UNIVERSITY OF TARTU
1632

Unravelling Unstructured Process Models

Marlon Dumas, Luciano García-Bañuelos

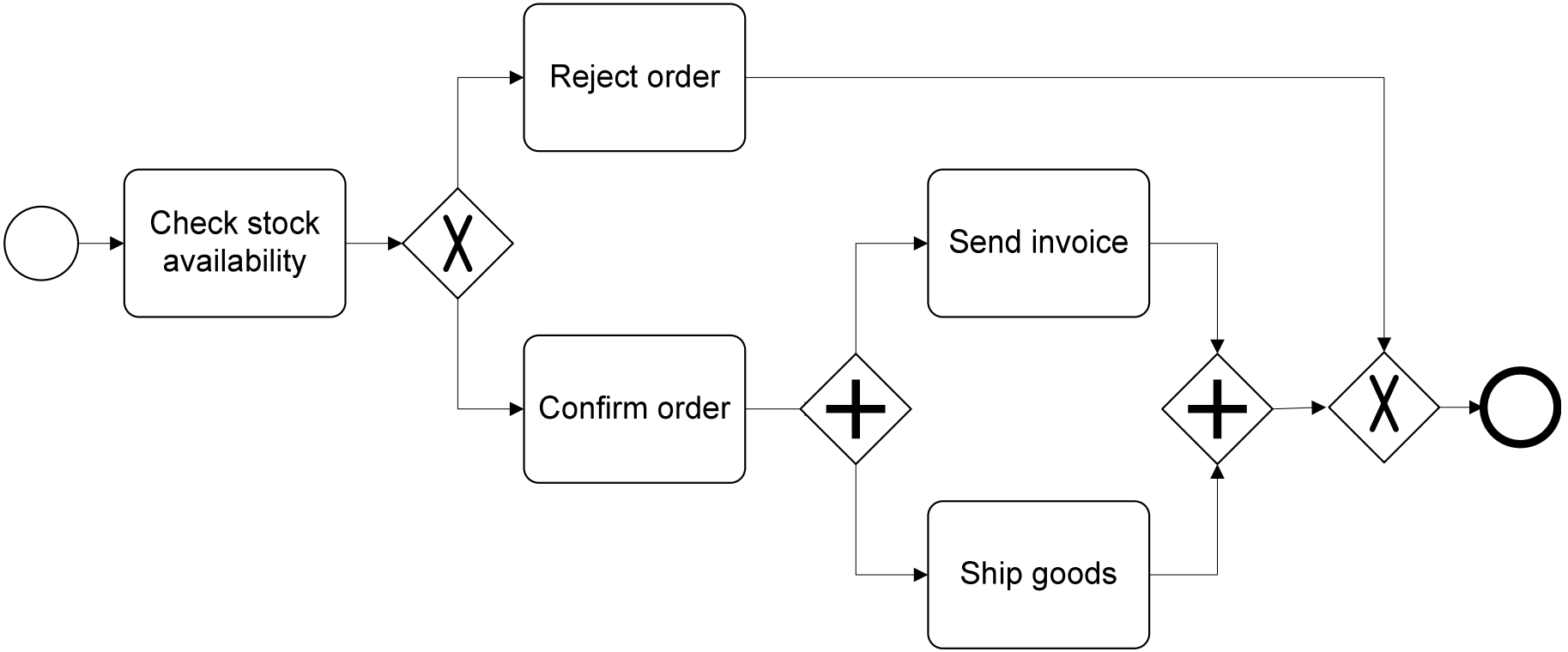
University of Tartu, Estonia

Artem Polyvyanyy

HPI, University of Potsdam, Germany

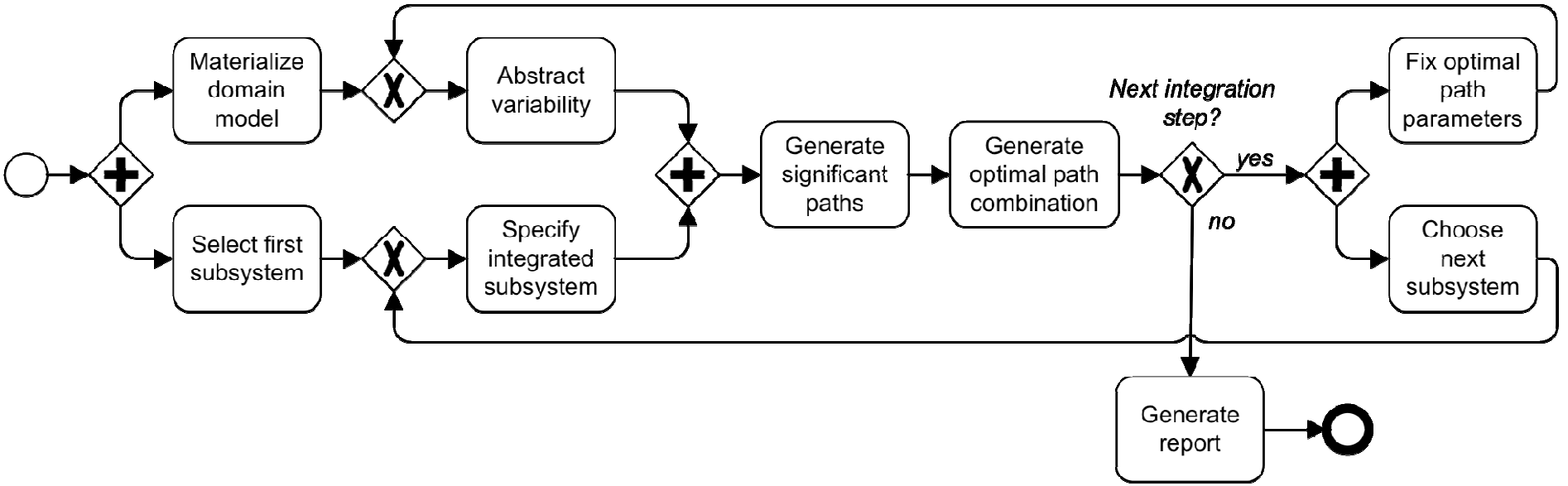
Theory Days, 1 Oct. 2010

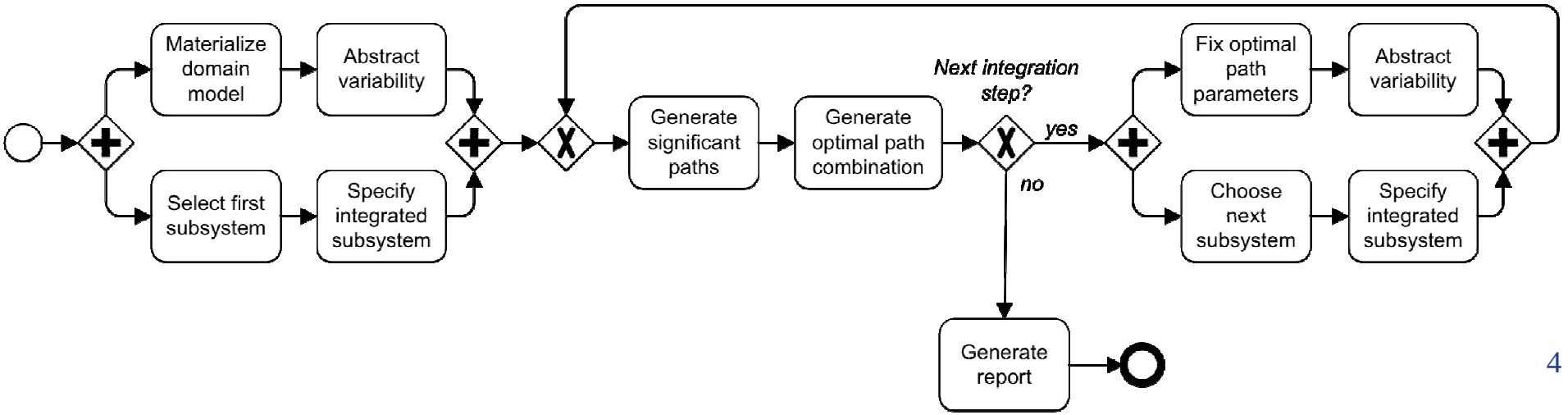
Background: Process Models



Poll: Which desk do you prefer?

Poll: Which model do you prefer?



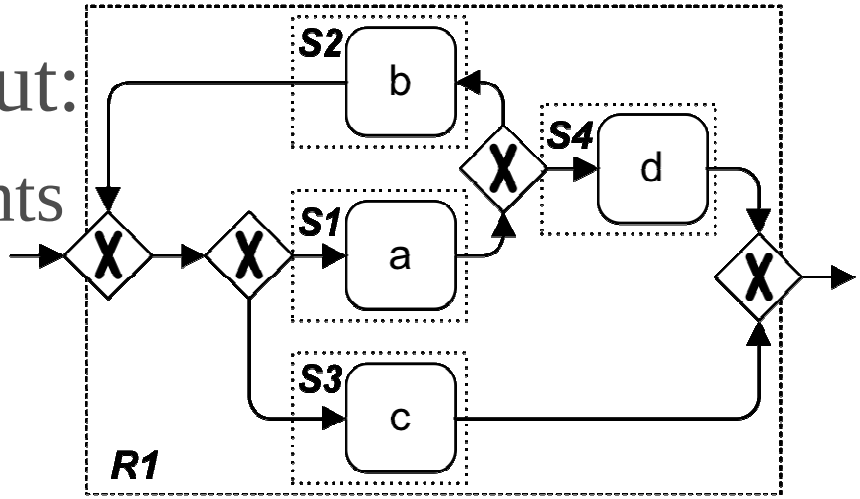


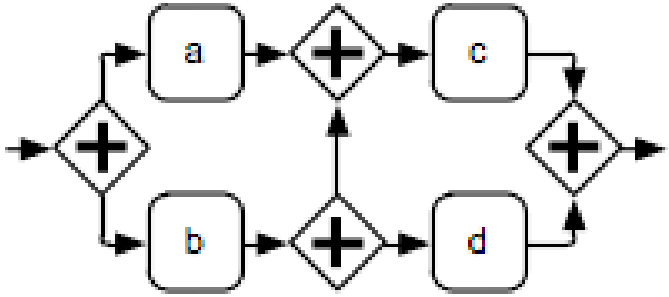
Problem Statement

- Premise: Structured is “better”
  - Easier to understand
  - Easier to analyze
  - Easier to automatically layout
  - Easier to abstract (zoom-out)
- We know not all models can be structured…
- Which ones can, which ones can’t?

Conclusion

- We can structure anything but:
  - Cycles with multiple exit points
  - Z-structures

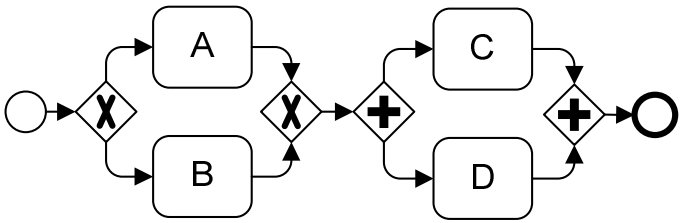




- Try it out: http://sep.cs.ut.ee/Main/bpstruct

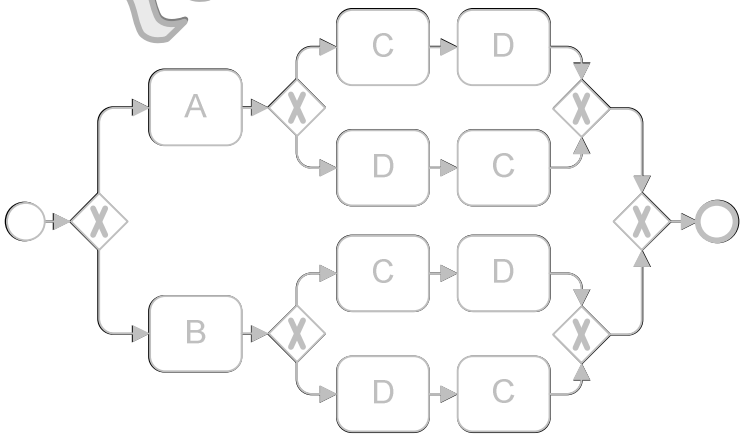
Behavioral Equivalence

Fully concurrent bisimilar (FCB)



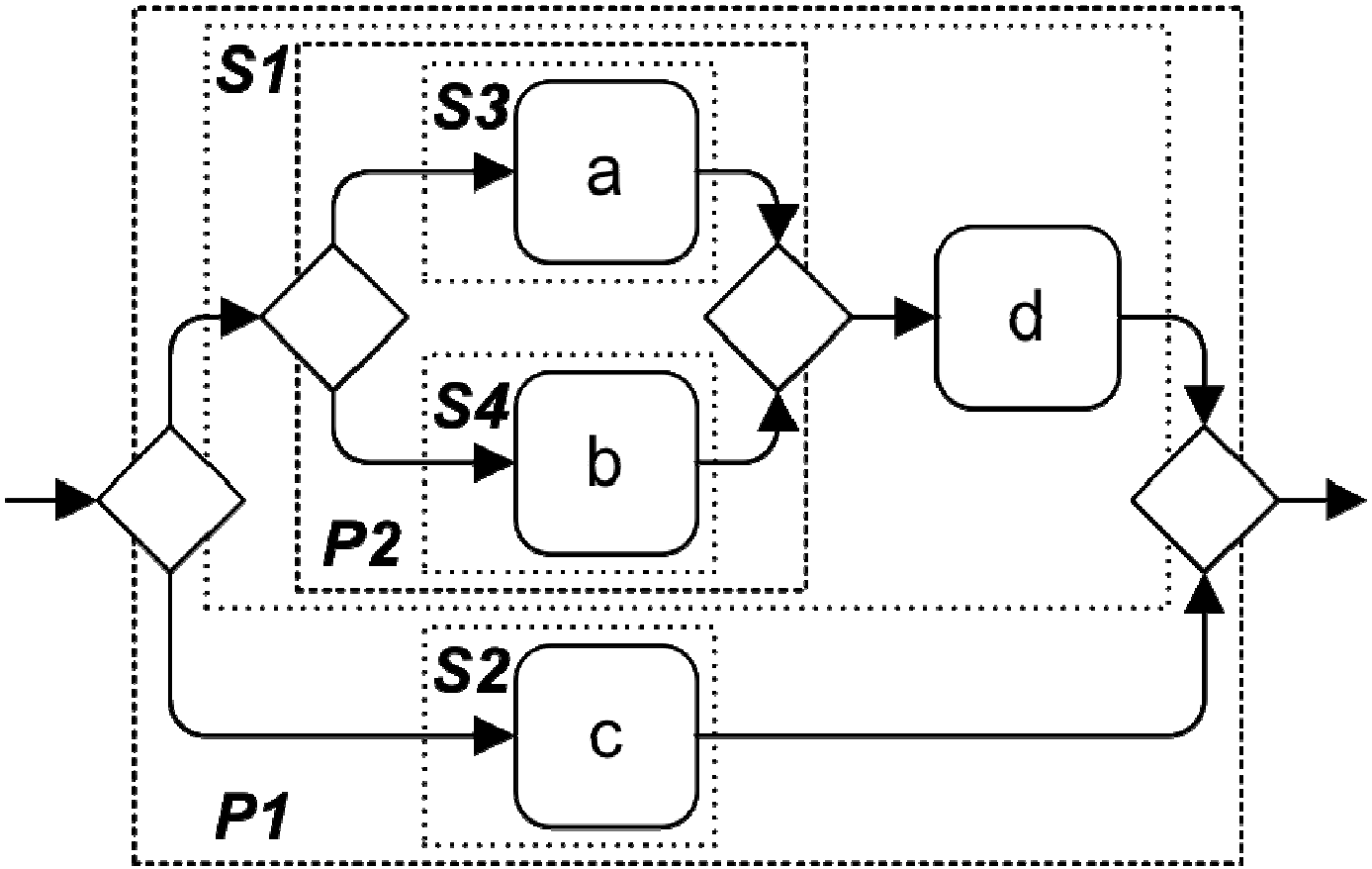
Preserves the level of concurrency in a process model

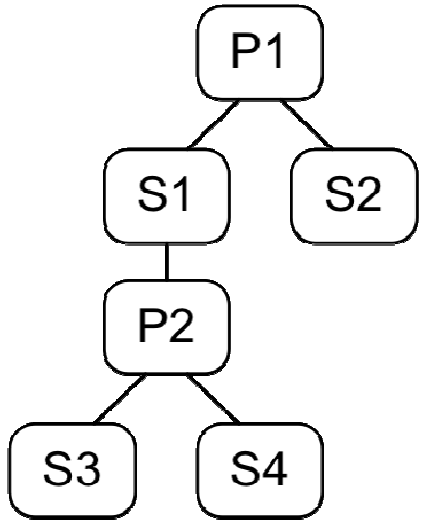
Weakly bisimilar



Sequential simulation of a process model

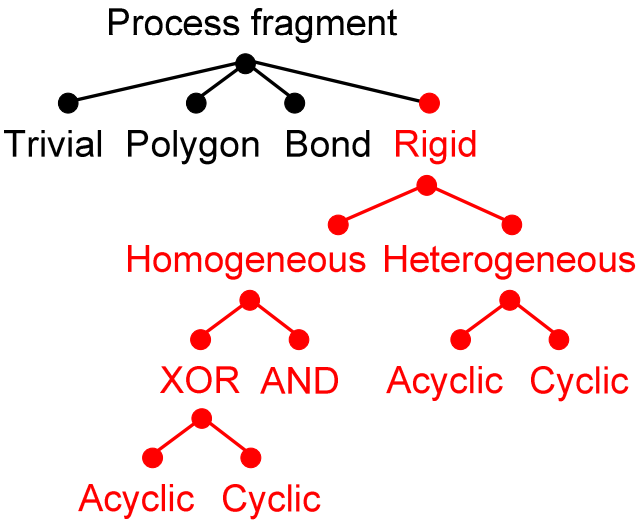
Starting Point – Process Structure Tree

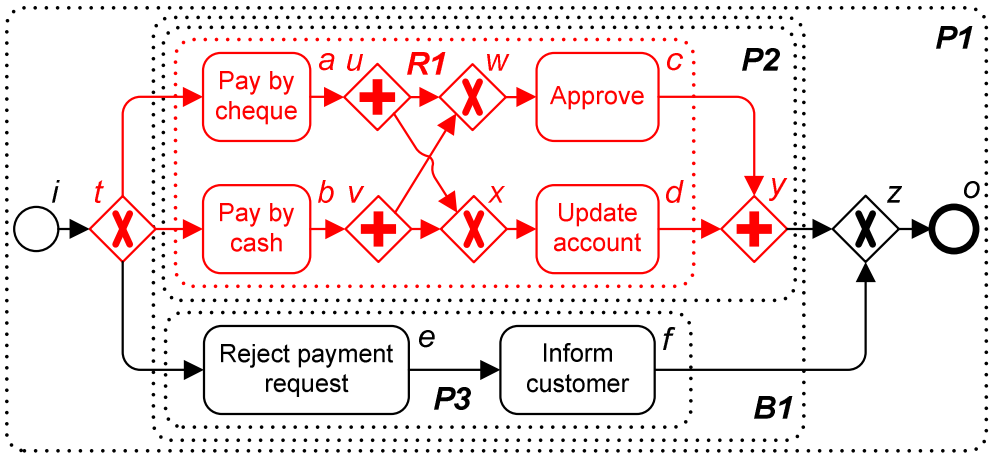




Taxonomy of Process Fragments

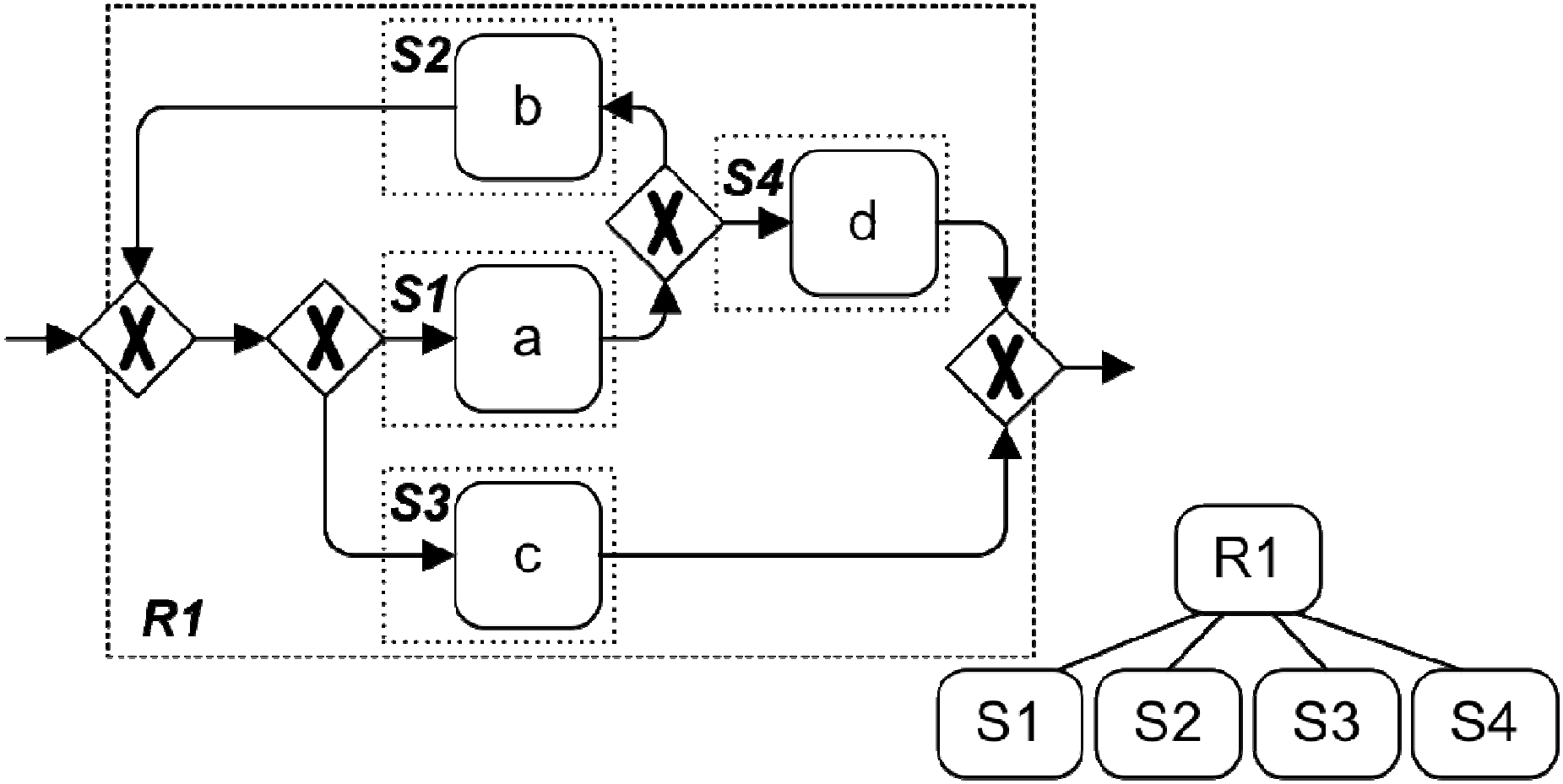
- Trivials, polygons, and bonds are structured fragments
- Rigids are "unstructured"



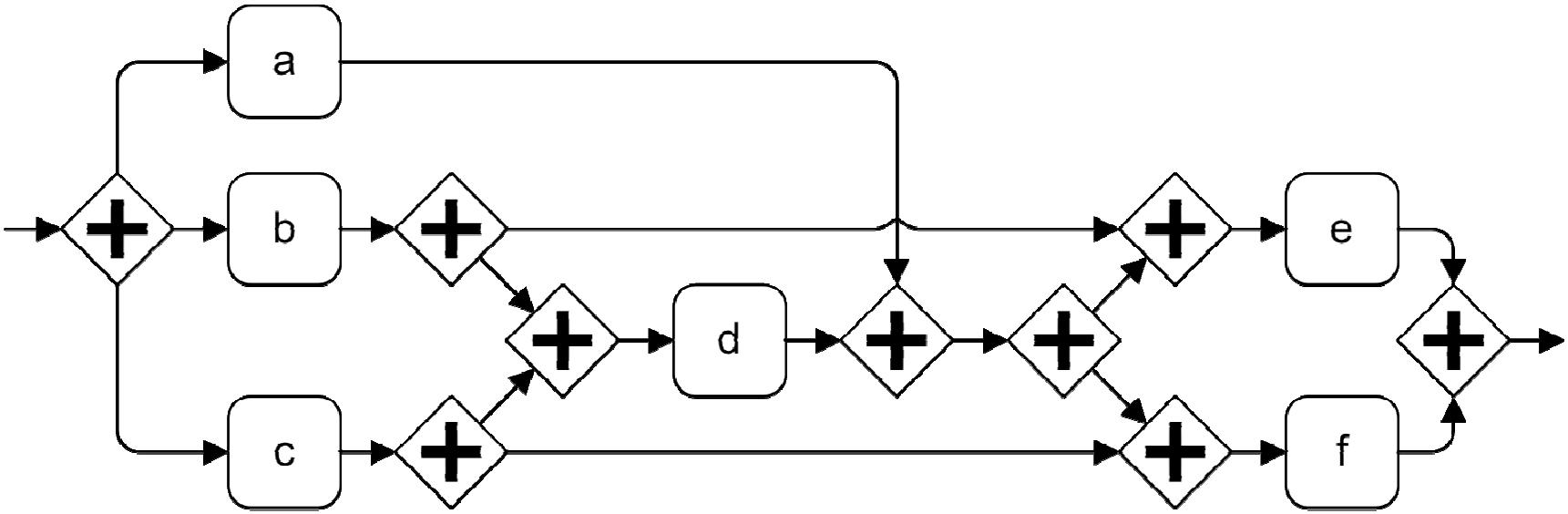


9

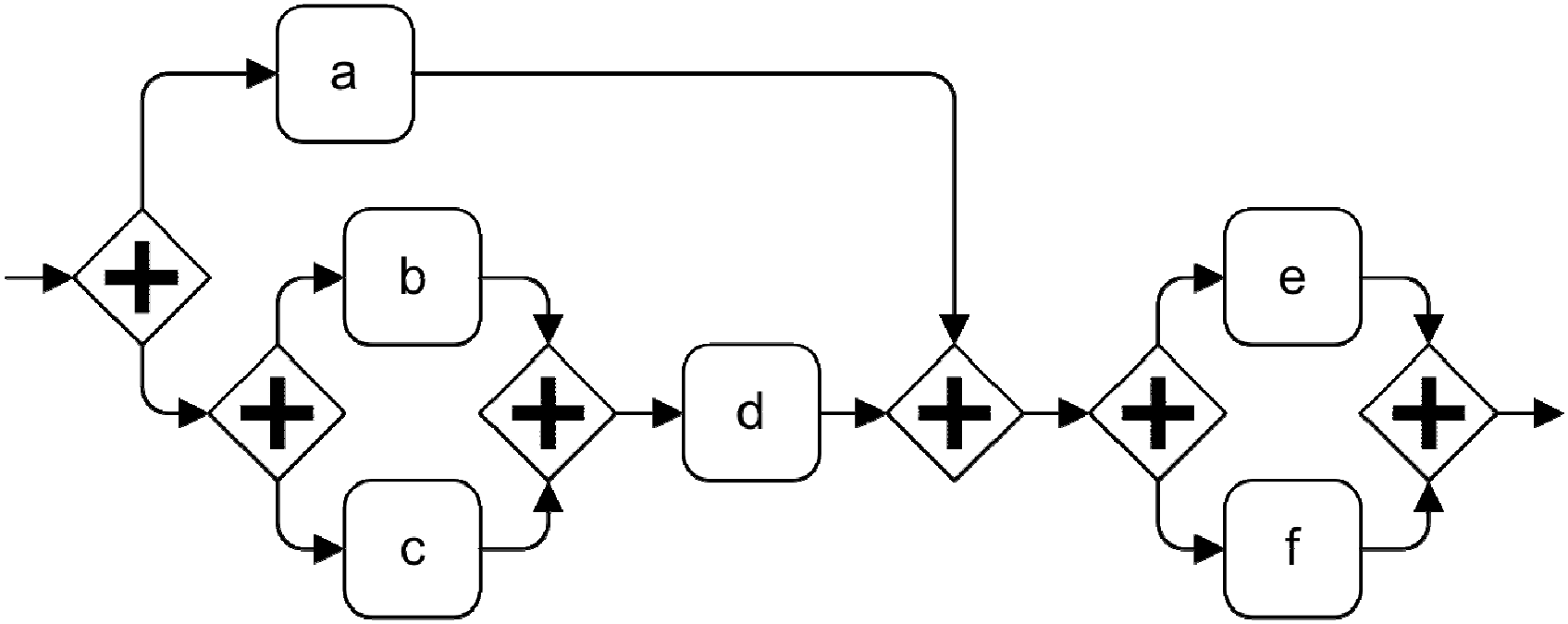
Homogeneous XOR Rigid



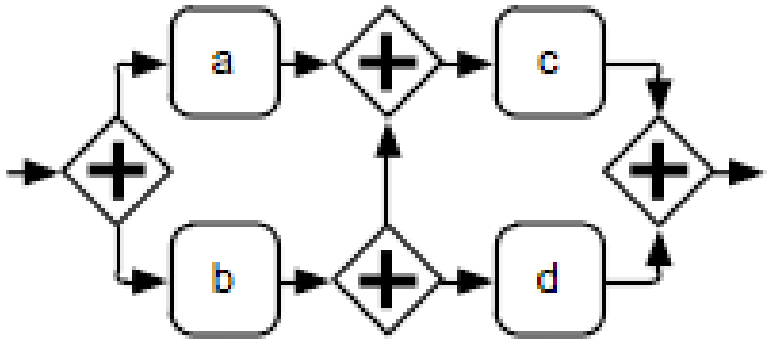
Homogeneous AND Rigid



Block-structured version…

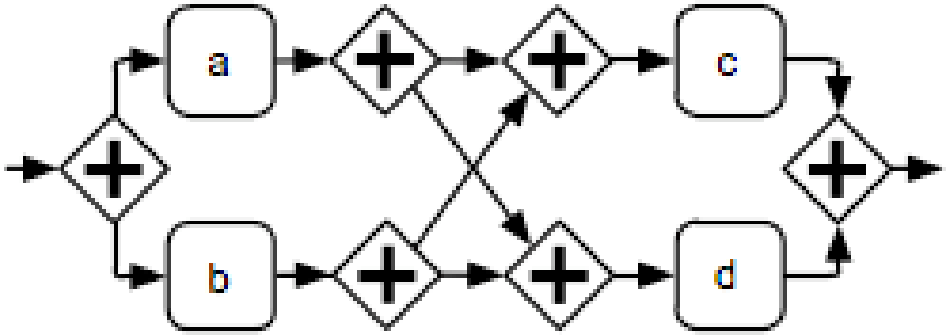


Homogeneous AND Rigid that cannot be structured



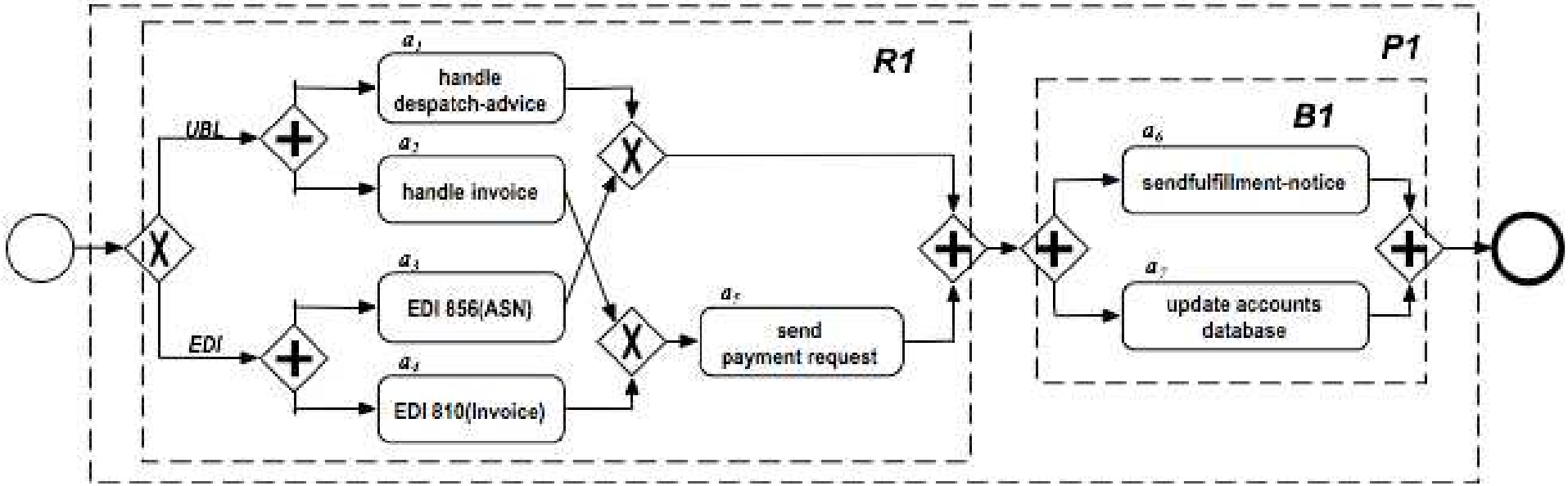
- Causal rules:
  - {A, B} → { C }
  - { B } → { D }
- Overlap on the left-hand side of the rules

Compare to this…

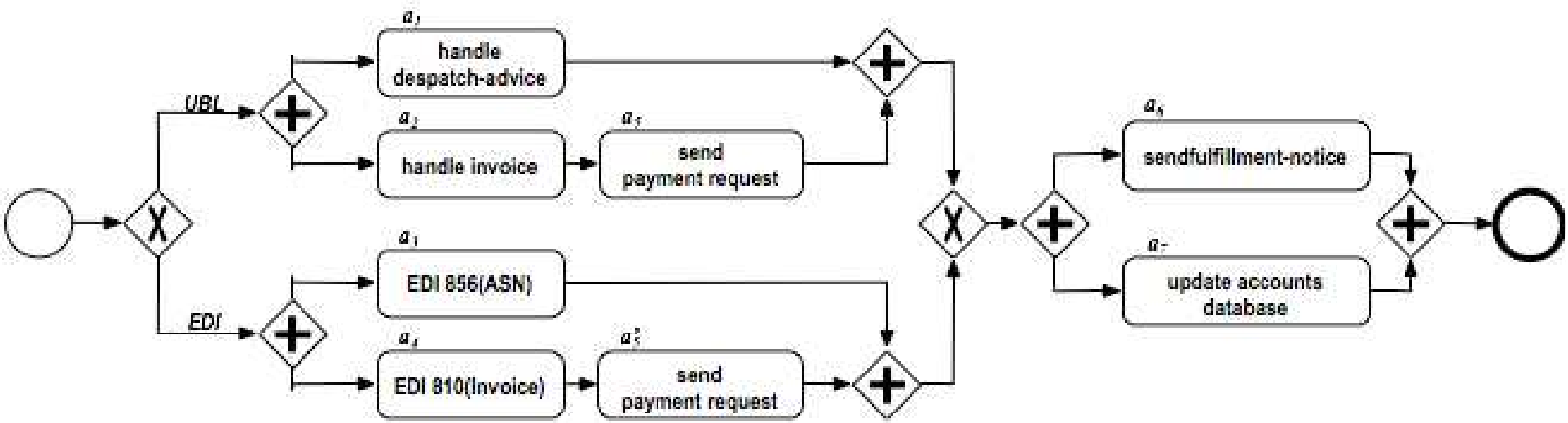


- Causal relations
  - {A, B} → {C, D}

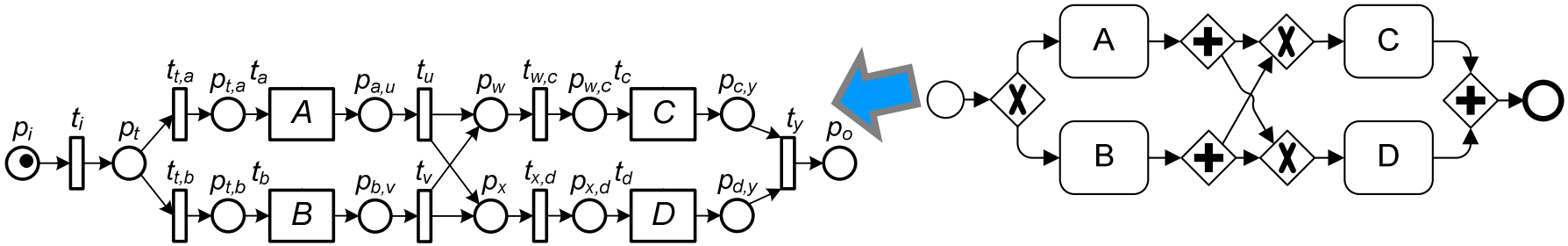
Heterogeneous Acyclic Rigid



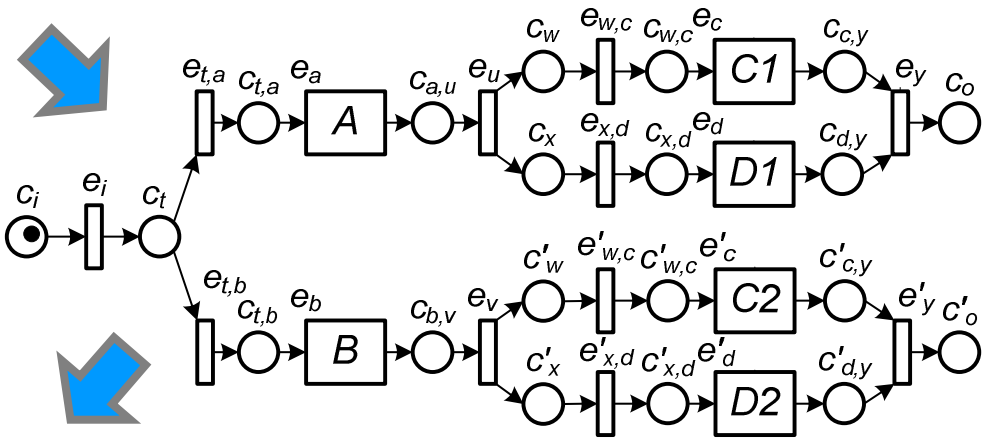
Equivalent Structured Fragment

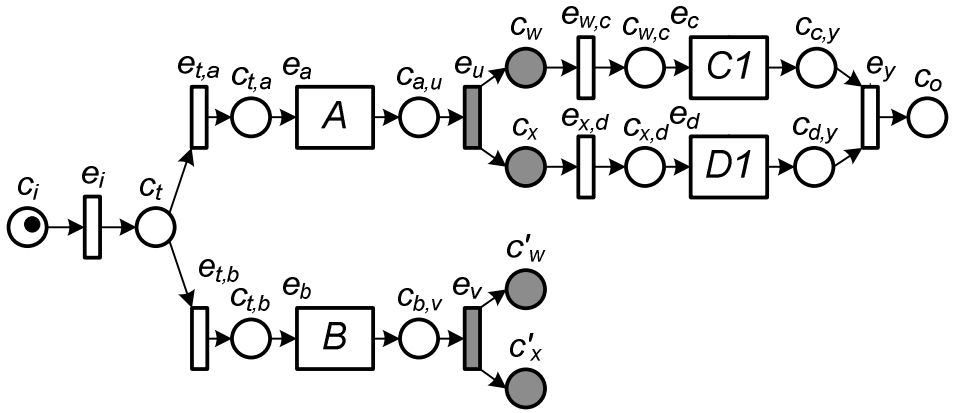


Complete Prefix Unfolding



An unfolding is a representation of a net without "merge" points





A complete prefix unfolding is a finite initial part of the unfolding that contains full information about the reachable states

Ordering Relations

A>C1

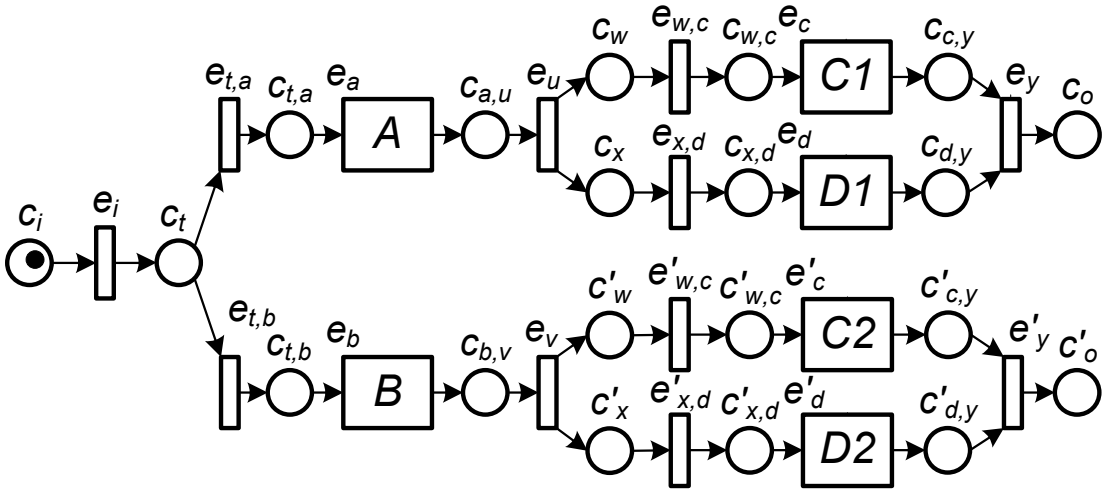
D2||C2

A#D2

B#A

C2||D2

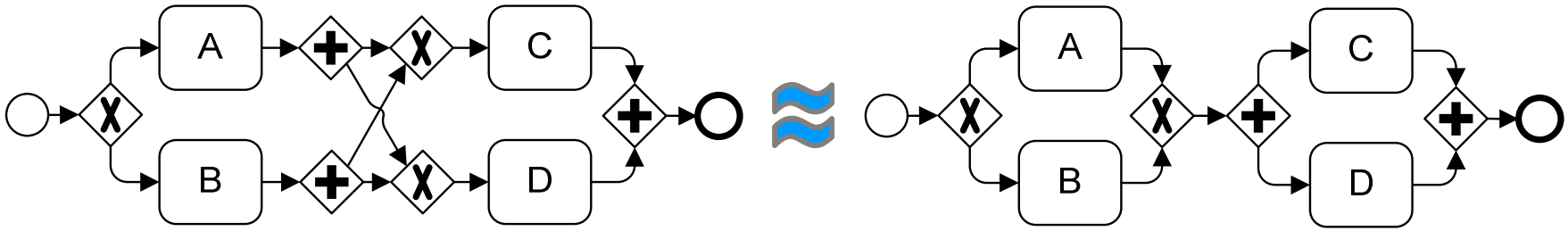
B>D2



- Two transitions of an occurrence net are in one of the following relations:
  - A and B are in *causal* relation (A>B), iff there exists a path from A to B
  - A and B are in *conflict* (A#B), iff there are two transitions t1, t2 that share an input place and there is a path from t1 to A and a path from t2 to B
  - A and B are in *concurrency* (A||B) relation iff A and B are neither in causal, nor in conflict relation

FCB and Ordering Relations

Two process models are FCB-equivalent …



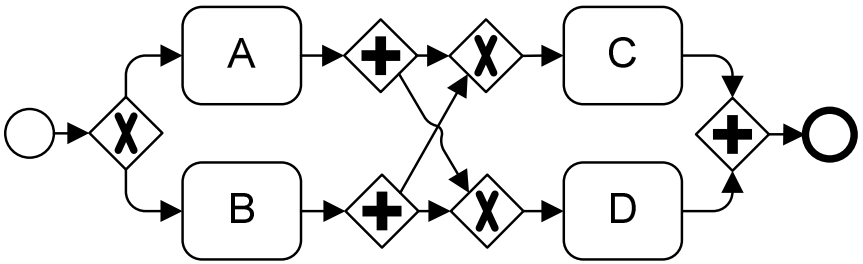
… if and only if, (complete prefix) unfoldings of both models expose same ordering relations

	A	B	C	D
A		#	>	>
B	#		>	>
C	<	<		||
D	<	<	||	

≈

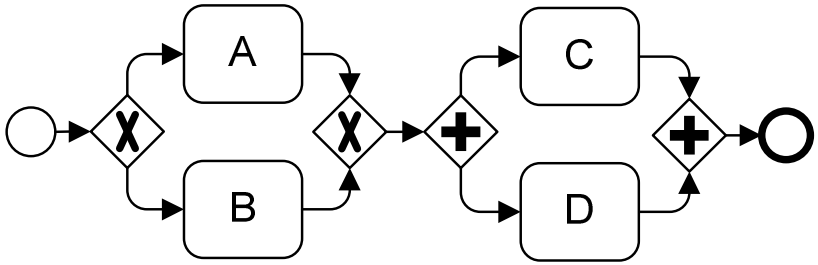
	A	B	C	D
A		#	>	>
B	#		>	>
C	<	<		||
D	<	<	||	

Structuring Process Models



Compute ordering relations of a graph-structured process model

	A	B	C	D
A		#	>	>
B	#		>	>
C	<	<		||
D	<	<	||	

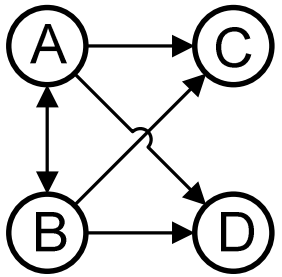


Construct a block-structured process model from ordering relations

20

Ordering Relations Graph

	A	B	C	D
A		#	>	>
B	#		>	>
C	<	<		||
D	<	<	||	

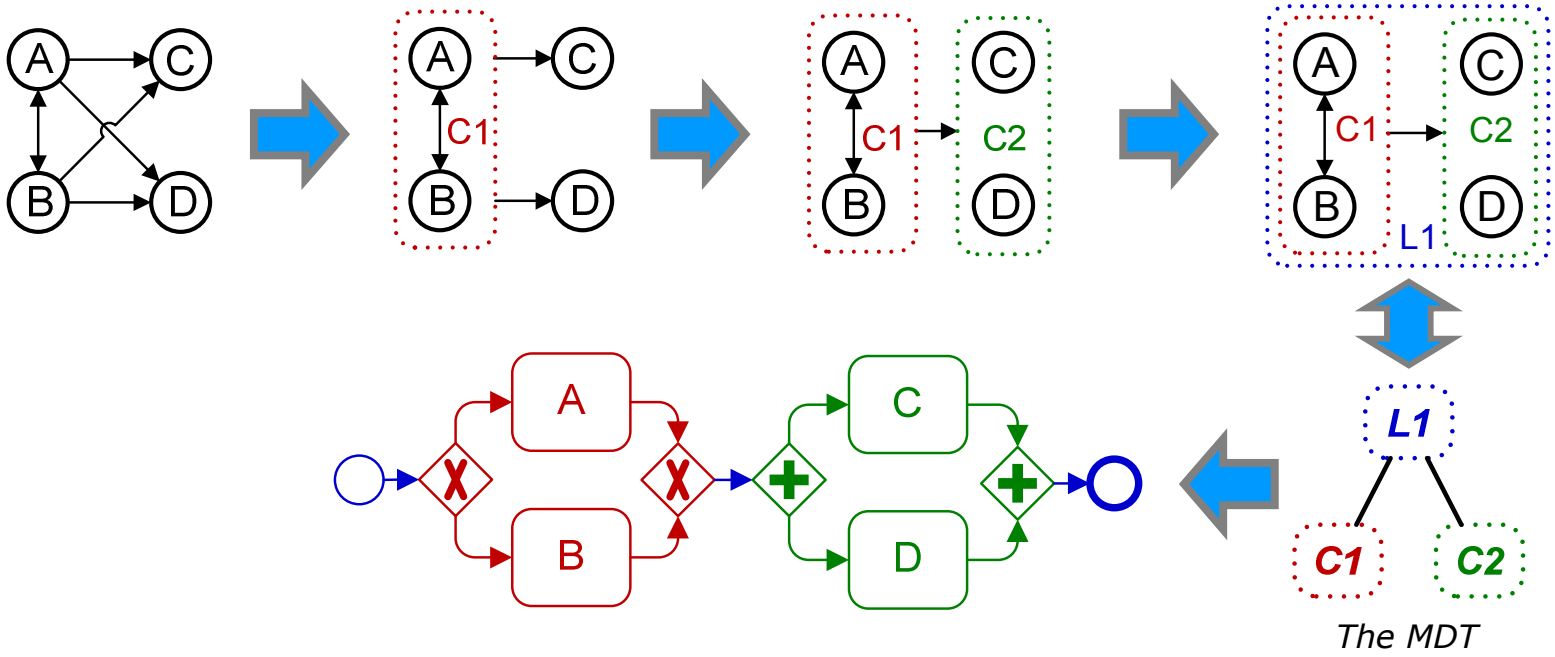


An ordering relations graph

Modular Decomposition Tree (MDT)

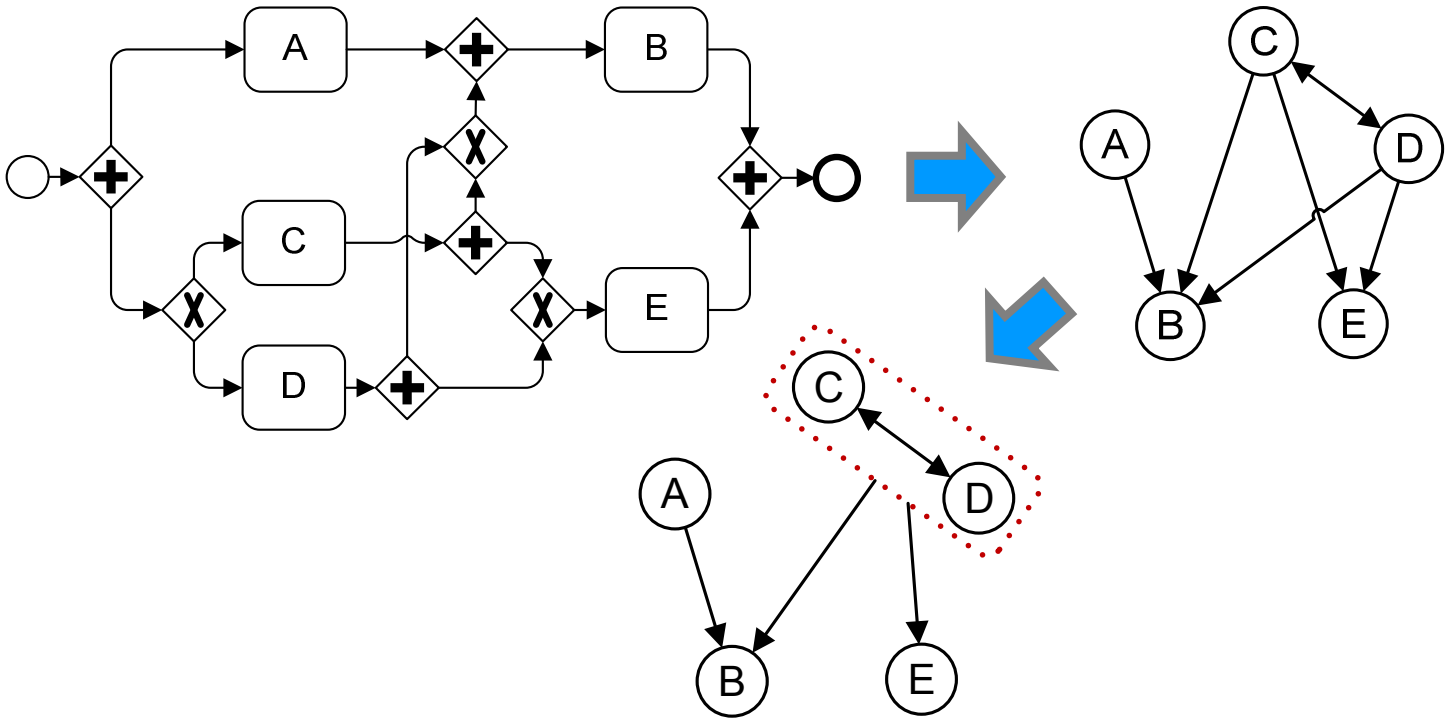
- A module is a set of edges with uniform
- A *linear* (L) module is a total order on a set of nodes of a graph
- A *complete* (C) module is a complete graph, or a clique
- A *primitive* (P) module is neither trivial, nor linear, nor complete

- The MDT is *unique* and can be computed in *linear time*



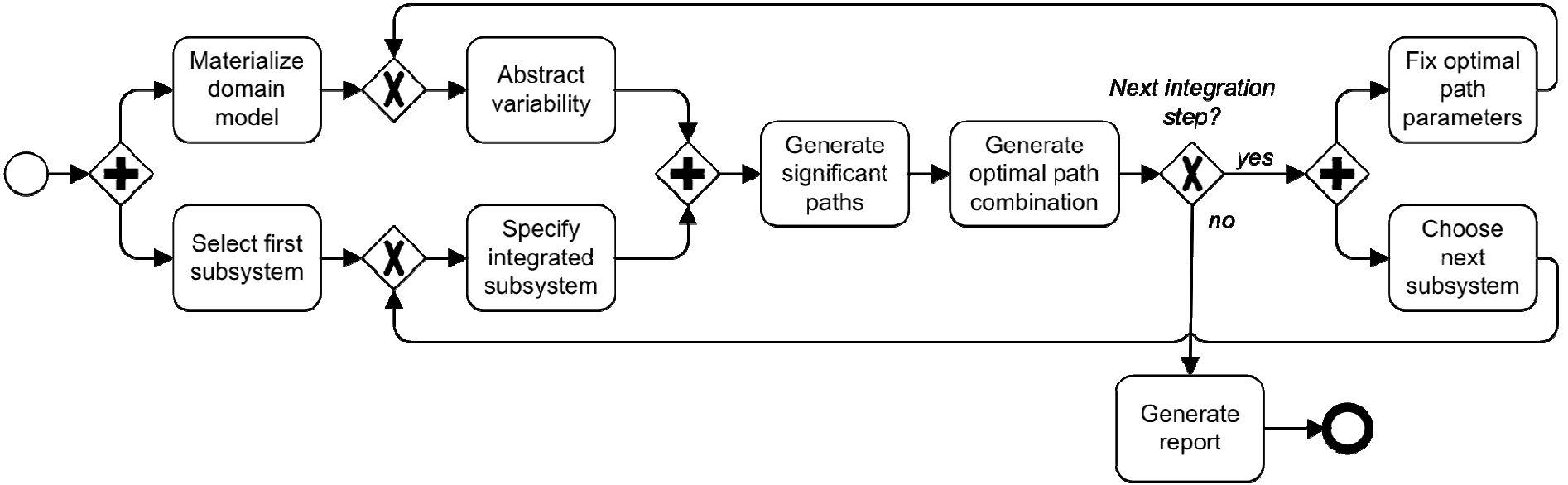
Structuring Acyclic Process Models

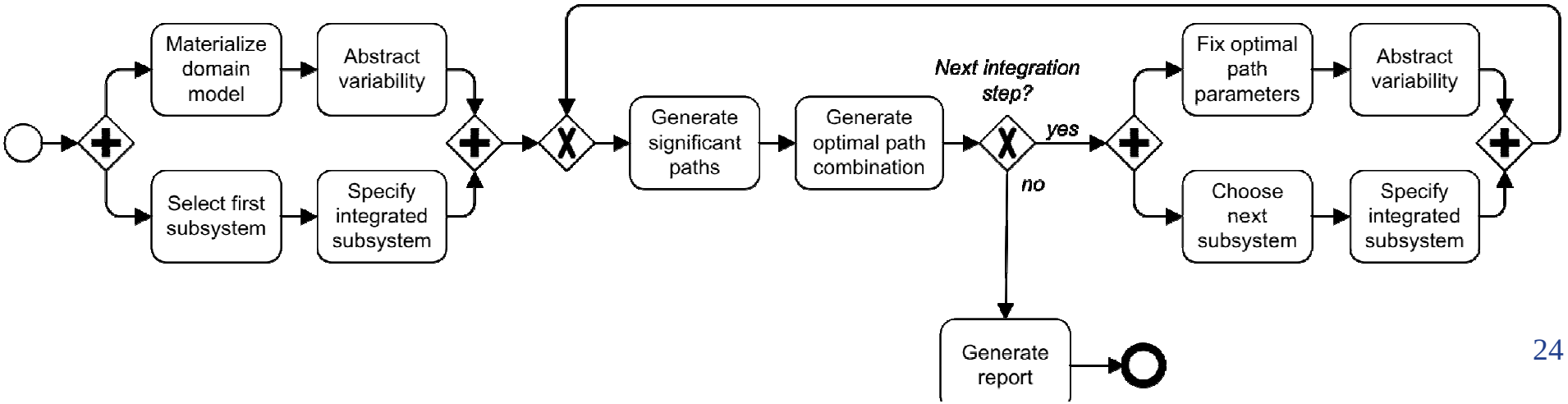
Let G be an ordering relations graph. The MDT of G has no primitive module, iff there exists a well-structured process model W such that G is the ordering relations graph of W.



A primitive module

Heterogeneous Cyclic Rigid





For further details…

- A. Polyvyanyy, L. García-Bañuelos, M. Dumas. "Structuring Acyclic Process Models". In *Proc. of the 8th Int. Conf. on Business Process Managament (BPM'2010)*, Hoboken, NJ, USA, September 2010, Springer LNCS.

- Tool available at: http://sep.cs.ut.ee/Main/bpstruct