“Detecting Frequently Recurring Structures in BPMN 2.0 Process Models”

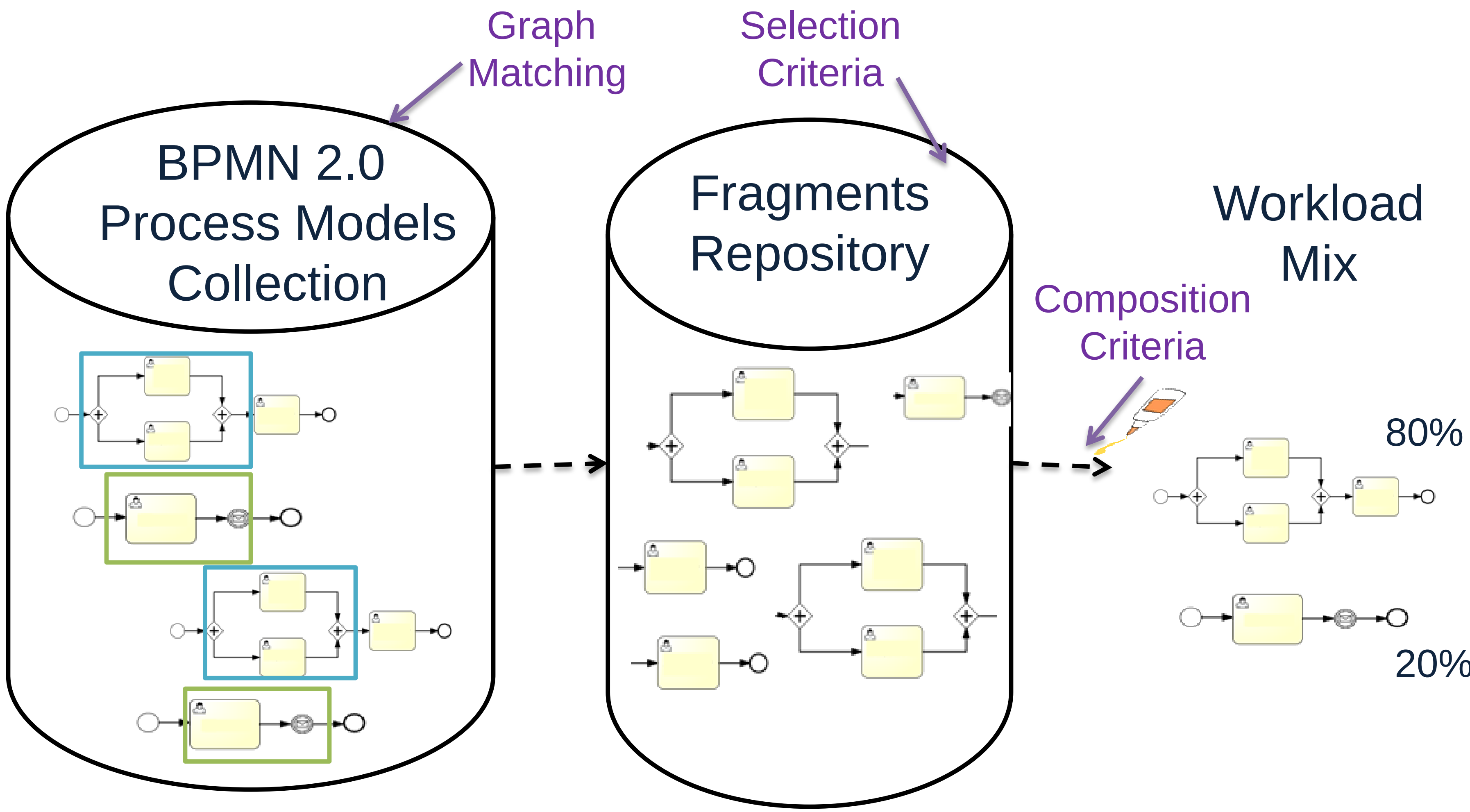
SummerSOC 2015

University of Stuttgart
Universitätsstr. 38
70569 Stuttgart
Germany

Marigianna Skouradaki, Frank Leymann
Institute of Architecture of Application Systems
{firstname.lastname}@iaas.uni-stuttgart.de

Phone +49-711-685 88477
Fax +49-711-685 88472

Motivation: Generation of Realistic Workload



Agenda

- Process Model Matching
- Basic Concepts
- Algorithms
- Validation & Discussion
- Conclusions & Outlook

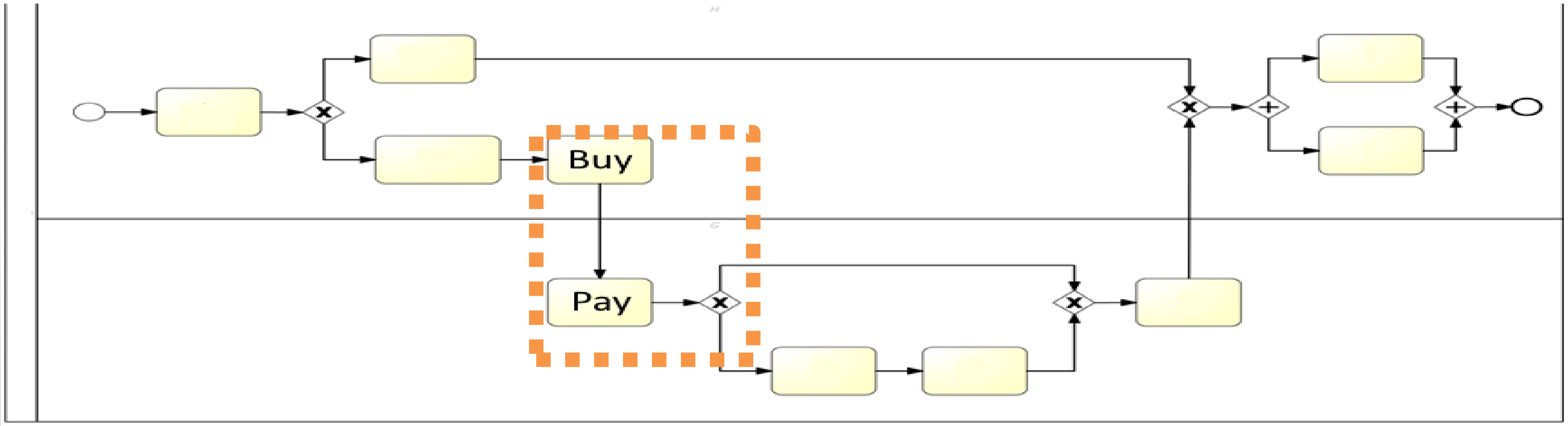
Process Model Matching

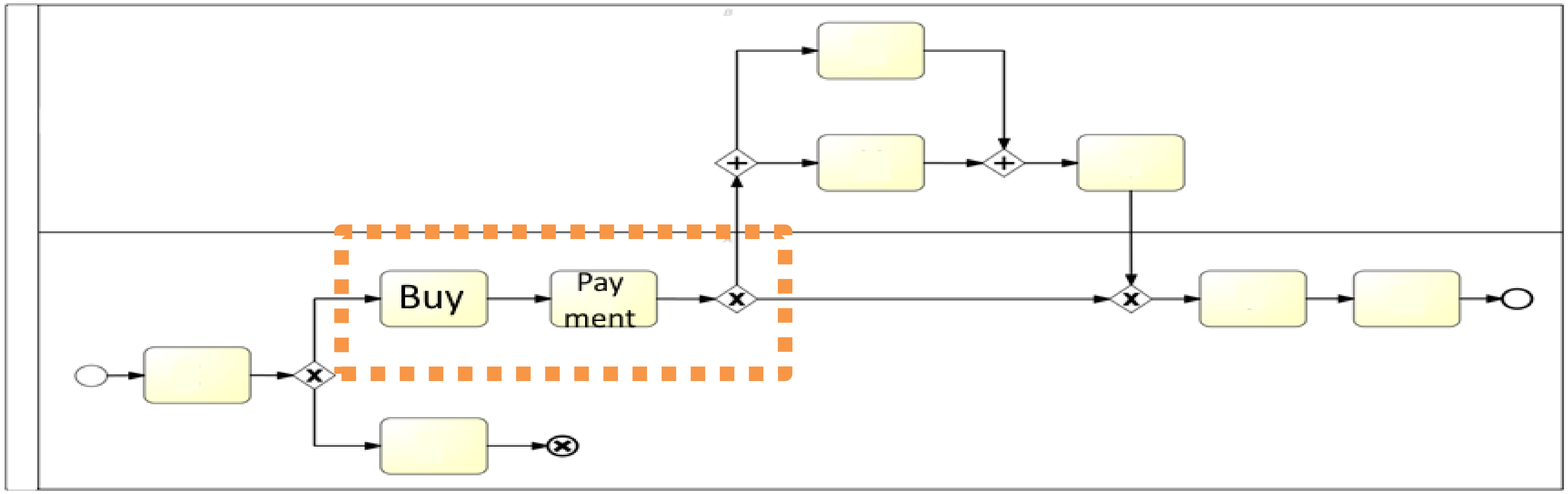
Research

BPMN 2.0 Collection Characteristics

- Detect the reoccurring structures on BPMN 2.0 process models which:
  - Might be anonymized (*no text information available*)
  - Might be mock-up models (*non-executable*)

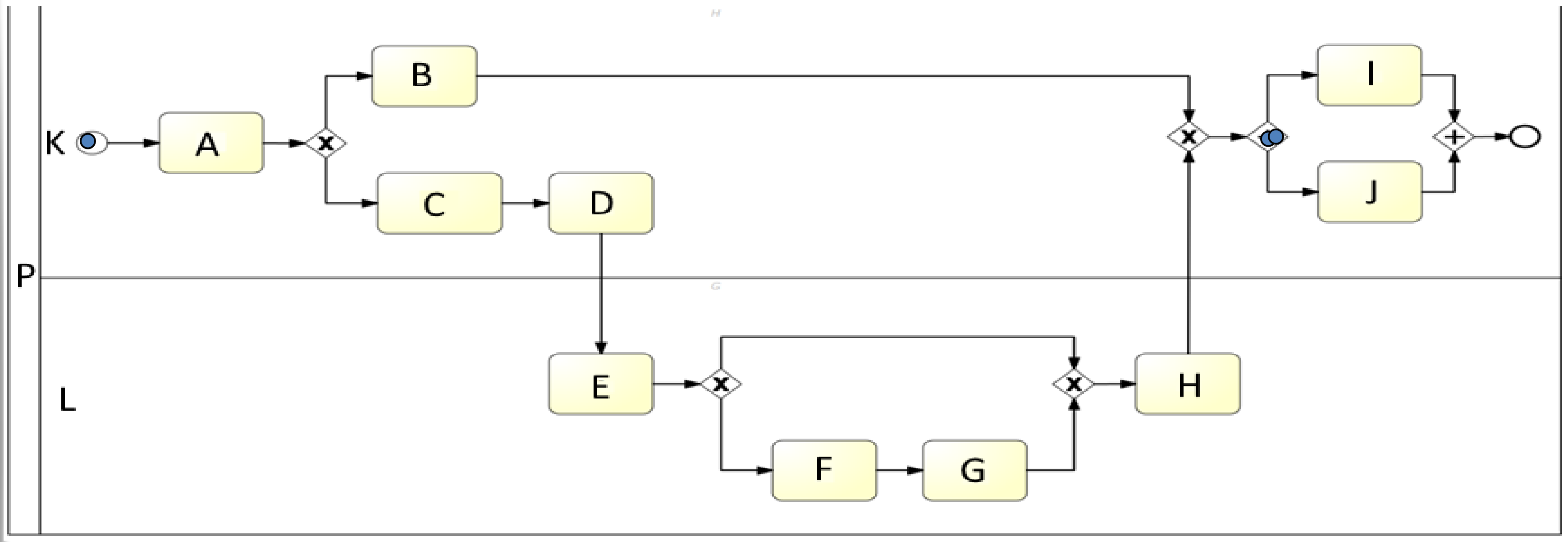
Text Matching

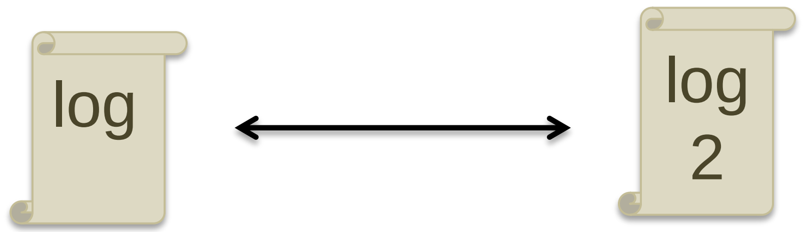




☹ Cannot be applied to anonymized models

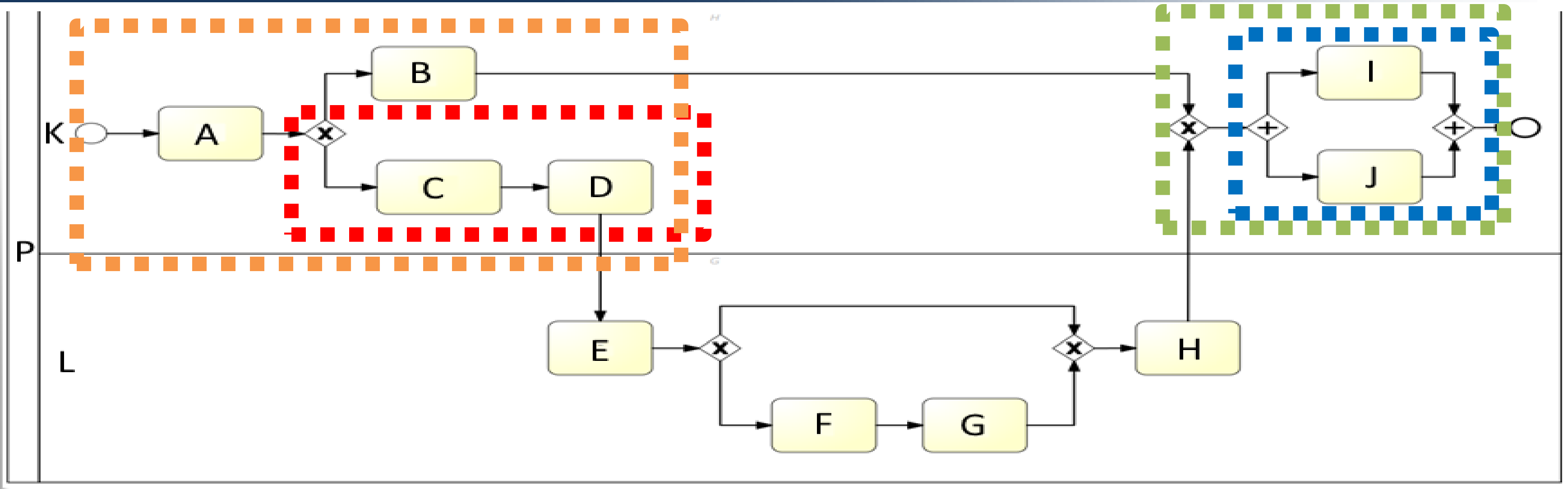
Behavioral Matching

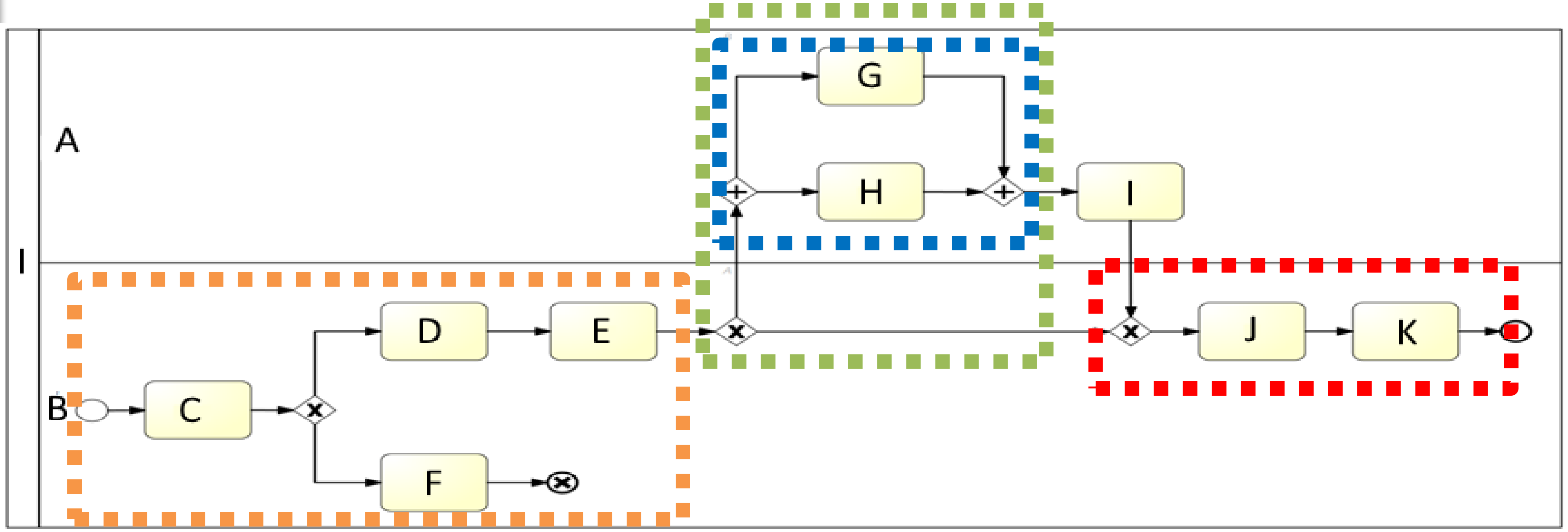




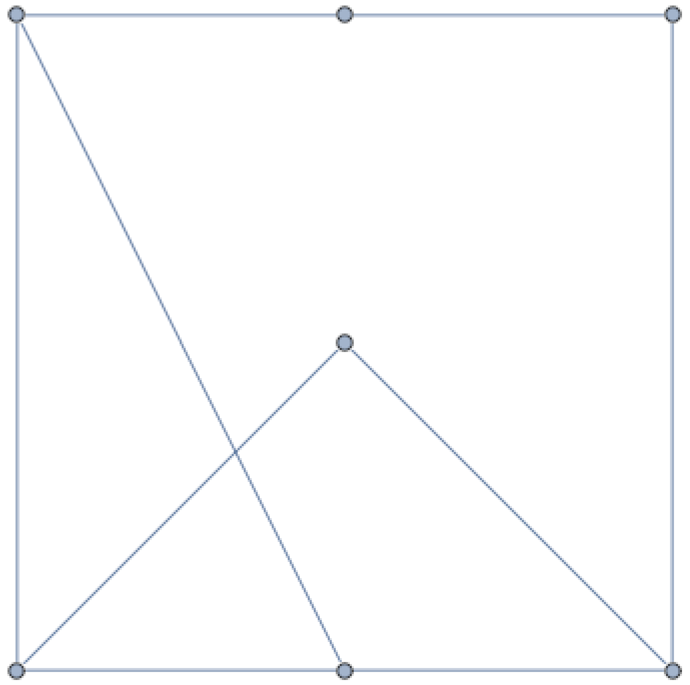
☹ Cannot be applied on mock-up models

Structural Matching





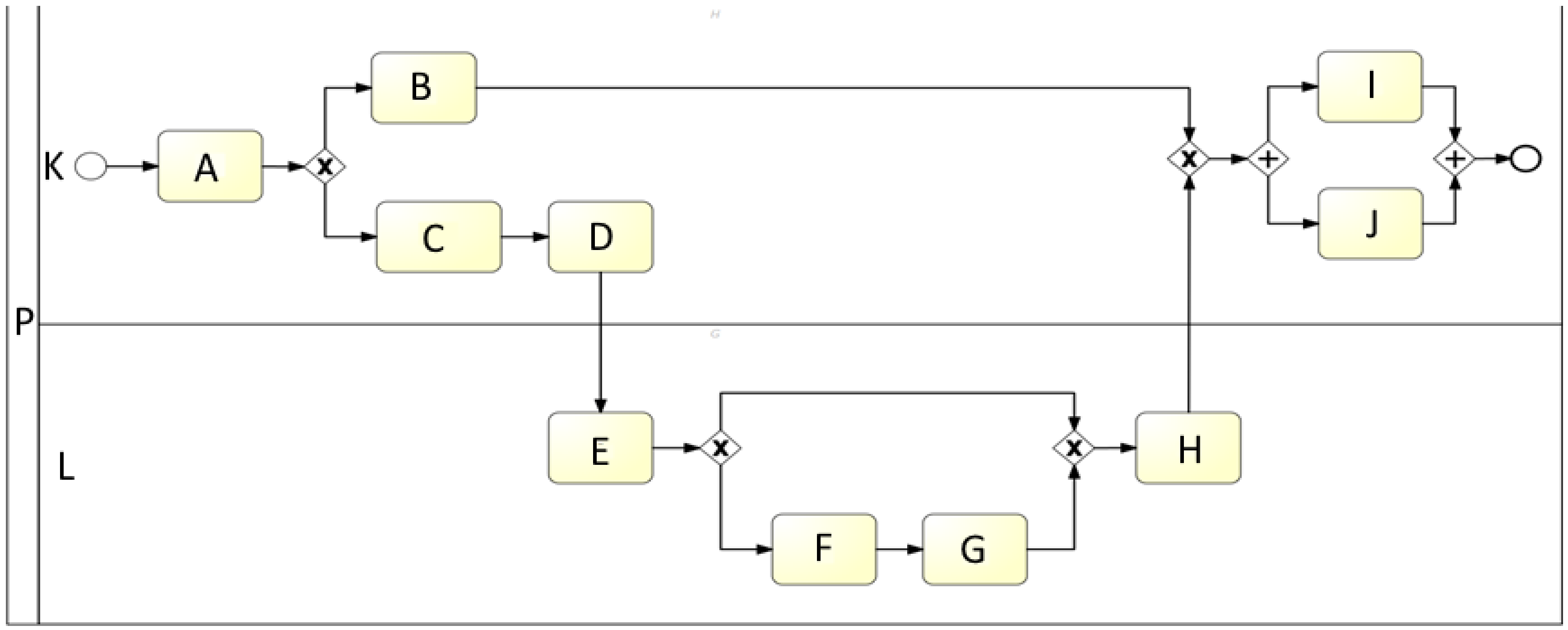
The Challenge of Graph Isomorphism



NonDeterministic Polynomial Time
(NP – Complete)

The time required to solve the problem using any currently known algorithm increases very quickly as the size of the problem grows.

Subgraph Isomorphism on BPMN 2.0 Process Models



BPMN 2.0 Process Models are **special types of graphs**

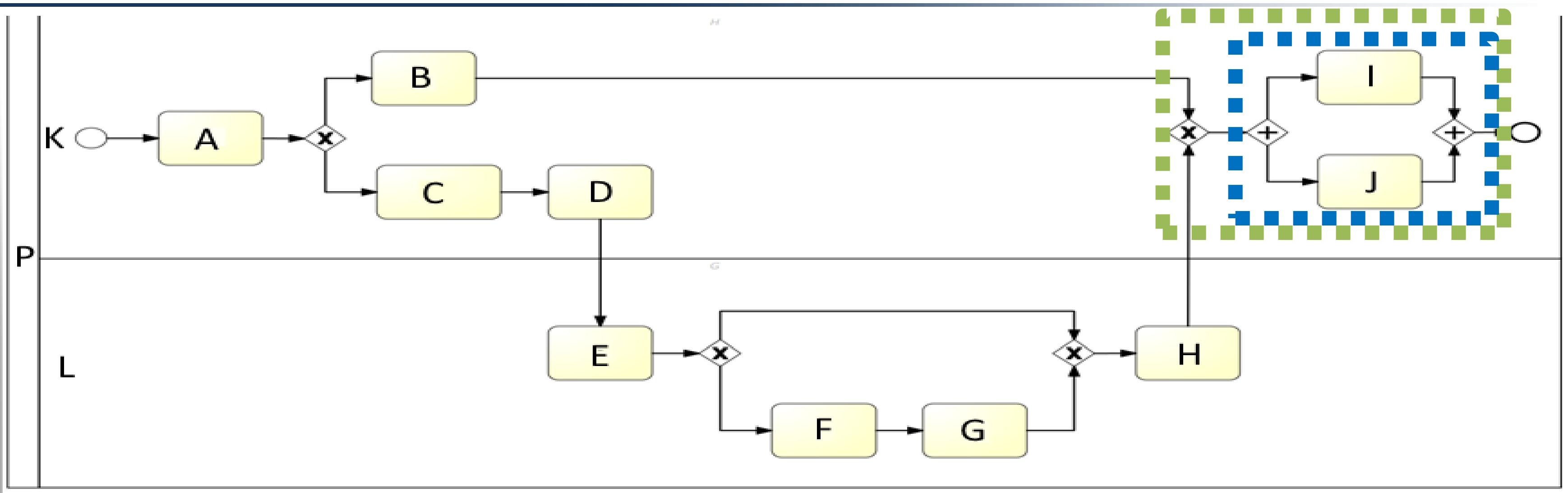
Subgraph isomorphism can be applied in **lower complexity**[1]

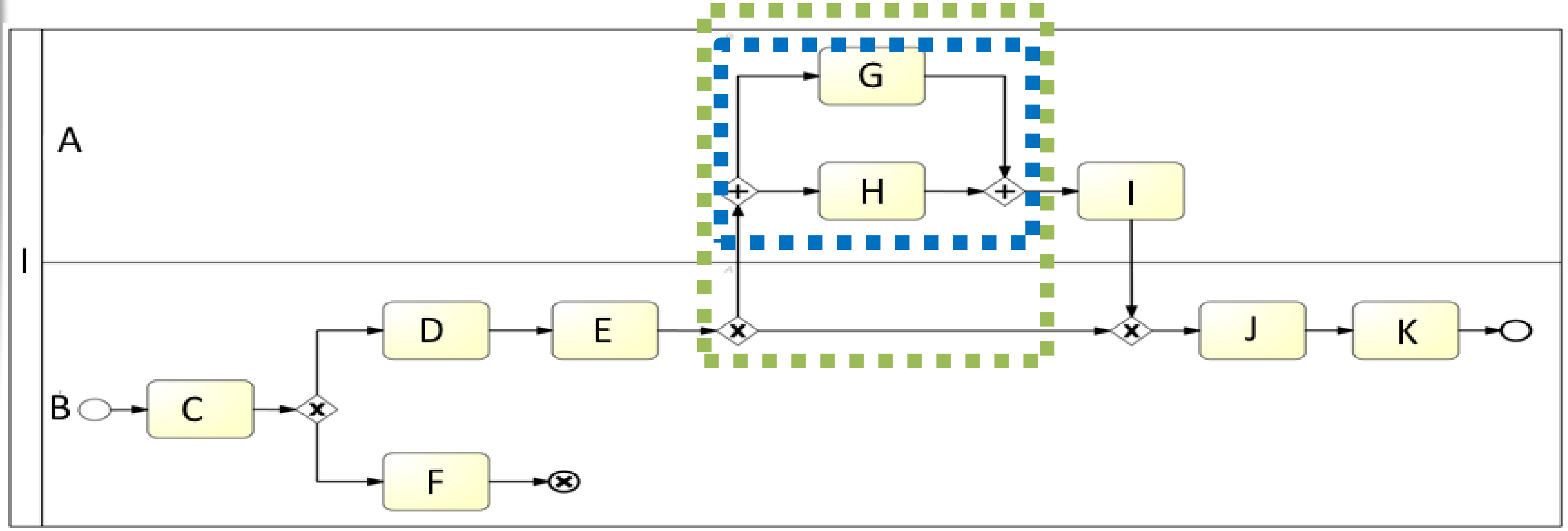
[1] R. M Verma.; and S. W. Reyner; “An analysis of a good algorithm for the subtree problem, correlated,” SIAM J. Comput., vol. 18, no. 5, pp. 906–908, Oct. 1989.

Basic Concepts

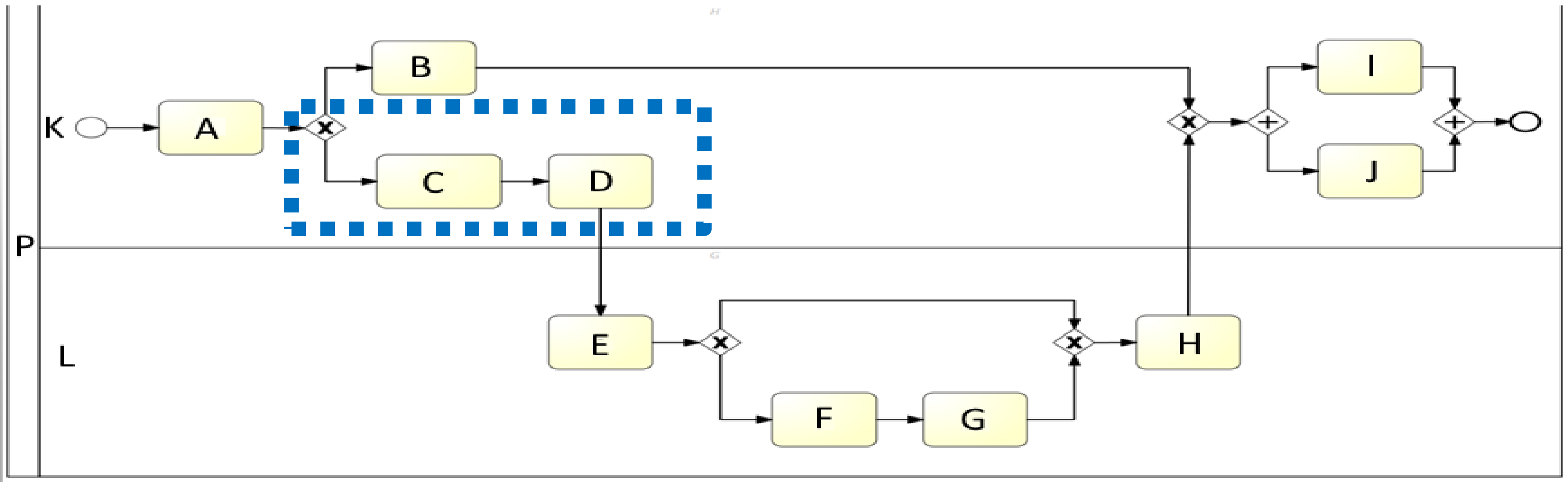
Research

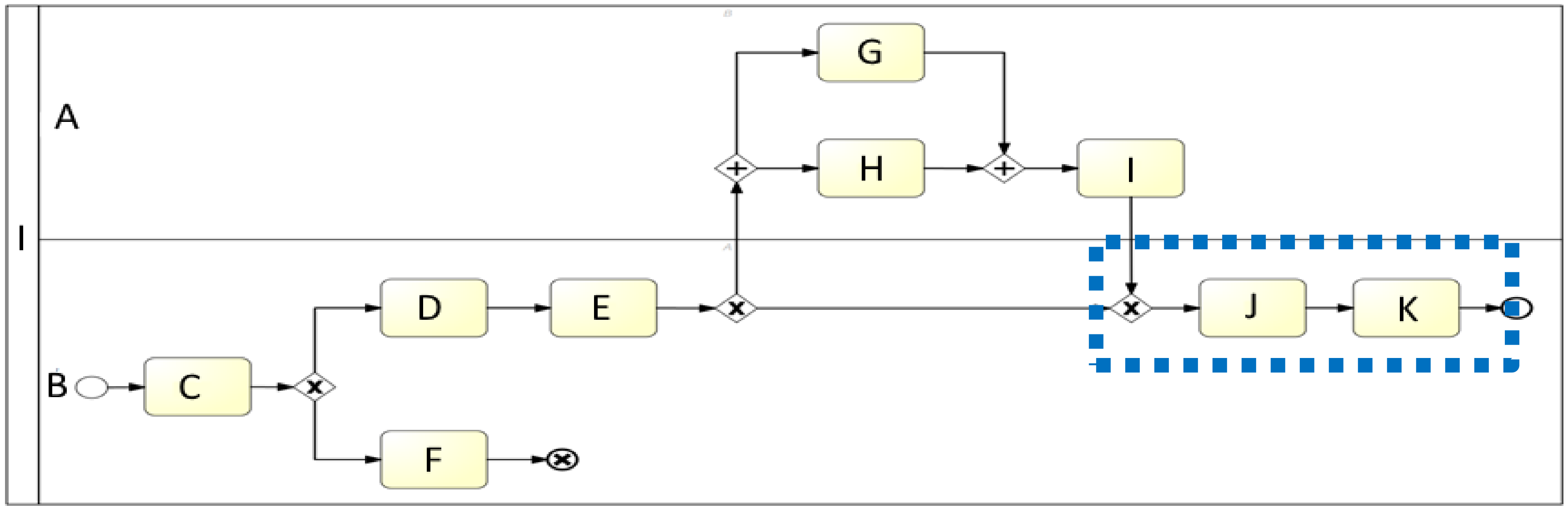
Exiting Attributes: Nested



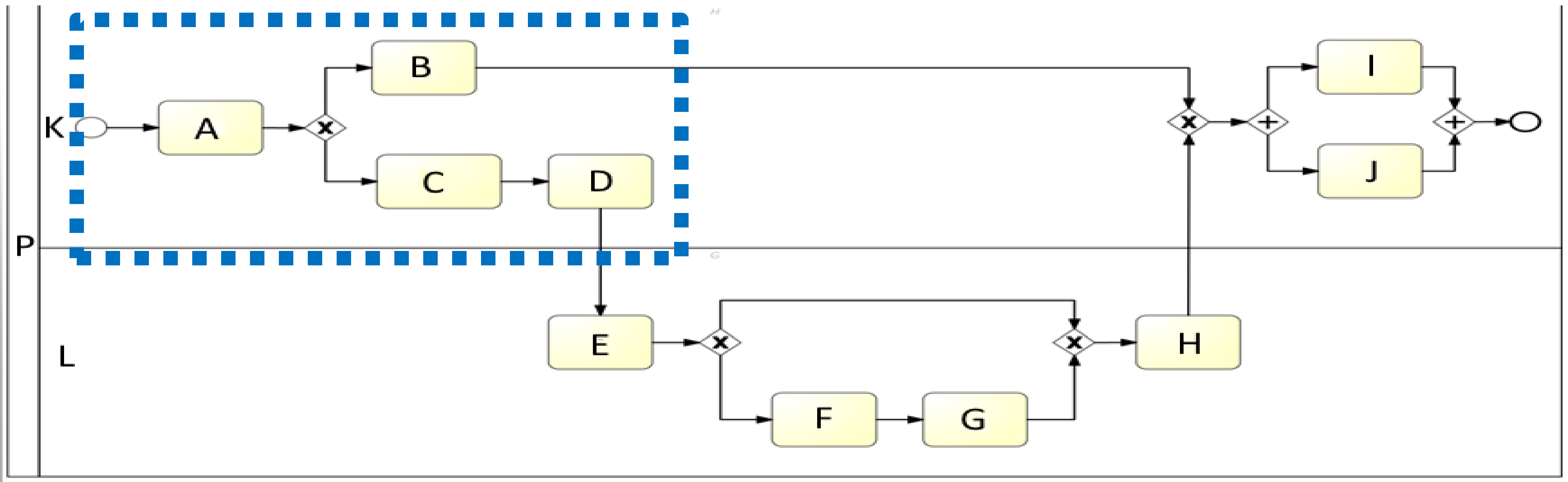


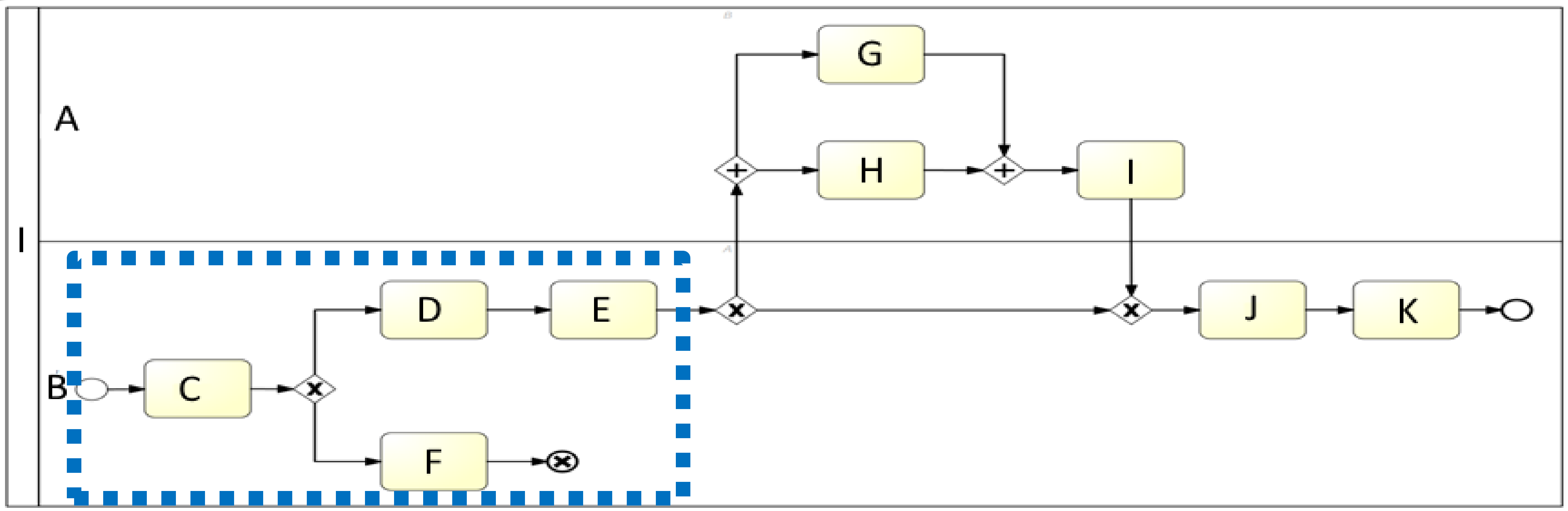
Exiting Attributes: Different Positions





Exiting Attributes: Partially Similar





Process Fragment

A Process Fragment is a piece of process model with loose completeness and consistency. The existence of process graph elements (start, end, activities, context etc.) is possible but not imperative in a process fragment. However, a process fragment must have ***at least one activity*** *and there must be a way to convert it to an executable process graph.*[2]

1. It is not necessarily related with a complete process model
2. A starting point is not defined
3. Existence of split, merge node or event is optional

[2]D. Schumm, F. Leymann, Z. Ma, T. Scheibler, and S. Strauch, "*Integrating Compliance into Business Processes: Process Fragments as Reusable Compliance Controls*" in MKWI'10, Göttingen, Germany, February 23-25, 2010, Ed., Conference Paper, pp. 2125–2137.

Checkpoints & Relevant Process Fragments (RPFs)

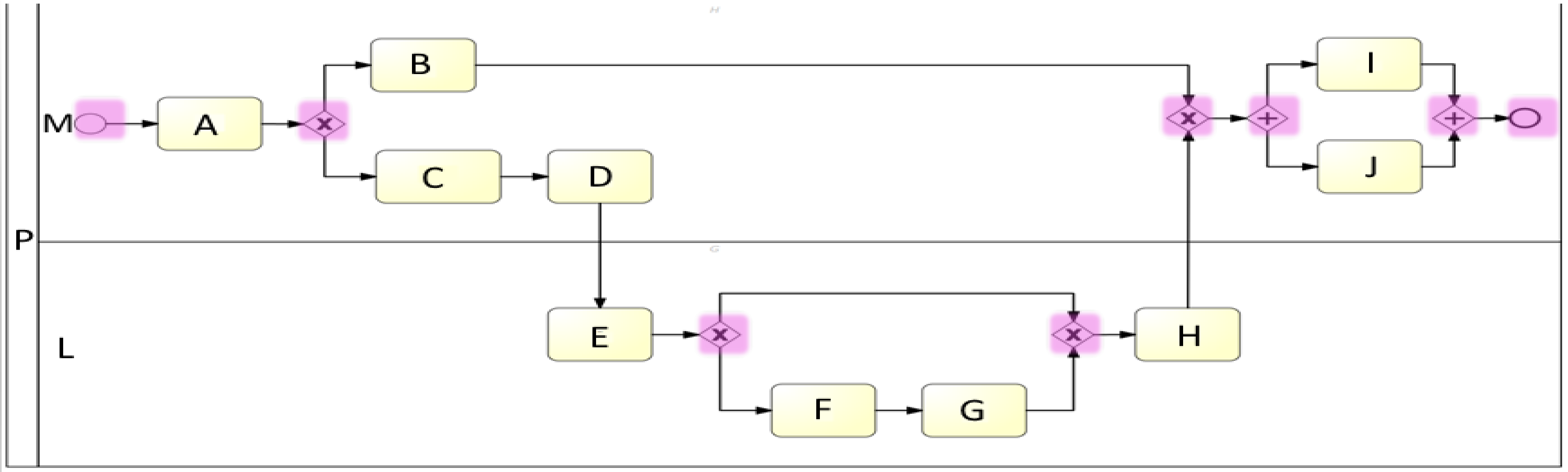
- **Checkpoint (the starting points)**
  - A pre-configured type of node that is used as **start point** of the “extended” process fragments
- **Relevant Process Fragment**
  - Exists in at least *K* business processes
  - Starts with a checkpoint
  - Has at least *N* nodes including the checkpoint
  - Contains at least one activity

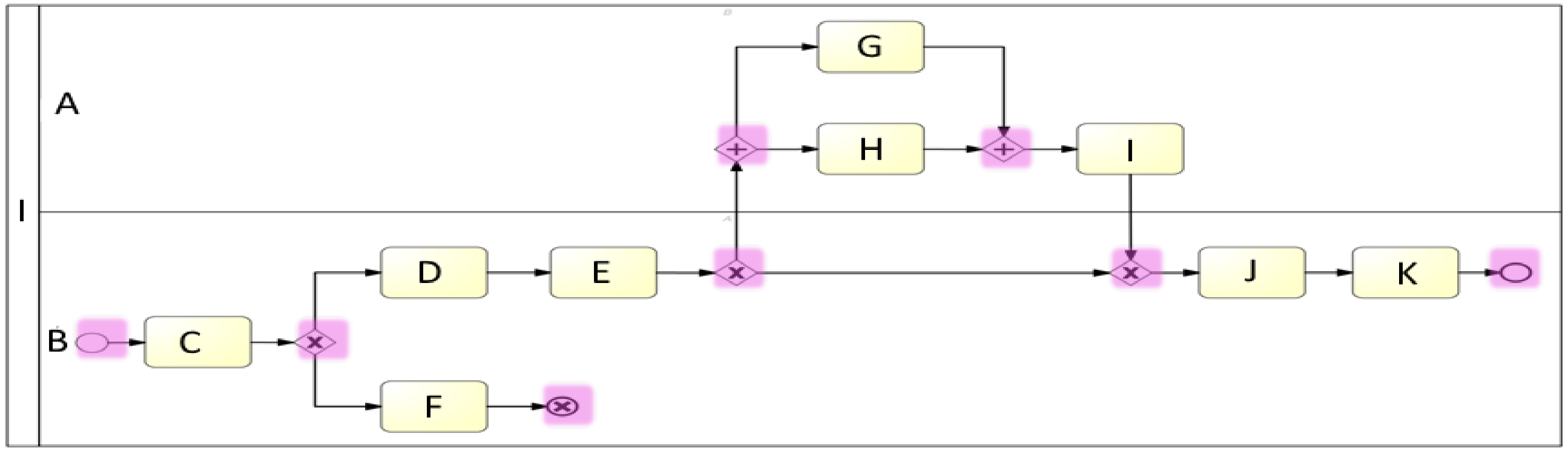
Checkpoints & Relevant Process Fragments

Checkpoints: Events, Gateways

K = 2 Process Models

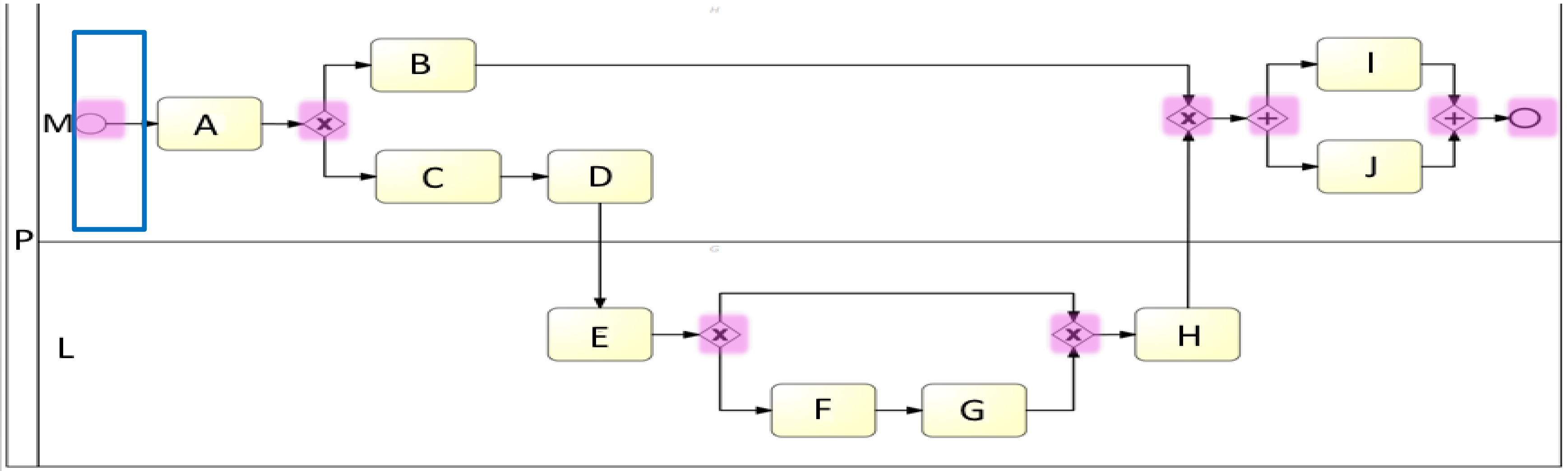
N = 3 Nodes

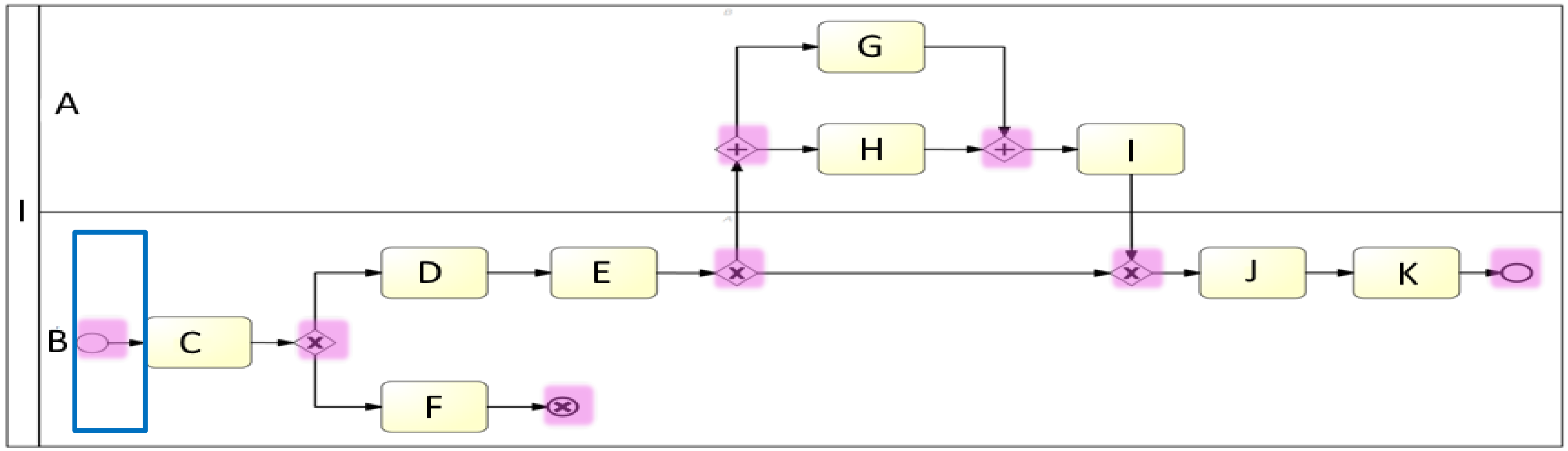




Checkpoints & Relevant Process Fragments

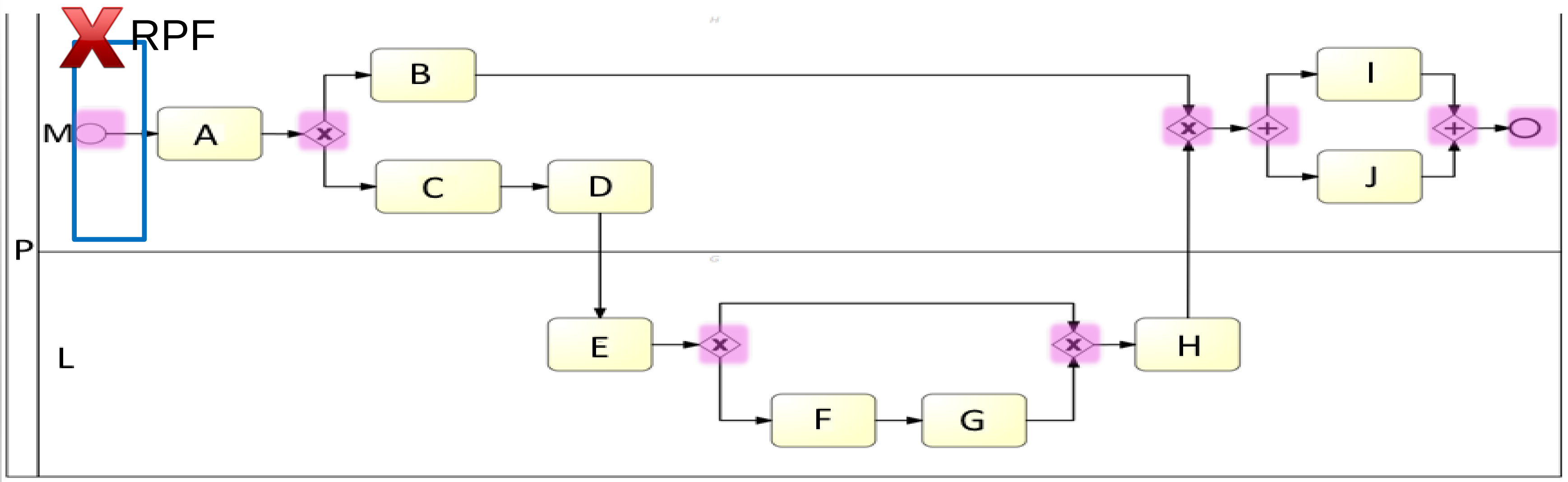
Checkpoints: Events, Gateways   K = 2 Process Models   N = 3 Nodes

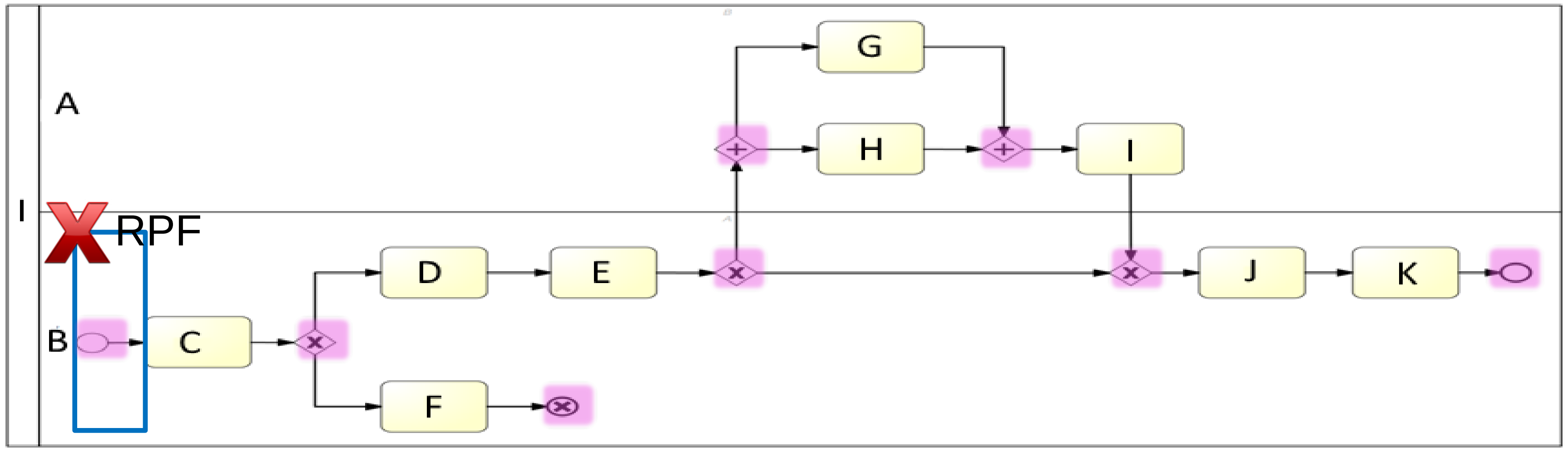




Checkpoints & Relevant Process Fragments

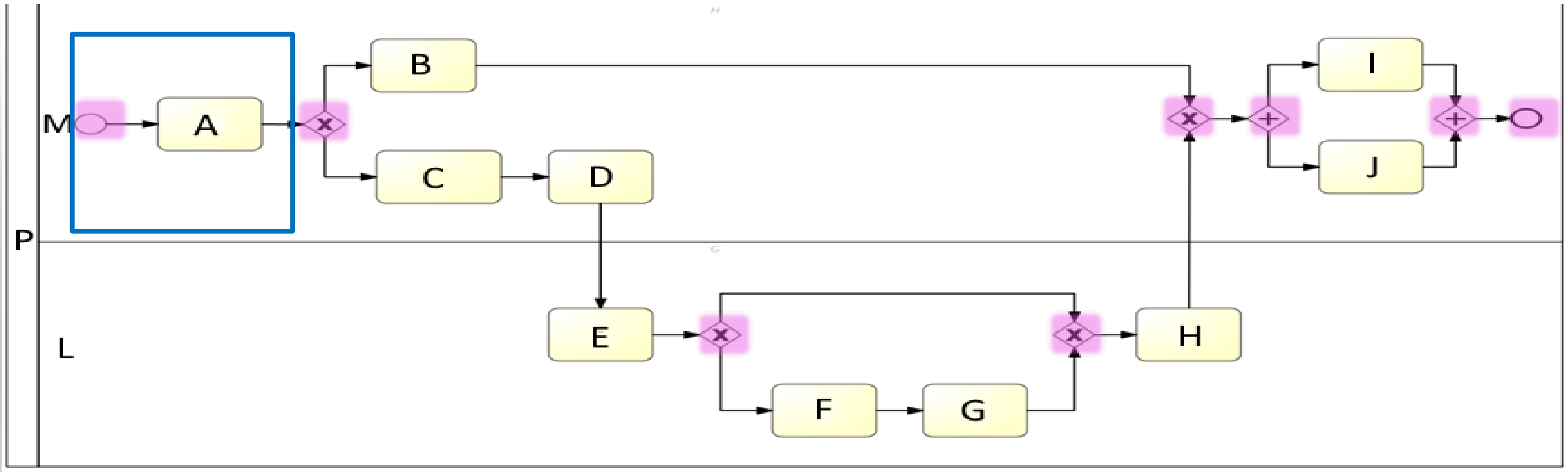
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

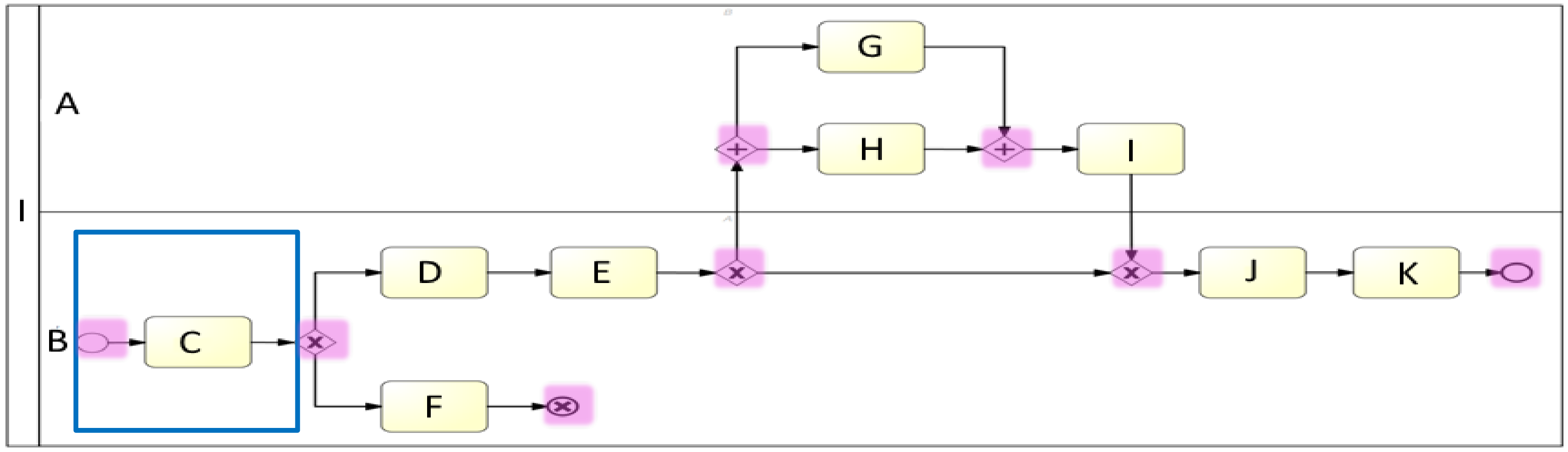




Checkpoints & Relevant Process Fragments

Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes



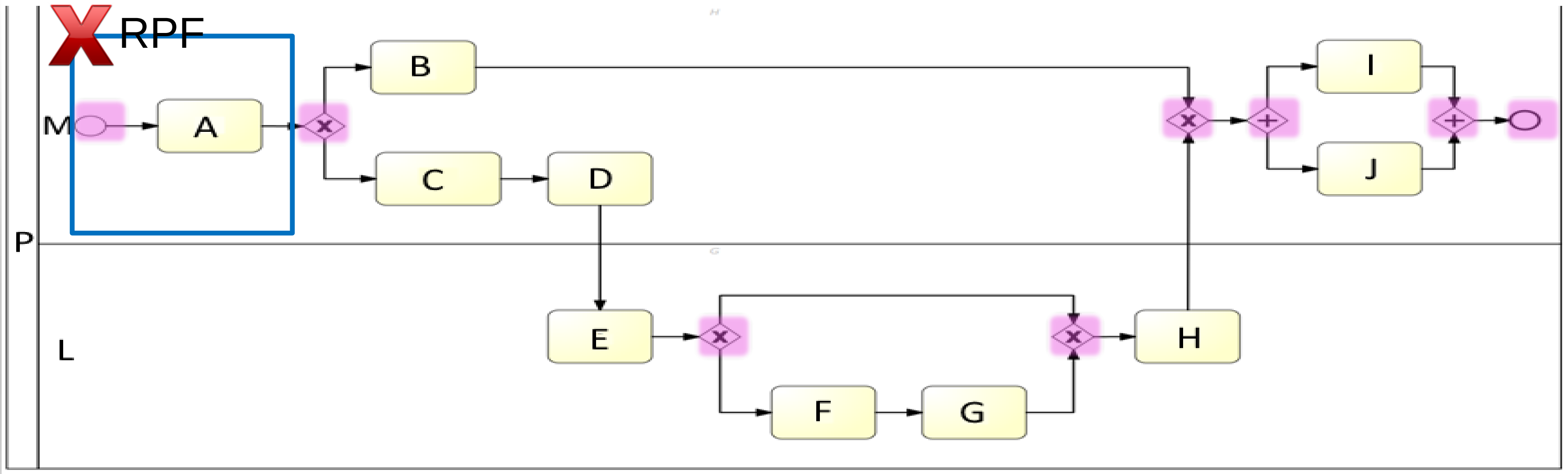


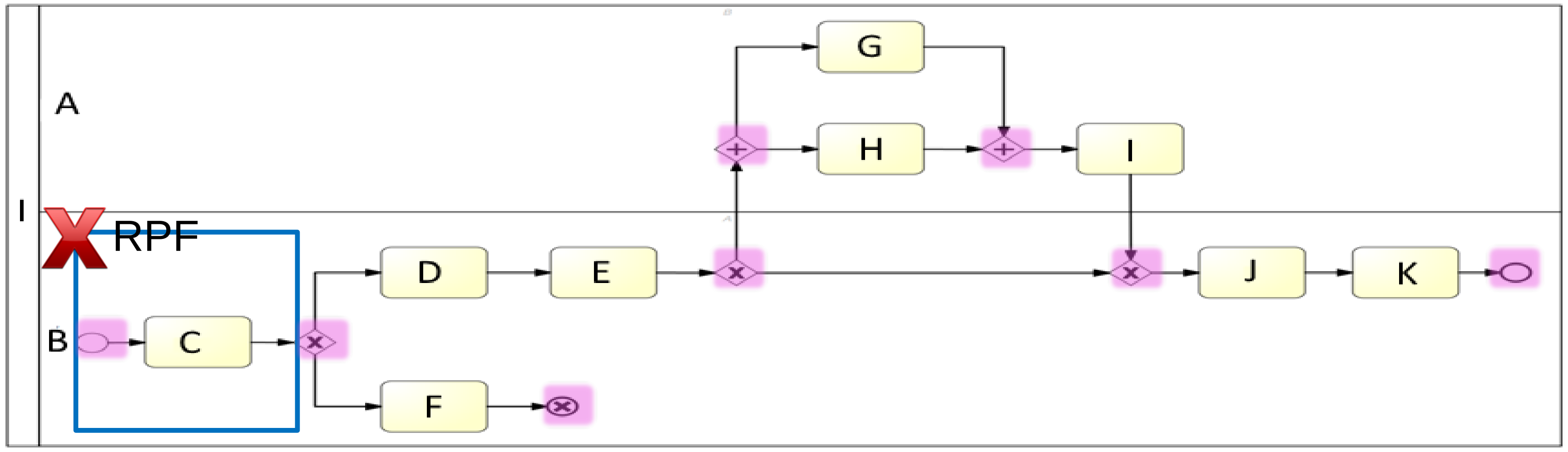
Checkpoints & Relevant Process Fragments

Checkpoints: Events, Gateways

K = 2 Process Models

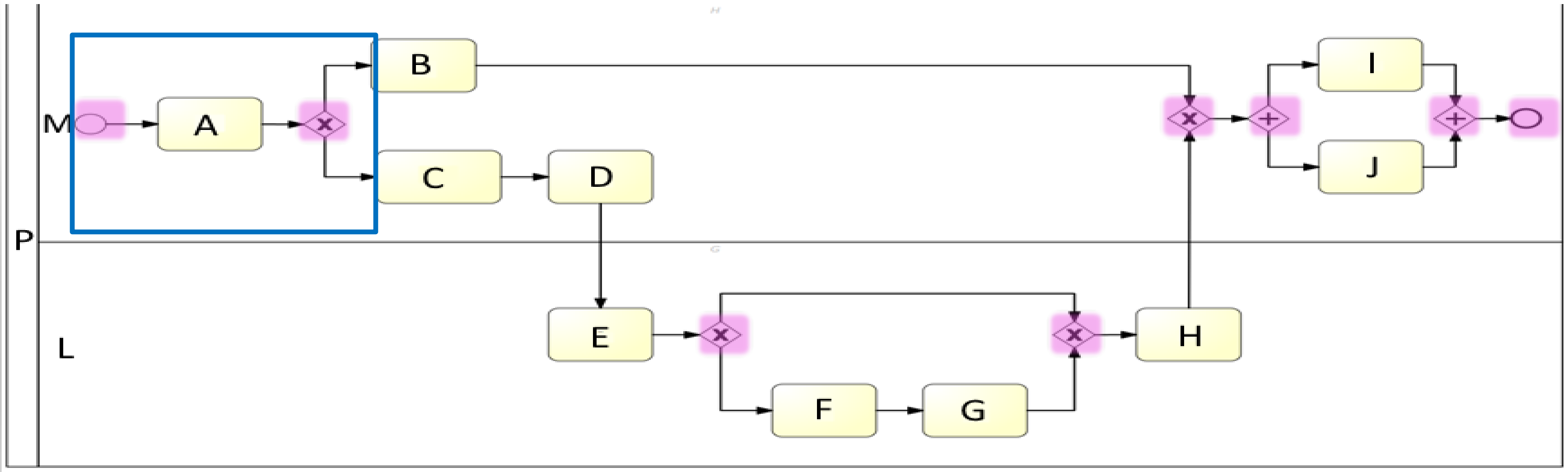
N = 3 Nodes

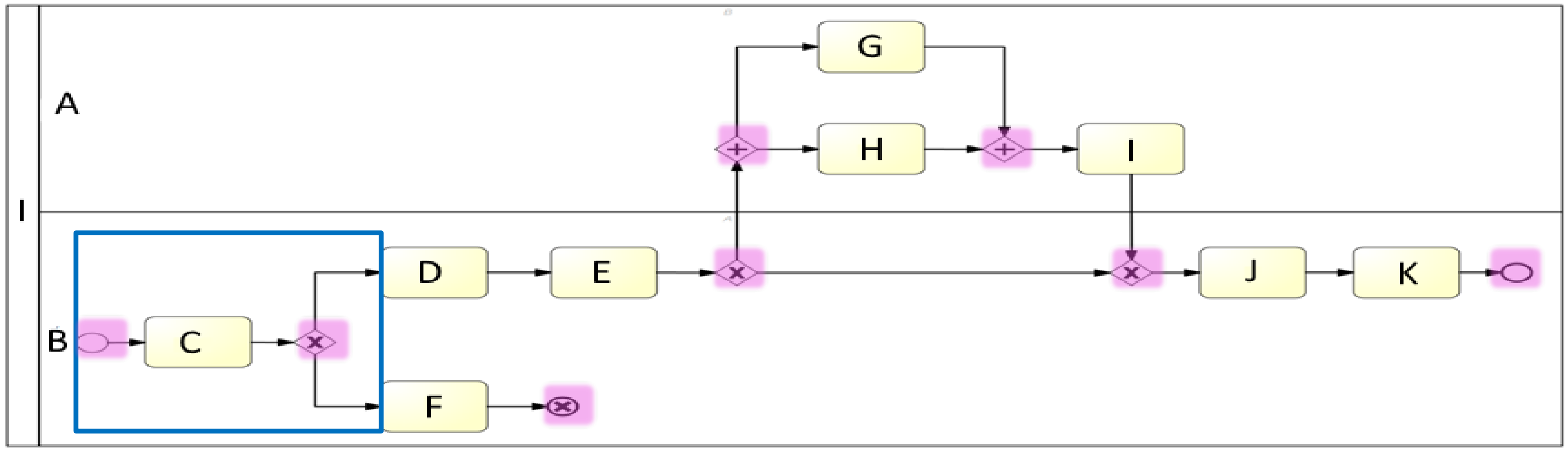




Checkpoints & Relevant Process Fragments

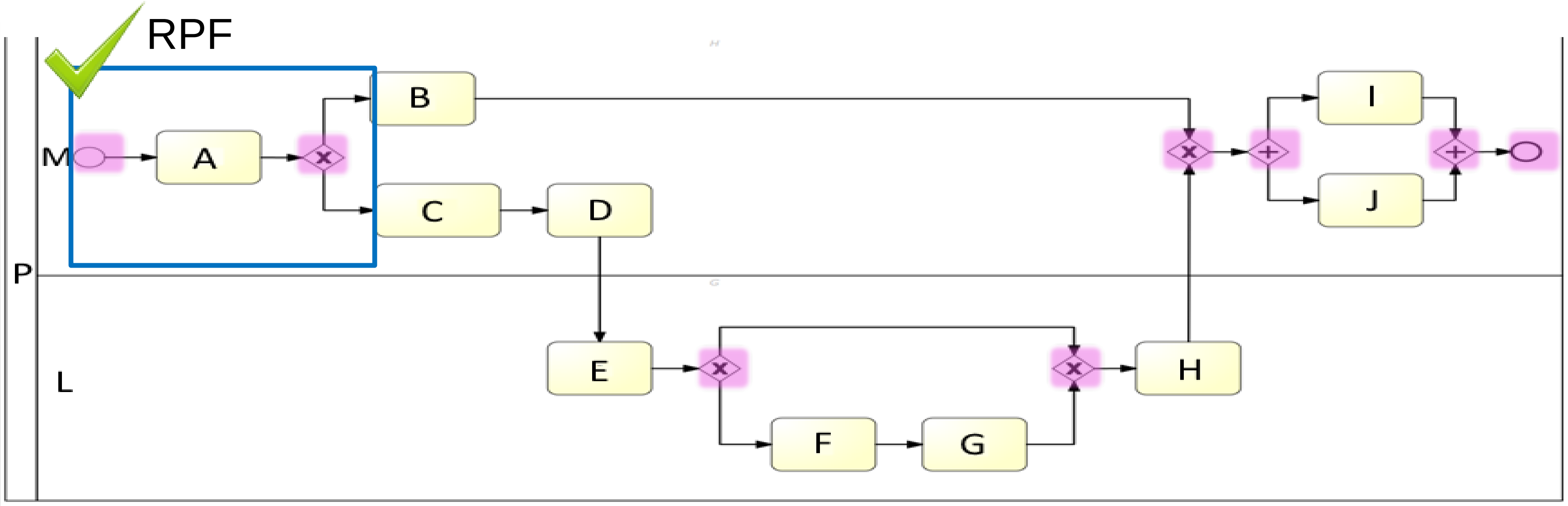
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

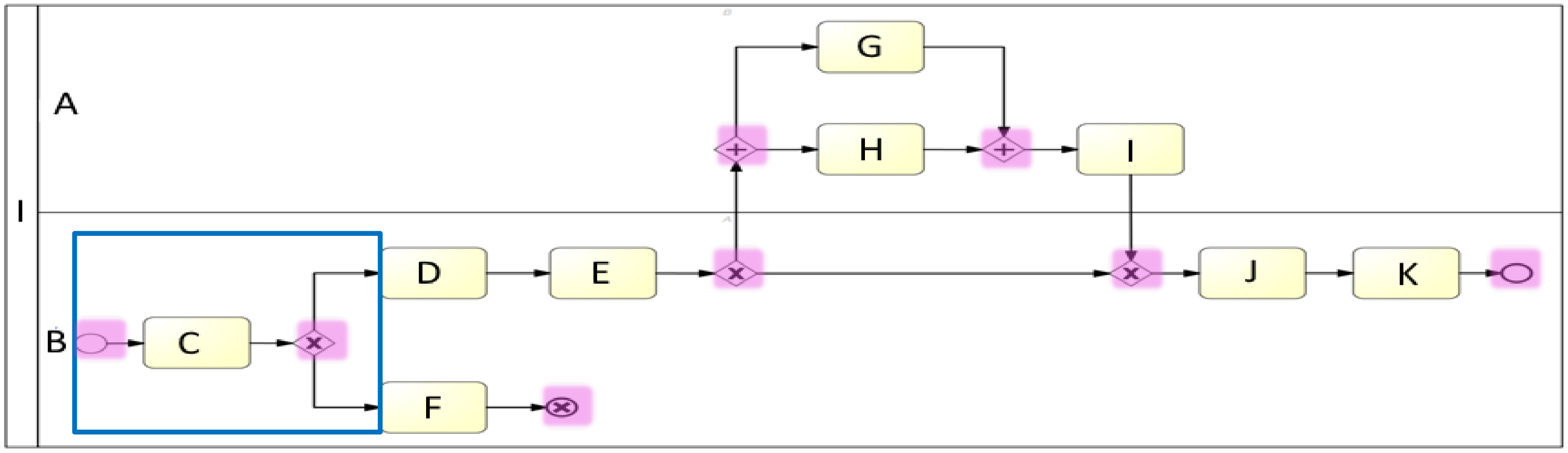




Checkpoints & Relevant Process Fragments

Checkpoints: Events, Gateways   K = 2 Process Models   N = 3 Nodes



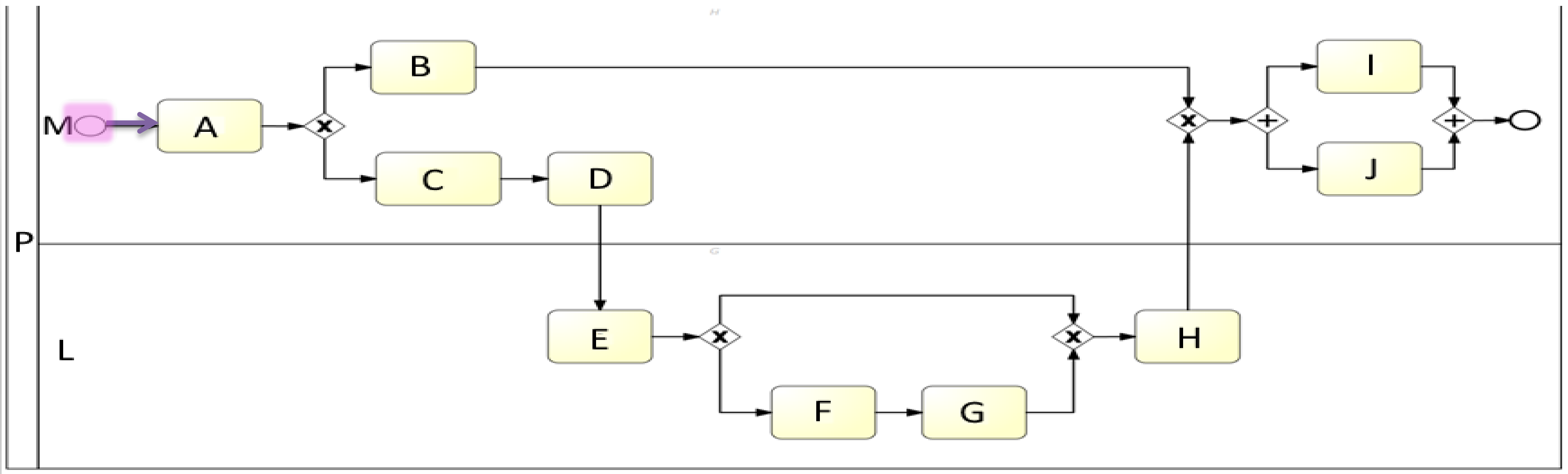


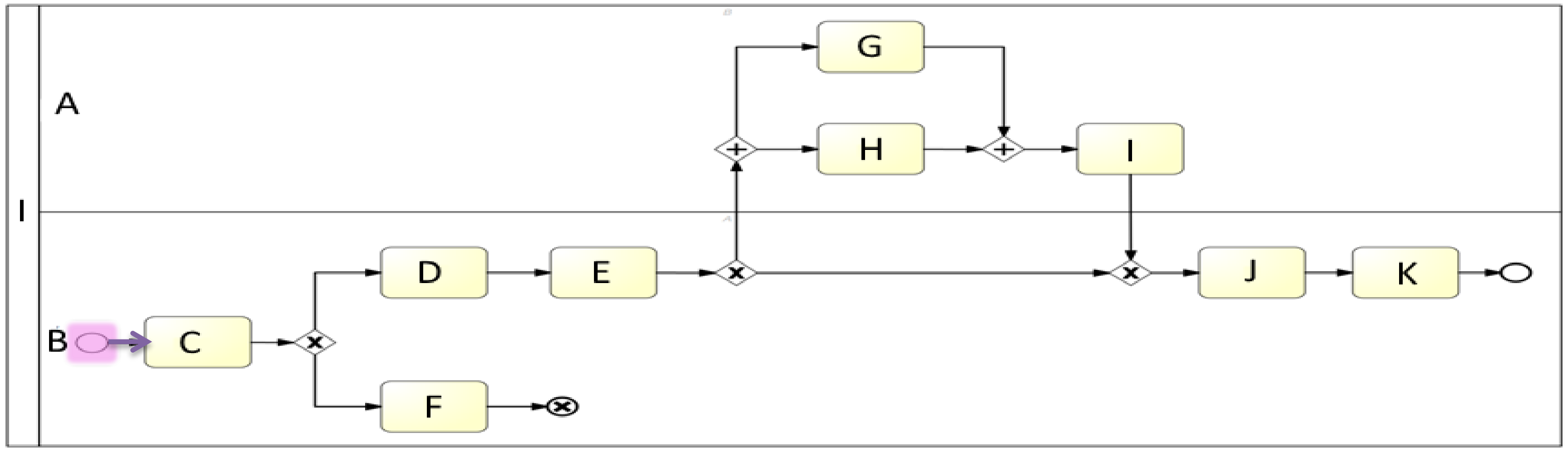
Algorithms

Research

Algorithm: Discovery of RPFs

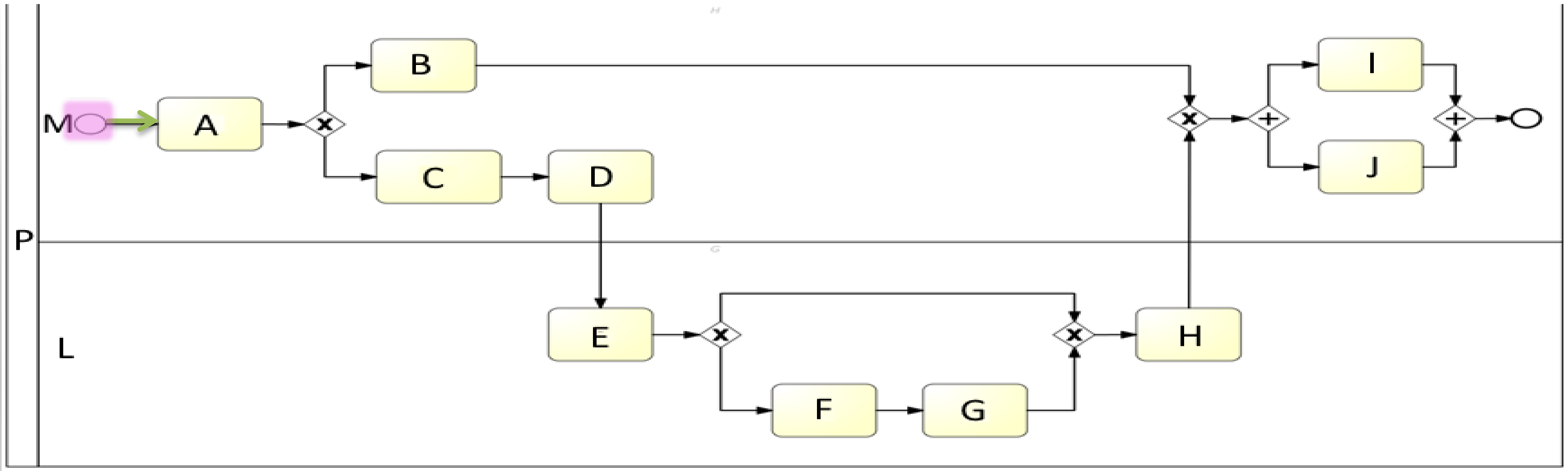
Checkpoints: Events, Gateways   K = 2 Process Models   N = 3 Nodes

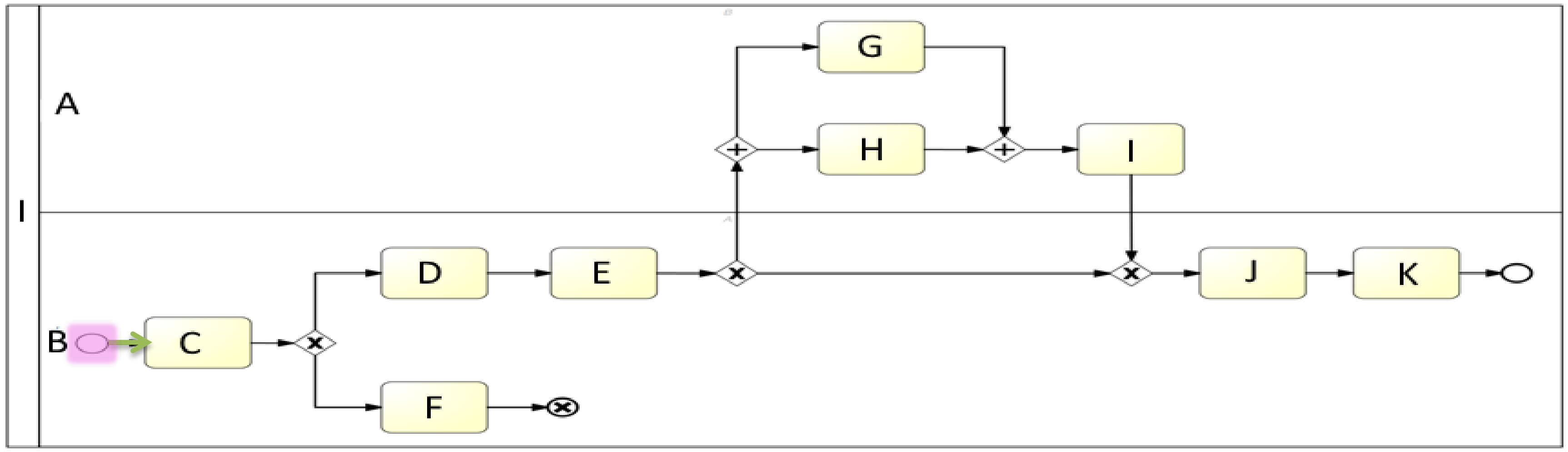




Algorithm: Discovery of RPFs

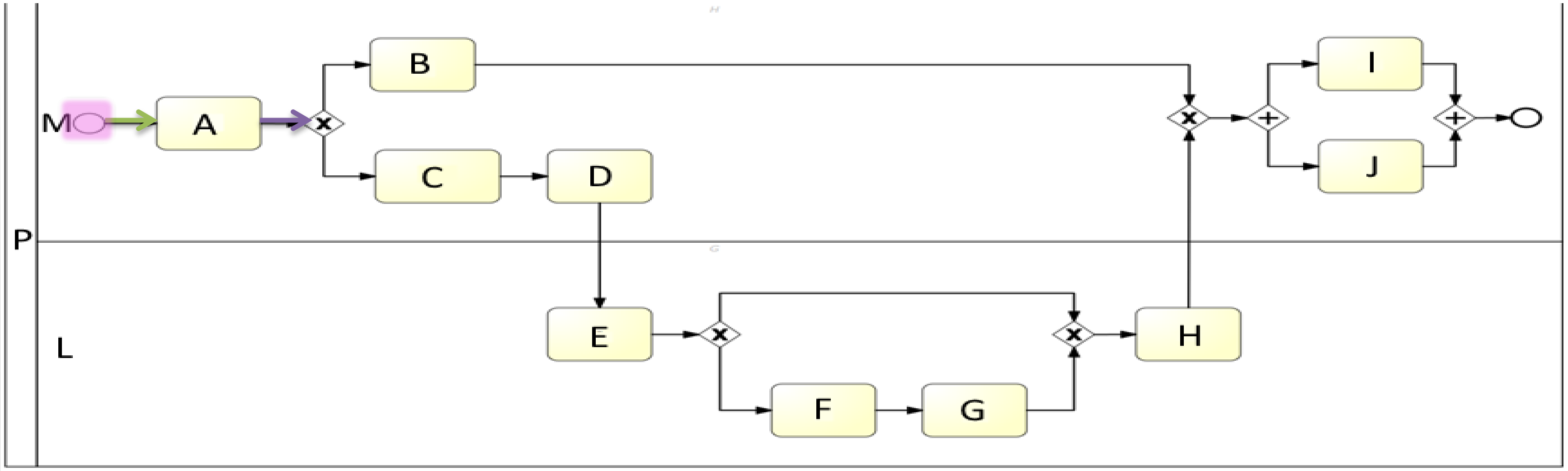
Checkpoints: Events, Gateways   K = 2 Process Models   N = 3 Nodes

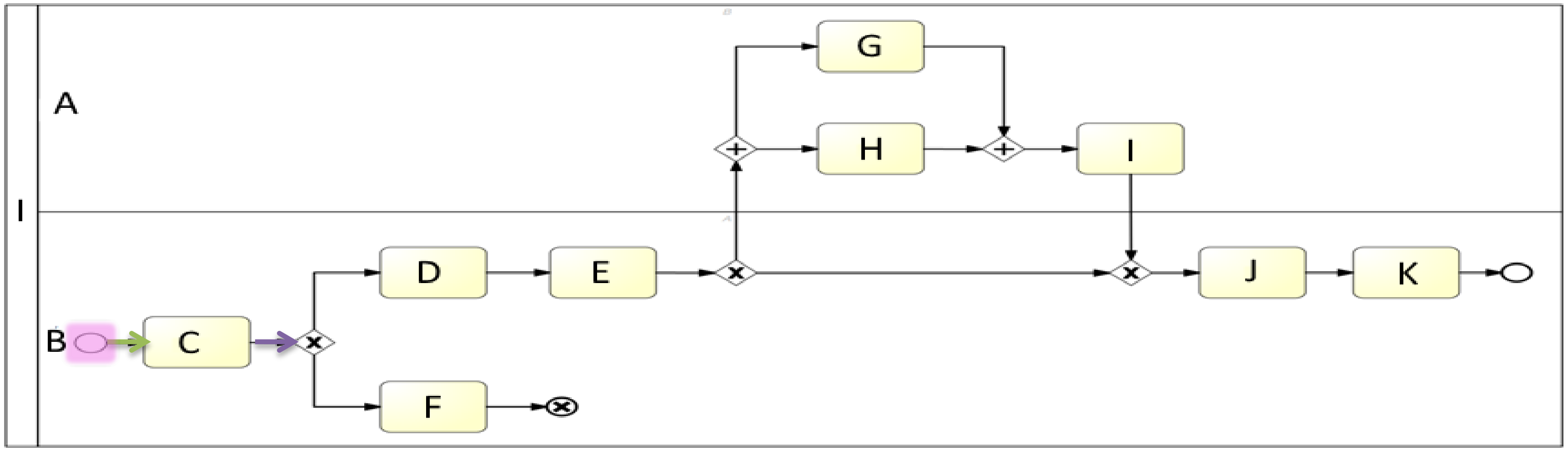




Algorithm: Discovery of RPFs

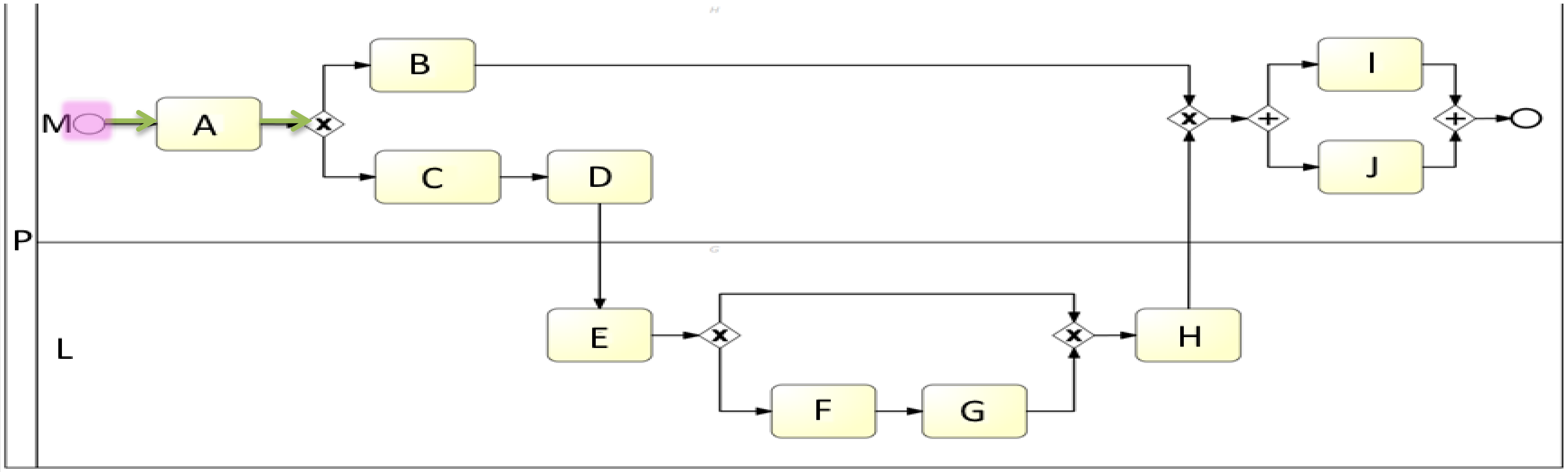
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

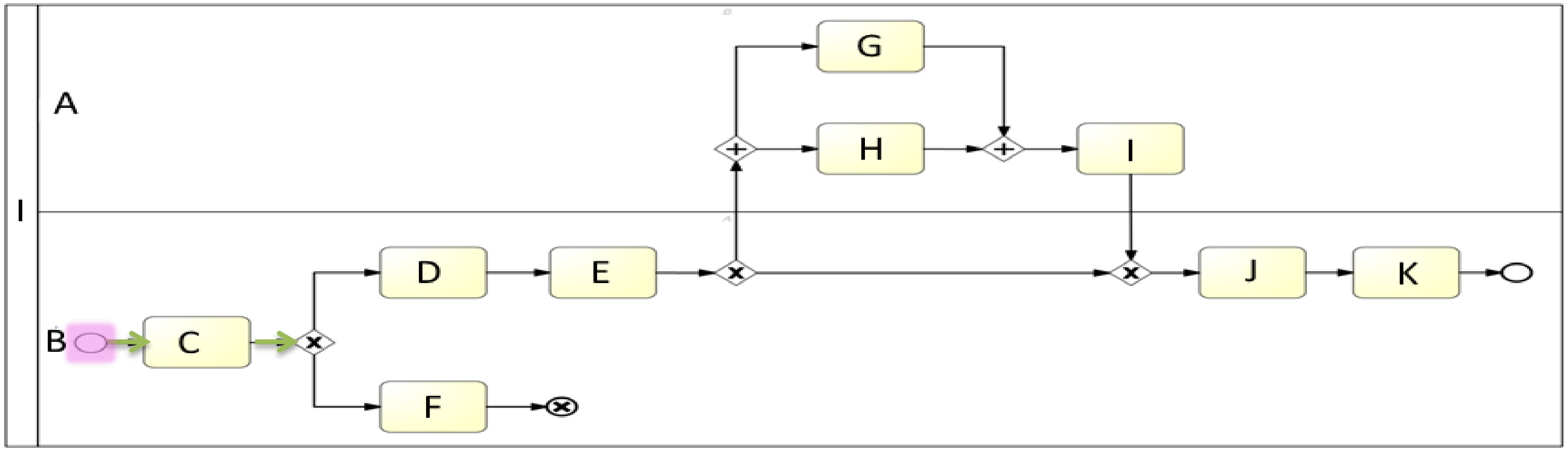




Algorithm: Discovery of RPFs

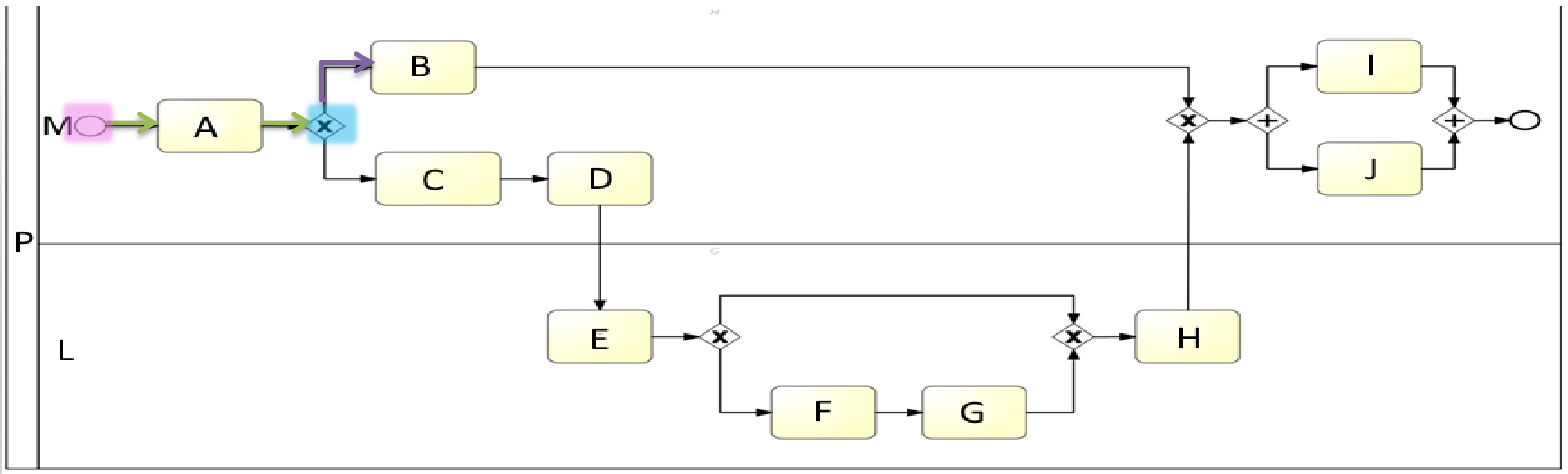
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

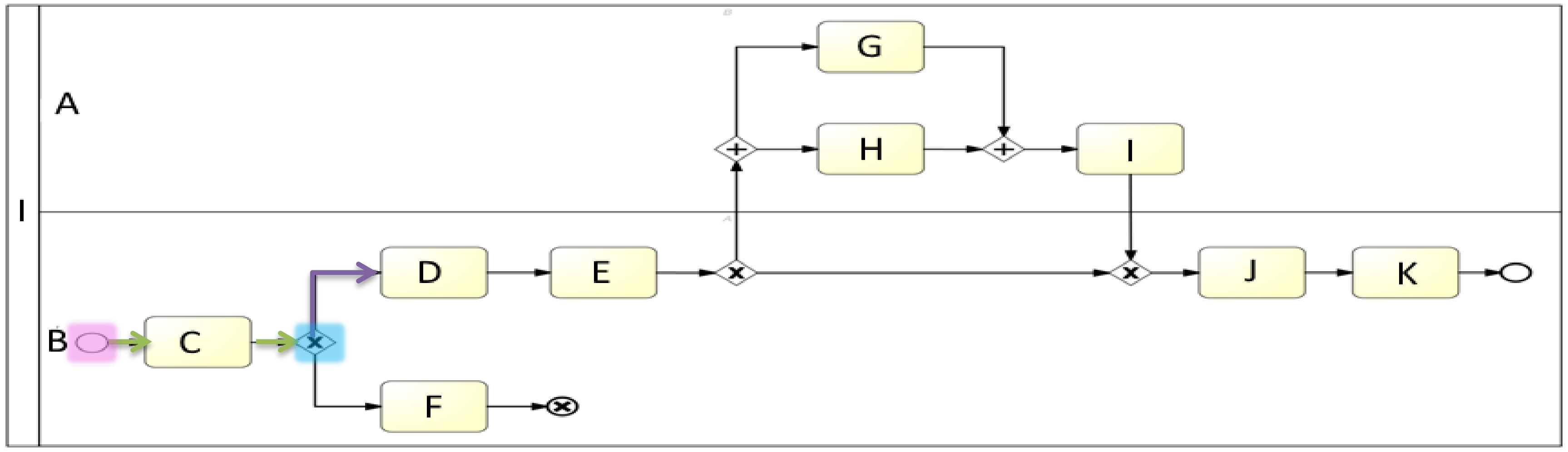




Algorithm: Discovery of RPFs

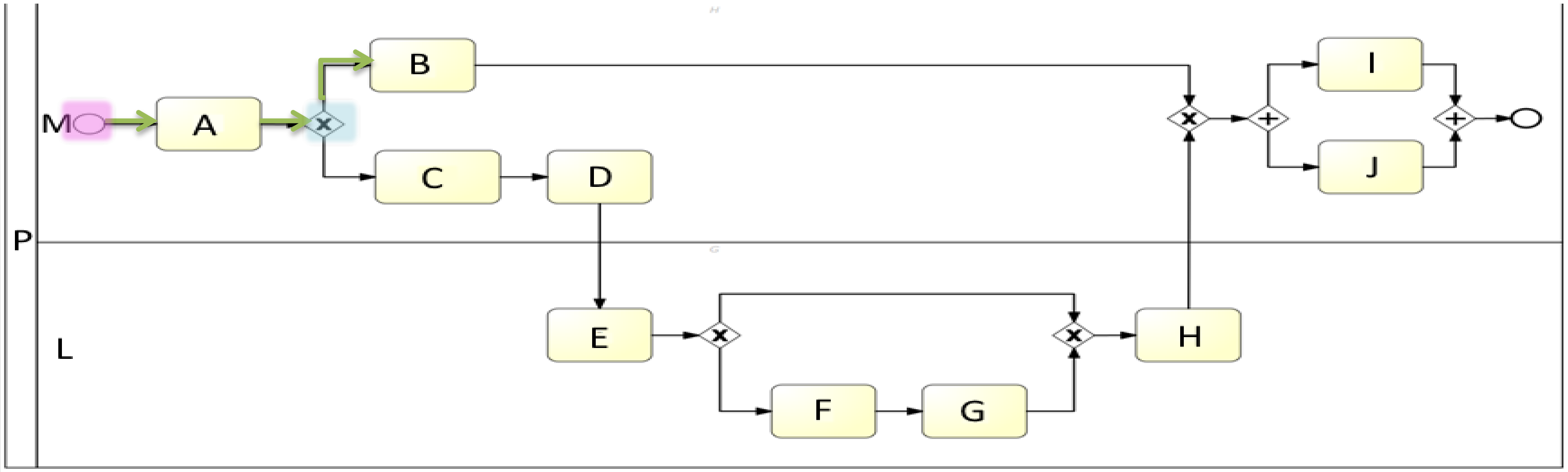
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

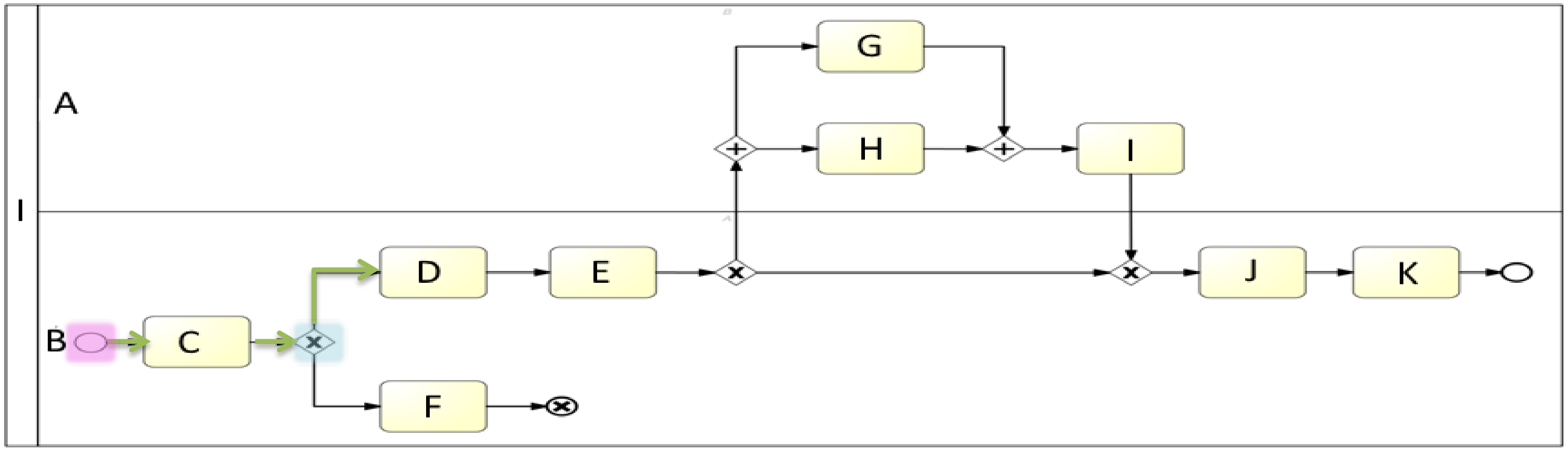




Algorithm: Discovery of RPFs

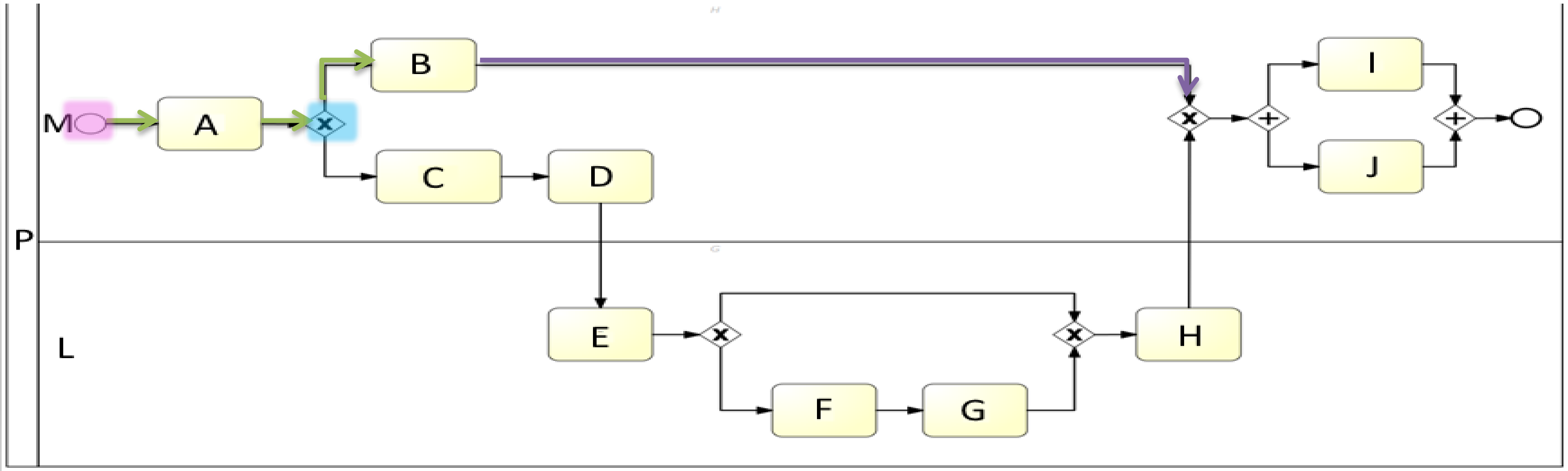
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

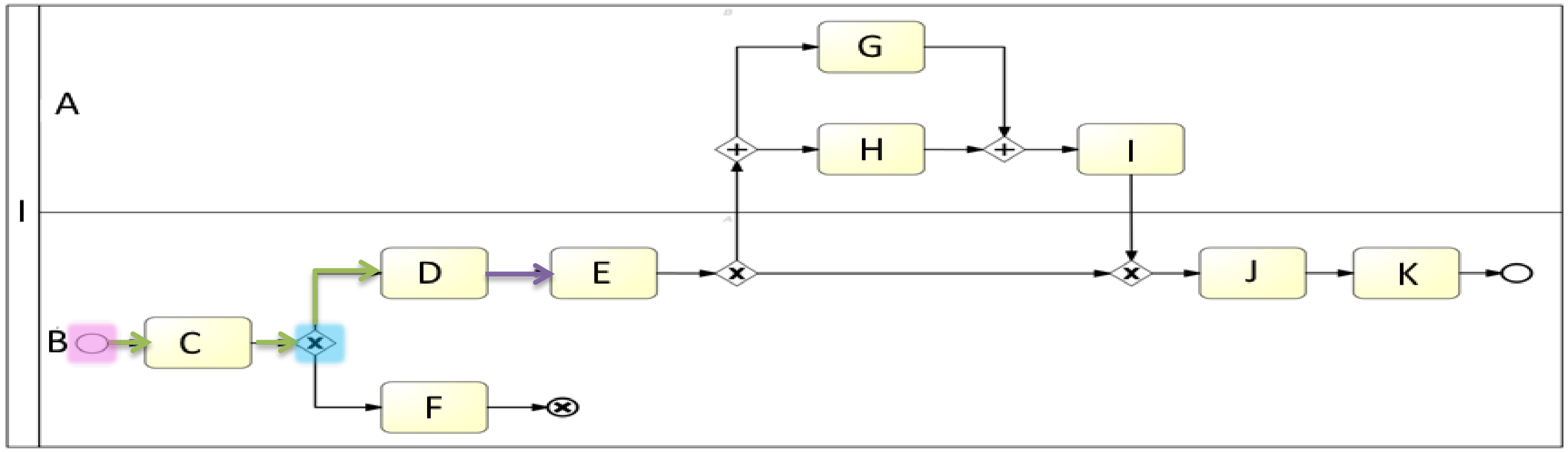




Algorithm: Discovery of RPFs

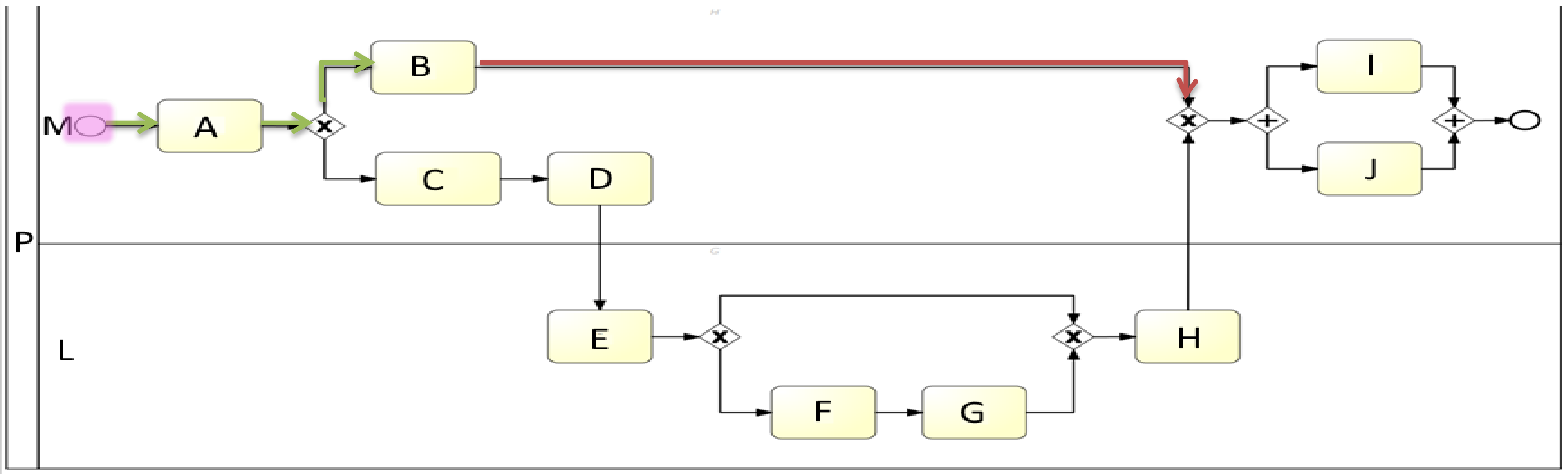
Checkpoints: Events, Gateways　　K = 2 Process Models　　N = 3 Nodes

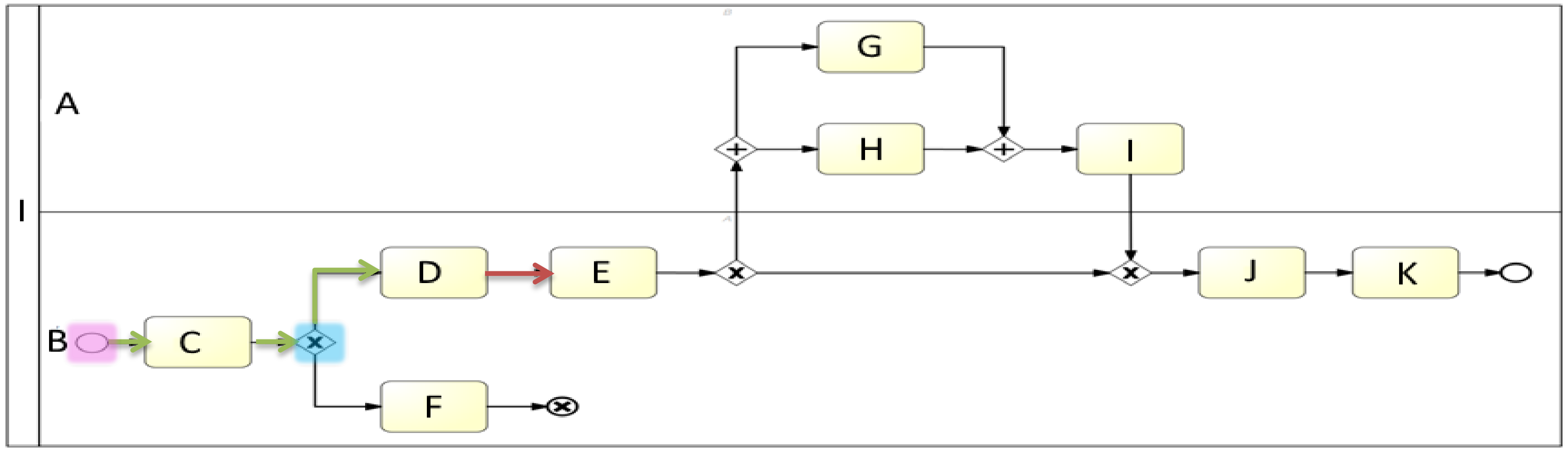




Algorithm: Discovery of RPFs

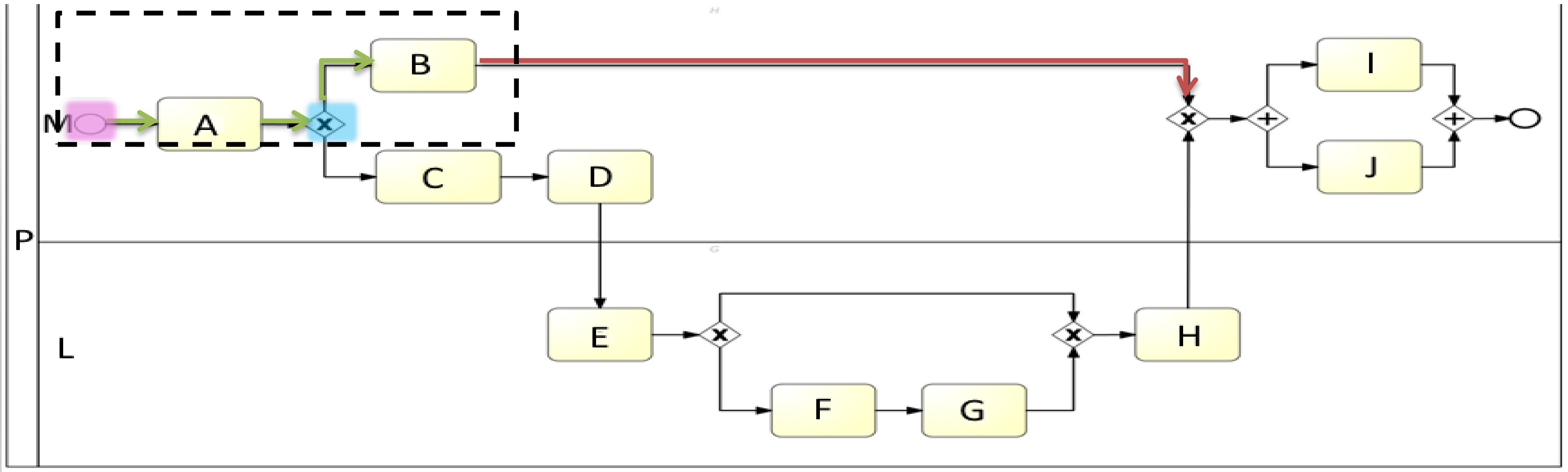
Checkpoints: Events, Gateways　　K = 2 Process Models　　N = 3 Nodes

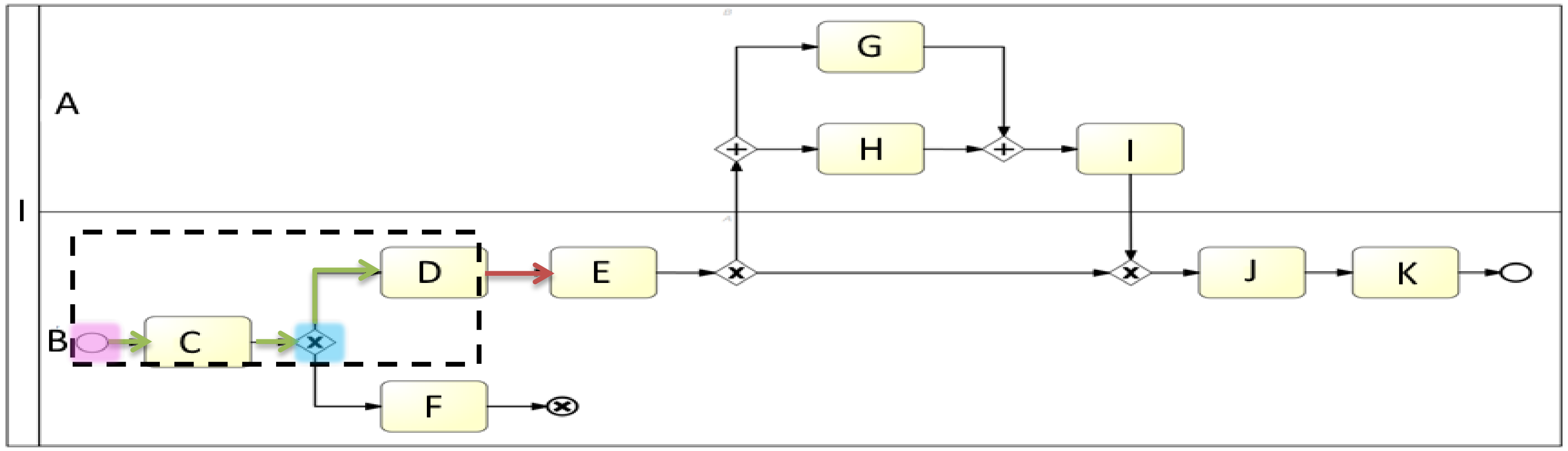




Algorithm: Discovery of RPFs

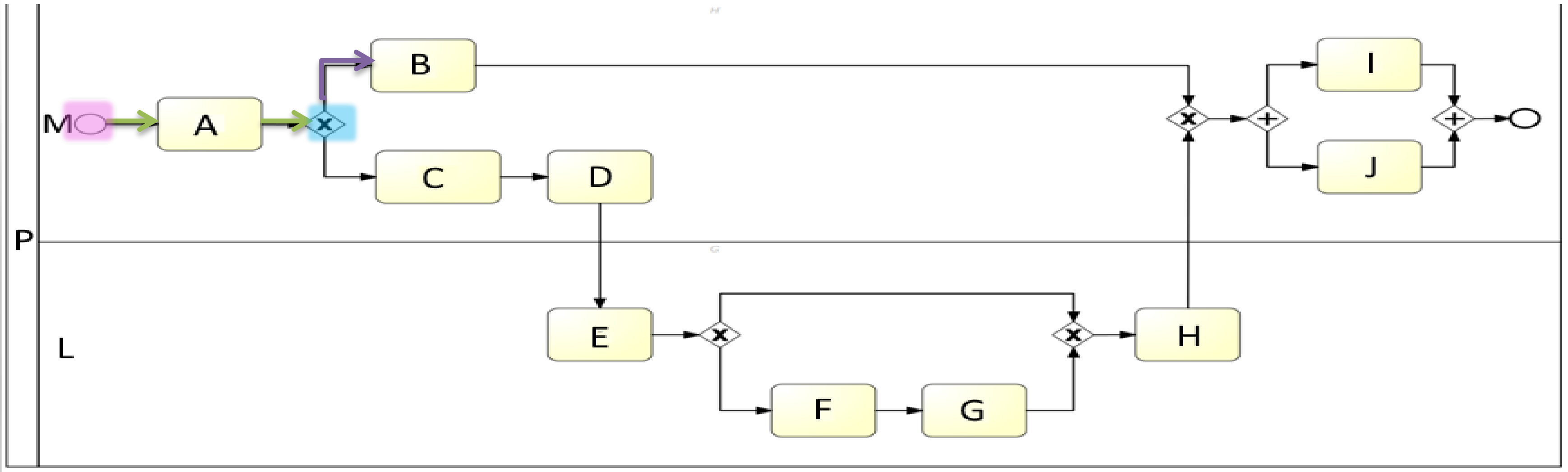
Checkpoints: Events, Gateways   K = 2 Process Models   N = 3 Nodes

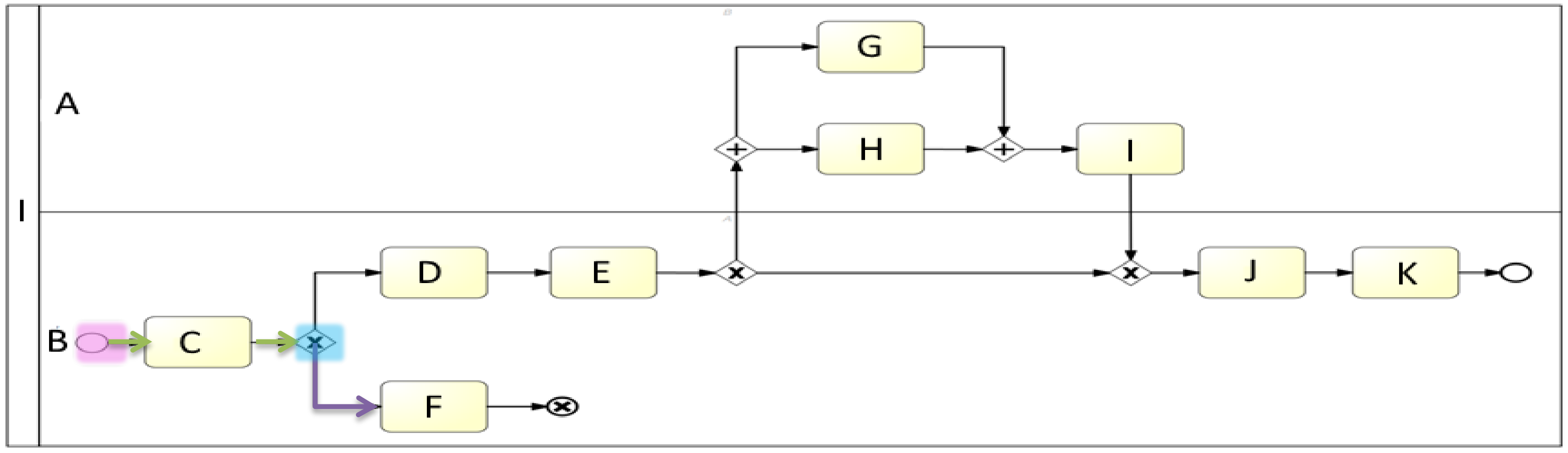




Algorithm: Discovery of RPFs

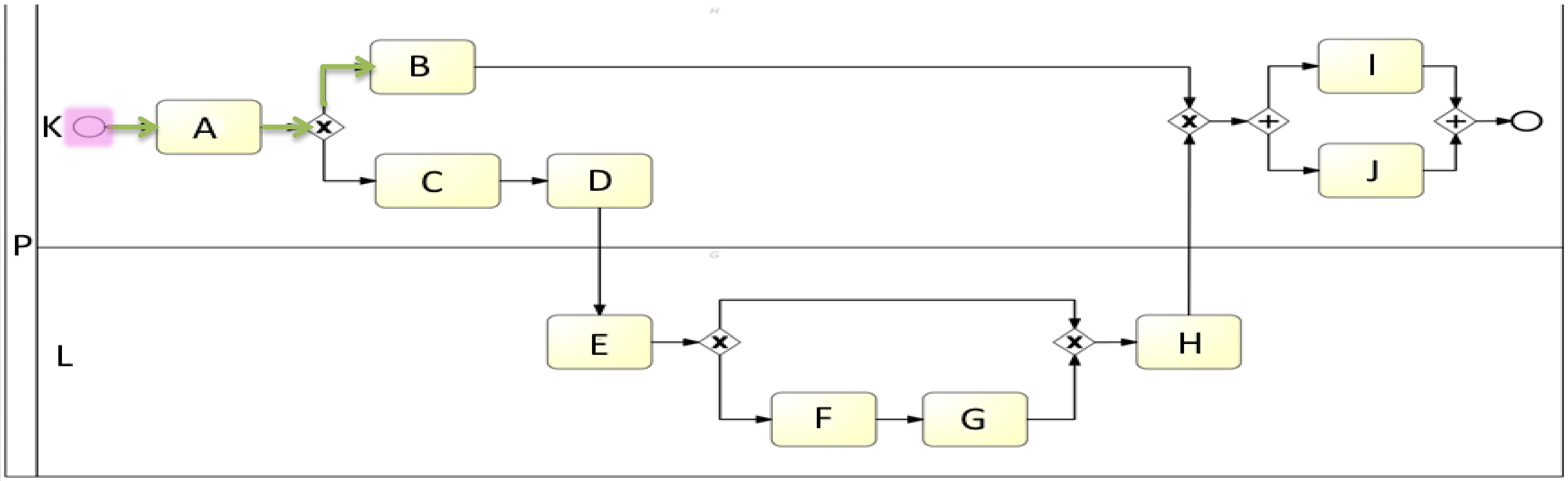
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

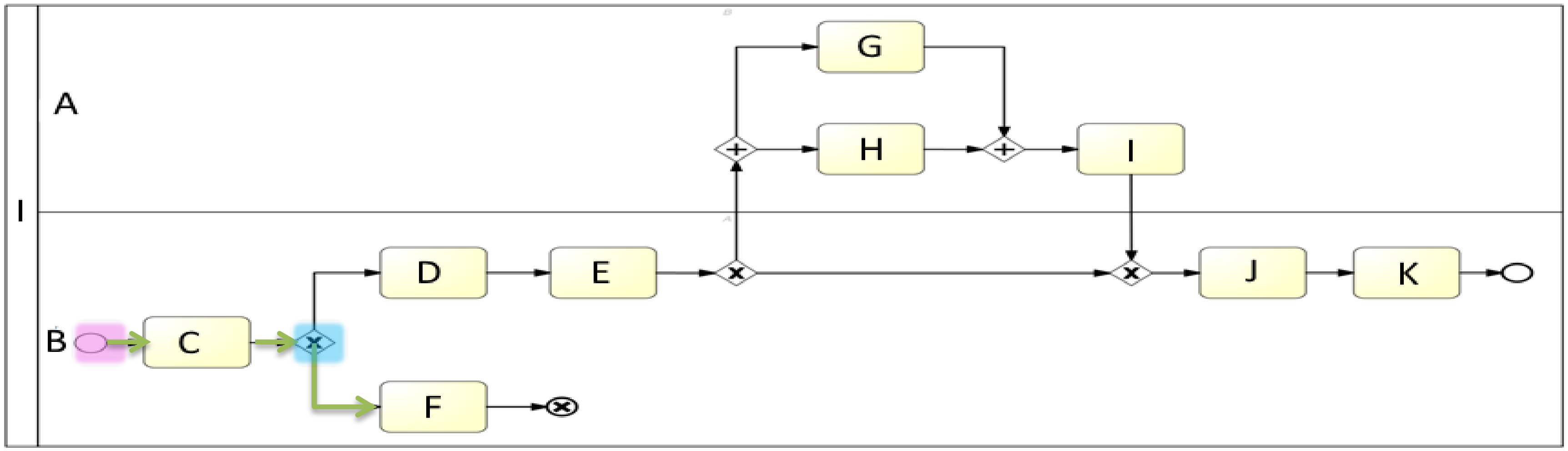




Algorithm: Discovery of RPFs

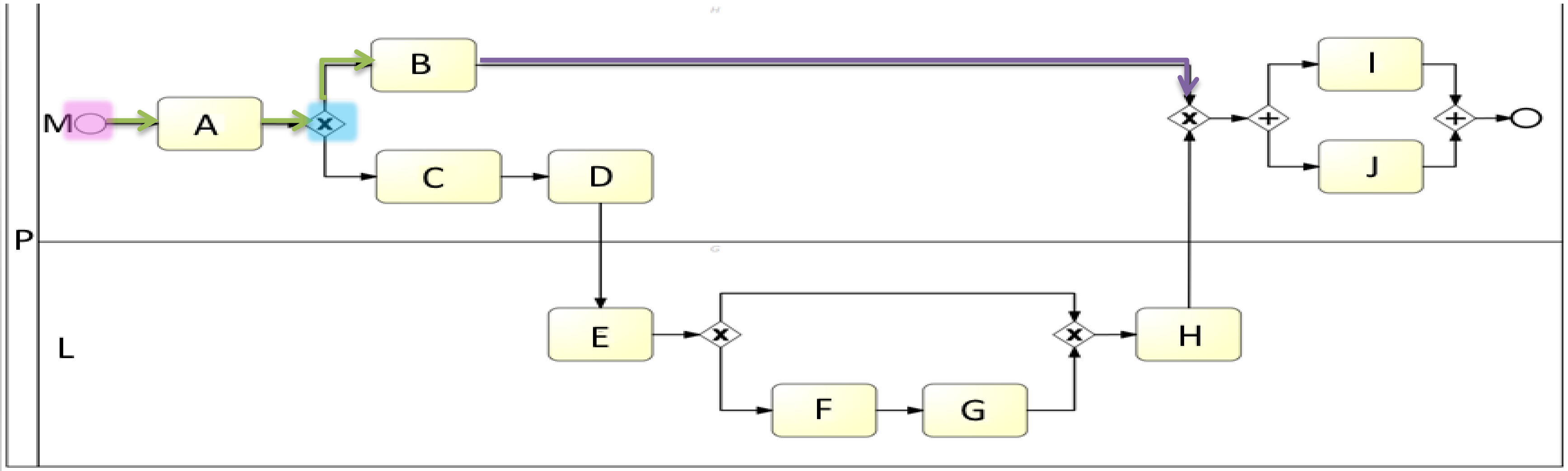
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

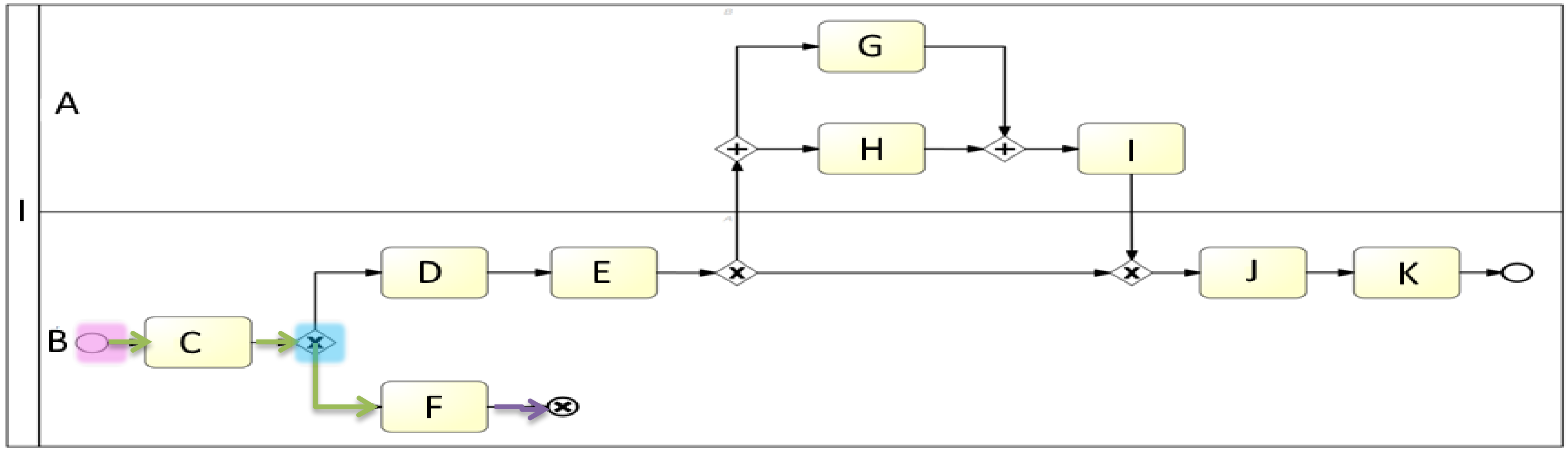




Algorithm: Discovery of RPFs

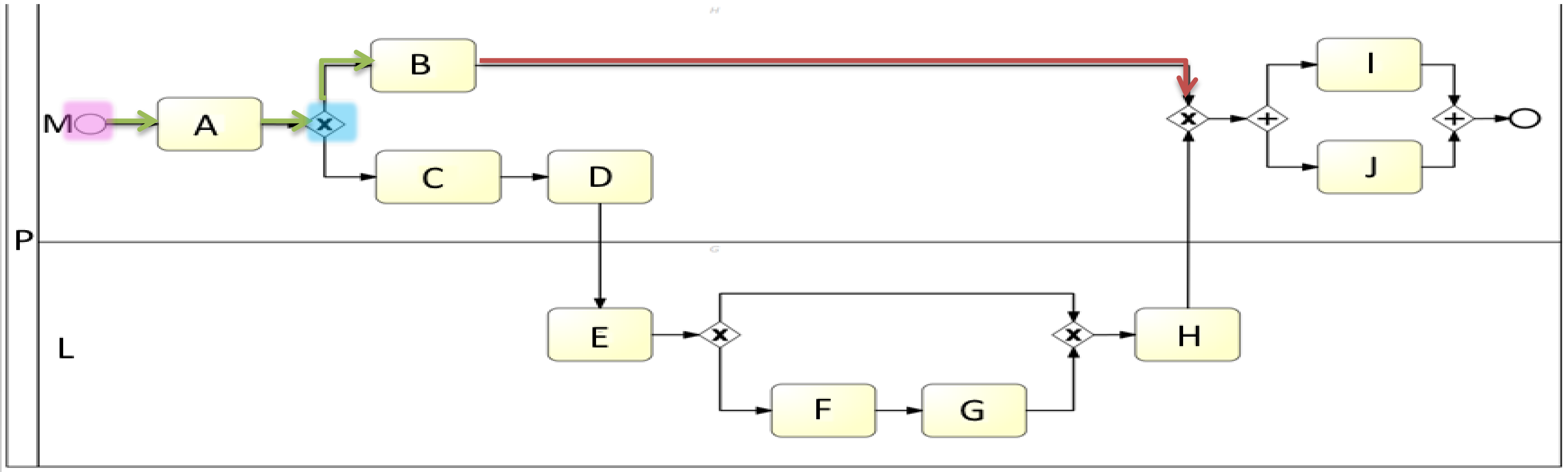
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

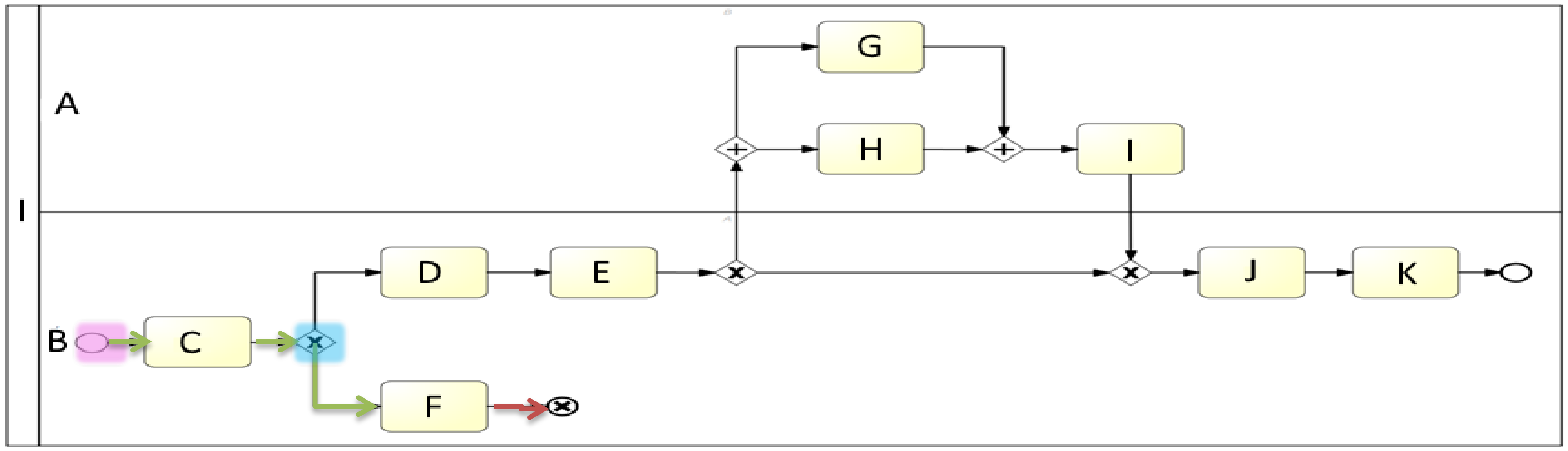




Algorithm: Discovery of RPFs

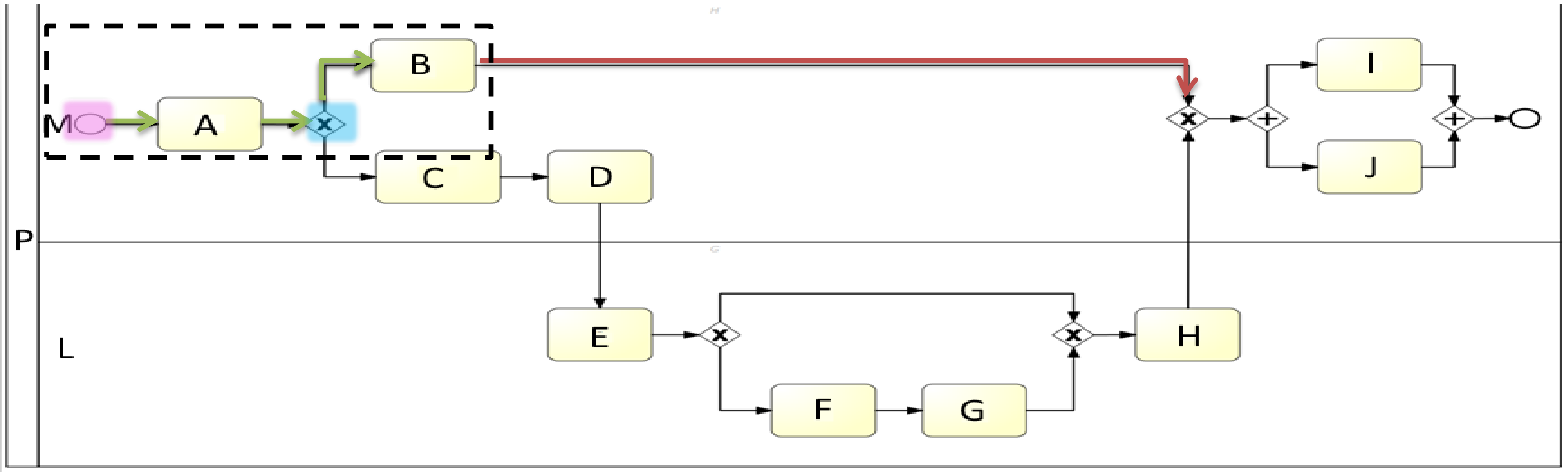
Checkpoints: Events, Gateways   K = 2 Process Models   N = 3 Nodes

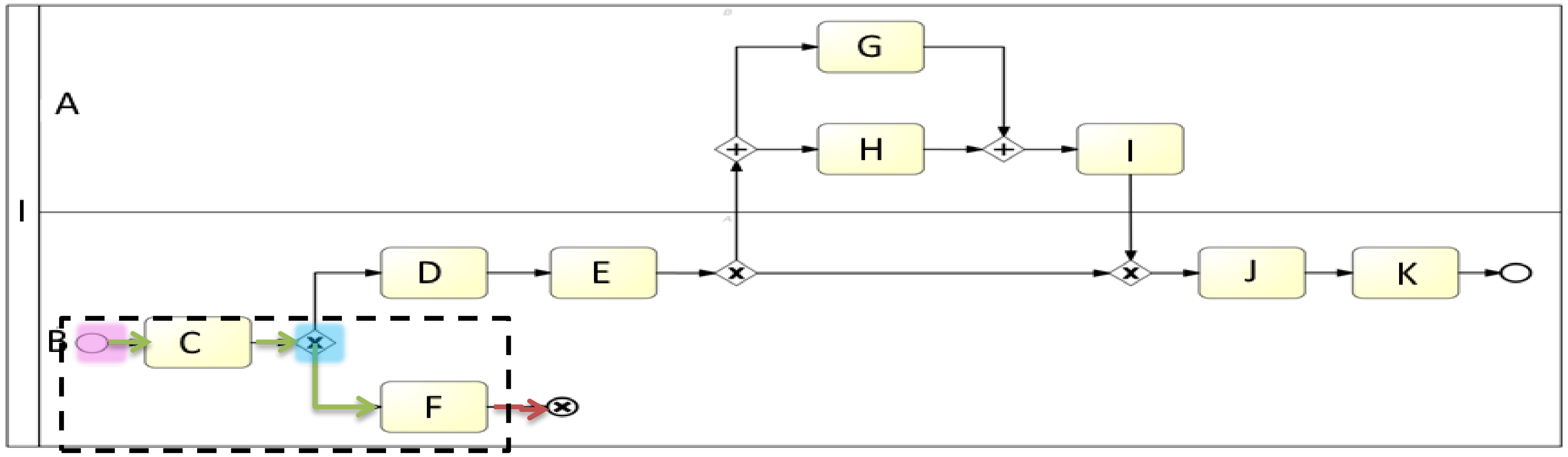




Algorithm: Discovery of RPFs

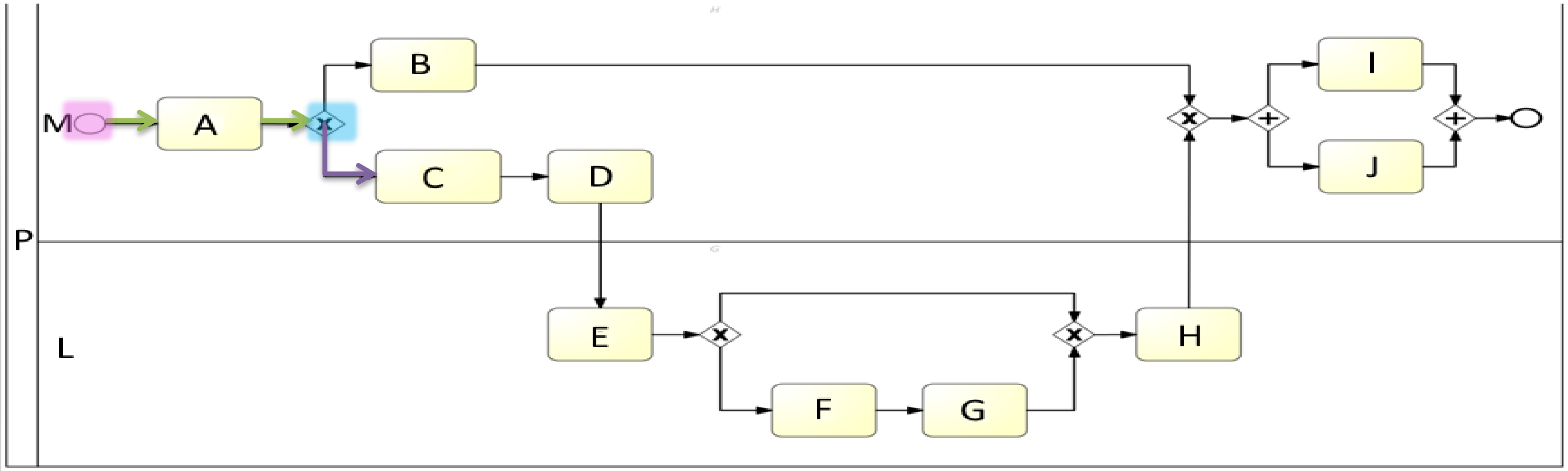
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

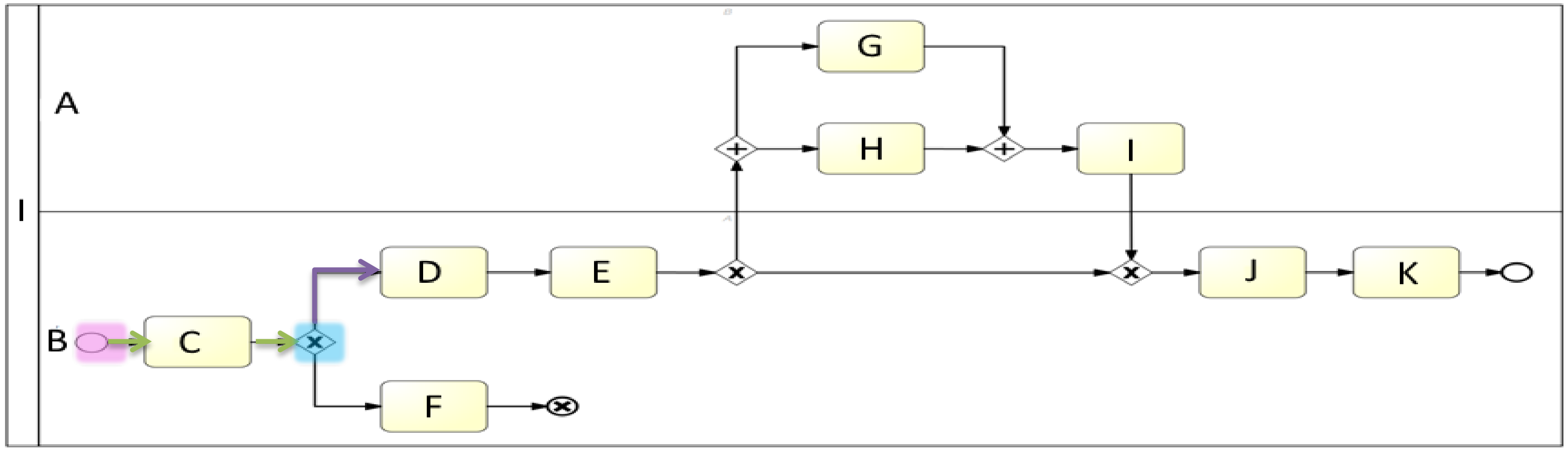




Algorithm: Discovery of RPFs

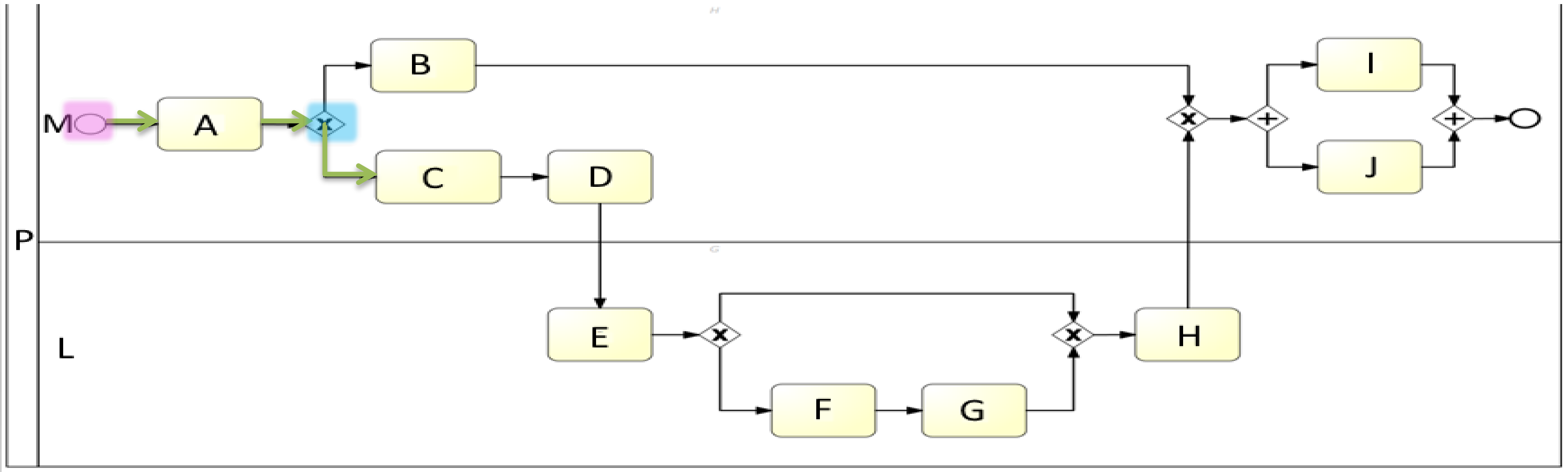
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

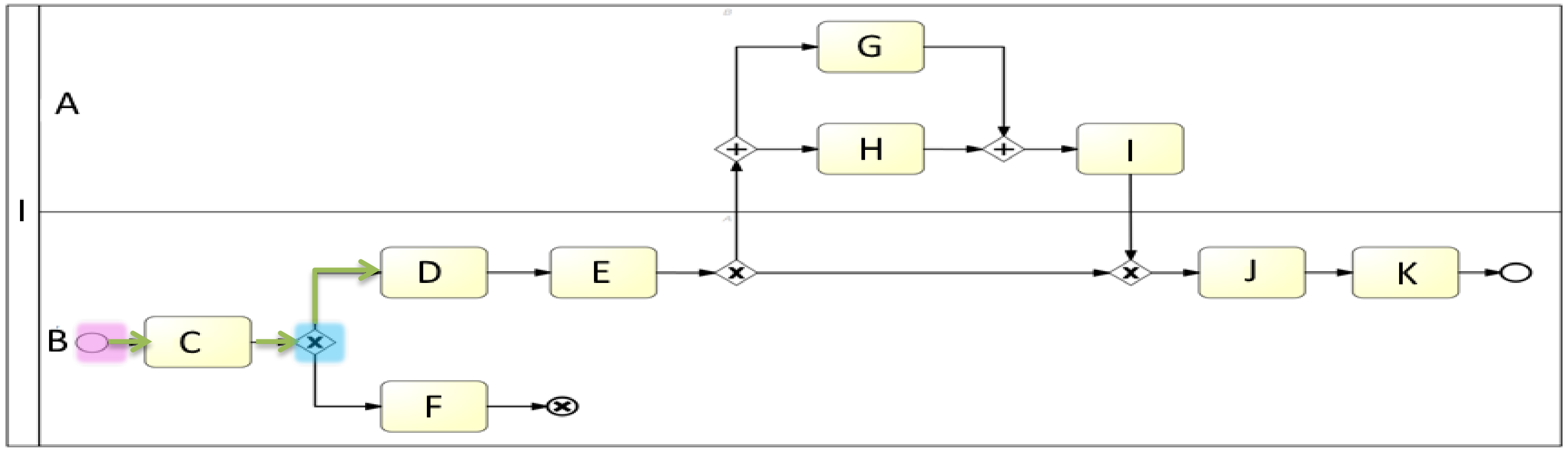




Algorithm: Discovery of RPFs

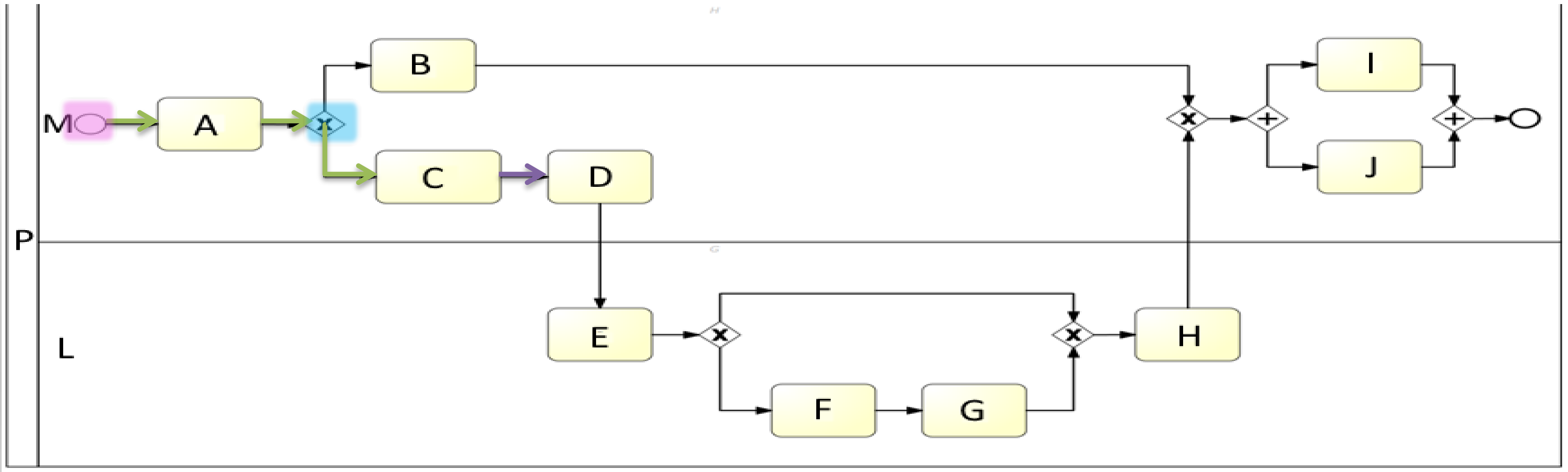
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

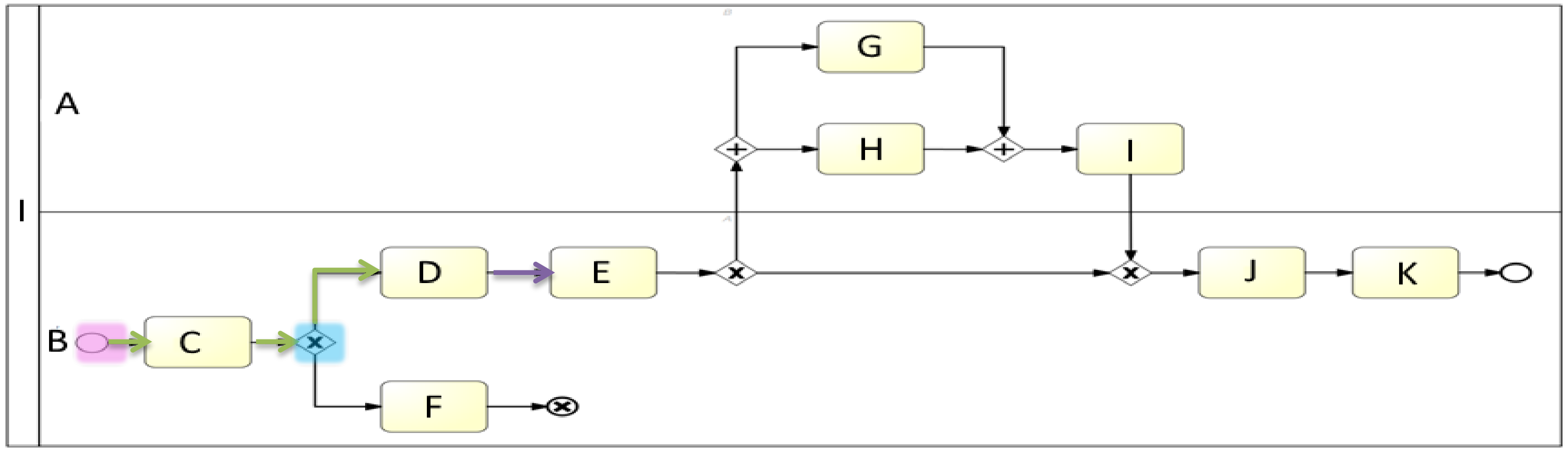




Algorithm: Discovery of RPFs

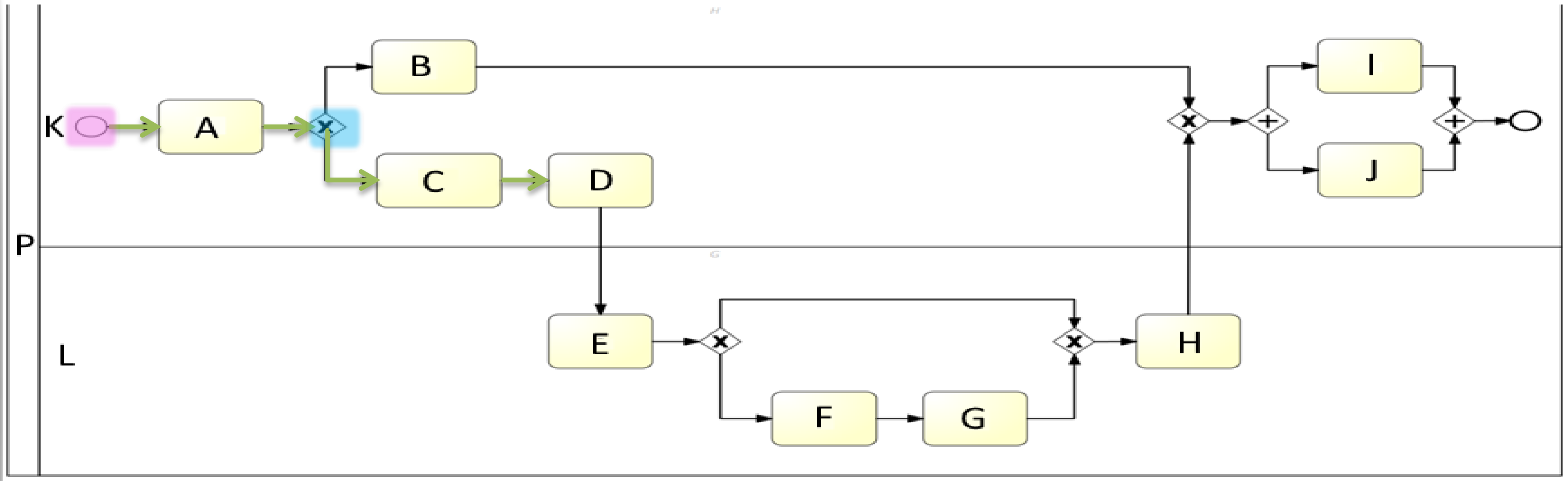
Checkpoints: Events, Gateways   K = 2 Process Models   N = 3 Nodes

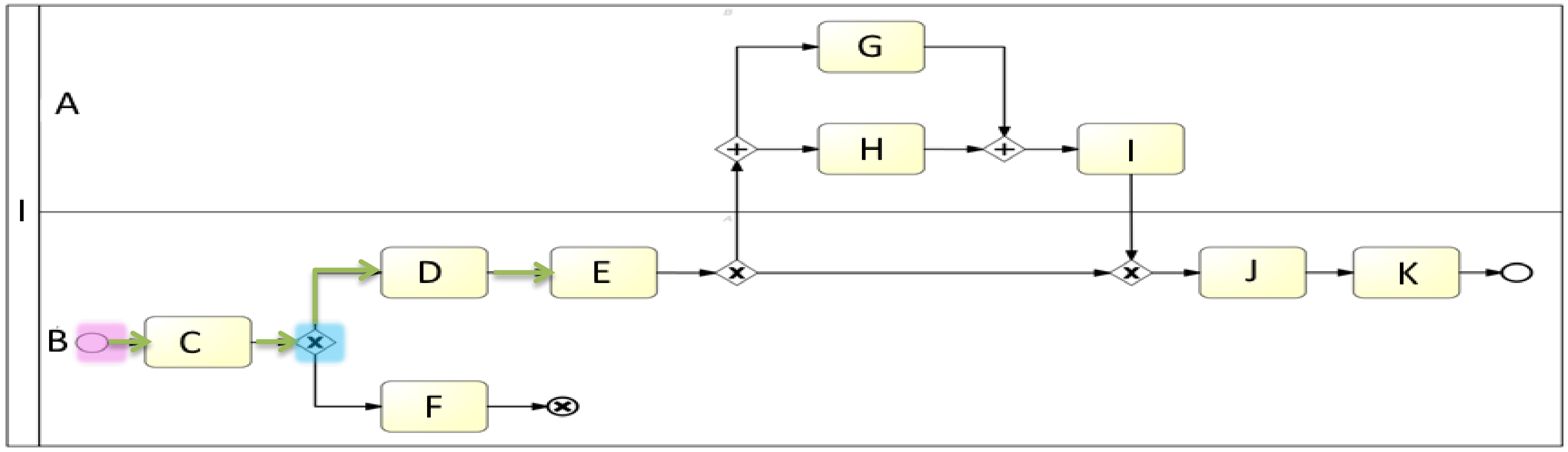




Algorithm: Discovery of RPFs

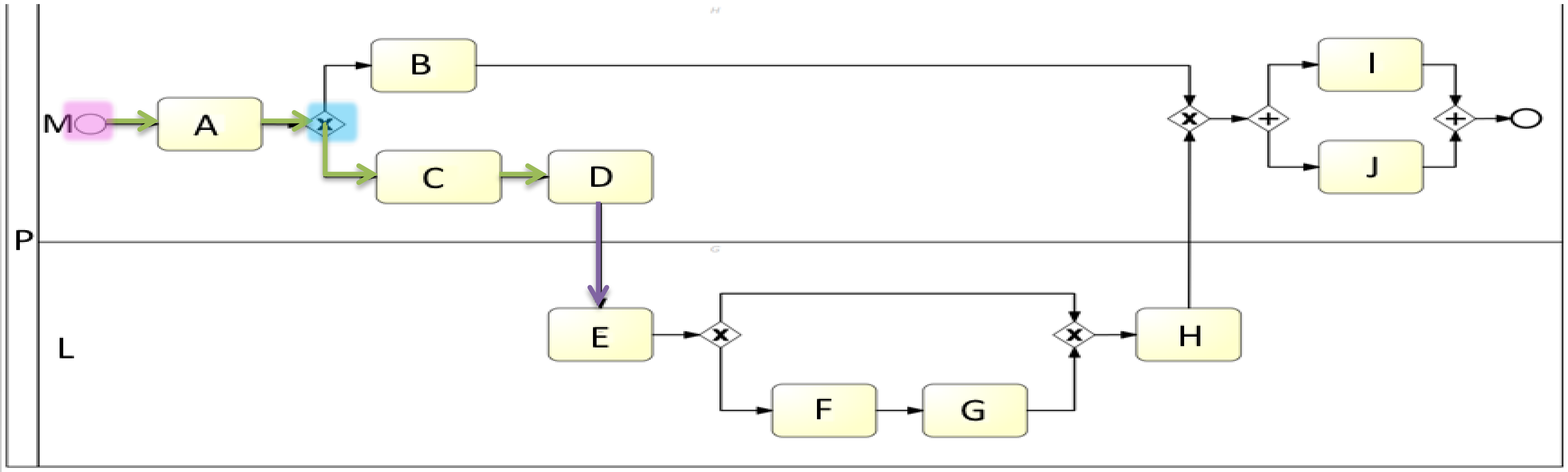
Checkpoints: Events, Gateways   K = 2 Process Models   N = 3 Nodes

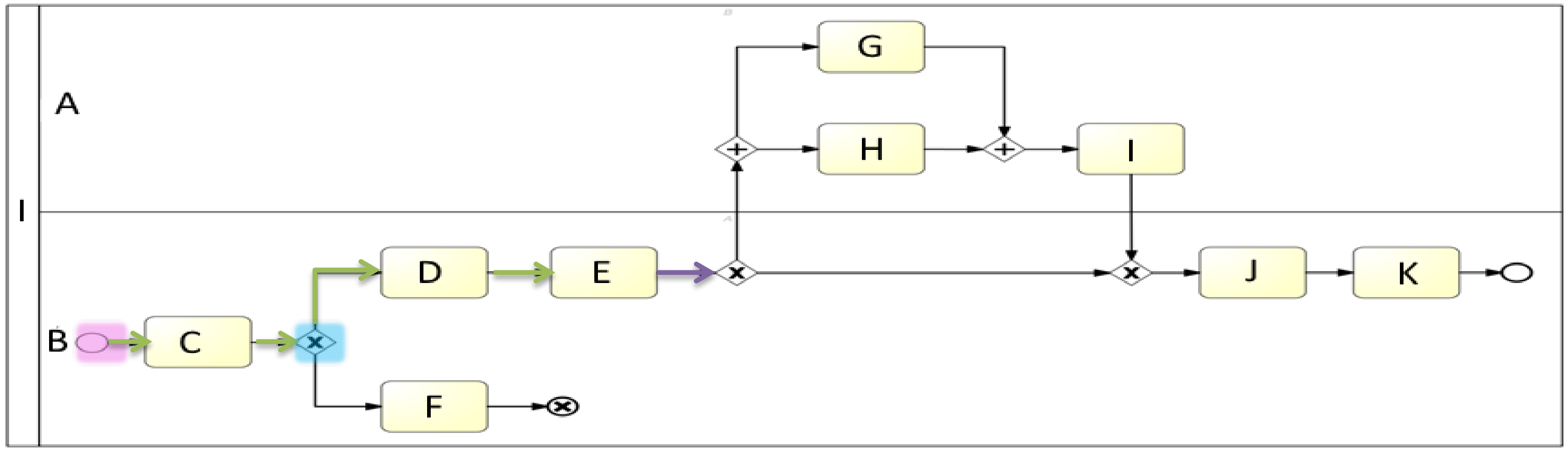




Algorithm: Discovery of RPFs

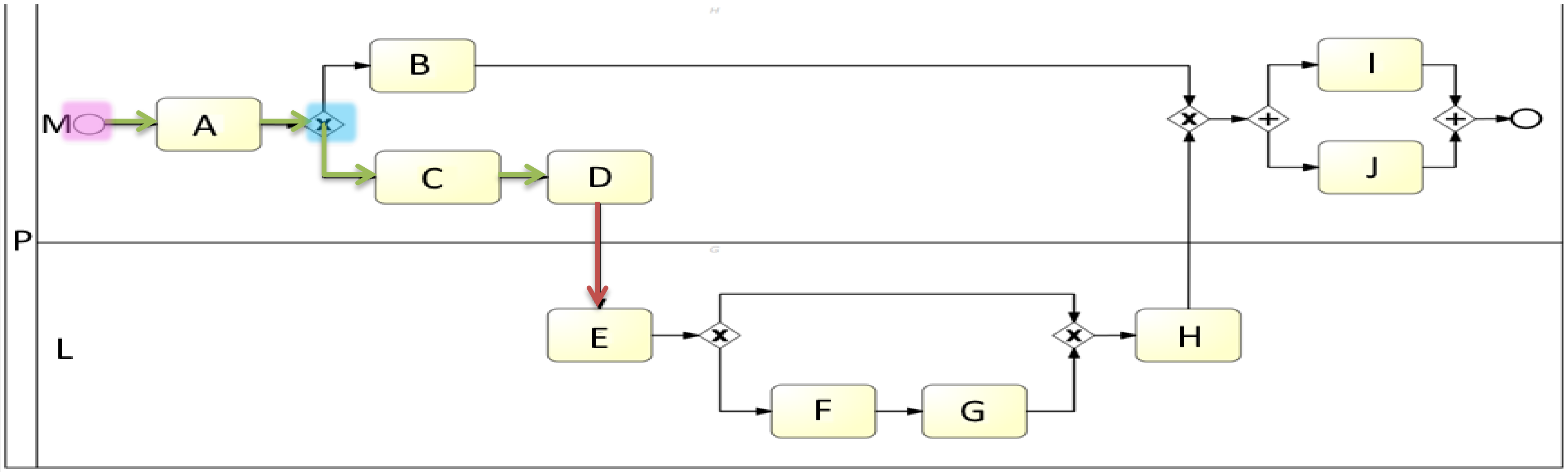
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

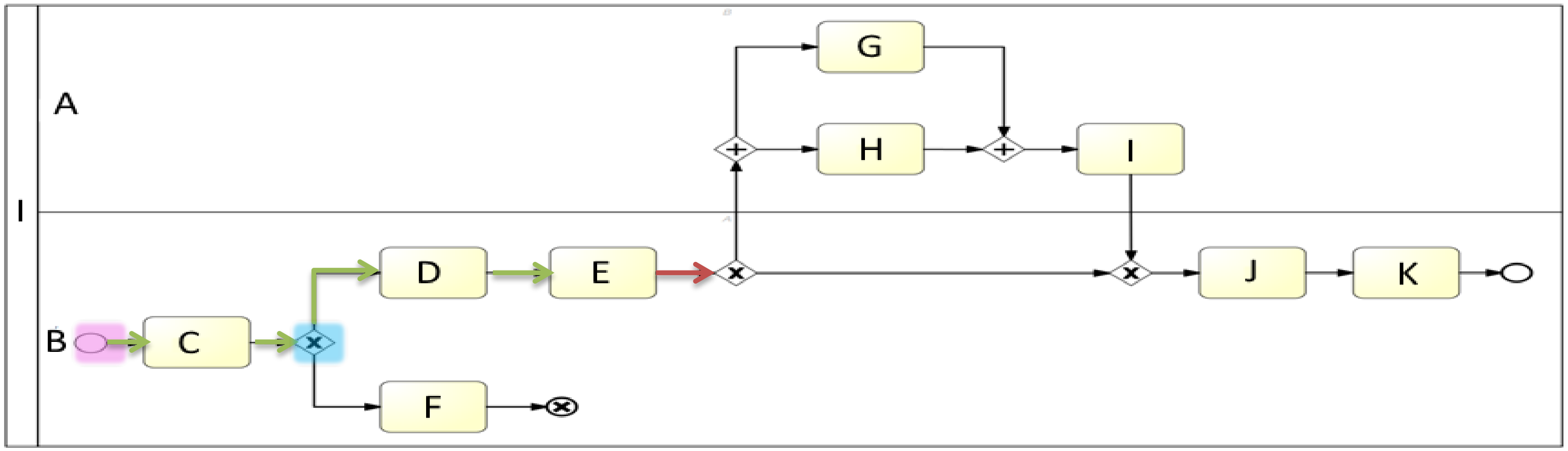




Algorithm: Discovery of RPFs

Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes



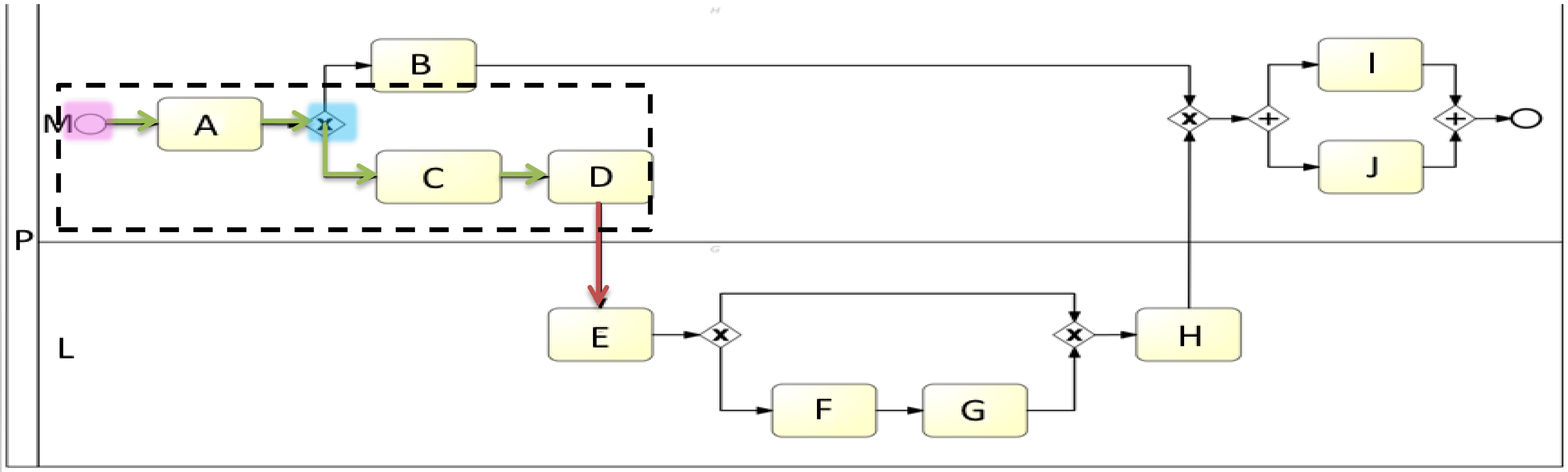


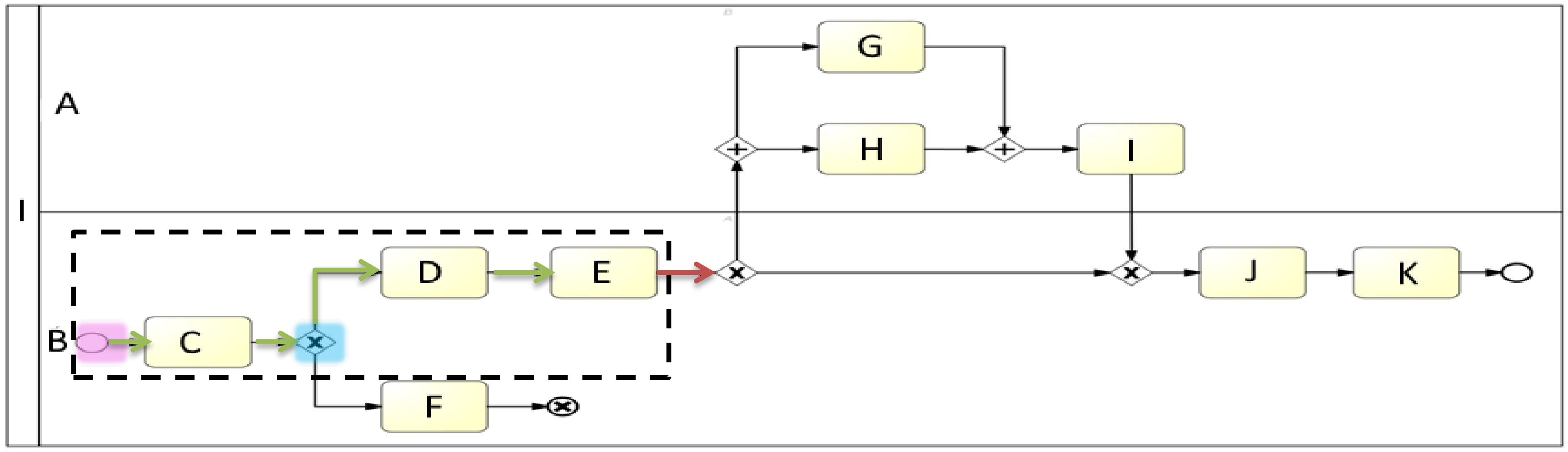
Algorithm: Discovery of RPFs

Checkpoints: Events, Gateways

K = 2 Process Models

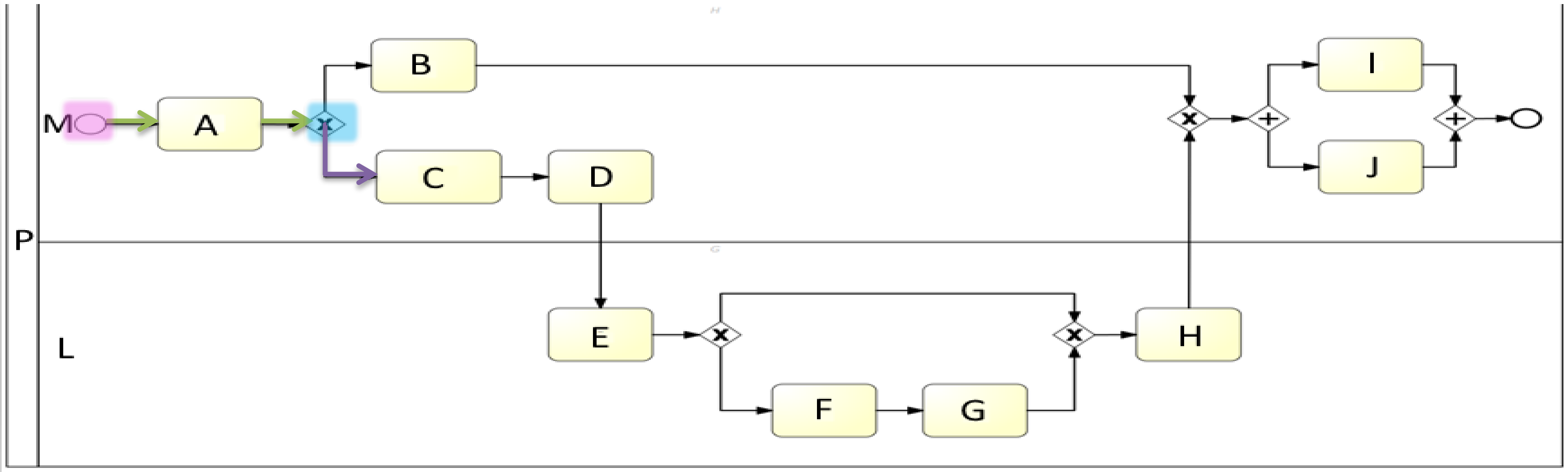
N = 3 Nodes

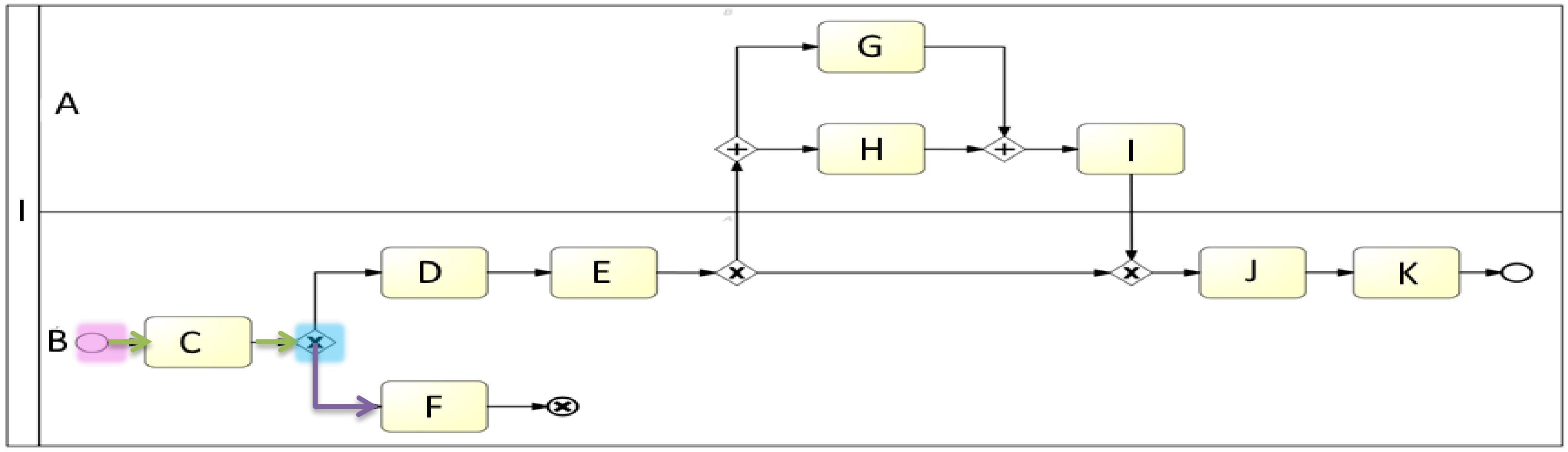




Algorithm: Discovery of RPFs

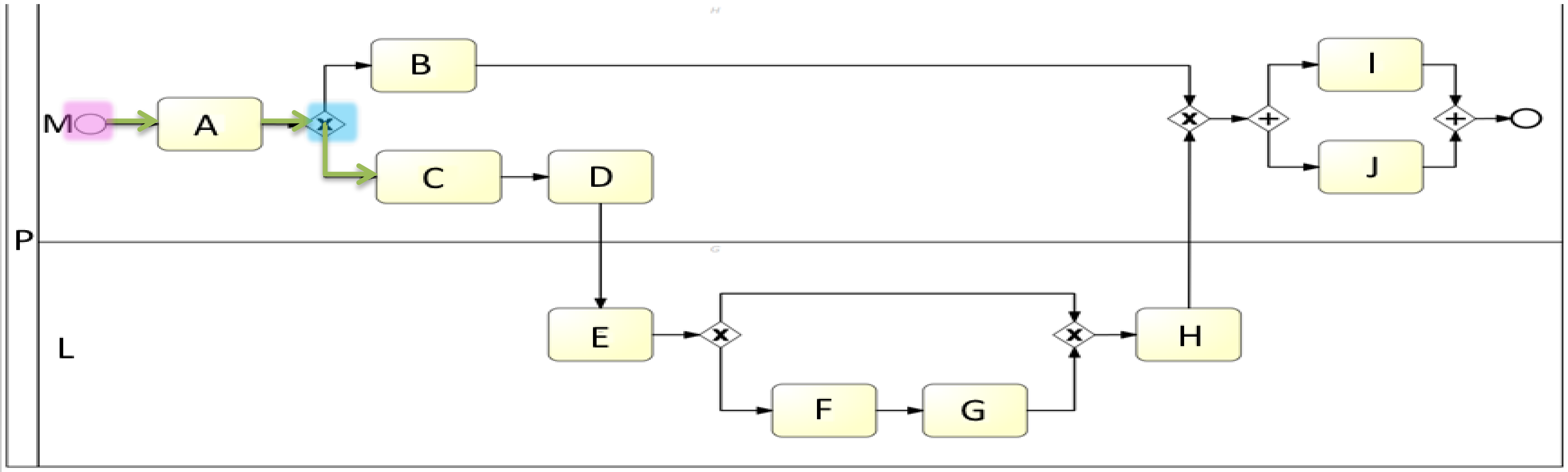
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

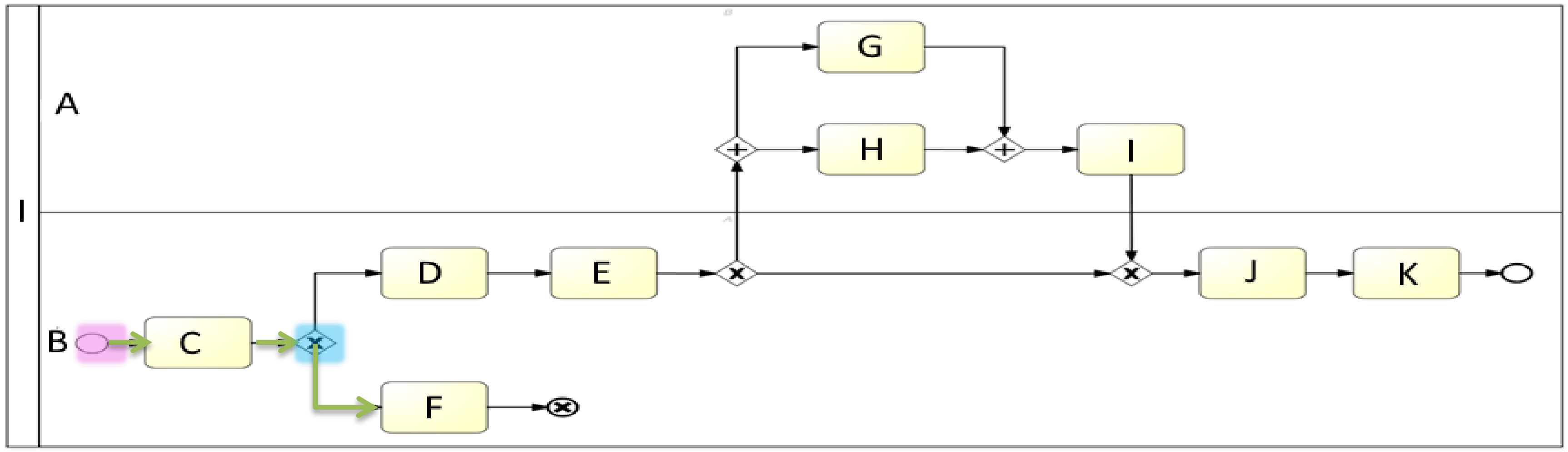




Algorithm: Discovery of RPFs

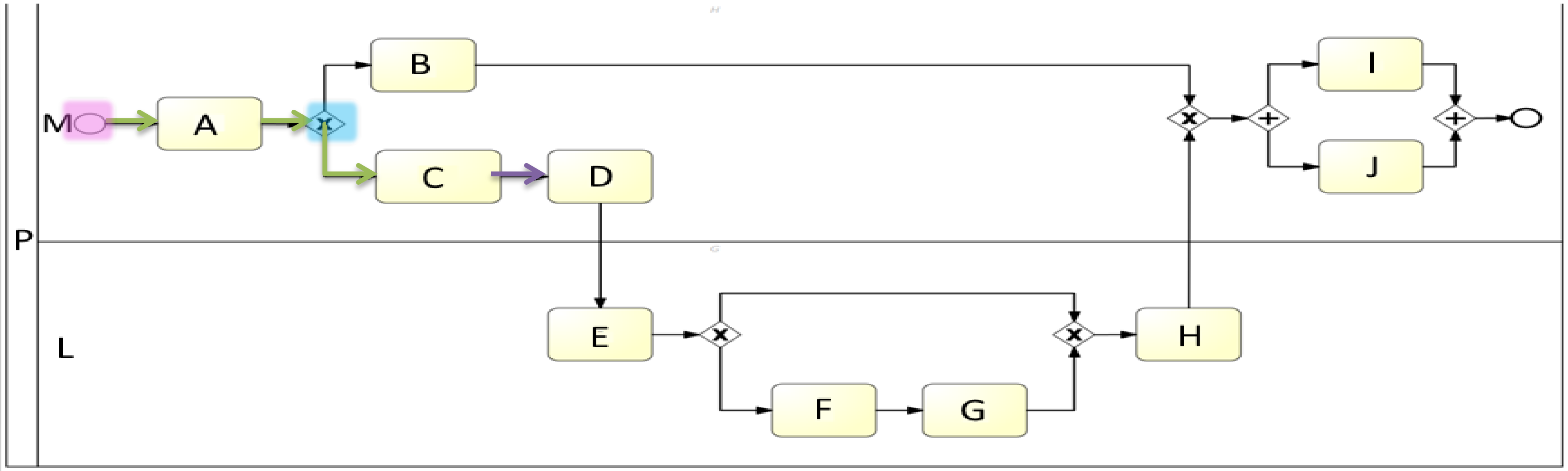
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

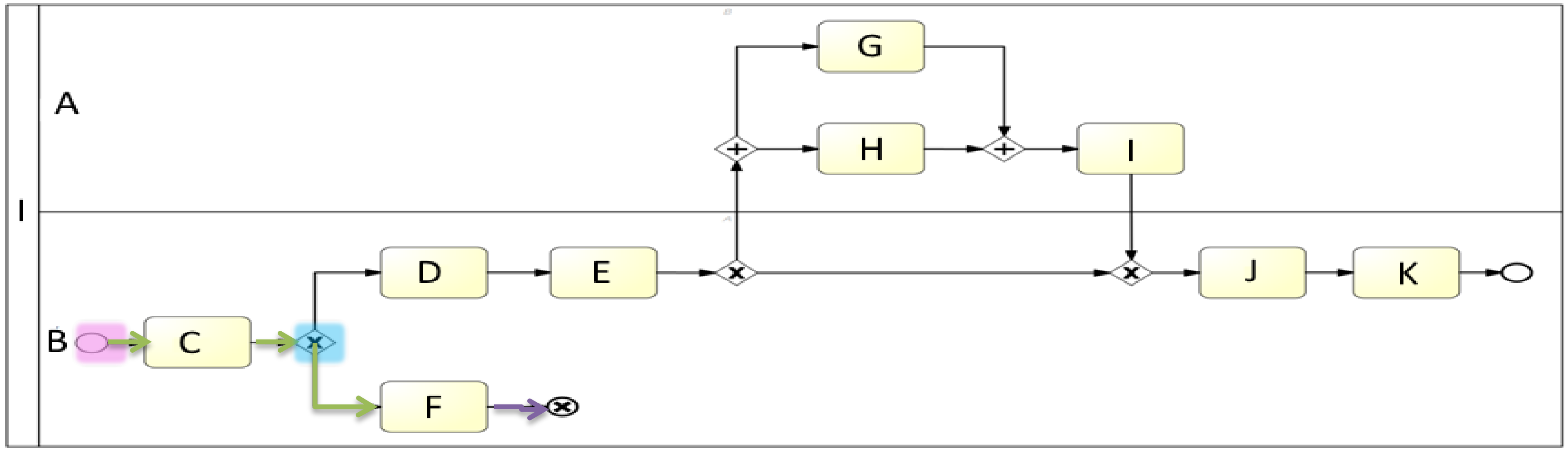




Algorithm: Discovery of RPFs

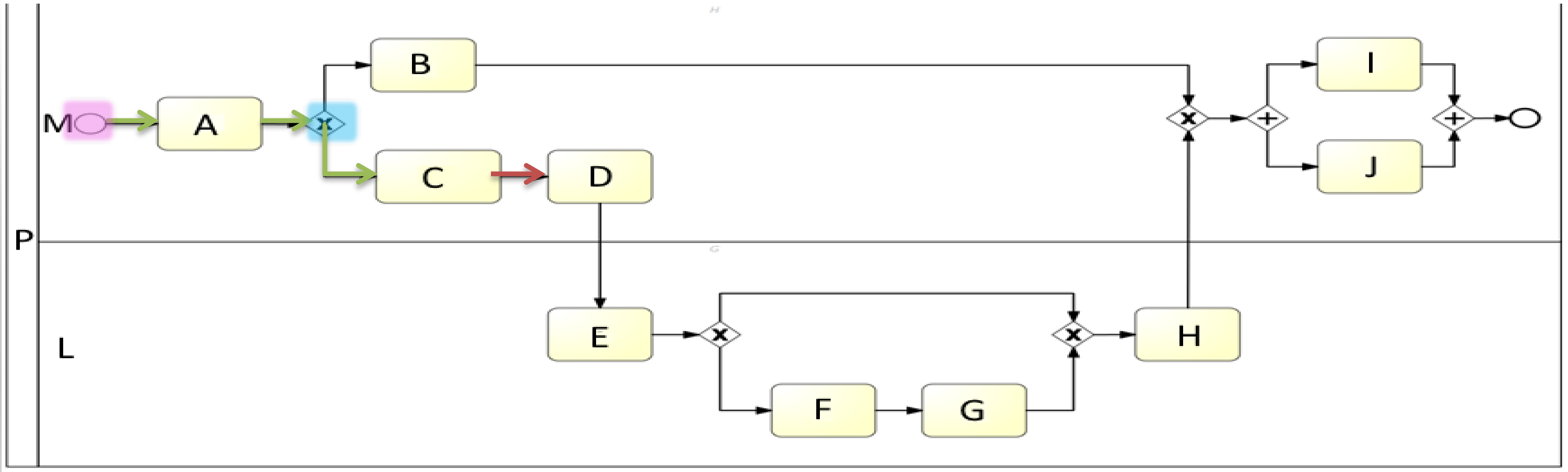
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

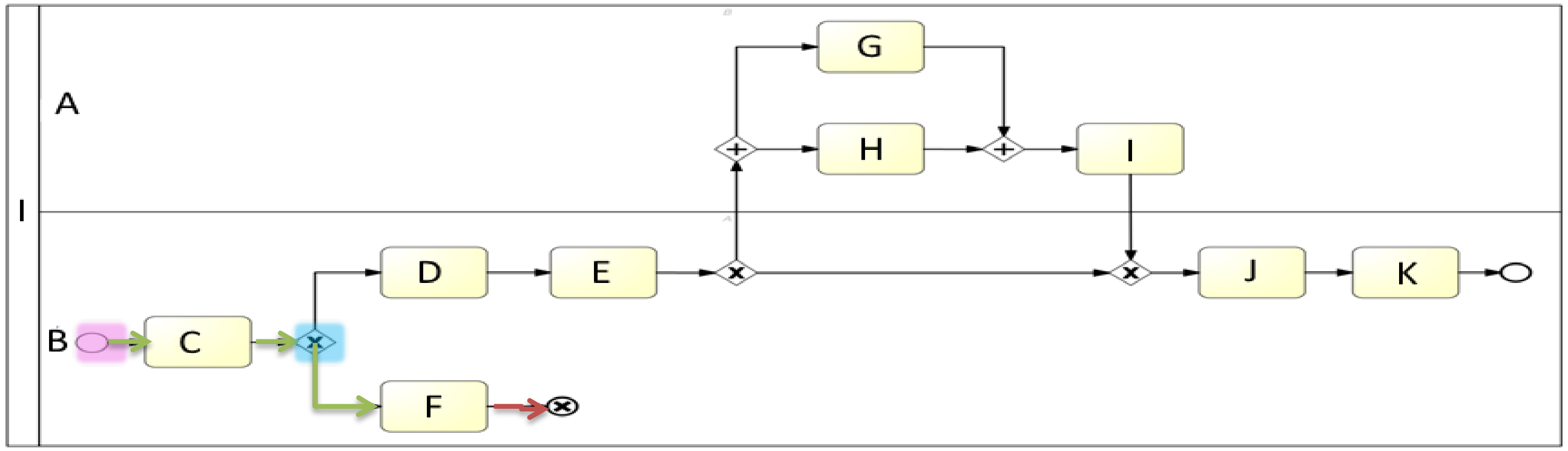




Algorithm: Discovery of RPFs

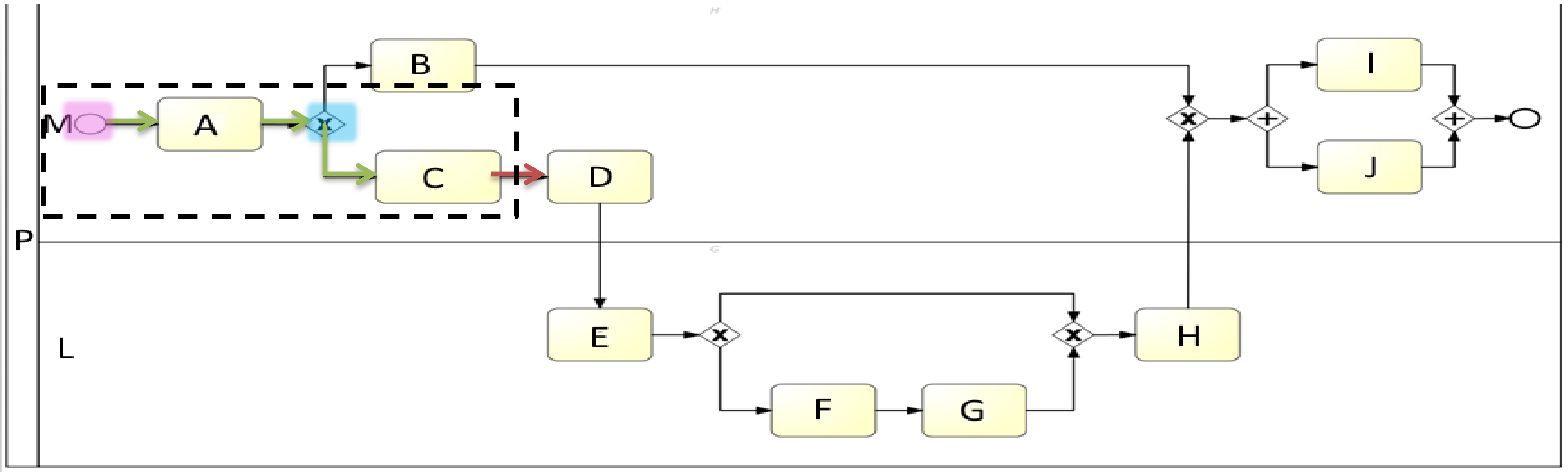
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

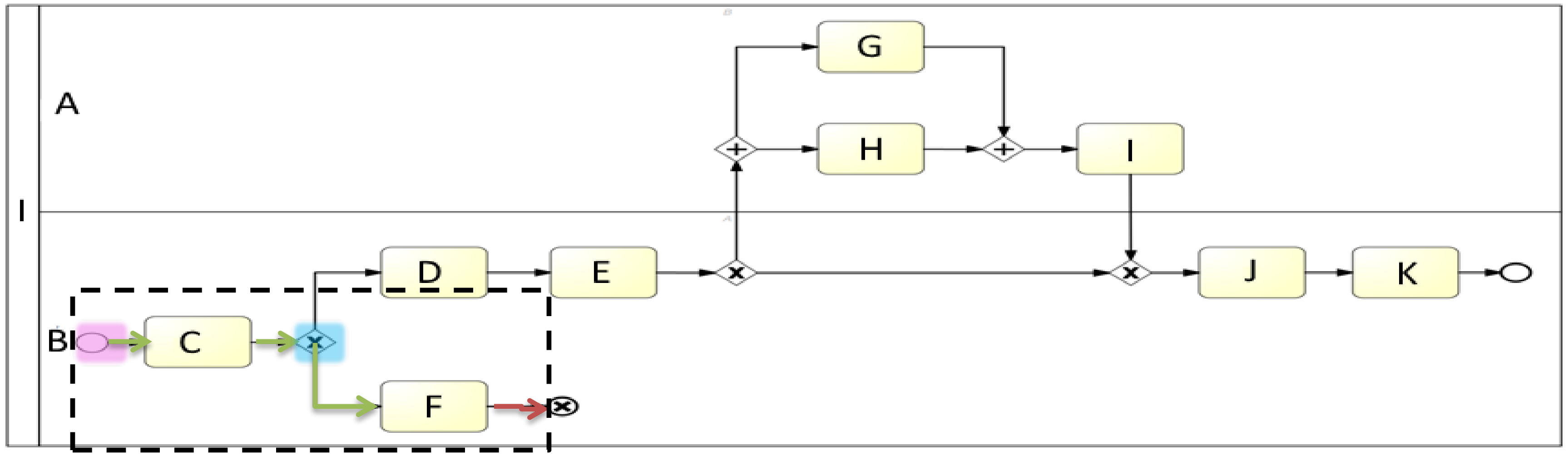




Algorithm: Discovery of RPFs

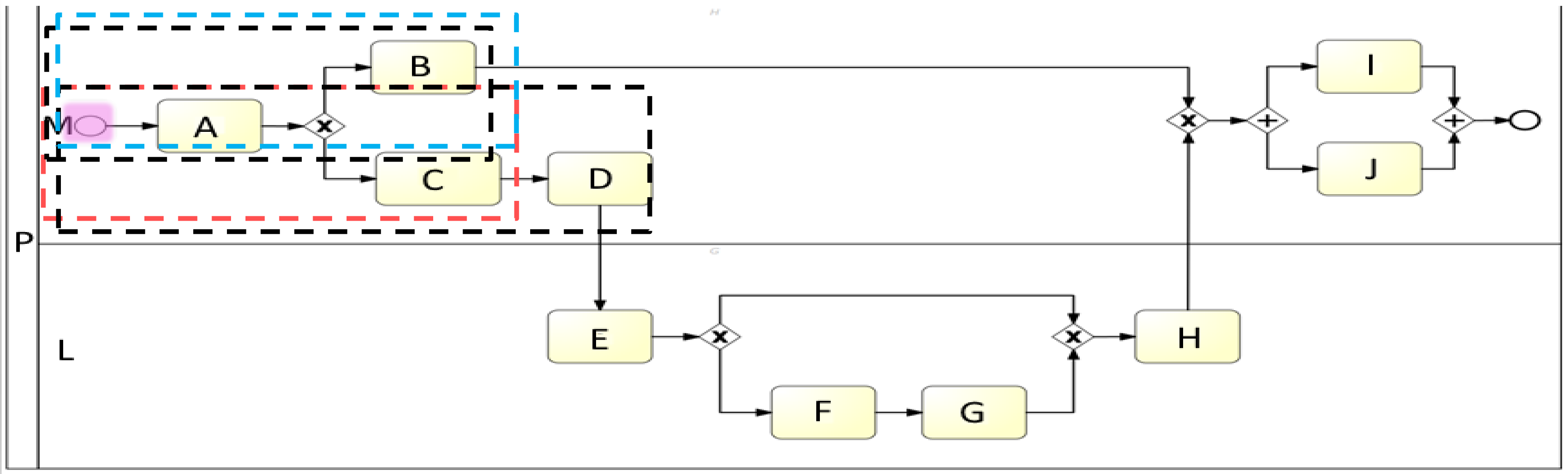
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

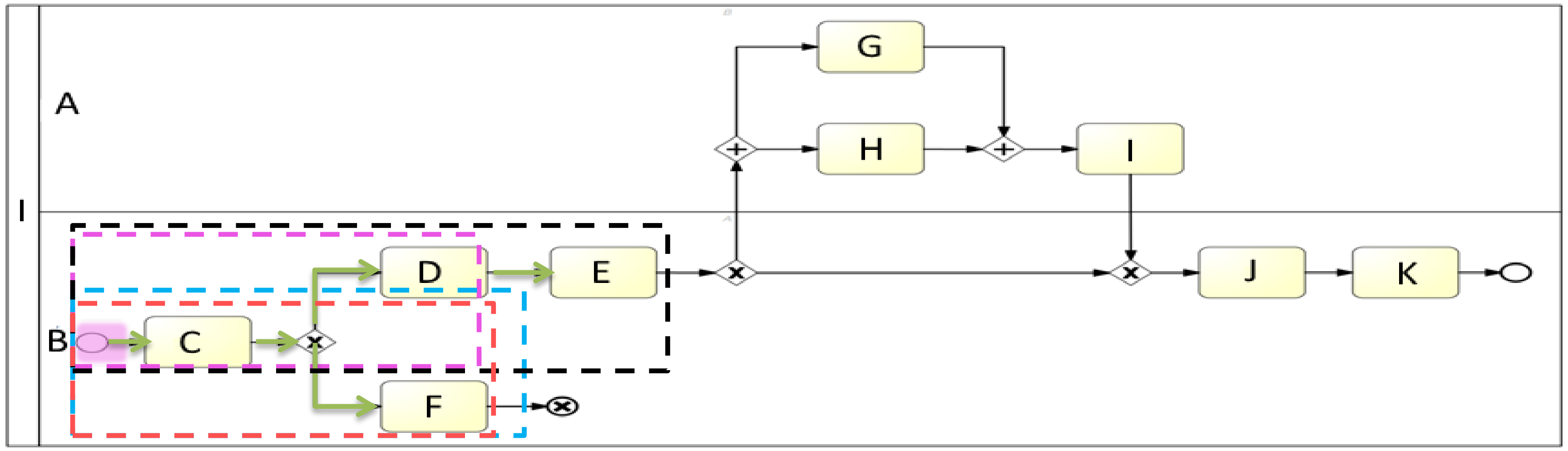




Algorithm: Discovery of RPFs

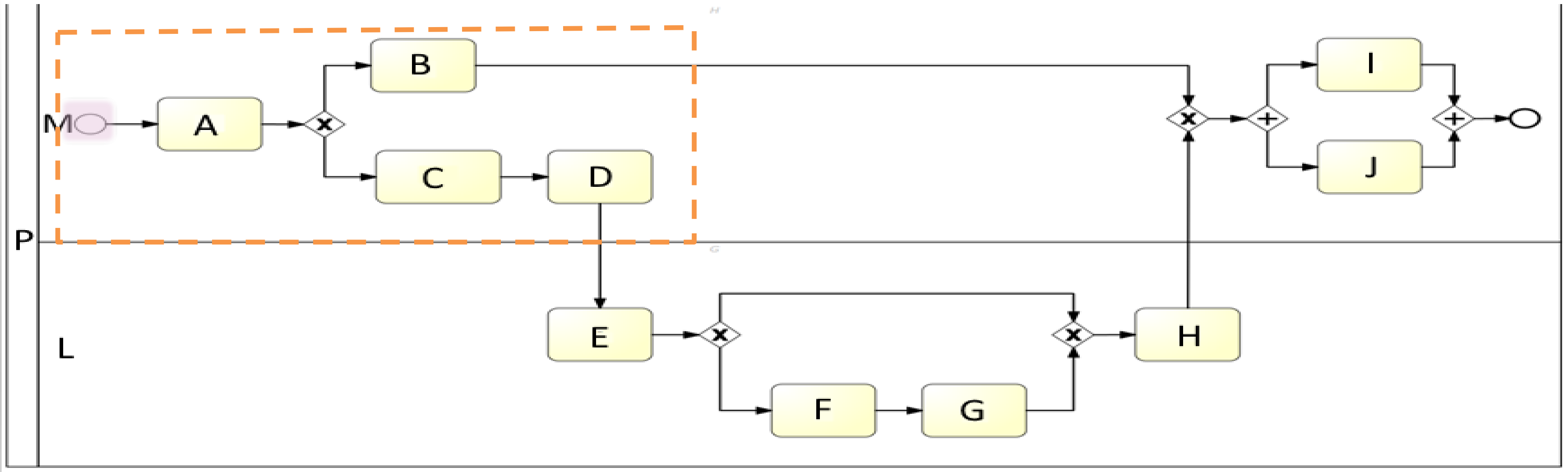
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

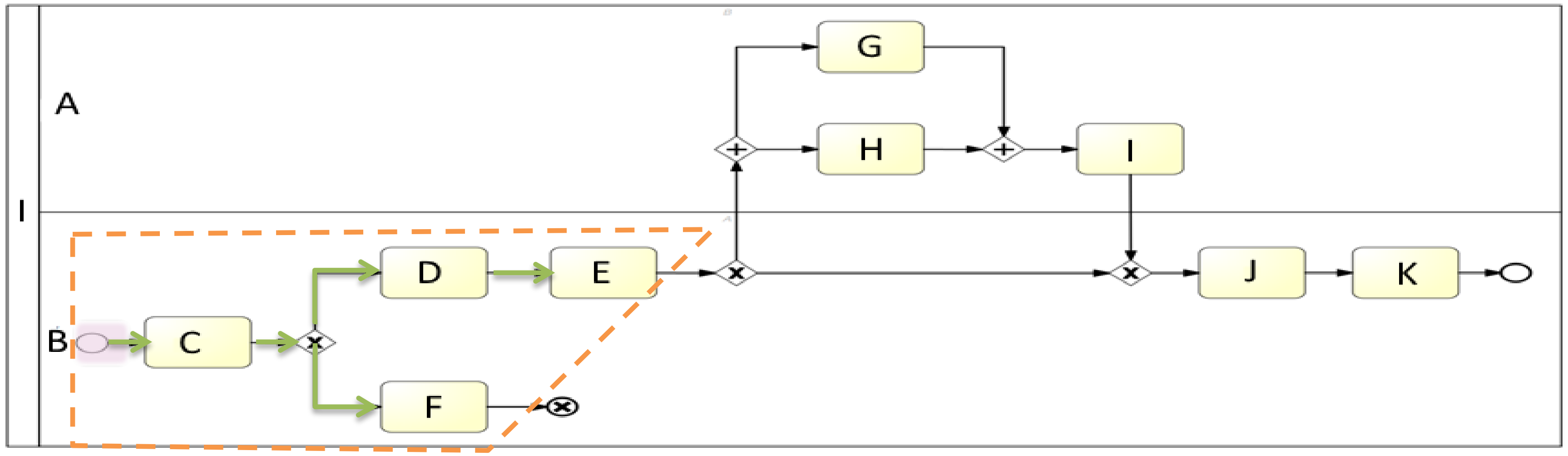




Algorithm: Discovery of RPFs

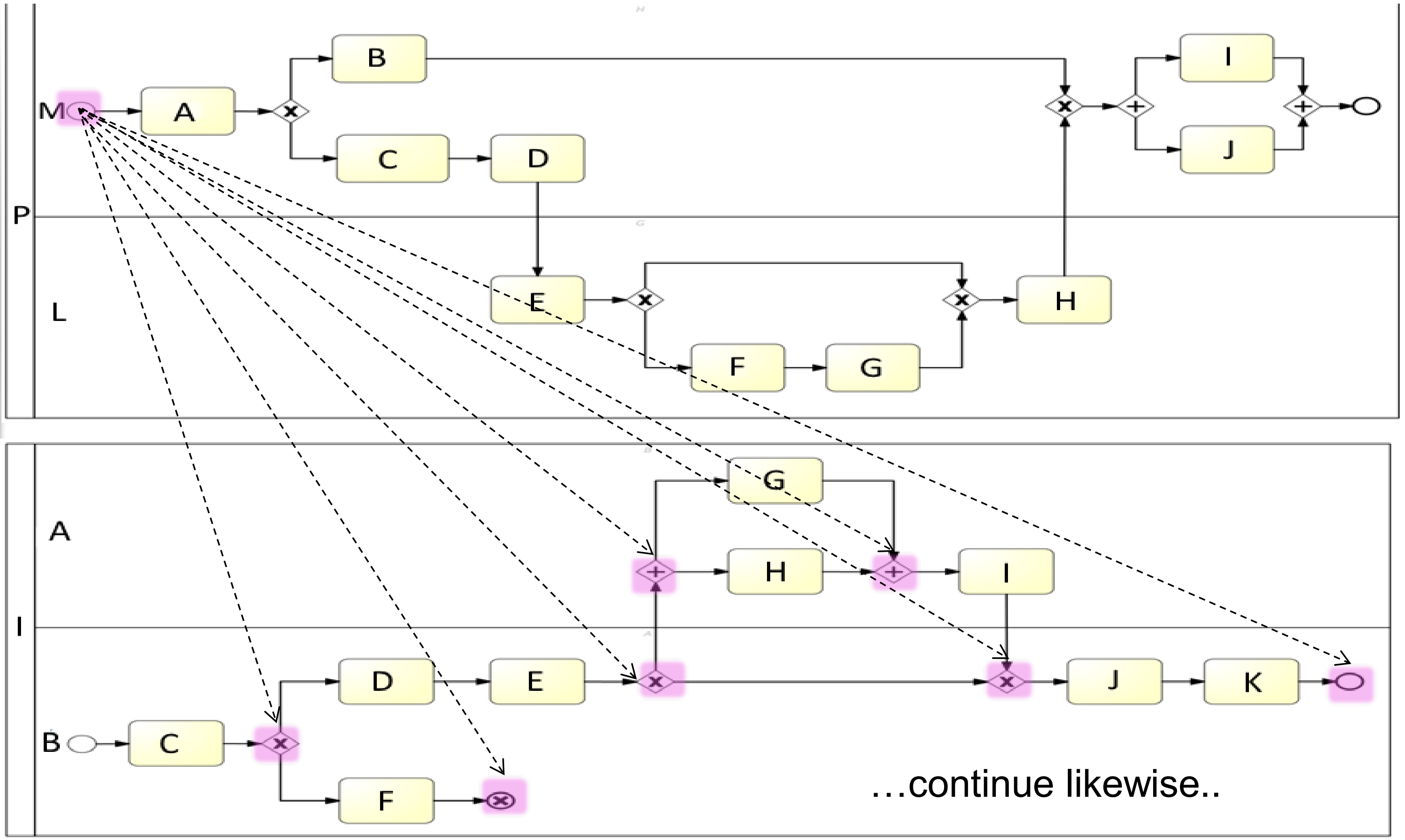
Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes





Algorithm: Discovery of RPFs

Checkpoints: Events, Gateways K = 2 Process Models N = 3 Nodes

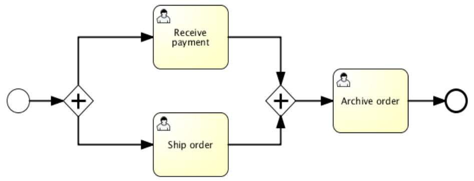


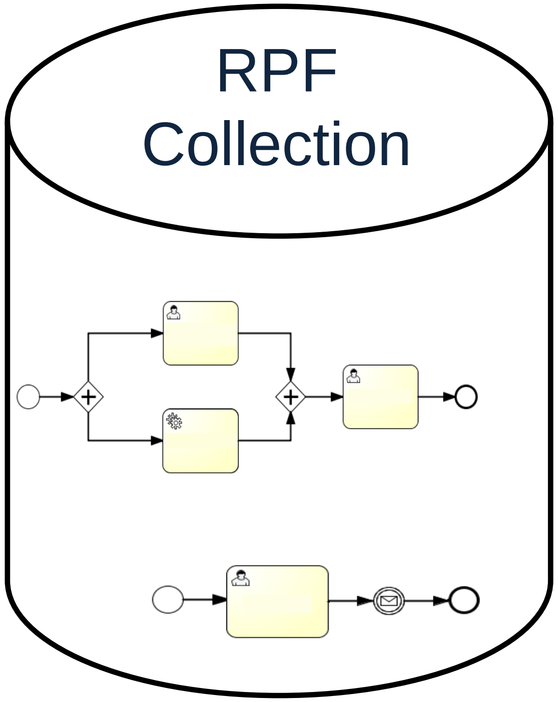
…continue likewise..

☹ **Will not work for cycles**

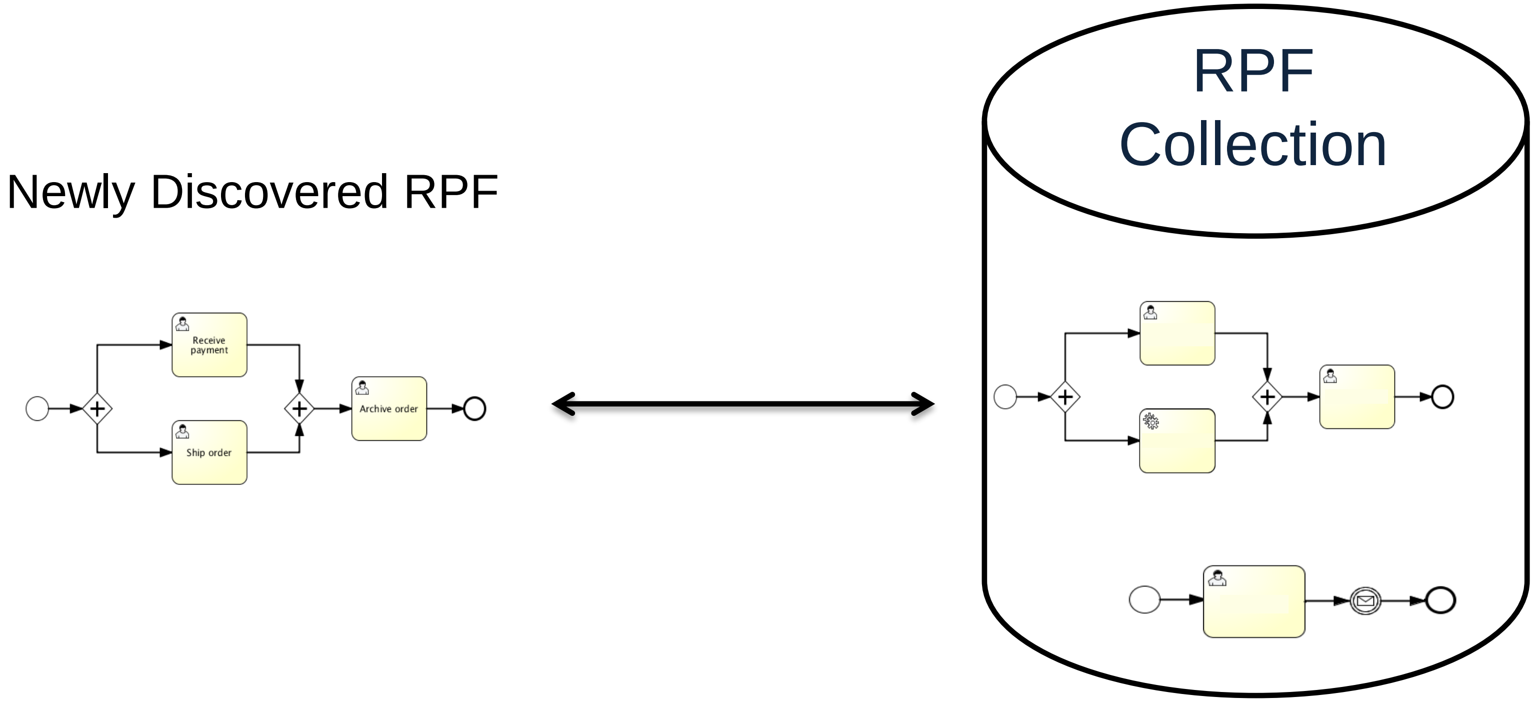
Discover Duplicates and Count Appearance

Newly Discovered RPF

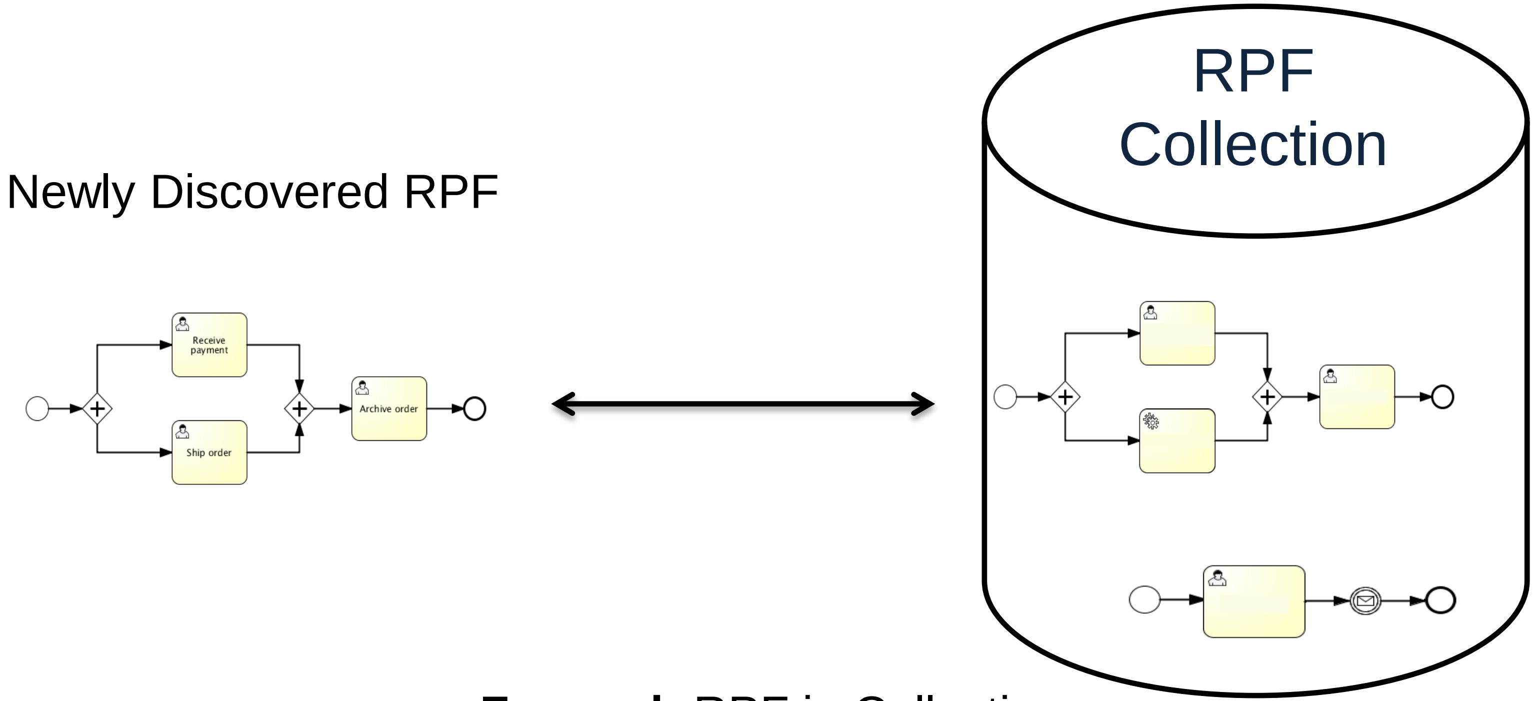




Discover Duplicates and Count Appearance

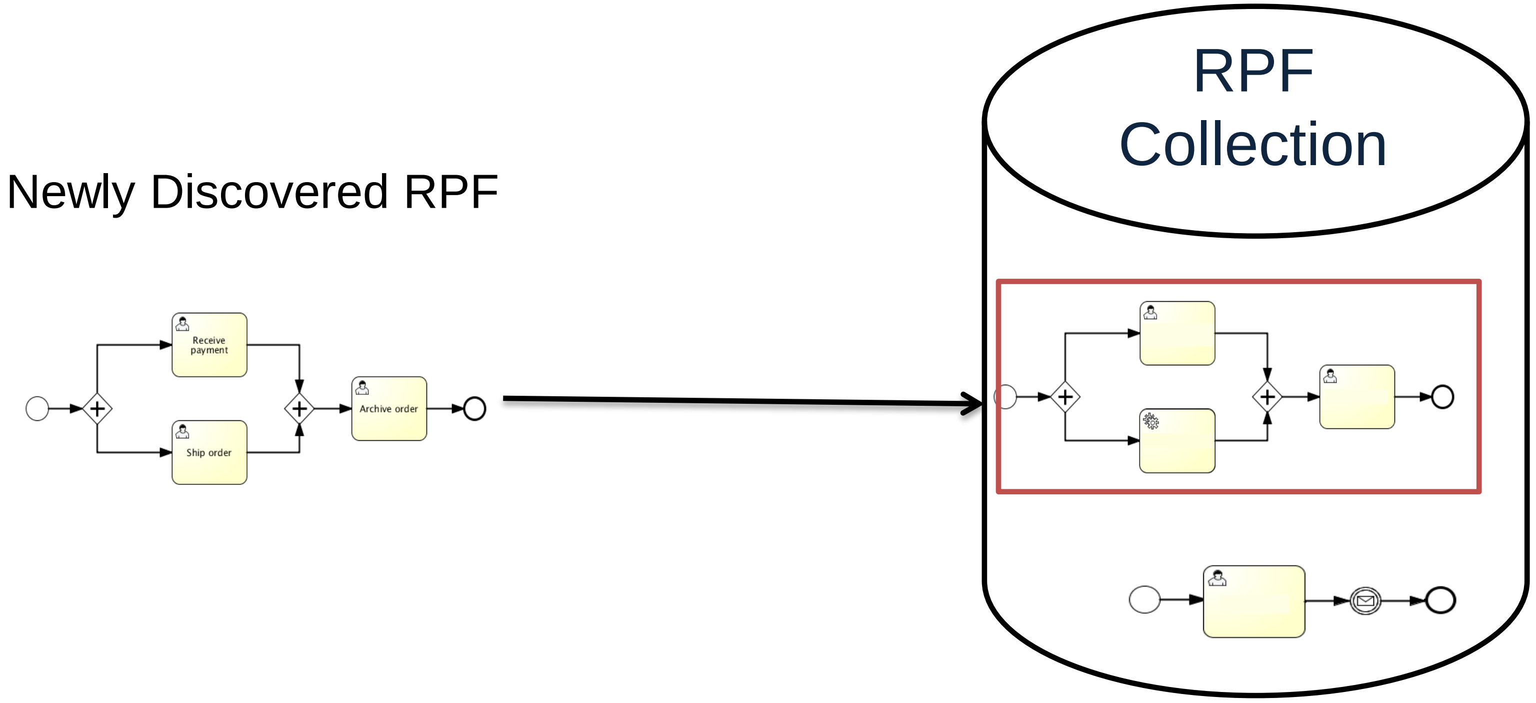


Discover Duplicates and Count Appearance

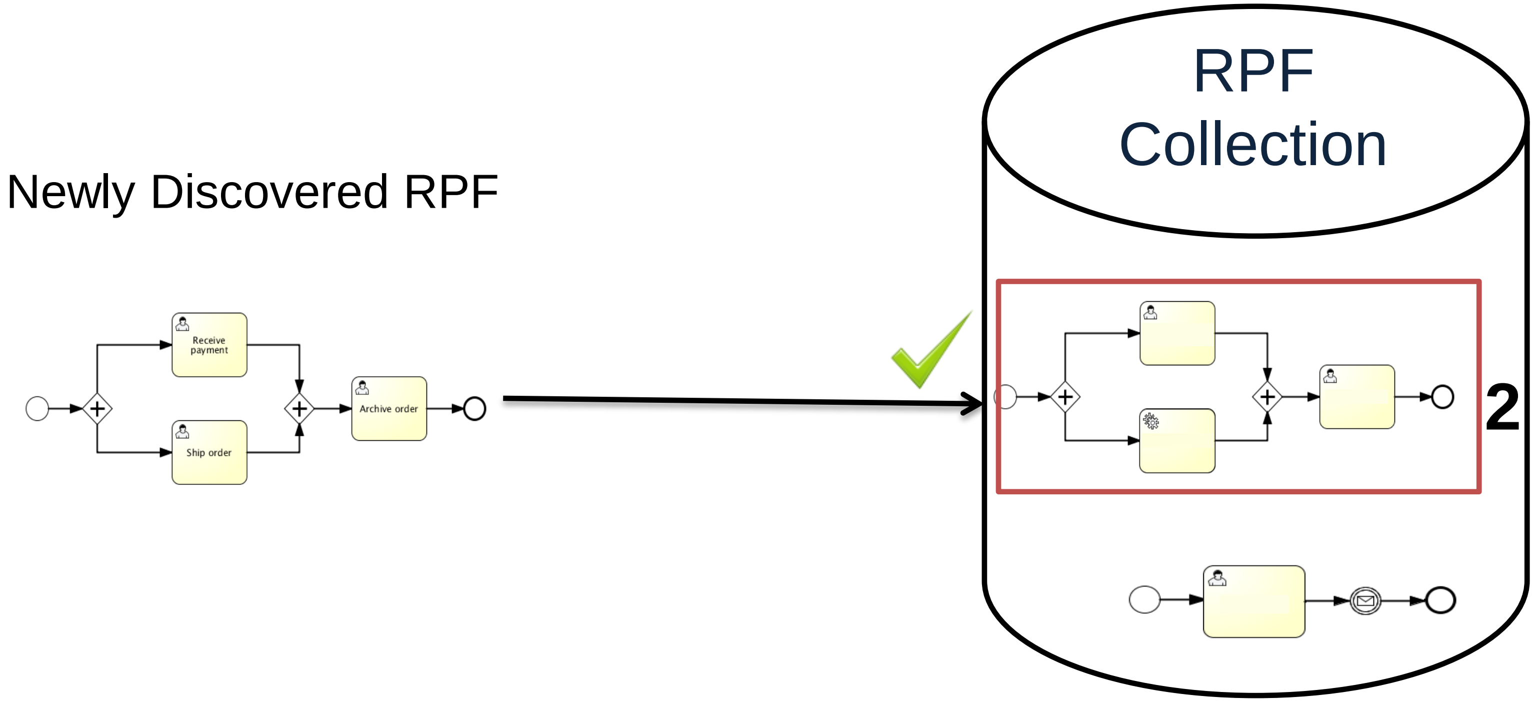


For each RPF in Collection:
If RPF have the same size:
COMPARE

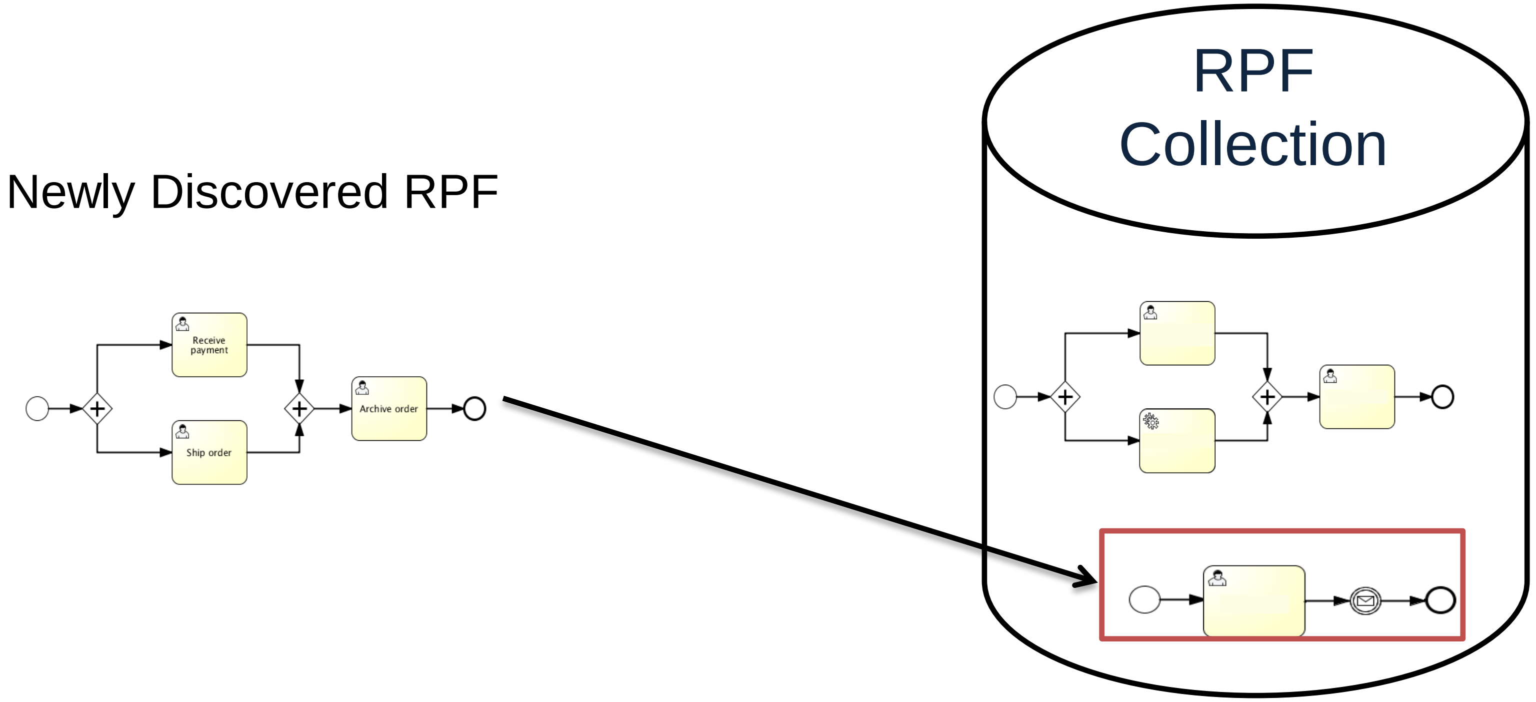
Discover Duplicates and Count Appearance



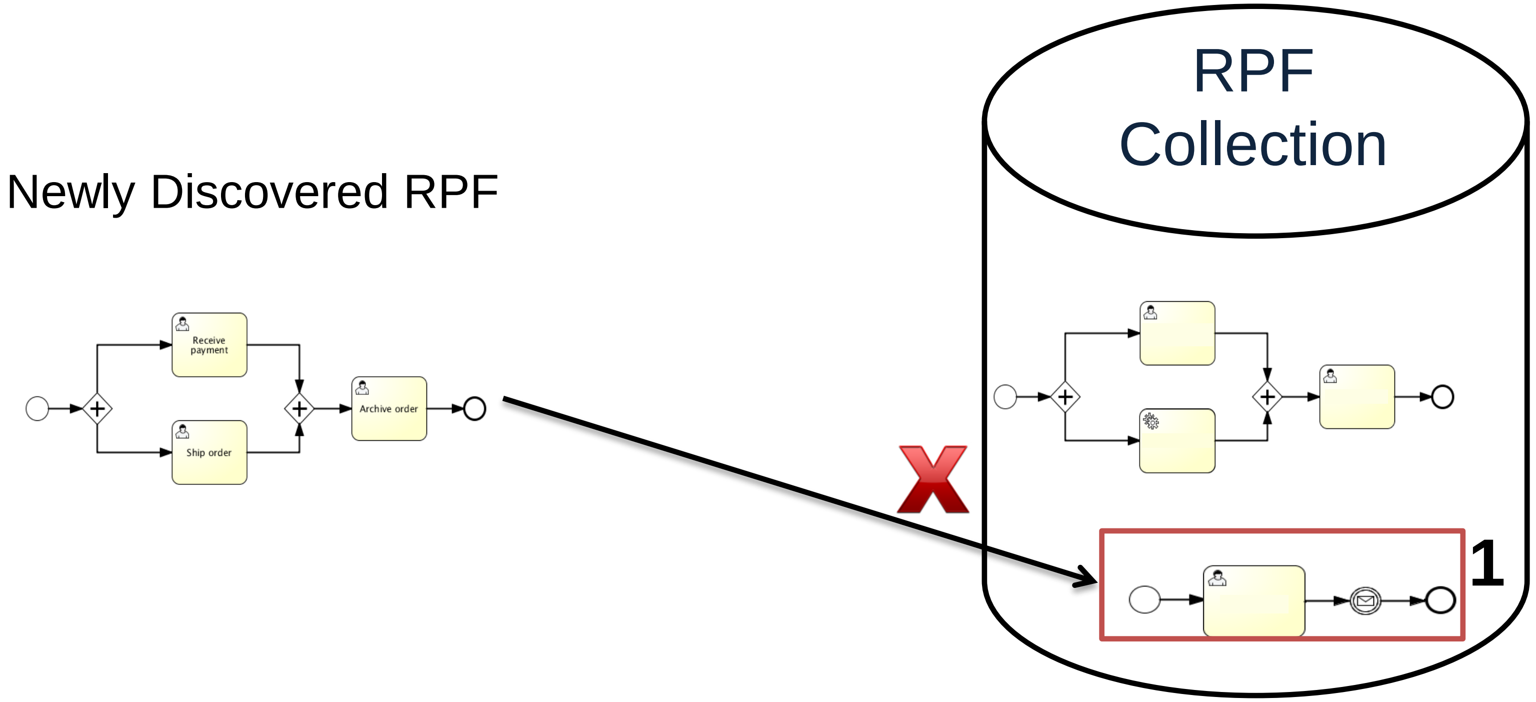
Discover Duplicates and Count Appearance



Discover Duplicates and Count Appearance



Discover Duplicates and Count Appearance



Validation and Discussion

Research

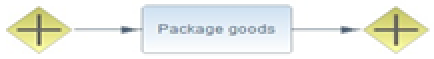
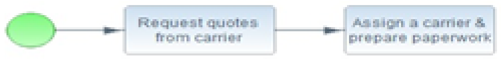
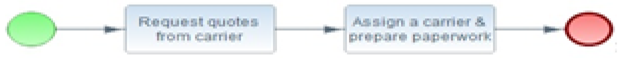
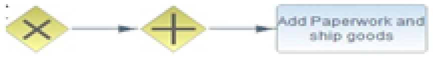
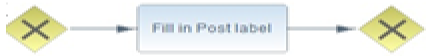
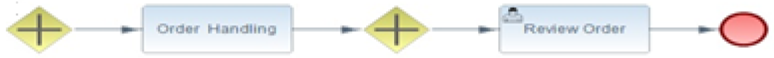
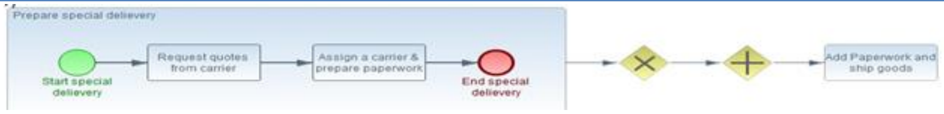
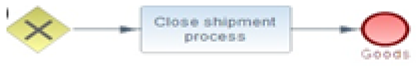
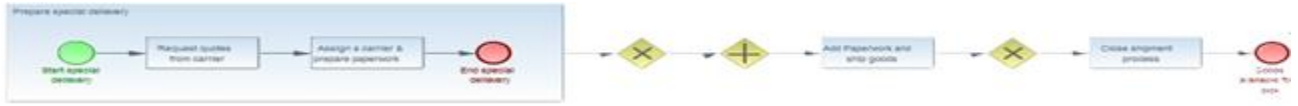
Validation

- 43 BPMN 2.0 Process Models
  - BPMN 2.0 Standard Example Processes
  - Models used in Pietsch and Wenzel, 2012
- 903 Comparisons
- 1544 non-filtered RPFs
- 83.22% decrease of results when filtering duplicates (259 RPFs)
- 54 RPF appear > 1 time

Median = Threshold = 14

- 27 RPFs with re-appearance rate above the *threshold*

Some Representative RPF and Number of Appearance

ID	Fragment	Count
1		178
2	 Request quotes from carrier Assign a carrier & prepare paperwork	169
3	 Request quotes from carrier Assign a carrier & prepare paperwork	117
4		101
5		62
6		60
7	 Prepare special delievery Start special delievery Request quotes from carrier Assign a carrier & prepare paperwork End special delievery Add Paperwork and ship goods	44
8		42
9	 Prepare special delievery Start special delievery Request quotes from carrier Assign a carrier & prepare paperwork End special delievery Add Paperwork and ship goods Close shipment process	42

IAAS Research

Conclusions & Outlook

- Extension of RPF Discovery algorithm
- Automatic count of the RPF appearance in collection
- We have evaluated the approach on 43 BPMN 2.0 process models
- Conclusions on frequently used structures (best practices)
- Conclusions for collection’s special characteristics

- Extend the algorithms for the complete set of BPMN 2.0
- Apply to thousands real world BPMN 2.0 process models and execute thorough analysis
- Implement the prototype for process synthesizing methodology

Thank You!!!

Marigianna Skouradaki: skourama@iaas.uni-stuttgart.de

☺