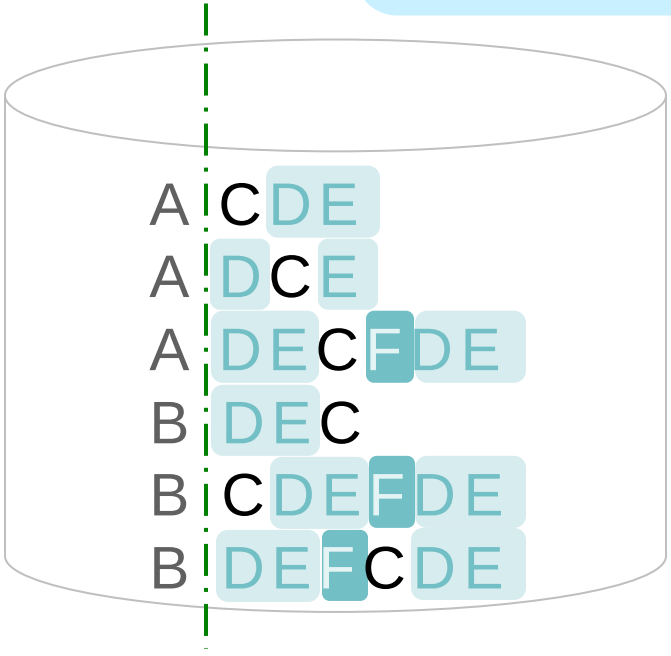


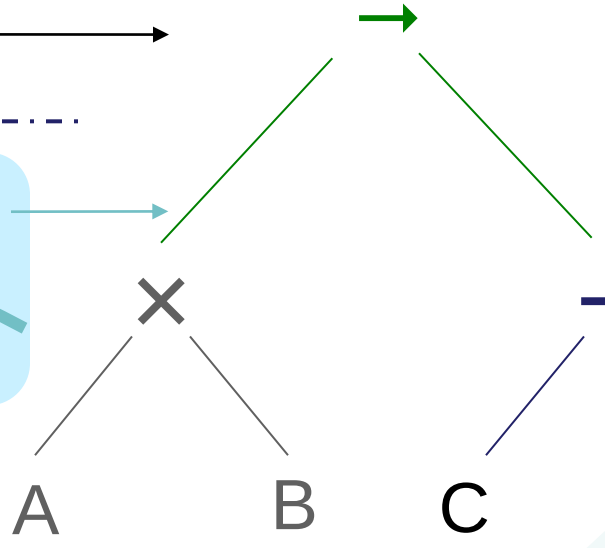


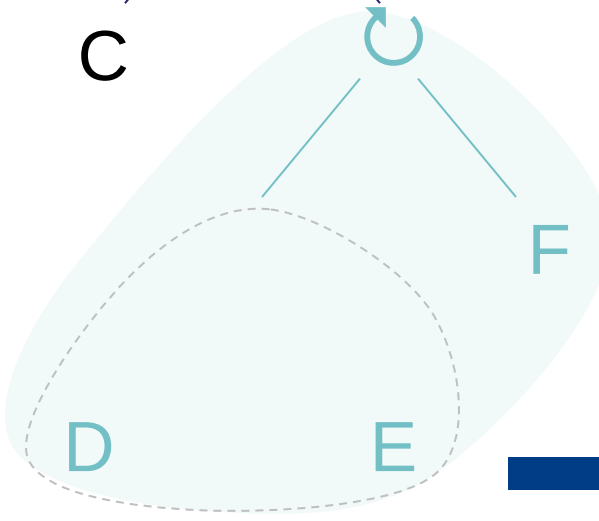
Loop Cut

Christian-Albrechts-Universität zu Kiel



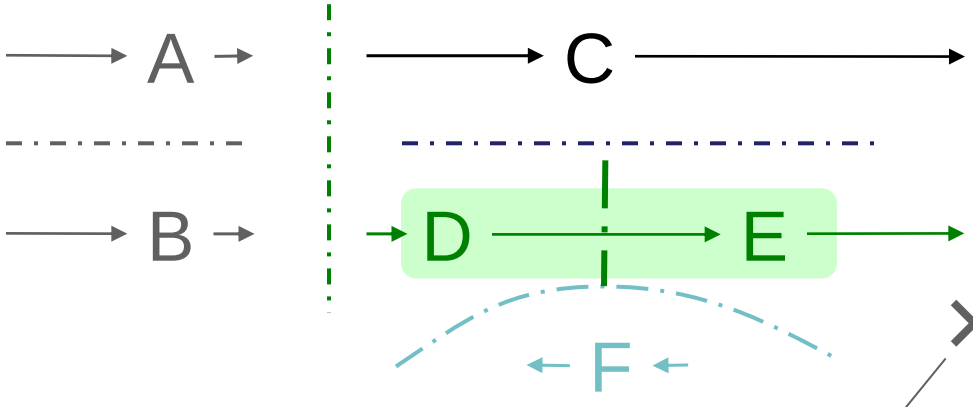


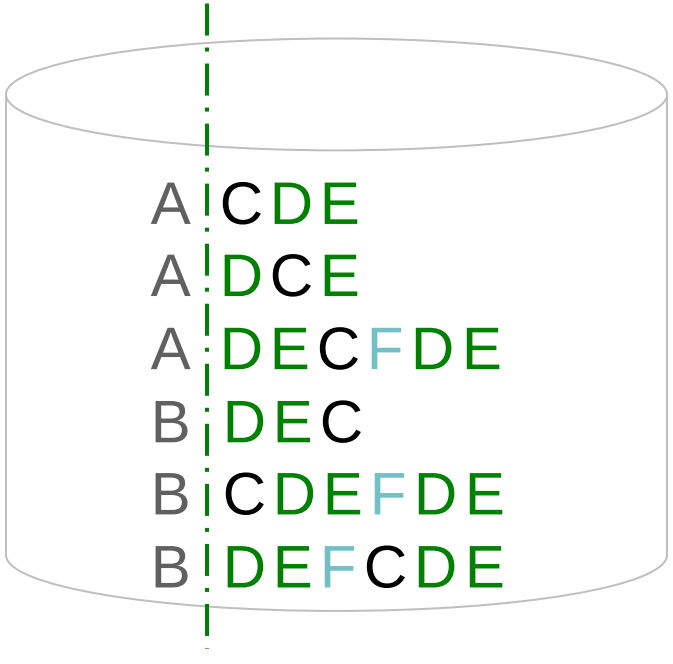


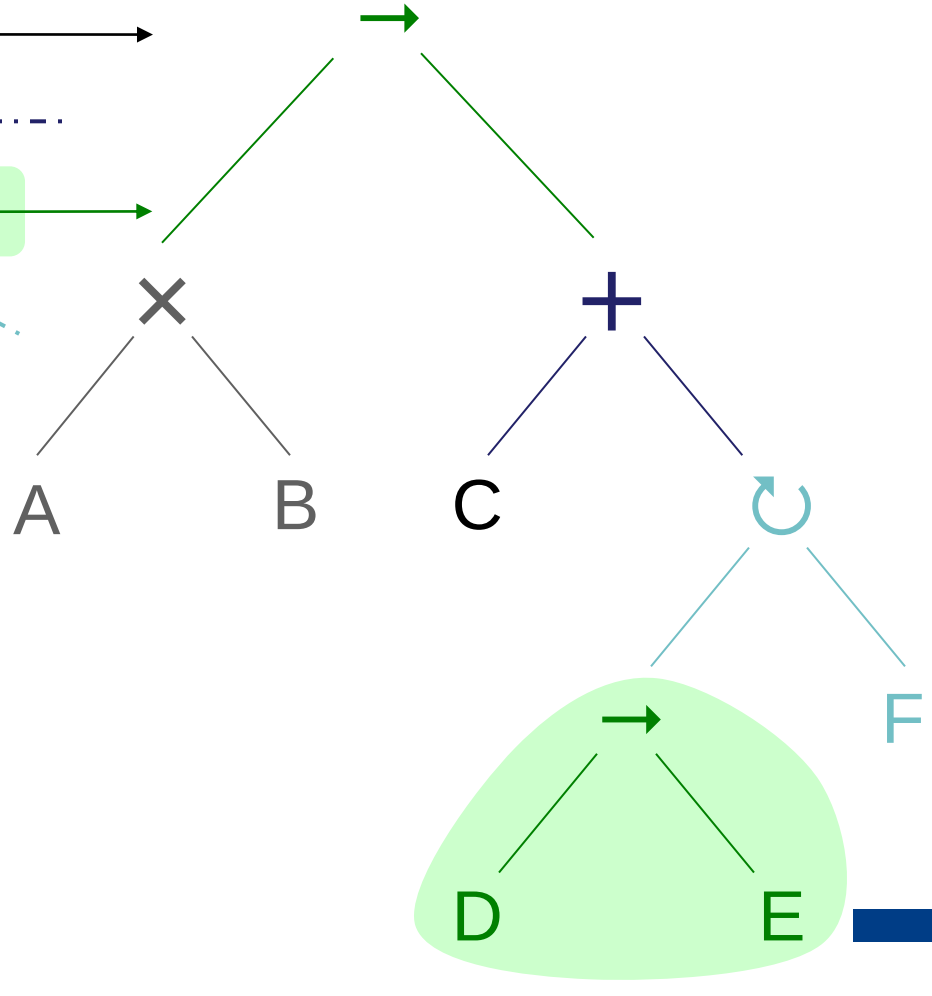


… until All Bases Reached

Christian-Albrechts-Universität zu Kiel

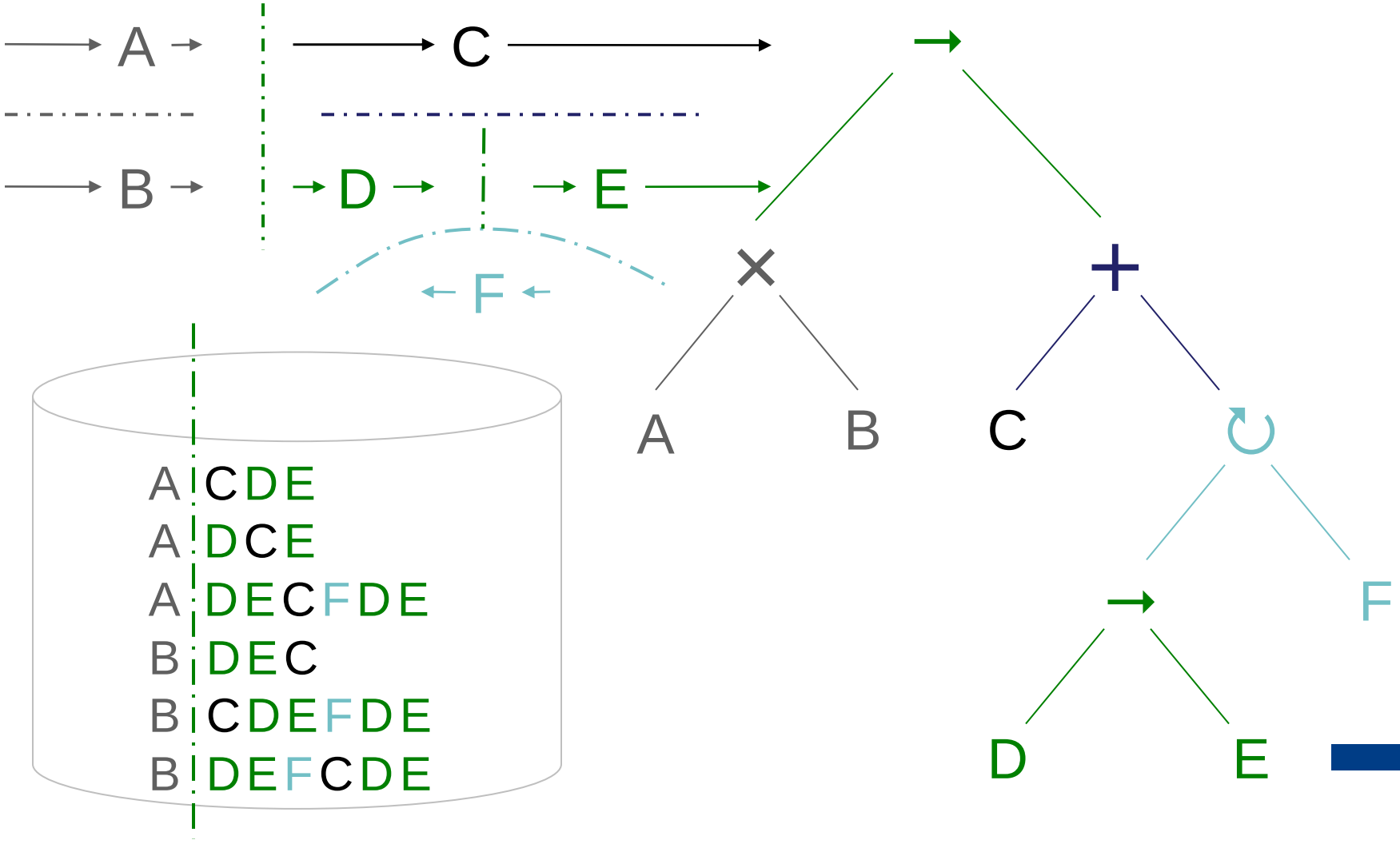






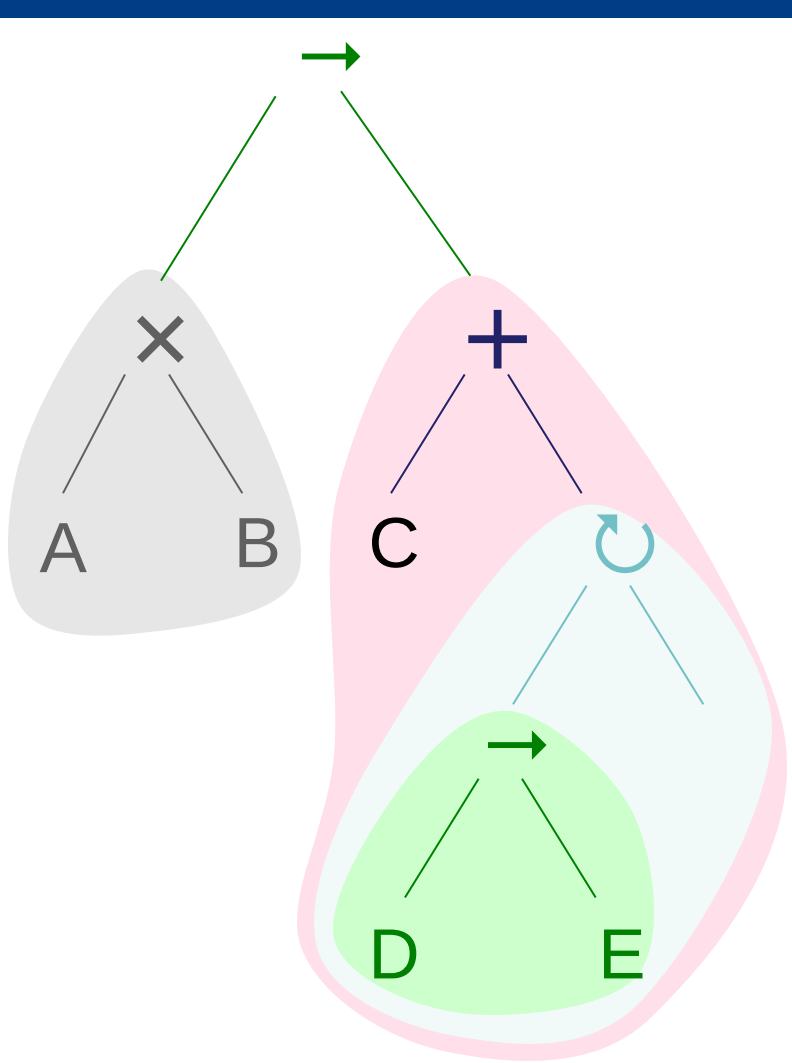
Sequence, Choice, Parallel, Loop (or “Flower”)

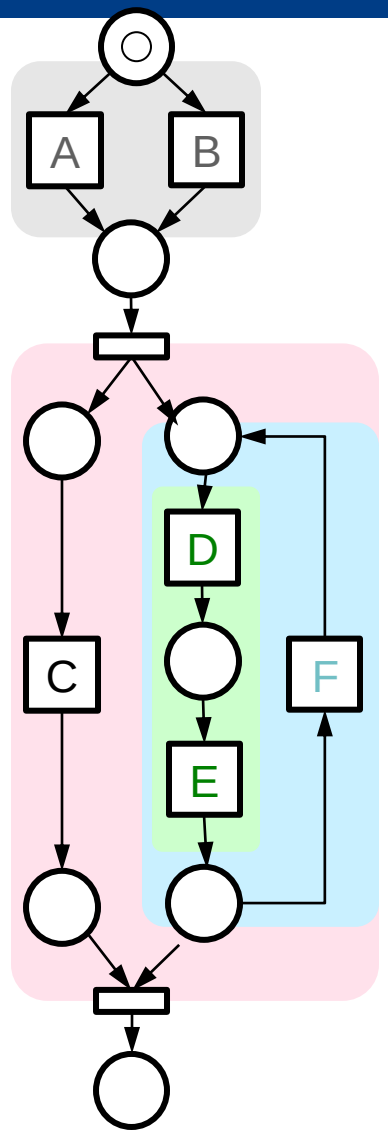
Christian-Albrechts-Universität zu Kiel

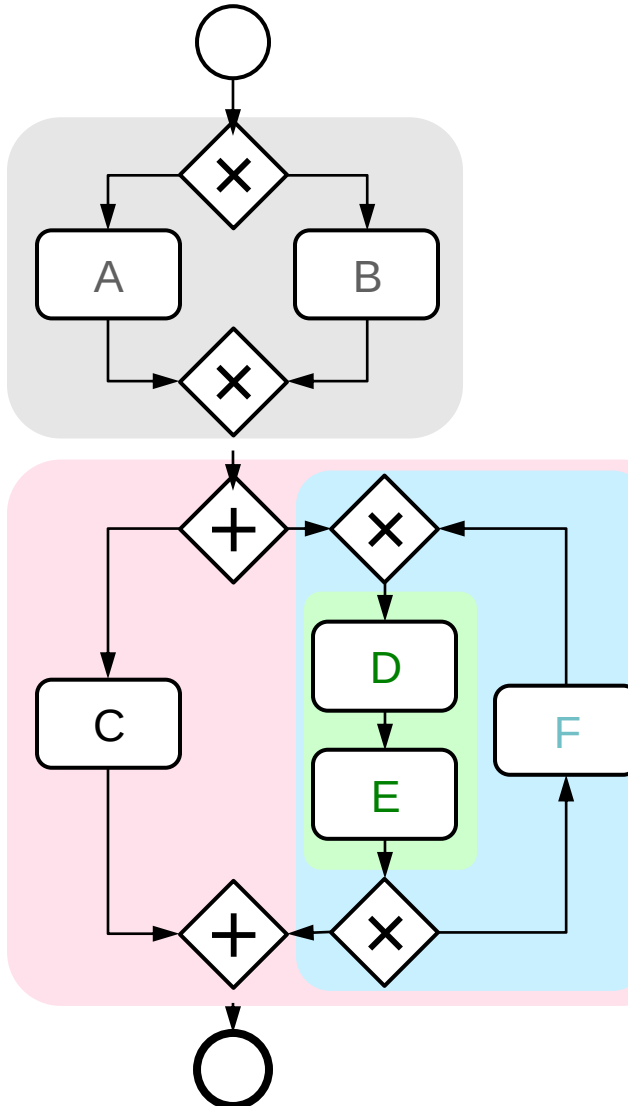


Process Tree = Block-Structured Model

Christian-Albrechts-Universität zu Kiel







Process Discovery

Organisationsperspektive

- Neben den vorgestellten Discovery-Verfahren für die Kontrollflussperspektive existieren auch verschiedene Discovery-Verfahren für die Organisationsperspektive

- Ergebnisse
  - Soziale Netzwerke
  - Organisationsstrukturen

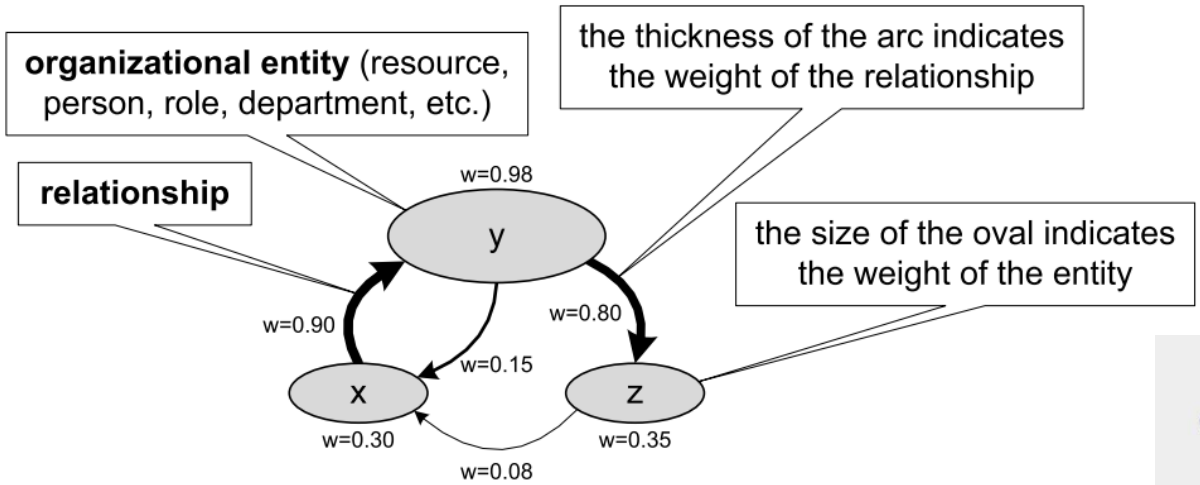
	Kontrollfluss	Organisation	Zeit	Daten
Discovery	X	X		
Conformance	X	X	X	X
Enhancement		X	X	X

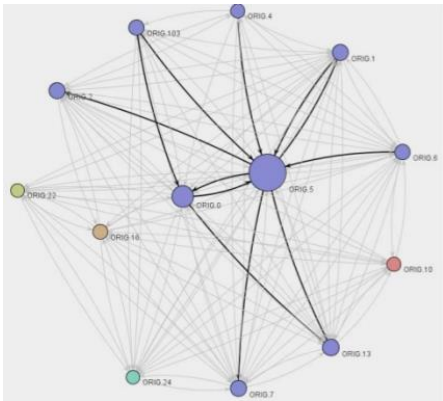
Process Discovery

Christian-Albrechts-Universität zu Kiel

Social Network Analysis

- Erstellung sozialer Netzwerke auf Basis verschiedener Metriken





- Knoten: Entitäten der Organisation
- Kanten: Beziehungsart (siehe nächste Folie)
- Beispiel rechts erstellt mit ProM 6.1

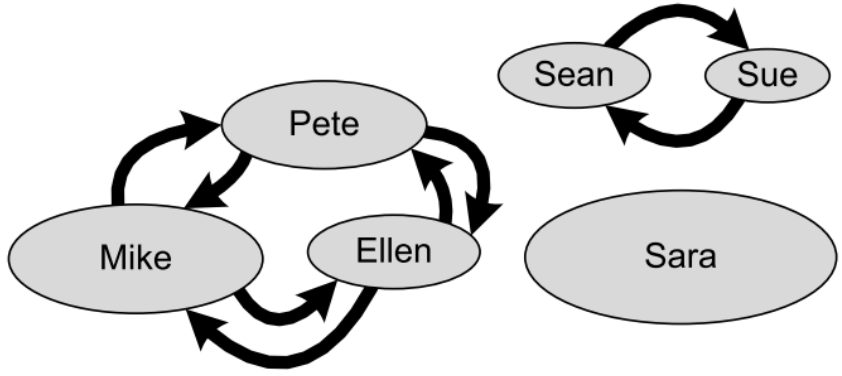
Quelle: Van der Aalst, Wil M. P: Process Mining: Data Science in Action. Springer. S. 283, 2016.

Process Discovery

Social Network Analysis - Metriken

- Mit sozialen Netzwerken können über die Kanten verschiedene Beziehungsarten zwischen Entitäten einer Organisation dargestellt werden
- Mögliche Metriken:
  - Zusammenarbeit (Welche Mitarbeiter arbeiten gemeinsam an Fällen?)
  - Übergabe von Arbeit (Welcher Mitarbeiter führt Folgeaktivitäten aus?)
  - Ähnliche Aufgaben (Welche Mitarbeiter führen ähnliche Aufgaben aus?)

Beispiel: Mitarbeiter, die ähnliche Sammlungen von Aktivitäten ausführen, sind „verwandt“. Sara ist die einzige Mitarbeiterin, die bestimmte Aktivitäten ausführt. Daher ist sie nicht mit anderen Mitarbeitern verbunden.



Quelle: Van der Aalst, Wil M. P: Process Mining: Data Science in Action. Springer. S. 286, 2016.

Process Discovery

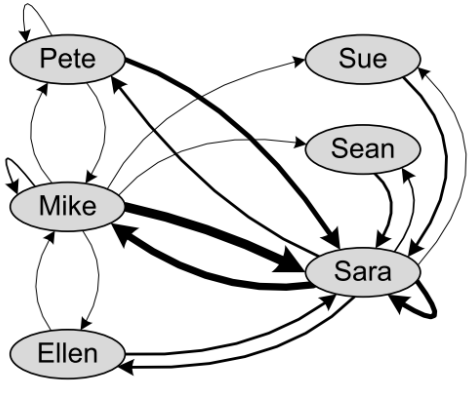
Christian-Albrechts-Universität zu Kiel

Social Network Analysis – Übergabe von Arbeit

- Metrik: Übergabe von Arbeit (handover of work) (Welcher Mitarbeiter führt Folgeaktivitäten aus?)

	Pete	Mike	Ellen	Sue	Sean	Sara
Pete	0.135	0.225	0.09	0.06	0.09	1.035
Mike	0.225	0.375	0.15	0.1	0.15	1.725
Ellen	0.09	0.15	0.06	0.04	0.06	0.69
Sue	0	0	0	0	0	0.46
Sean	0	0	0	0	0	0.69
Sara	0.885	1.475	0.59	0.26	0.39	1.3

- Die Matrix in der Tabelle zeigt die durchschnittliche Anzahl an „Arbeitsübergaben“ von Mitarbeiter zu Mitarbeiter pro Fall
- Die Abbildung zeigt das entsprechende soziale Netzwerk (Schwellwert von 0.1)
- Die Stärke der Kanten basiert auf der Häufigkeit der Übergabe von Arbeit von einem Mitarbeiter zu einem anderen



Quelle: Van der Aalst, Wil M. P: Process Mining: Data Science in Action. Springer. S. 284, 2016.

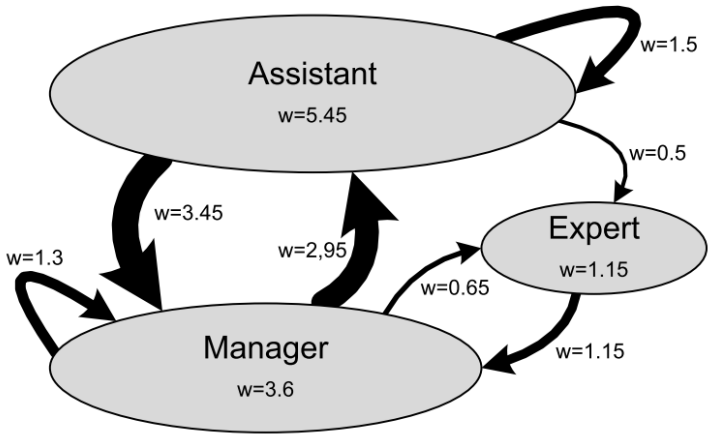
Process Discovery

Social Network Analysis – Übergabe von Arbeit

- Die Metrik „Übergabe von Arbeit“ (handover of work) kann auch auf Rollenebene ausgeführt werden, wenn Rollen bekannt sind

- Für das Beispiel auf der vorherigen Folie gelten folgende Rollenzuordnungen:
  - Assistant: Pete, Mike, Ellen
  - Expert: Sue, Sean
  - Manager: Sara
- Die Abbildung zeigt das entsprechende soziale Netzwerk (Die Knotengewichte basieren auf der Häufigkeit, mit welcher die Rolle eine Aktivität ausführt)

	Assistant	Expert	Manager
Assistant	1.5	0.5	3.45
Expert	0	0	1.15
Manager	2.95	0.65	1.3



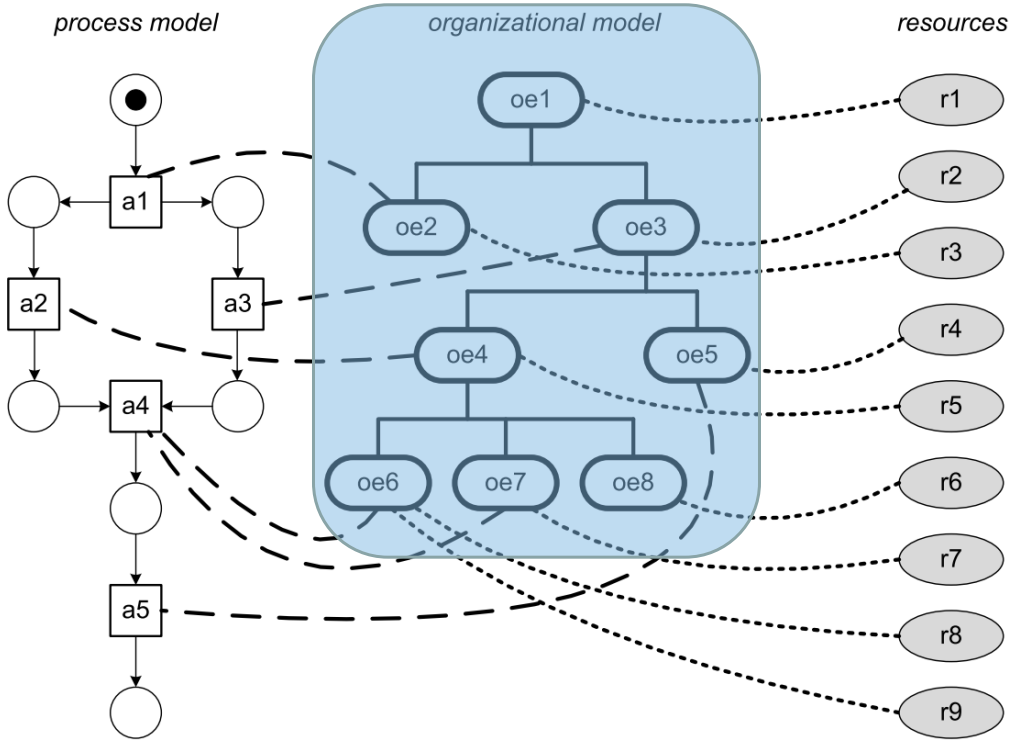
Quelle: Van der Aalst, Wil M. P: Process Mining: Data Science in Action. Springer. S. 285, 2016.

Process Discovery

Christian-Albrechts-Universität zu Kiel

Organisationsstruktur / Rollenmodell

- Durch das Clustering von Mitarbeitern mit ähnlichen Aktivitäten können auch Modelle gebildet werden um die Organisationsstruktur der Prozessbeteiligten abzubilden
- Das Verfahren basiert auf dem agglomerativen hierarchischen Clustering und bildet ein Dendrogram (Baumdiagramm)



Quelle: Van der Aalst, Wil M. P: Process Mining: Data Science in Action. Springer. S. 288, 2016.

Lernziel Kapitel 5

Christian-Albrechts-Universität zu Kiel