CS 4072 - Topics in CS
Process Mining

Lecture # 13

April 05, 2022

Spring 2022

FAST - NUCES, CFD Campus

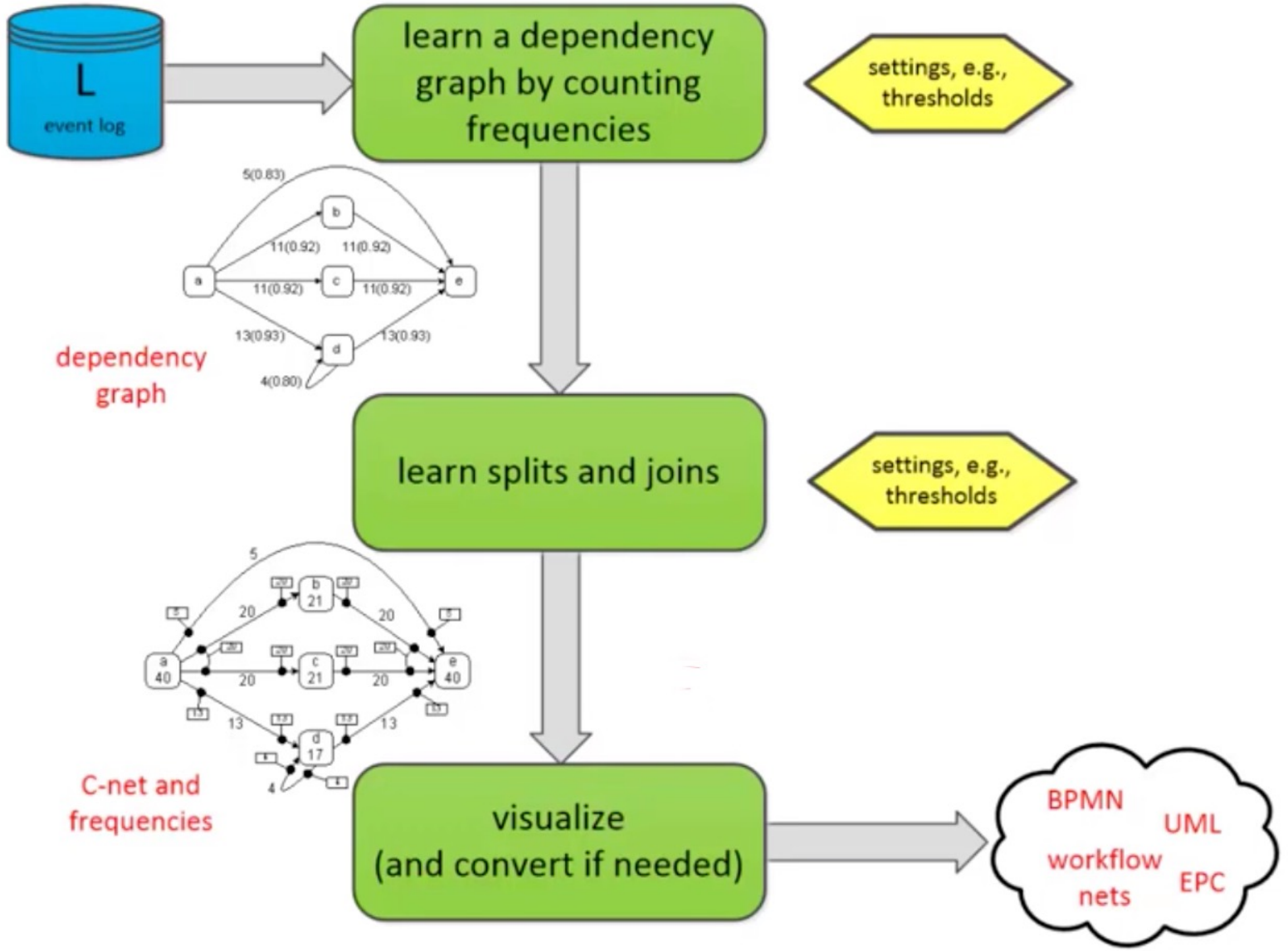
Dr. Rabia Maqsood

rabia.maqsood@nu.edu.pk

Today's Topics

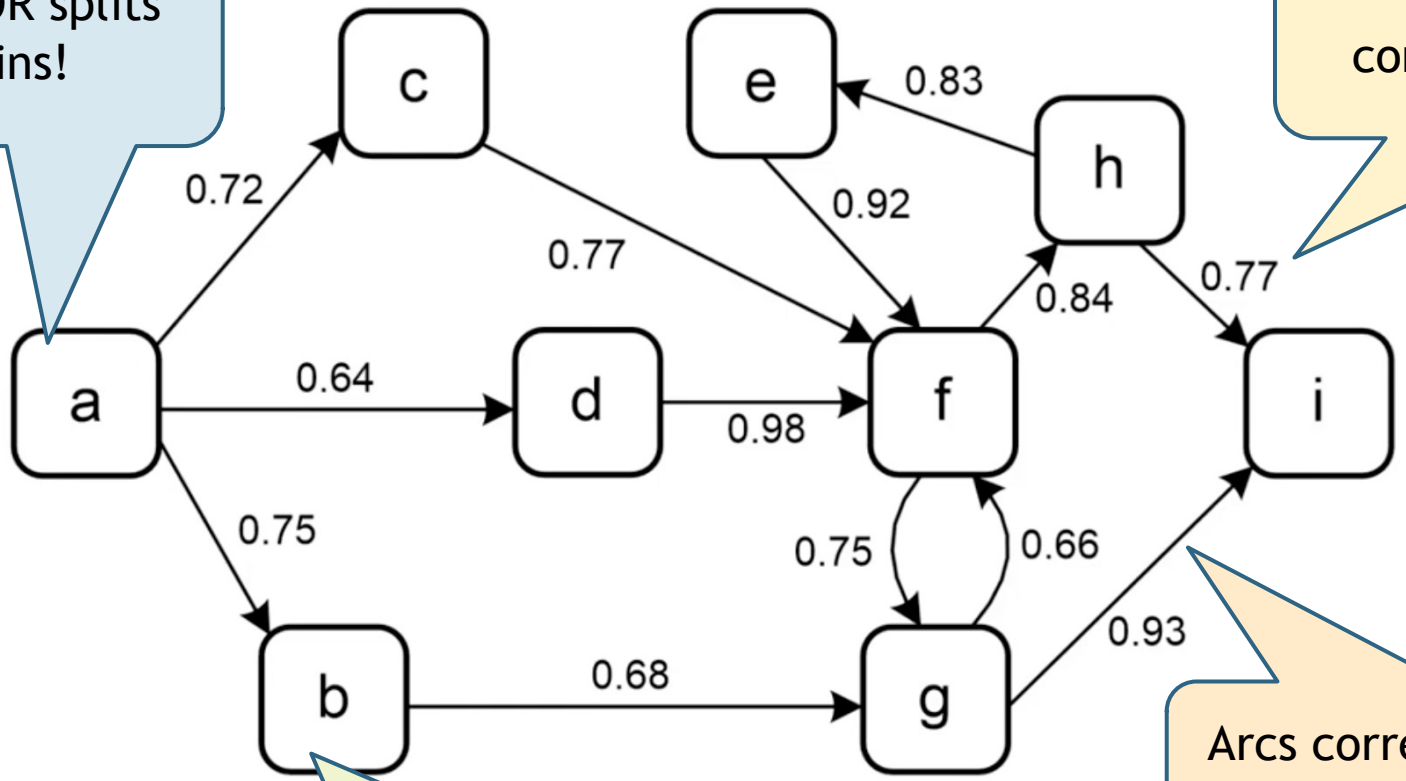
- Dependency Graph
- Causal Nets

Heuristic Mining: two main phases



Dependency Graph

No AND/XOR splits and joins!



Arcs may be annotated with frequency and/or confidence/uncertainty

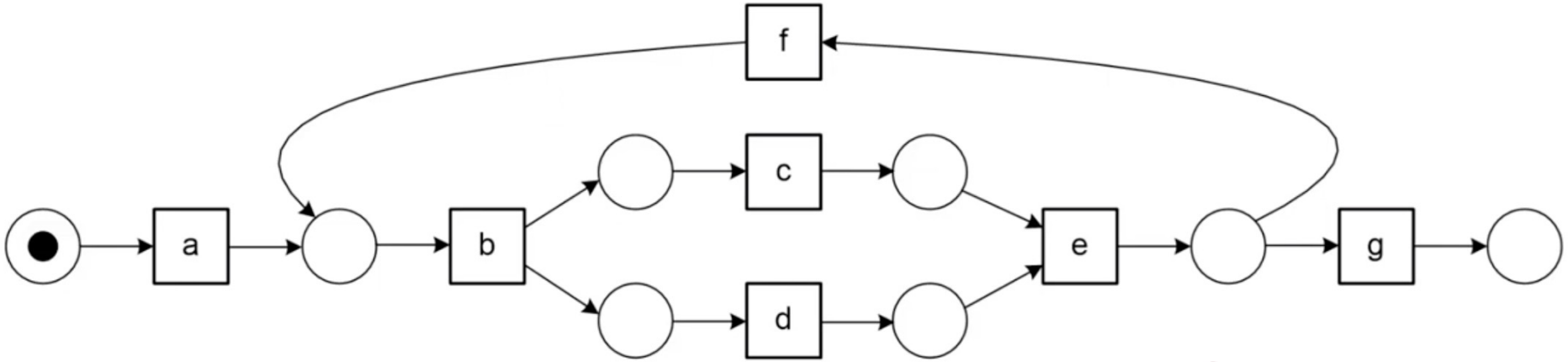
Arcs corresponds to causal dependencies

Nodes corresponds to activities

Intuition: Causality relations in footprints

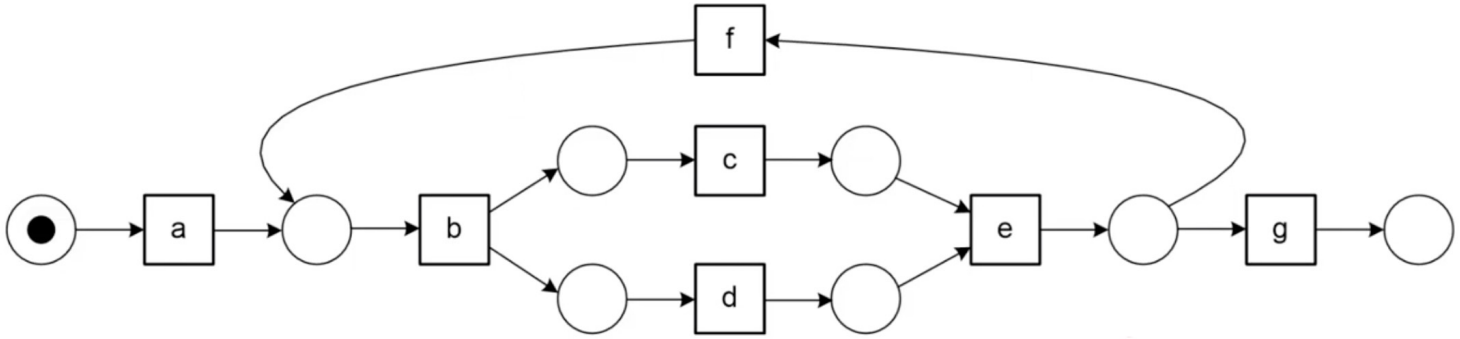
	a	b	c	d	e	f	g
a	#	$\rightarrow$	#	#	#	#	#
b	$\leftarrow$	#	$\rightarrow$	$\rightarrow$	#	$\leftarrow$	#
c	#	$\leftarrow$	#	$\parallel$	$\rightarrow$	#	#
d	#	$\leftarrow$	$\parallel$	#	$\rightarrow$	#	#
e	#	#	$\leftarrow$	$\leftarrow$	#	$\rightarrow$	$\rightarrow$
f	#	$\rightarrow$	#	#	$\leftarrow$	#	#
g	#	#	#	#	$\leftarrow$	#	#

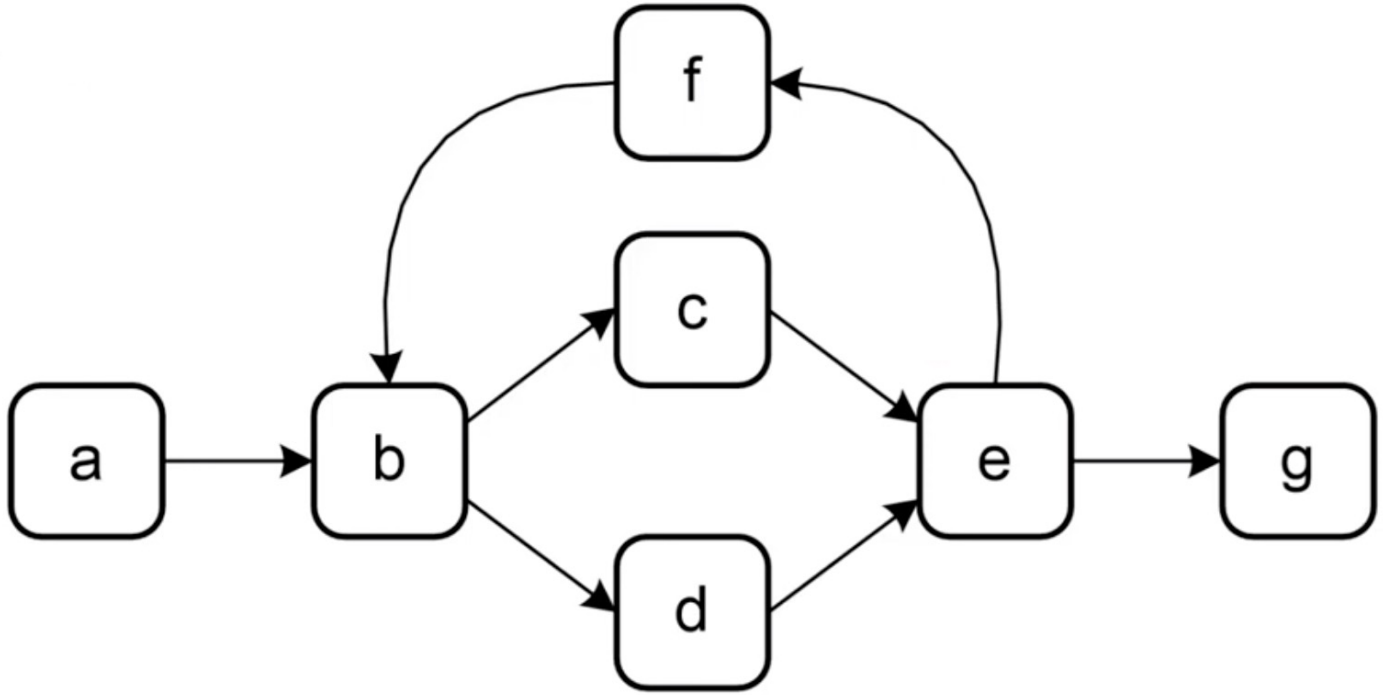
Intuition: Causality relations in Petri net



	a	b	c	d	e	f	g
a	#	$\rightarrow$	#	#	#	#	#
b	$\leftarrow$	#	$\rightarrow$	$\rightarrow$	#	$\leftarrow$	#
c	#	$\leftarrow$	#	$\parallel$	$\rightarrow$	#	#
d	#	$\leftarrow$	$\parallel$	#	$\rightarrow$	#	#
e	#	#	$\leftarrow$	$\leftarrow$	#	$\rightarrow$	$\rightarrow$
f	#	$\rightarrow$	#	#	$\leftarrow$	#	#
g	#	#	#	#	$\leftarrow$	#	#

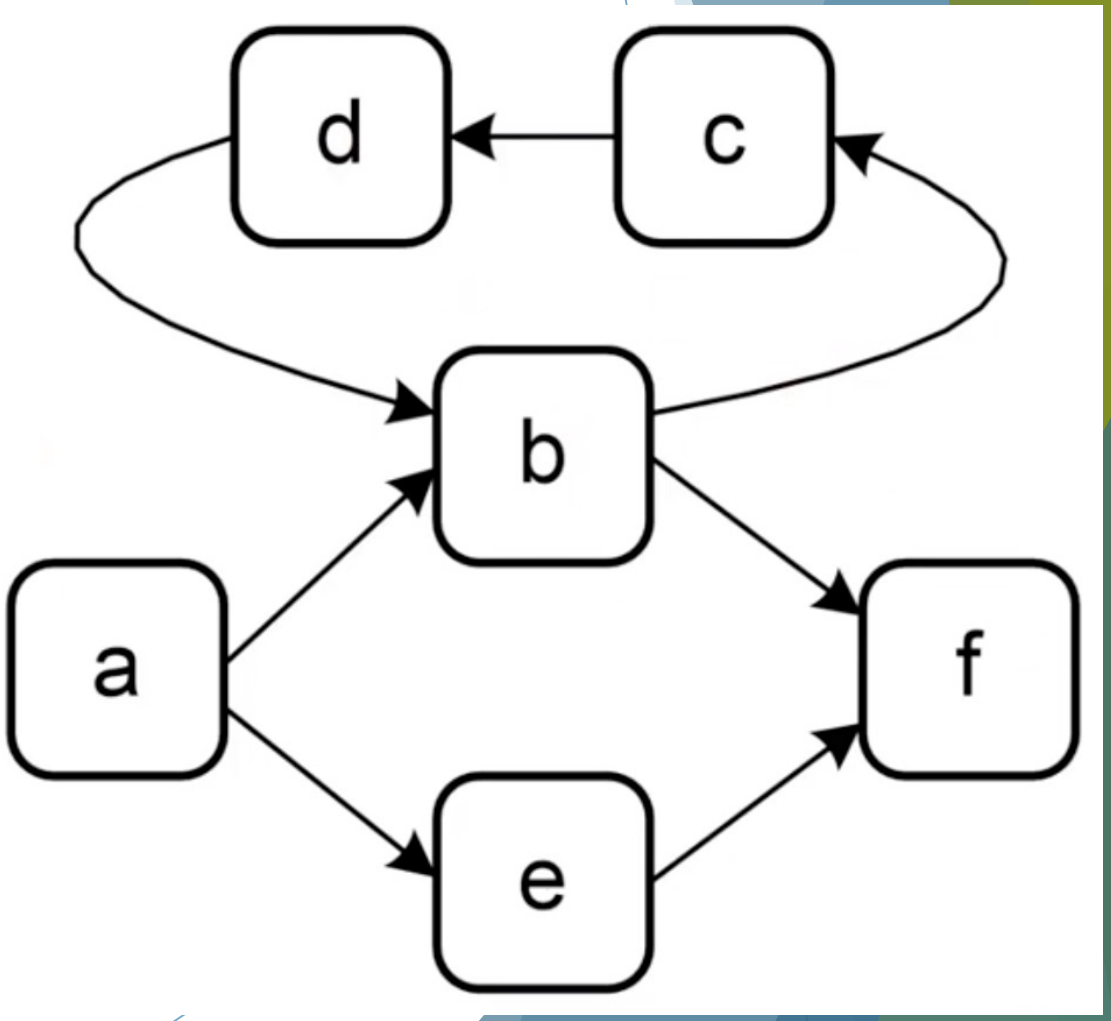
Dependency Graph based on Petri net



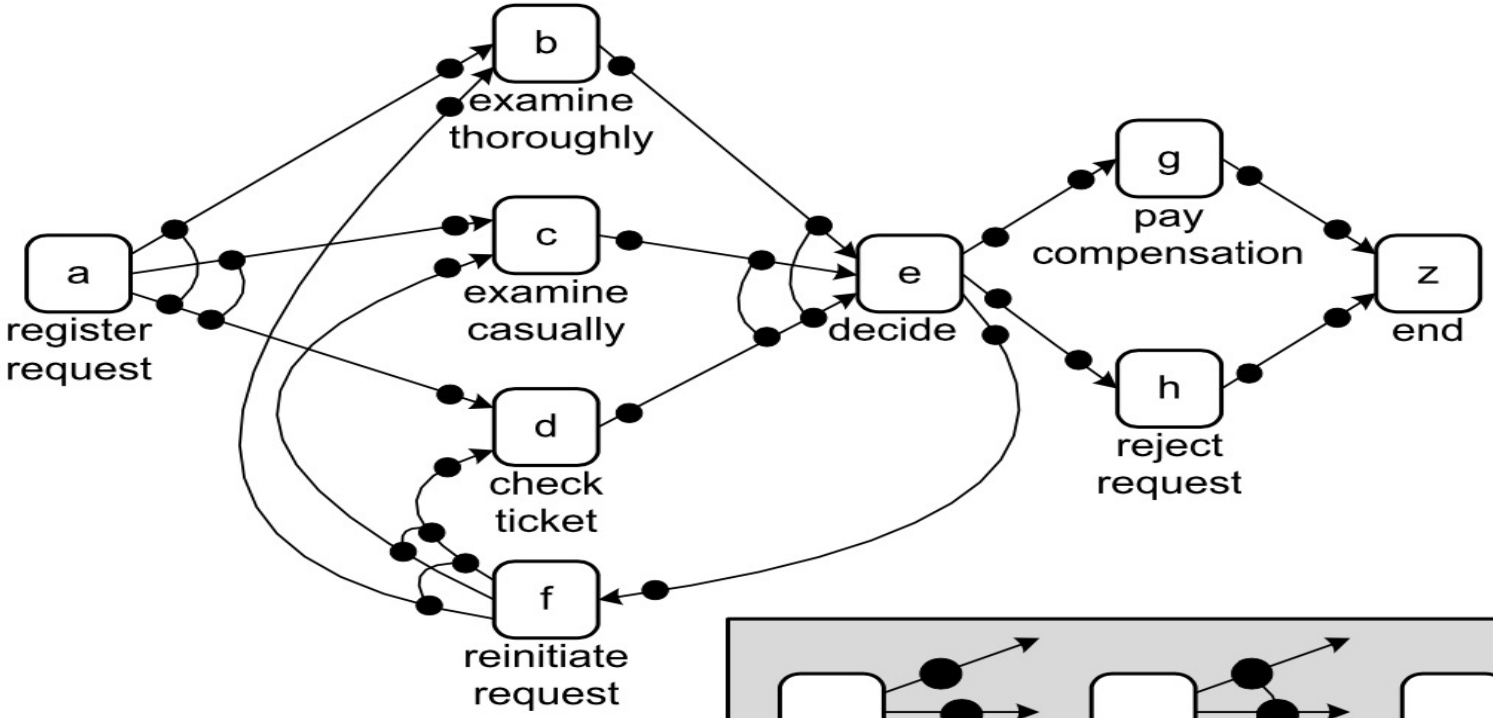


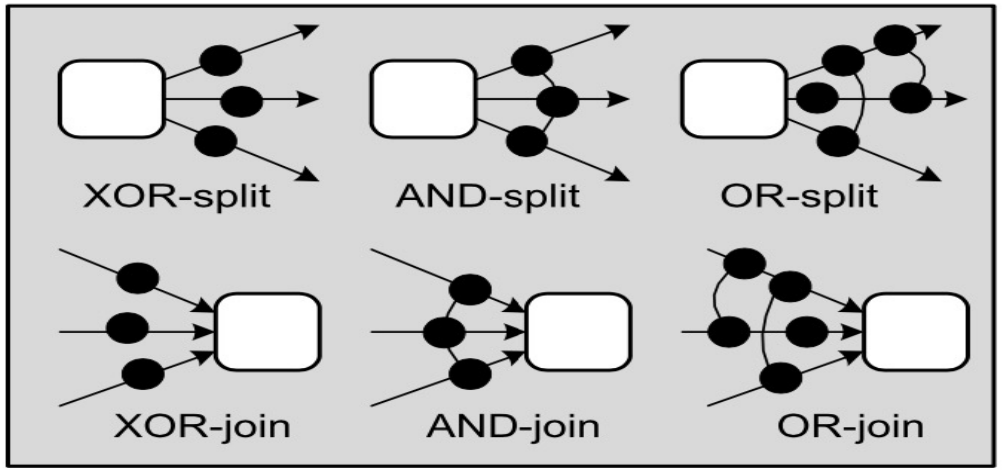
Dependency Graph based on Petri net: another example

	a	b	c	d	e	f
a	#	$\rightarrow$	#	#	$\rightarrow$	#
b	$\leftarrow$	#	$\rightarrow$	$\leftarrow$	$\|\|$	$\rightarrow$
c	#	$\leftarrow$	#	$\rightarrow$	$\|\|$	#
d	#	$\rightarrow$	$\leftarrow$	#	$\|\|$	#
e	$\leftarrow$	$\|\|$	$\|\|$	$\|\|$	#	$\rightarrow$
f	#	$\leftarrow$	#	#	$\leftarrow$	#



Causal Nets (C-nets)

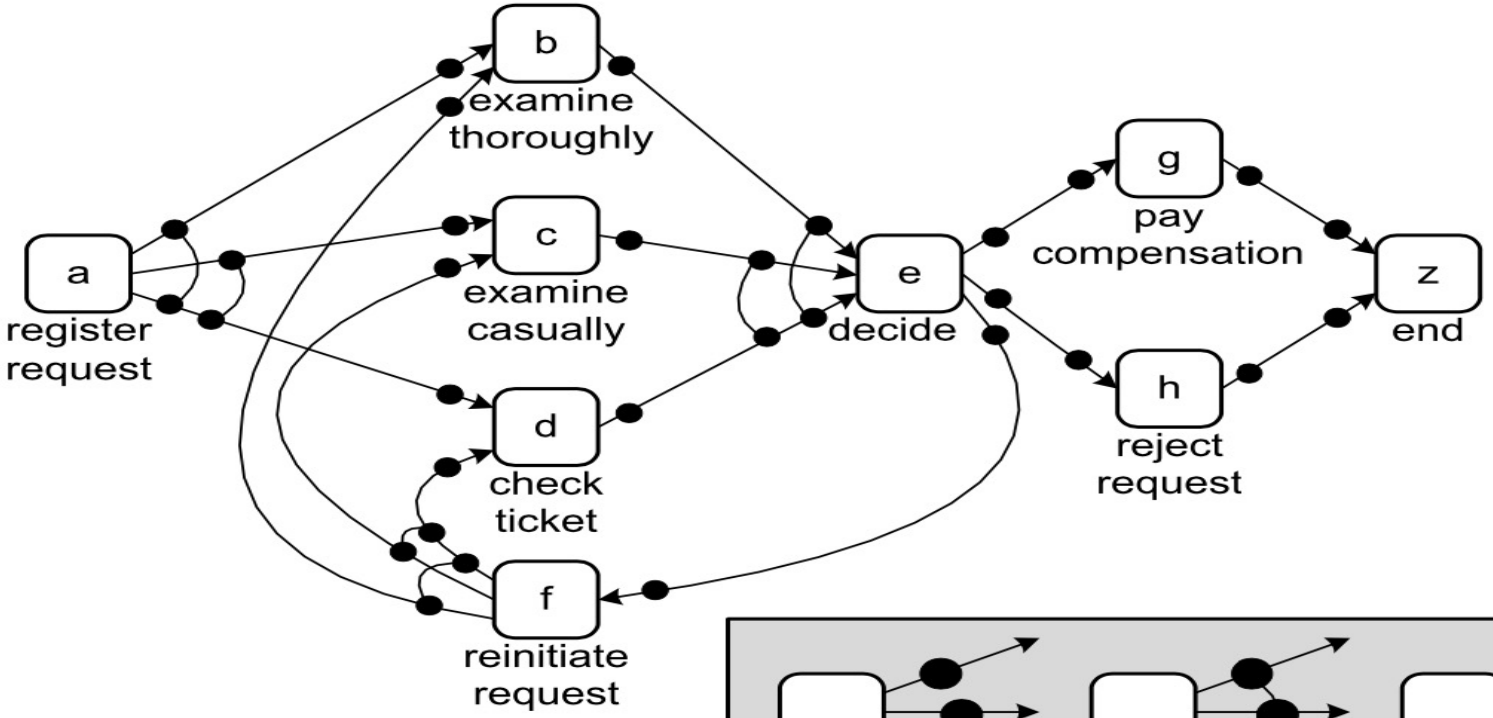


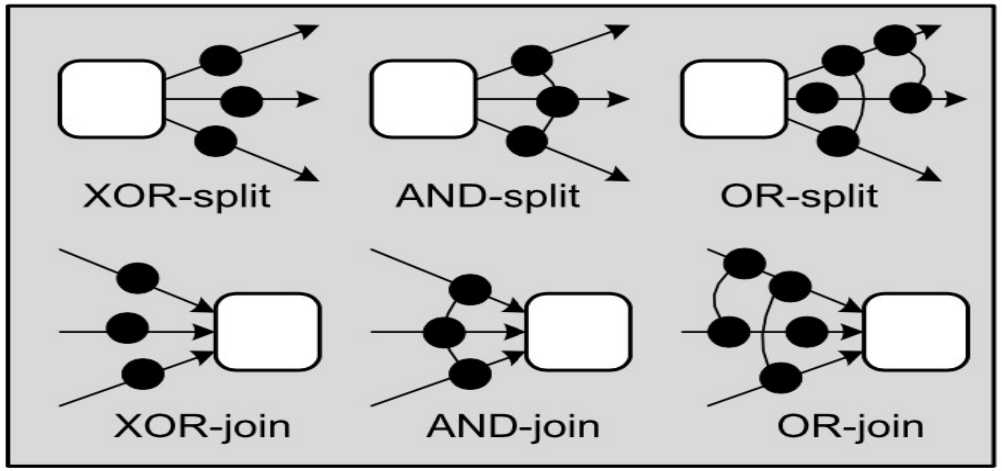


Why C-nets?

- Output of many mining techniques, e.g., heuristic miner.
- Fits well with mainstream languages (e.g., BPMN and UML activity diagrams)
- Able to model XOR, AND, and OR, no silent steps or duplicate activities needed.
- Avoiding non-sound models.

Semantics: Loose interpretation





Provides replay semantics rather than execution semantics, e.g., the moment of choice is not fixed.

C-nets: formal definition

Definition 3.8 (Causal net) A *Causal net* (C-net) is a tuple $C = (A, a_i, a_o, D, I, O)$ where:

- $A \subseteq \mathscr{A}$ is a finite set of *activities*;
- $a_i \in A$ is the *start activity*;
- $a_o \in A$ is the *end activity*;
- $D \subseteq A \times A$ is the *dependency relation*,
- $AS = \{X \subseteq \mathscr{P}(A) \mid X = \{\emptyset\} \vee \emptyset \notin X\}$;[2]

[2] $\mathscr{P}(A) = \{A' \mid A' \subseteq A\}$ is the powerset of A. Hence, elements of AS are *sets of sets* of activities.

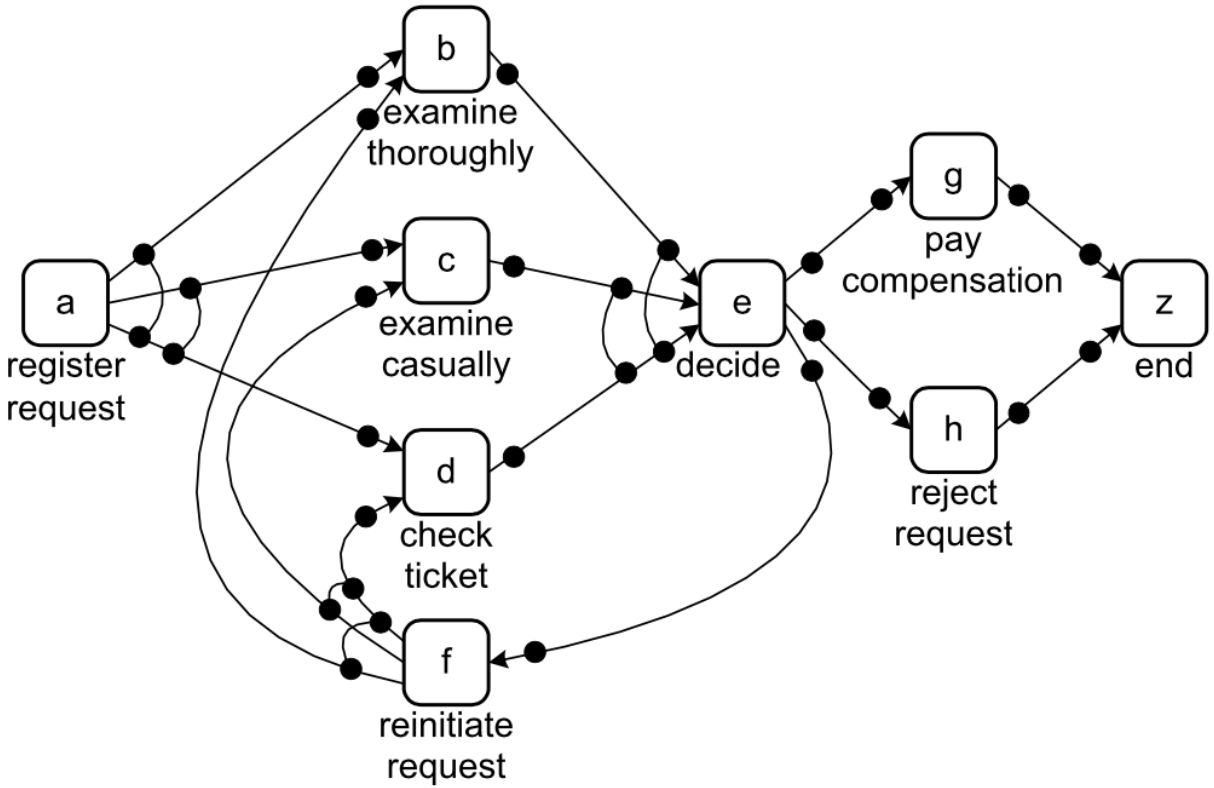
C-nets: formal definition

- $I \in A \rightarrow AS$ defines the set of possible *input bindings* per activity; and
- $O \in A \rightarrow AS$ defines the set of possible *output bindings* per activity,

such that

- $D = \{(a_1, a_2) \in A \times A \mid a_1 \in \bigcup_{as \in I(a_2)} as\}$;
- $D = \{(a_1, a_2) \in A \times A \mid a_2 \in \bigcup_{as \in O(a_1)} as\}$;
- $\{a_i\} = \{a \in A \mid I(a) = \{\emptyset\}\}$;
- $\{a_o\} = \{a \in A \mid O(a) = \{\emptyset\}\}$; and
- all activities in the graph (A, D) are on a path from a_i to a_o.

Example



A = {a, b, c, d, e, f, g, h, z}

a = a_i is the unique start activity

z = a_o is the unique end activity

D = {(a,b),(a,c),(a,d),(b,e),...,(g,z),(h,z)}

I (a) = {∅}
O(a) = {{b, d}, {c, d}}

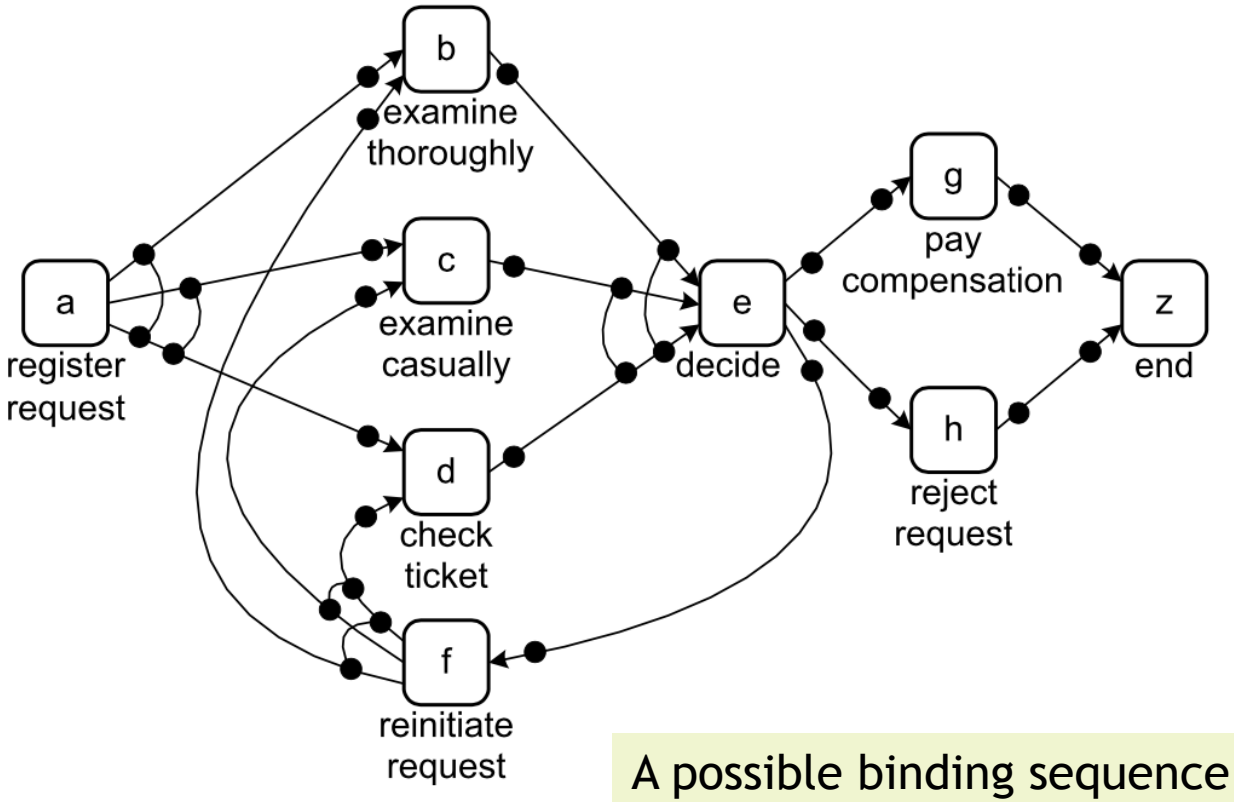
Any element of *AS* is a set of sets of activities, e.g., {{b, d }, {c, d }} ∈ *AS*

Activity Binding

- An ***activity binding*** is a tuple (a, as^I , as^O) denoting the occurrence of activity **a** with input binding as^I and output binding as^O.
- For example, **(e,{b,d},{f})** denotes the occurrence of activity **e** while being preceded by **b** and **d**, and succeeded by **f**.

Definition 3.9 (Binding) Let $C = (A, a_i, a_o, D, I, O)$ be a C-net. $B = \{(a, as^I, as^O) \in A \times \mathscr{P}(A) \times \mathscr{P}(A) \mid as^I \in I(a) \wedge as^O \in O(a)\}$ is the set of *activity bindings*. A *binding sequence* σ is a sequence of activity bindings, i.e., $\sigma \in B^*$.

Example



A = {a, b, c, d, e, f, g, h, z}

a = a_i is the unique start activity

z = a_o is the unique end activity

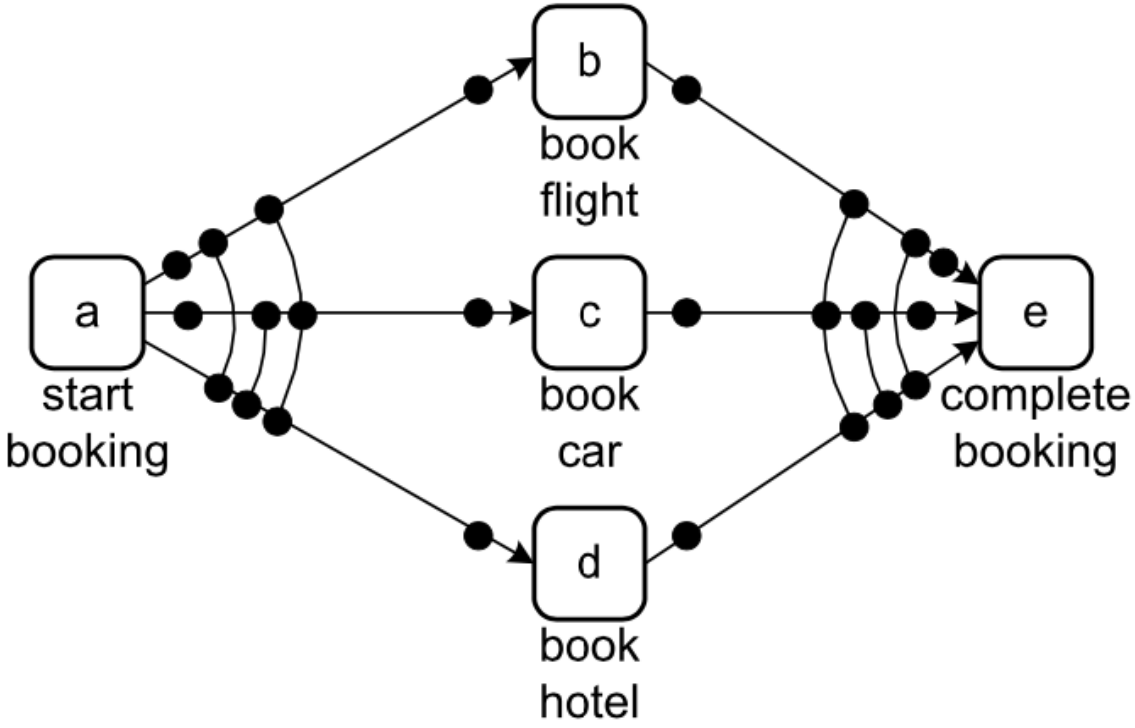
D = {(a,b),(a,c),(a,d),(b,e),...,(g,z),(h,z)}

I (a) = {∅}
O(a) = {{b, d}, {c, d}}

Any element of *AS* is a set of sets of activities, e.g., {{b, d }, {c, d }} ∈ *AS*

A possible binding sequence for this C-net is: ⟨(a,∅,{b,d}), (b, {a}, {e}), (d, {a}, {e}), (e, {b, d}, {g}), (g, {e}, {z}), (z, {g}, ∅)⟩.

Another Example



A = {a, b, c, d, e}

a(start booking) = a_i is the unique start activity

e(complete booking) = a_o is the unique end activity

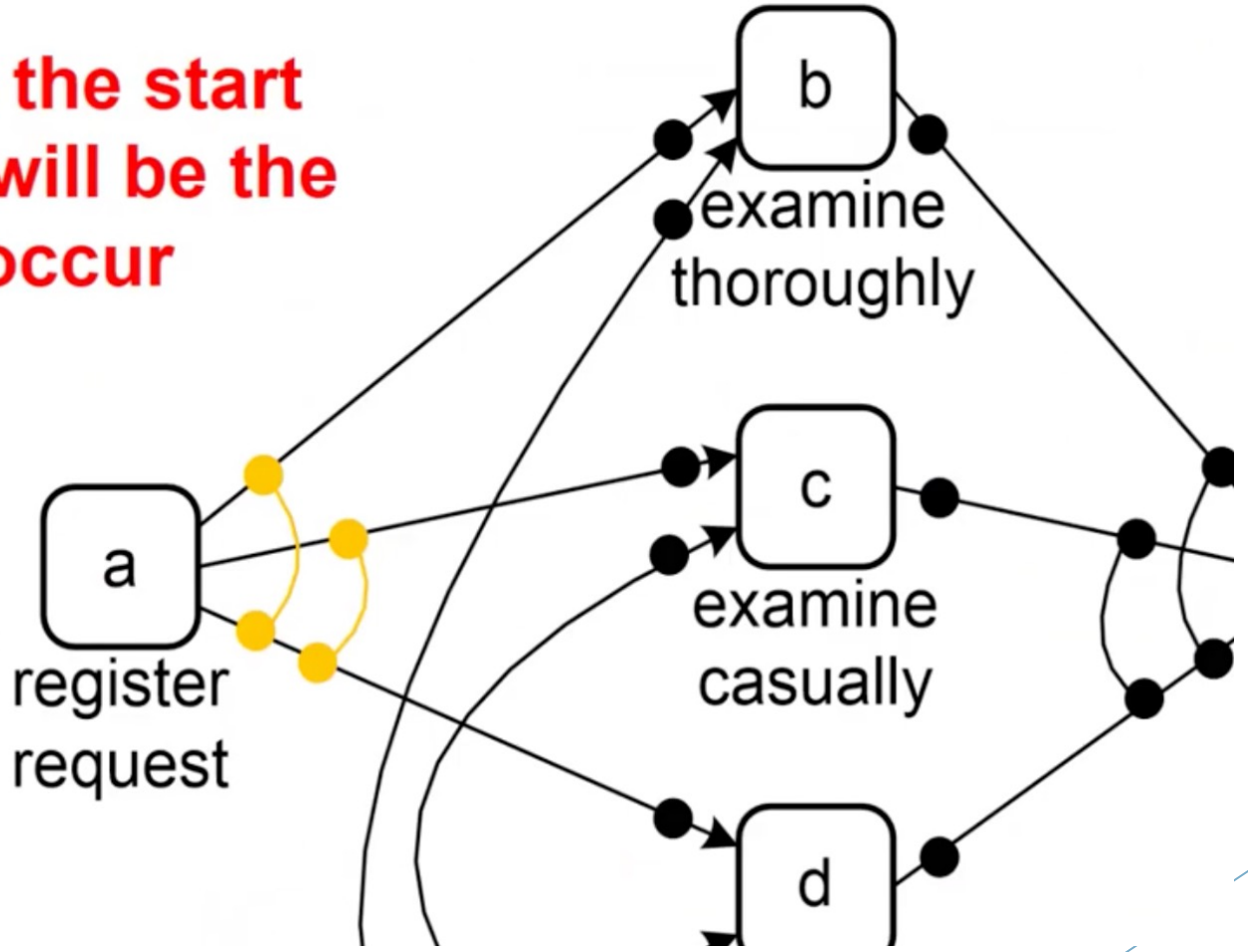
D = {(a,b),(a,c),(a,d),(b,e),...}

O(a) = I(e) = {{b},{c},{b,d},{c,d},{b,c,d}},
I(a) = O(e) = {∅},
I(b) = I(c) = I(d) = {{a}}, and
O(b) = O(c) = O(d) = {{e}}

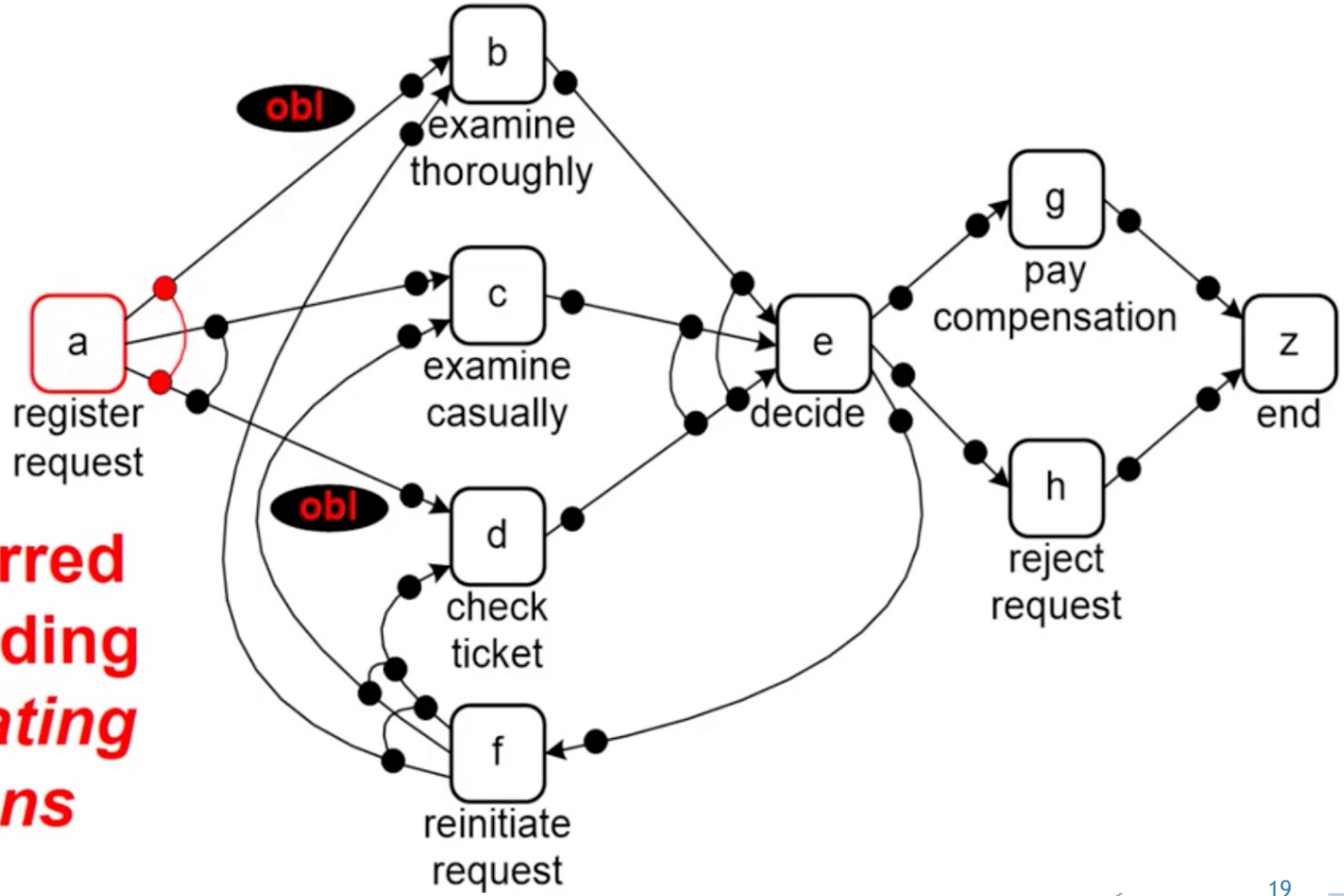
A possible binding sequence for this C-net is:
⟨(a, ∅, {b, d}), (d, {a}, {e}), (b, {a}, {e}), (e, {b, d}, ∅)⟩
the scenario in which a flight and a hotel are booked

Initial state: Only activity **a** can occur

activity *a* is the start activity and will be the first to occur

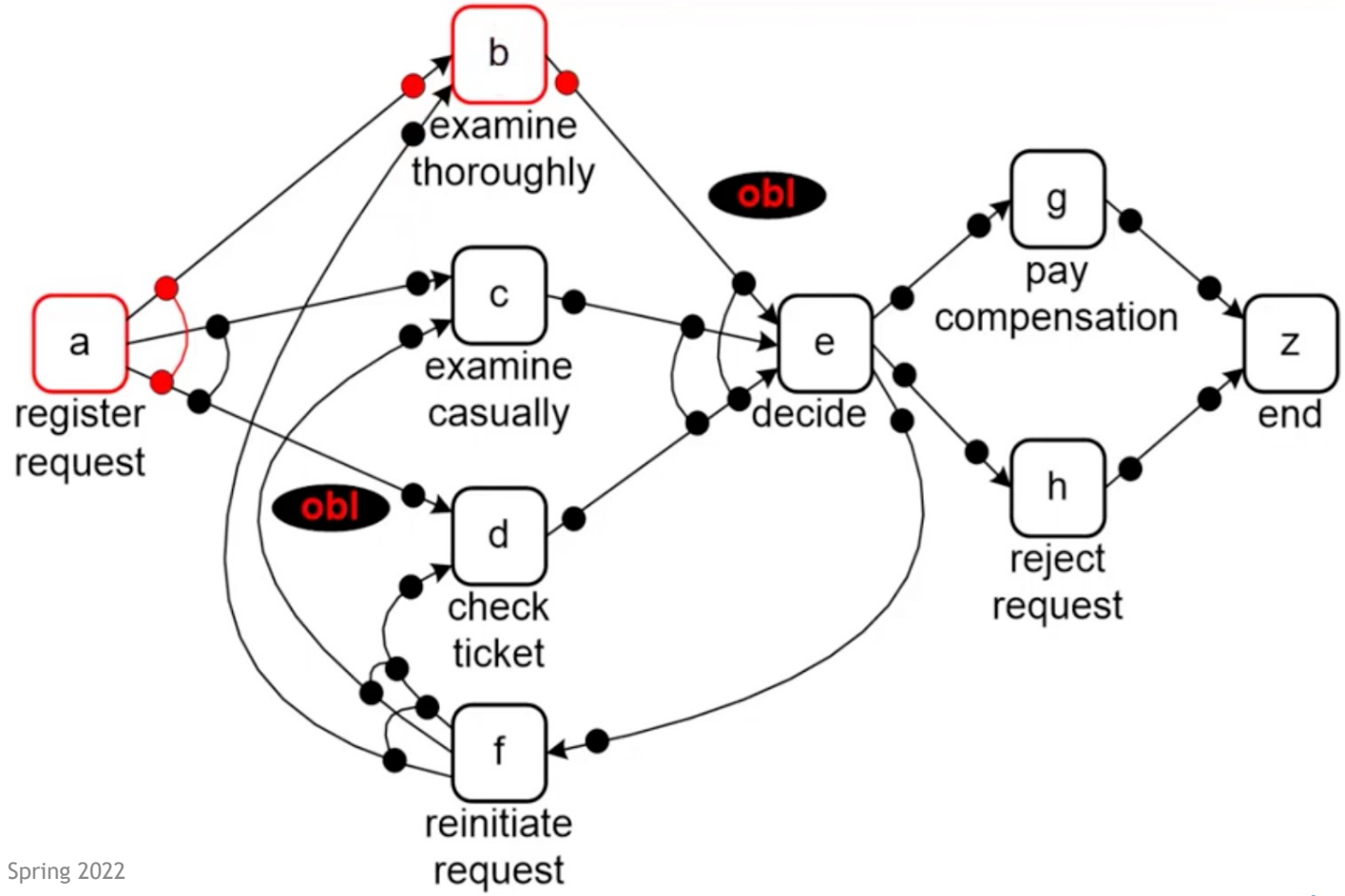


Assume activity **a** occurs with an output binding enabling **b** and **d**

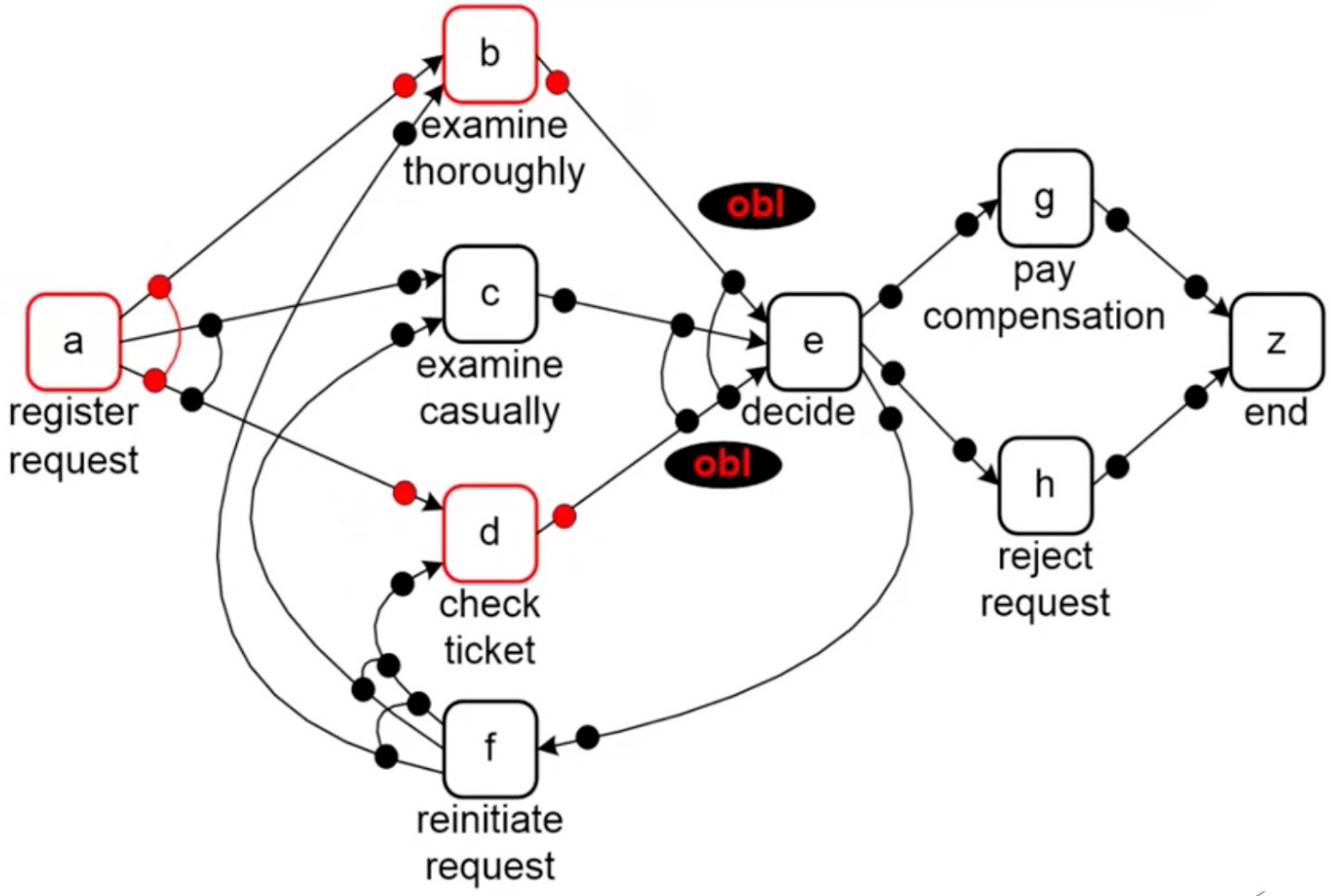


activity a* occurred with output binding *{b,d}*, thus creating *two obligations

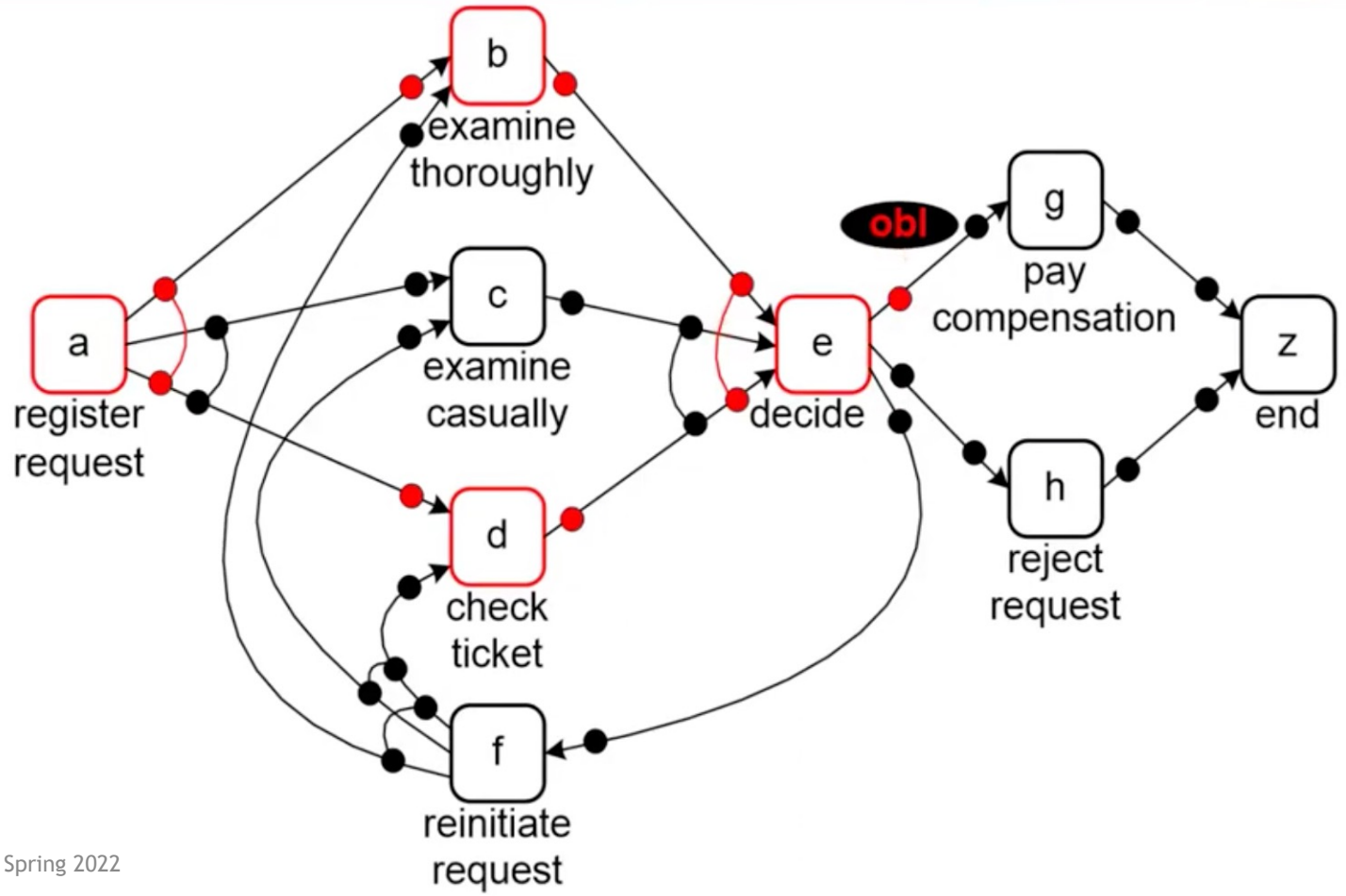
Activity **b** occurs, removes obligation (a,b) and creates obligation (b,e)



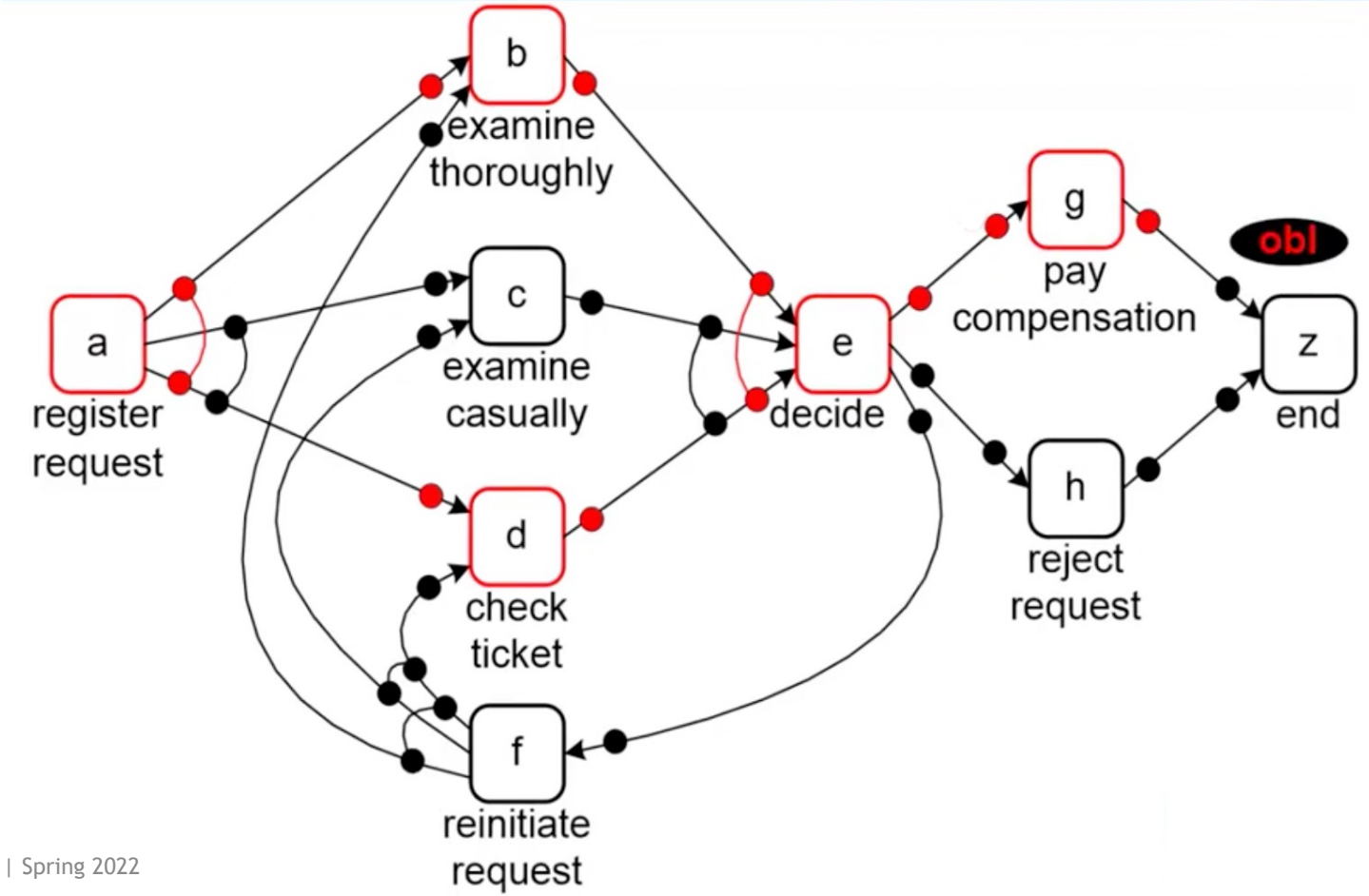
Activity **d** occurs, removes obligation (a,d) and creates obligation (d,e)



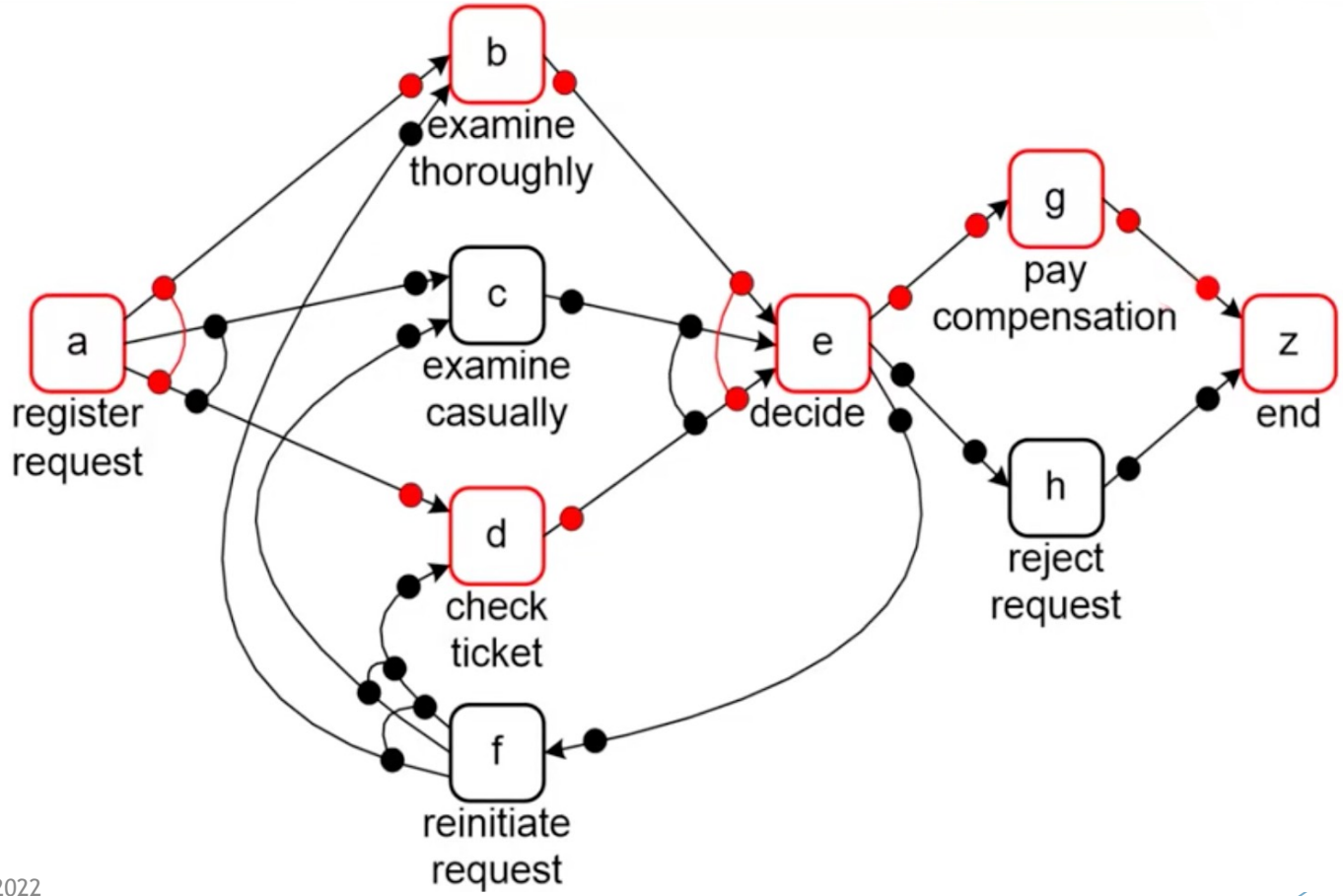
Activity **e** occurs, removes obligation (b,e) and (d,e) and creates obligation (e,g)



Activity g occurs, removes obligation (e,g) and creates obligation (g,z)



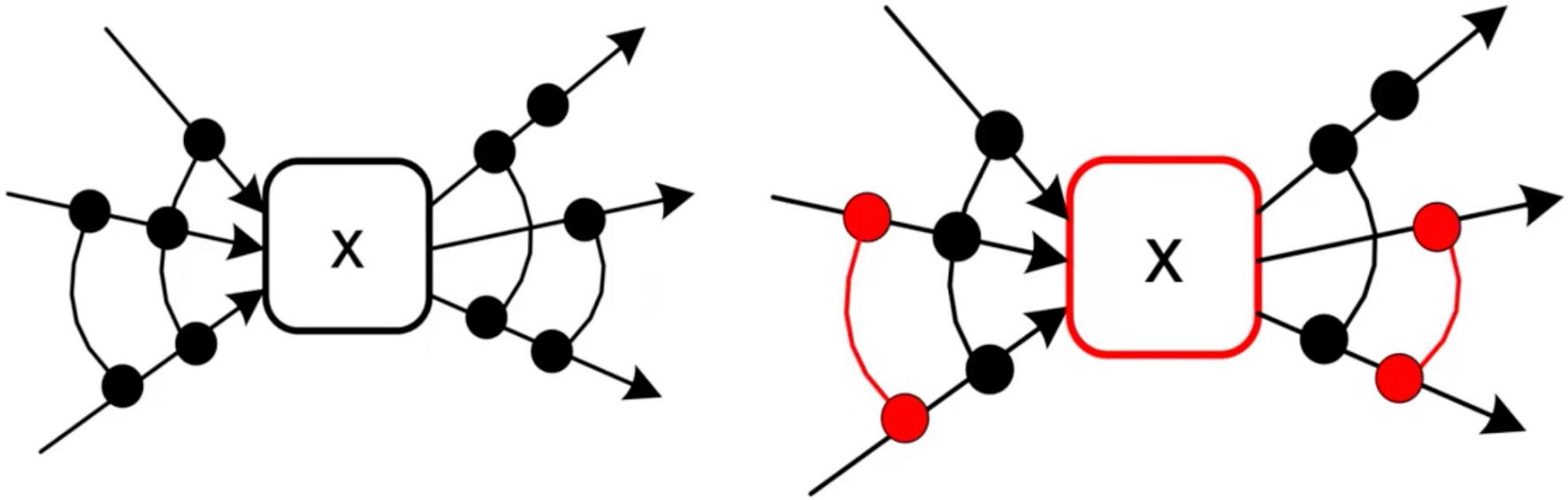
Activity z occurs, removes obligation (g,z) while leaving no other obligations



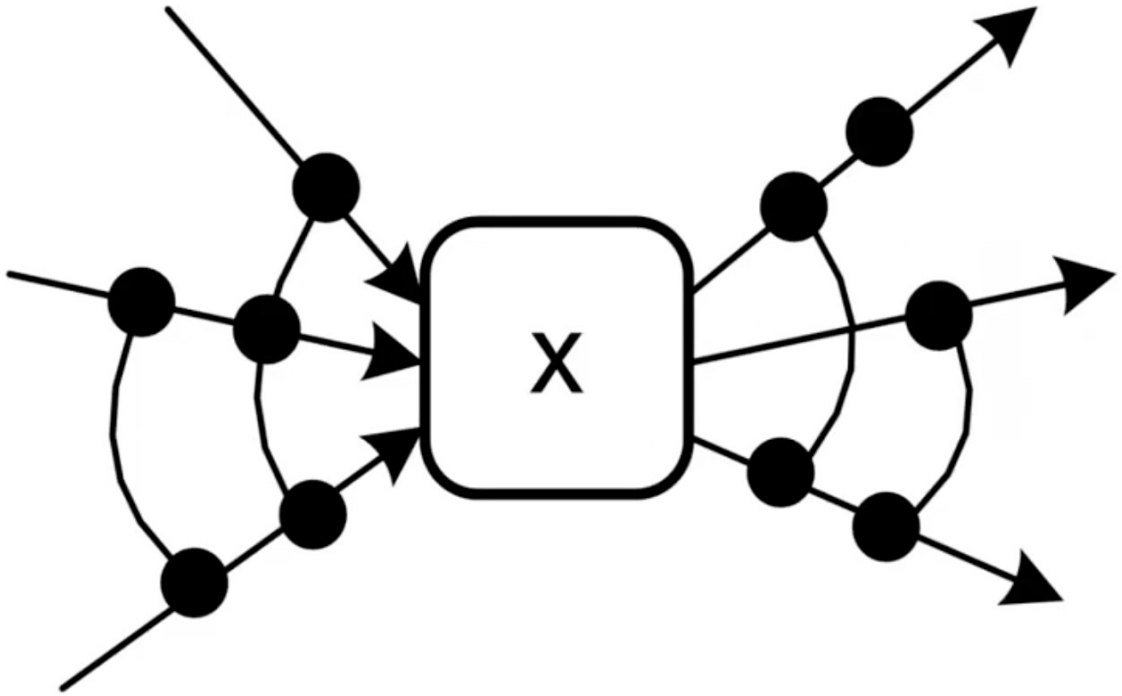
Rules of execution

- Start with the **start activity** of the C-net.
- End with the **end activity** of the C-net.
- The start and end activities **cannot** also happen in-between start and end.
- Obligations are like tokens (need to be there in order to be consumed).
- At the end, there should be no remaining obligations.

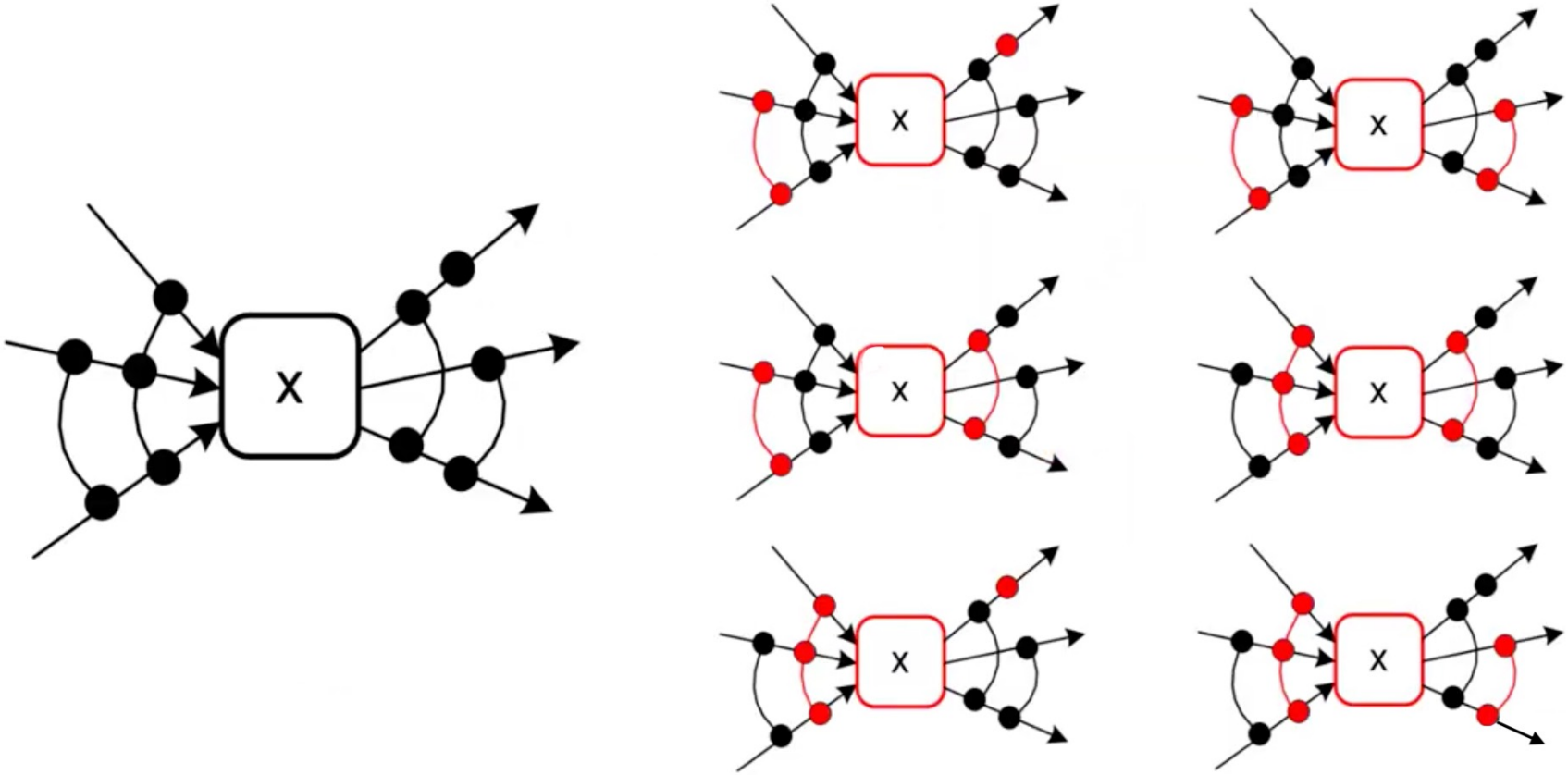
Example Binding



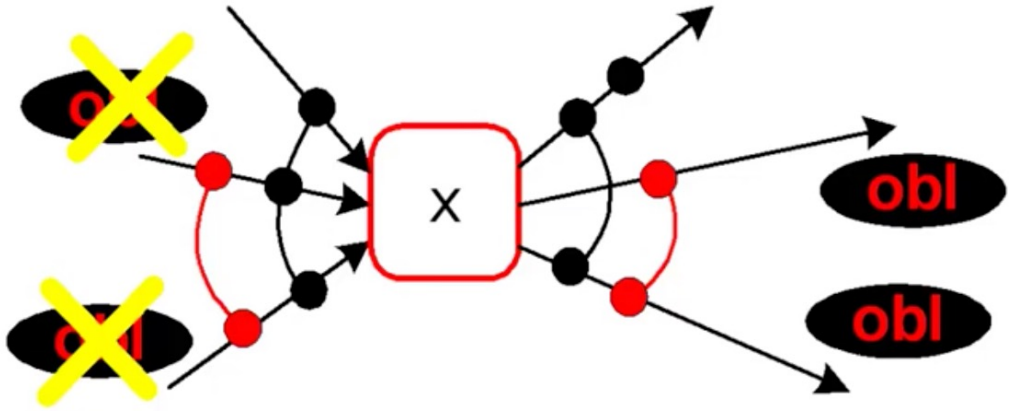
How many bindings are possible?



Six bindings

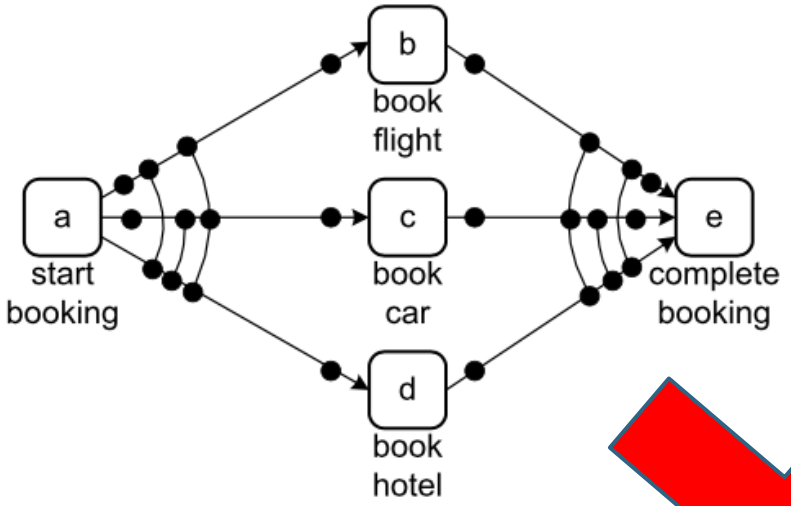


Valid Binding Sequences



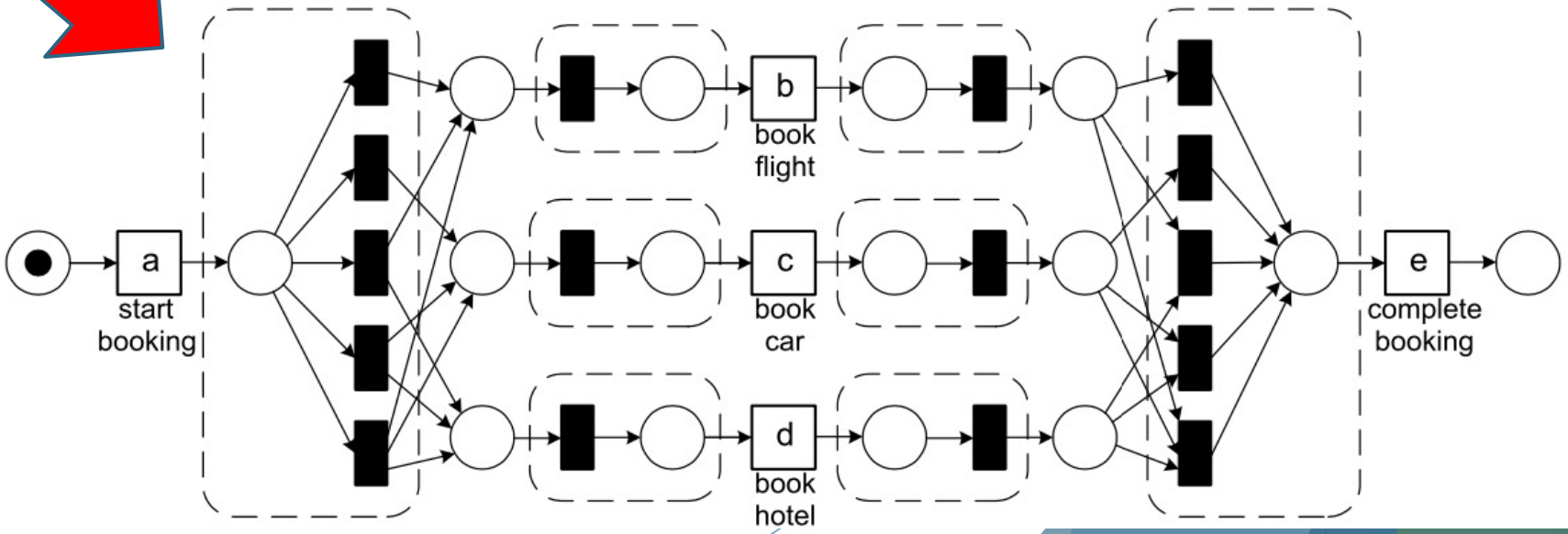
- Start with start activity without any obligations and end with end activity without any remaining obligations.
- Input bindings remove existing obligations and output bindings create obligations.

Relating C-nets to WF-nets

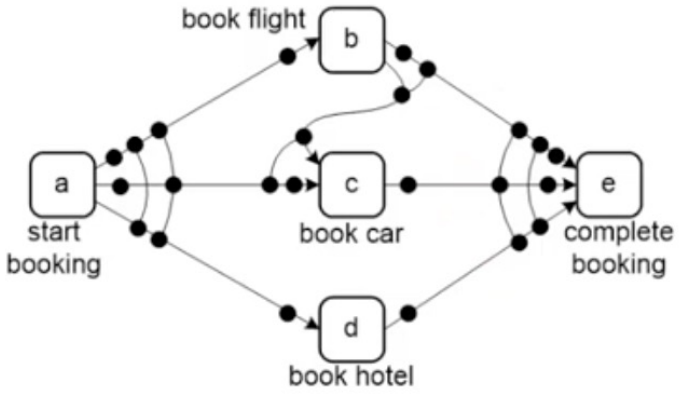


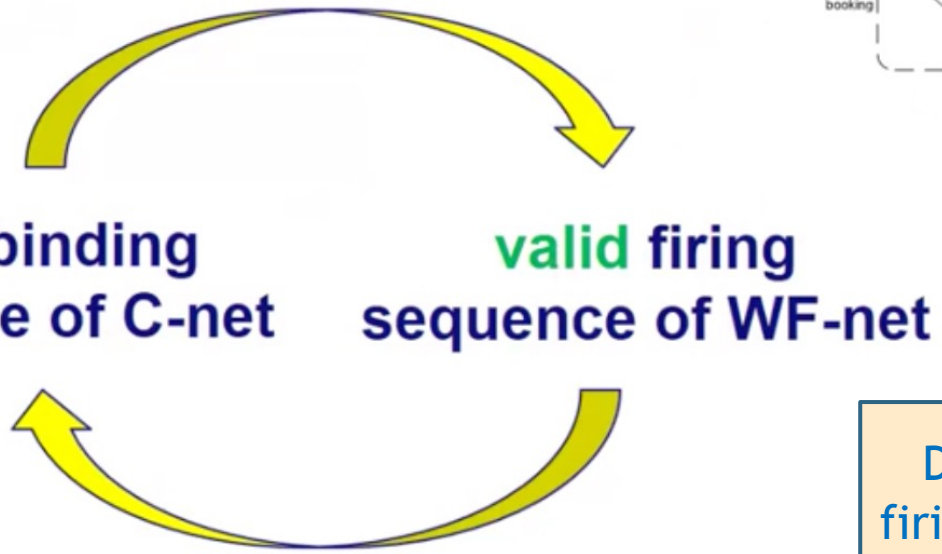
WF-net may have deadlocks, leave tokens behind, etc.

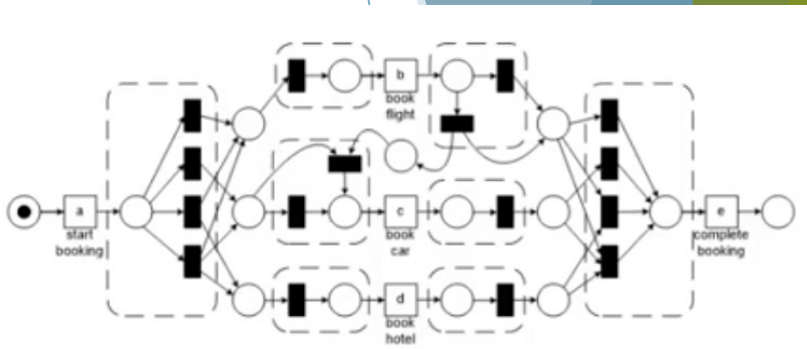
Only valid binding sequences are considered in C-nets.



Relating C-nets to WF-nets



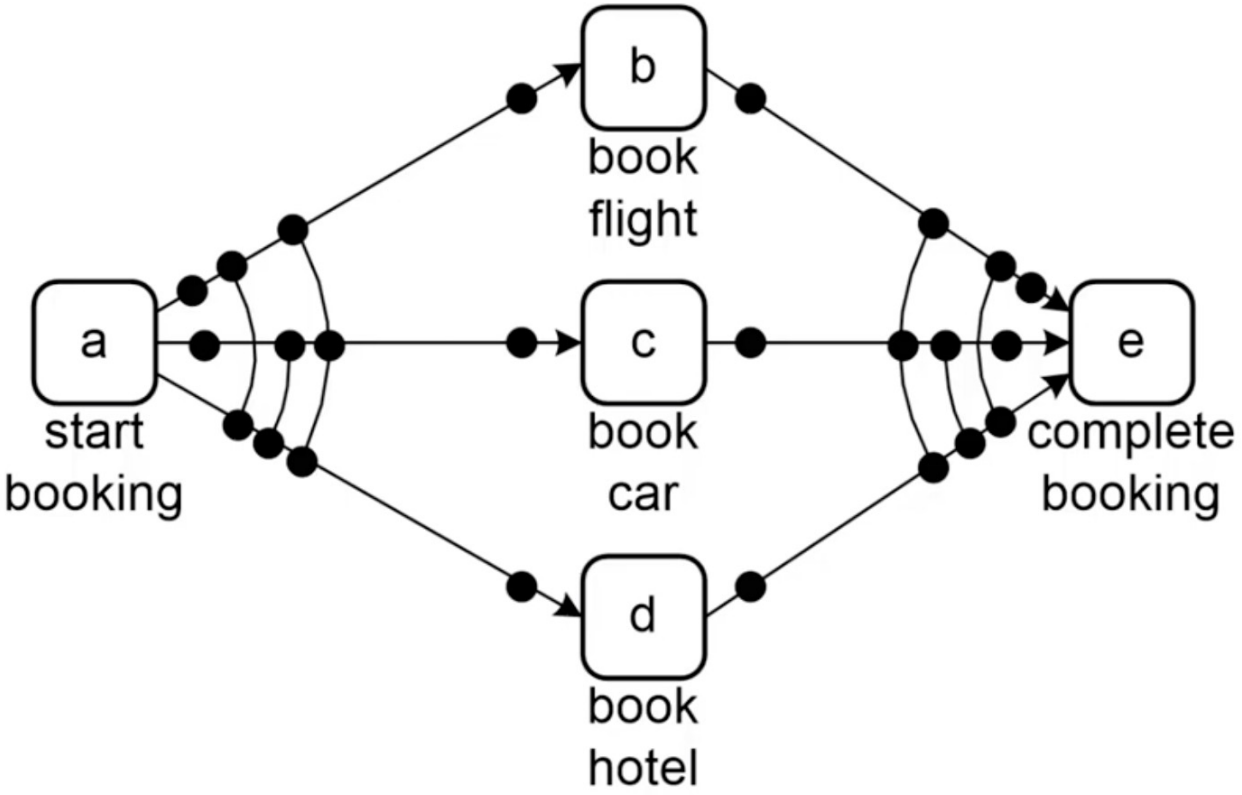




Deadlocking and livelocking firing sequences are considered

WF-net may not be sound

Homework: how may valid binding sequences?



ProM

Download ProM 6.11 from https://www.promtools.org/doku.php

Reading Material

- Chapter 3 & 7: Aalst