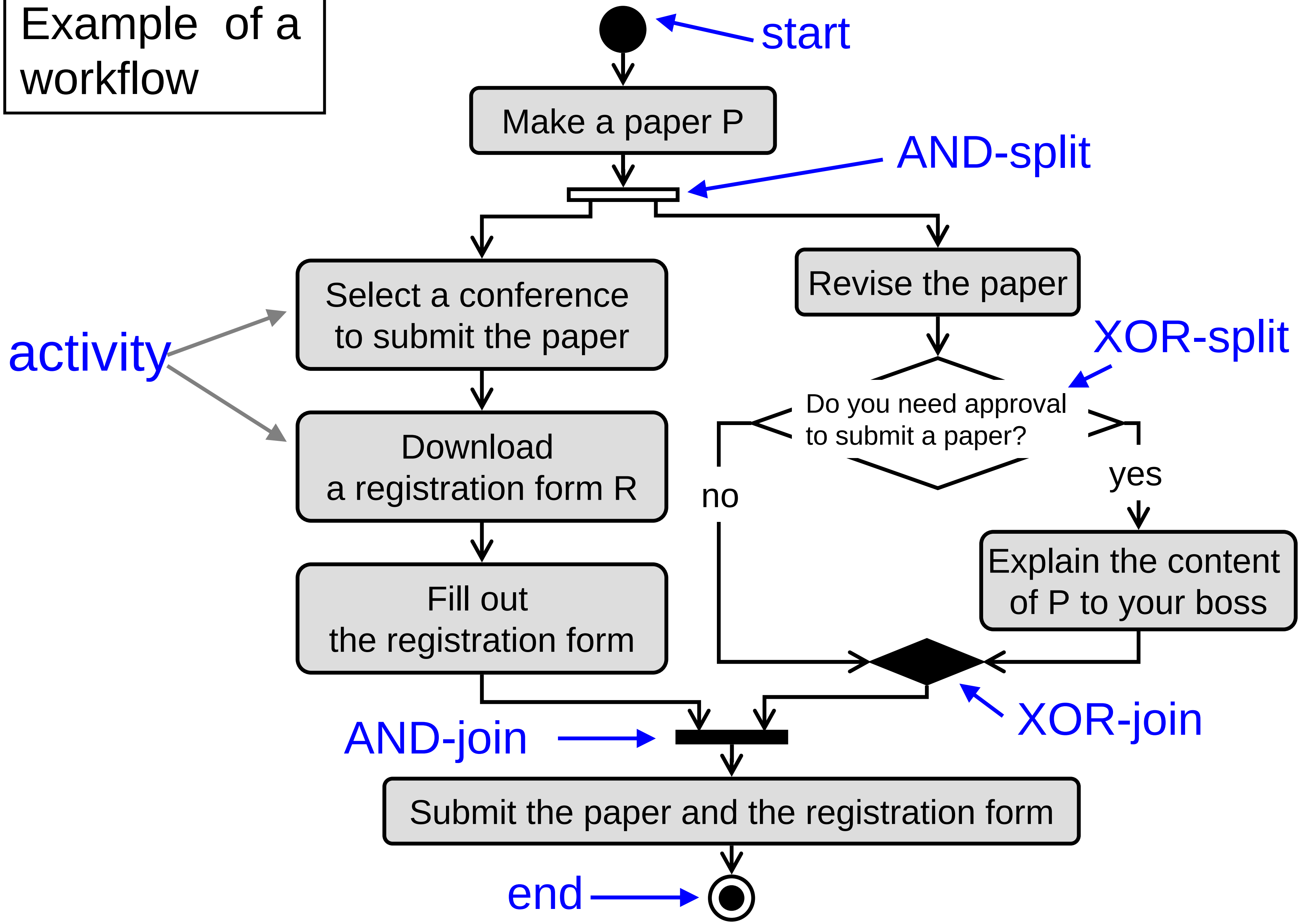
Correctness Properties for Workflows with Multiple Starts and/or Ends

O.Takaki, I.Takeuti, T.Seino,
N.Izumi, K.Takahshi
AIST

- Contents of this talk
  - Motivation
    - Correctness of Workflows
    - Why do we consider multiple starts and/or ends?
  - General Correctness of Workflows with Multiple starts and/or Ends

Example of a workflow



- Workflow (Nodes, Arcs):

  a simple connected directed graph

  – Nodes = Starts U Ends U Activities

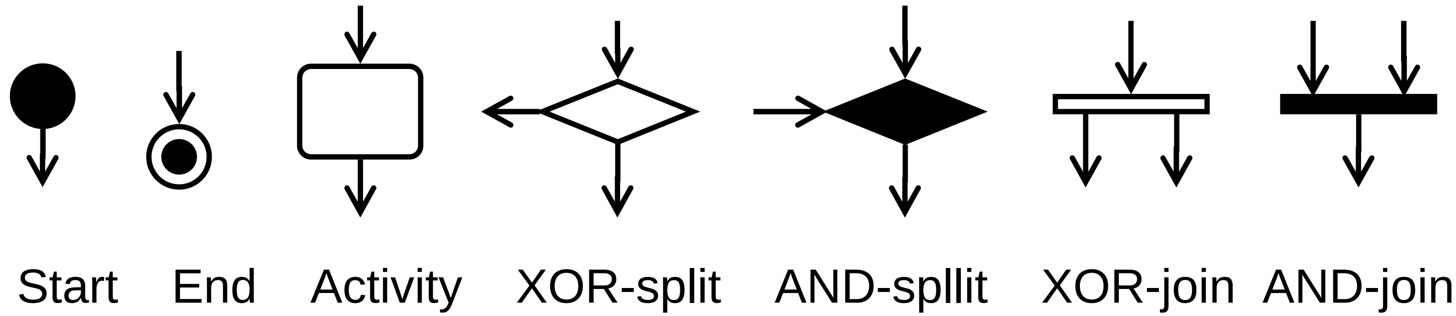
    U XOR-splies U XOR-joins

    U AND-splies U AND-joins

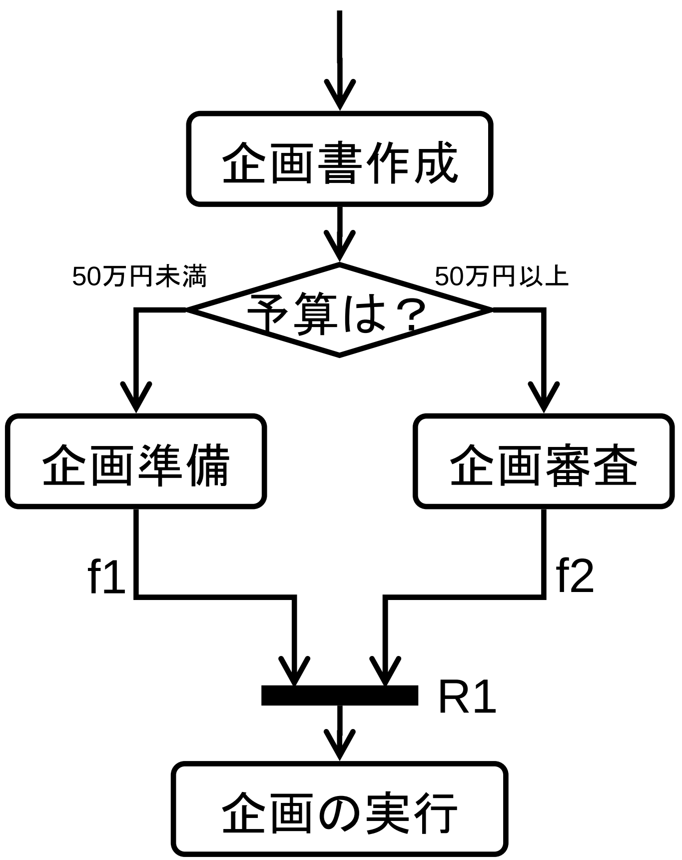
  – For each n∈Nodes there exists a path from a start to n.

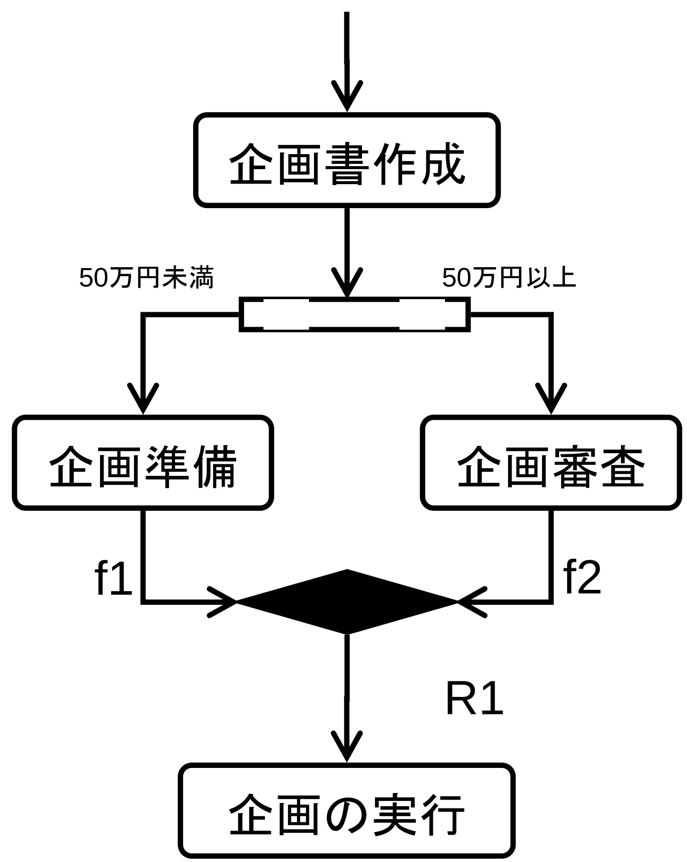
  – For each n∈Nodes there exists a path from n to an end.

  – In this talk, we consider only acyclic workflows.



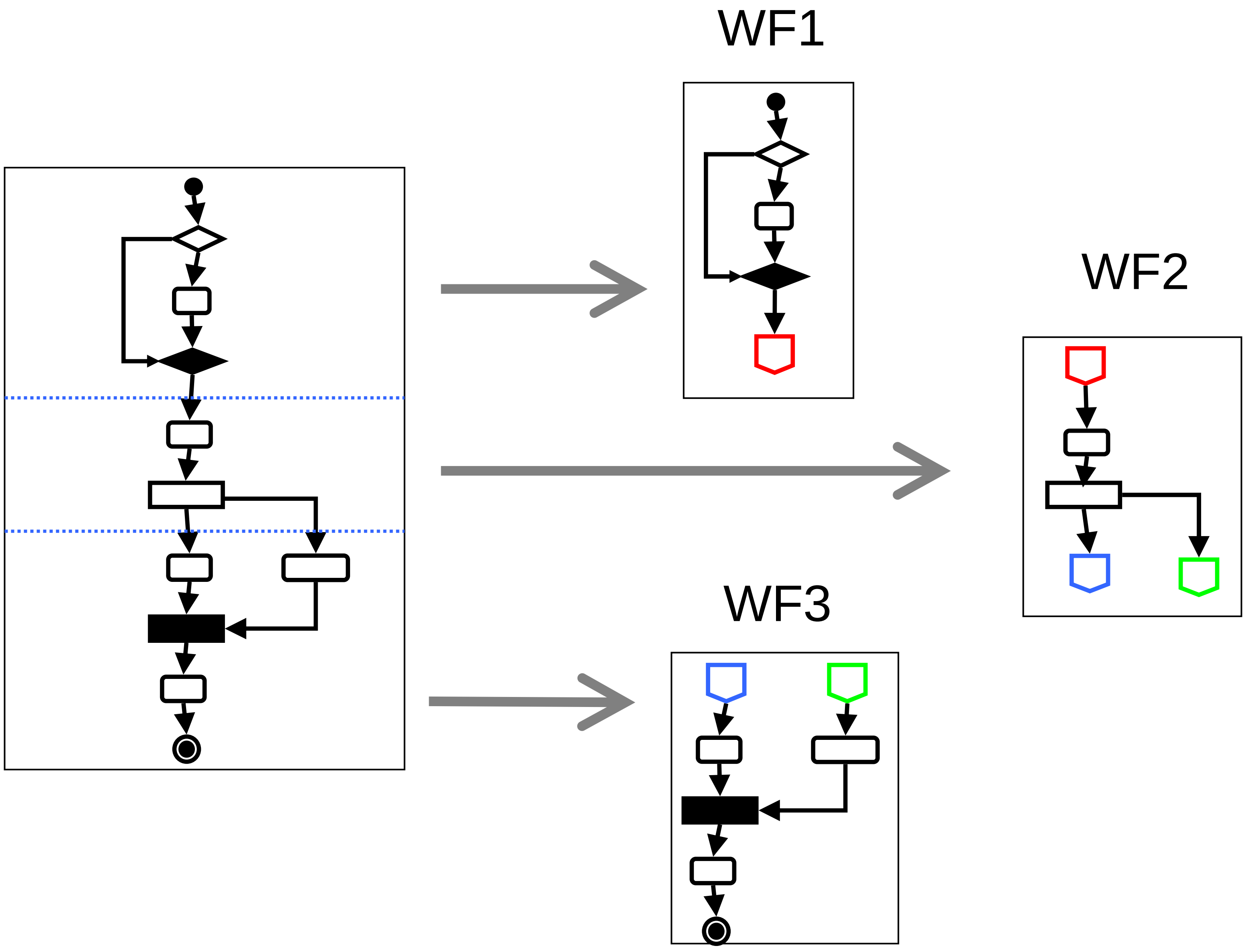
- Correctness of Workflow with one start and one end [Sadiq & Owlrowska 00]
  - – Deadlock free
  - – Lack of synchronization free



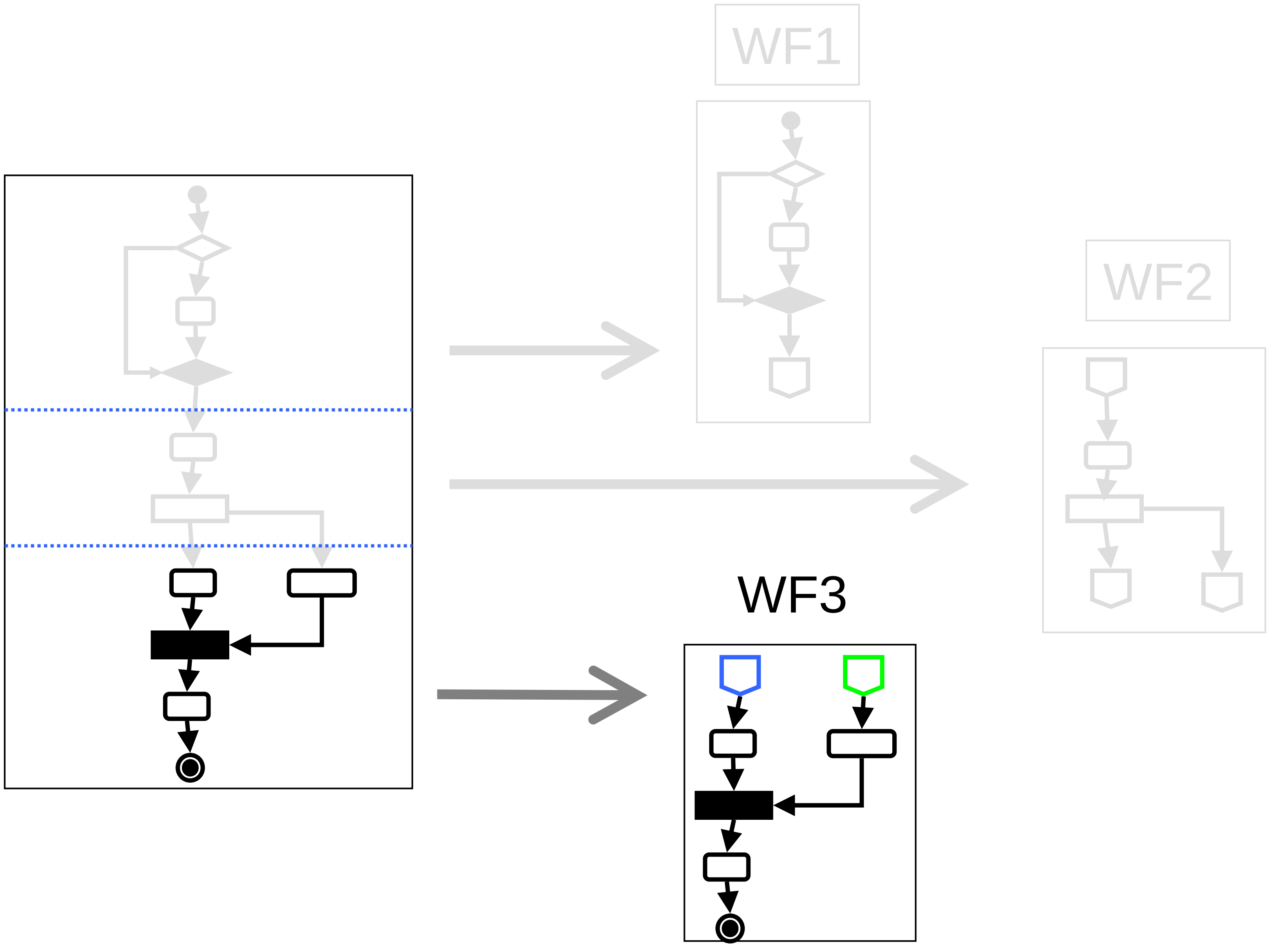


- Verification of Correctness
  - Graph Reductions
    (Sadiq and Orlowska 1998, 2000)
  - WF-nets (van der Aalst 1997, 1998)
  - Grobal-Local Correctness
    (Kindler, Martens and Reisig 2000)
  - Woflan (Verbeek, Basten and van der Aalst 2001)
  - Improvement of Sadiq-Orlowska’s works
    (Lin, Zhao, Li and Chen 2002)
  - Standard Workflow Models (Kiepuszewski, ter Hofstede and van der Aalst 2003)
  - EPCs (Dehnert and van der Aalst 2004～2006)

Why do we consider Multiple starts/ends?



Why do we consider Multiple starts/ends?



- Our purpose
  - Extend the concept of correctness over workflows with one start and end to that over workflows with multiple starts and/or end.
  - Verification Algorithms of the extended crrectness of given workflows.
  - Imprement of the algorithms to develop design assistant system of workflow.

- Our purpose
  - **Extend the concept of correctness over workflows with one start and end to that over workflows with multiple starts and/or end.**
  - Verification Algorithms of the extended crrectness of given workflows.
  - Imprement of the algorithms to develop design assistant system of workflow.

- General Correctness
  - We define ``general correctness'', that is a generalized version of correctness of workflow to satisfies the following properties.

1. General correctness is a natural extension of correctness, thar is, for a workflow with one start, general correctness is the same as original one.
2. General correctness is preserved by the operation of connection and/or division of workflows.
3. General correctness assures the possibility for a workflow to be completed to a correct workflow.

- **start**(W): the set of starts in W
- **end**(W): the set of ends in W

Definition 0.1 For a workflow W, an intermediary graph of W denotes the minimal subgraph V of W that satisfies the following properties.

1. V contains just one start of W.
2. If V contains an XOR-split c, then V contains just one out-degree of c.
3. If V contains a node c other than XOR-split, then V contains all out-degrees.

- **IG**(W) denotes the set of intermediary graph of W.
- **IG**(W, s) the set of intermediary graph of W with start s.

Definition 0.2 (Sadiq and Orlowska 2000) Let W be a workflow with one start.

- An intermediary graph V of W is said to be deadlock free if, for every AND-join r in V, V contains all in-degrees of m.
- An intermediary graph V of W is said to be lack of synchronization free if, for every XOR-join m in V, V contains just one in-degree of m.

Definition 0.3 (Sadiq and Orlowska 2000) A workflow W with one start is said to be correct if every intermediary graph V of W is deadlock free and lack of synchronization free.

Definition 0.4 For a workflow W, a trace graph of W denotes a non-empty subgraph V of W that satisfies the following properties. Let n be a node in V.

1. If n is an XOR-split, then V contains just one out-degree of n as well as the in-degree of n.
2. If n is an XOR-join, then V contains just one in-degree of n as well as the out-degree of n.
3. Otherwise, V contains all in-degrees and all out-degrees of n.

- $\mathbf{TG}(W)$: the set of trace graphs of W
- $\mathbf{TG}(W, S)$: the set of trace graphs V of W with $\mathbf{start}(V) = S$

Definition 0.5 For a workflow W and $U_1, U_2 \in \mathbf{IG}(W)$, U_1 and U_2 are said to conflict on an XOR-split c if U_1 and U_2 share c but they do not share any out-degree of c.

Definition 0.6 Let W be a workflow, $\mathbf{U}$ a set of intermediary graphs of W and n an XOR-split. Then, $\mathbf{U}$ is said to conflict on n there exists a pair (U_i, U_j) on $\mathbf{U}$ that conflicts on n.

Definition 0.7 Let W be a workflow and $W_1. \ldots, W_n \in \mathbf{TG}(W)$ with $W_i \cap W_j = \emptyset$ for each $i \neq j$. Then, the non-connected graph $W_1 \cup \cdots \cup W_n$ is called a summation of trace graphs.

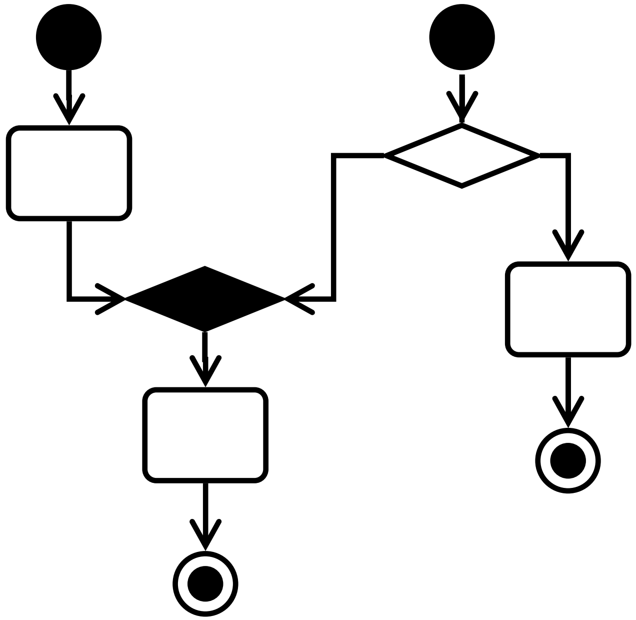
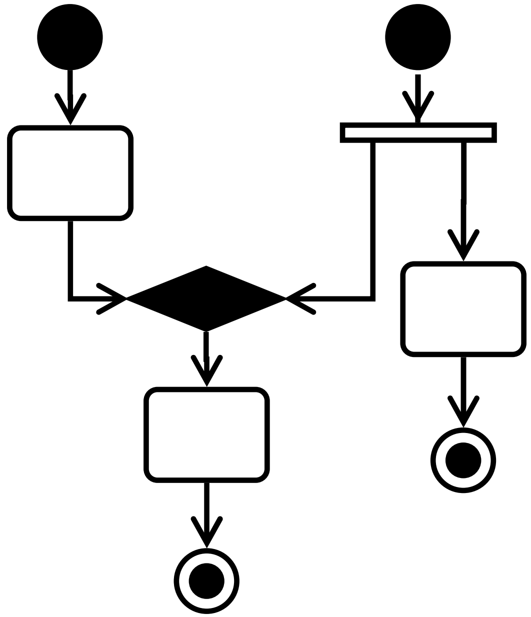
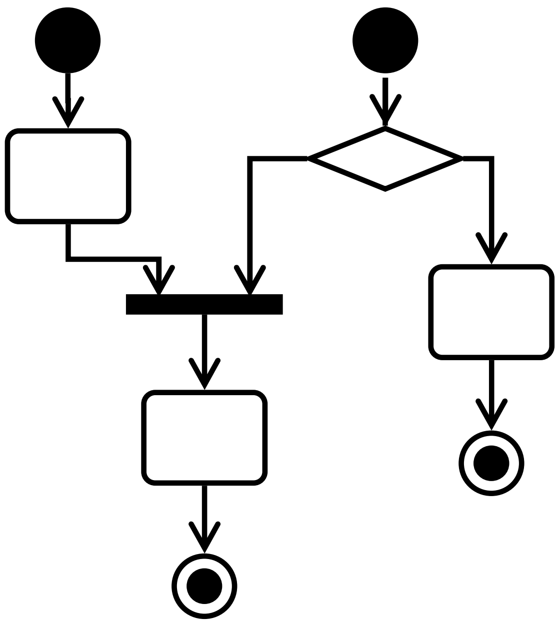
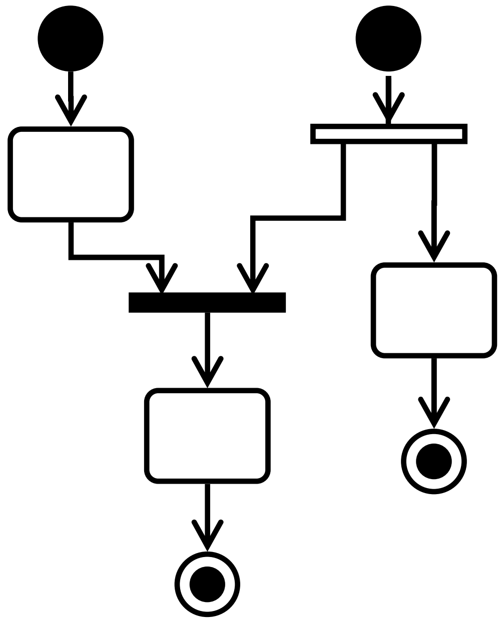
$\mathbf{TG}_s(W, S)$: the set of summations $W_1 \cup \cdots \cup W_n$ with

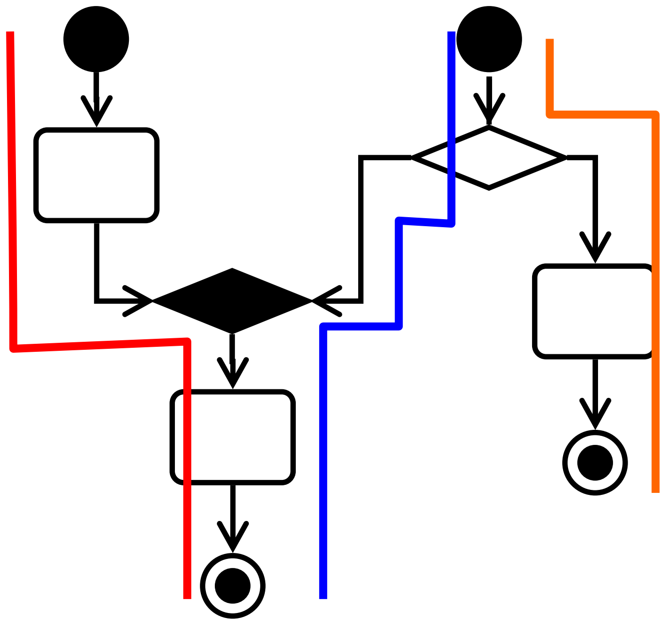
$$S = \mathbf{end}(W_1) \cup \cdots \cup \mathbf{end}(W_n).$$

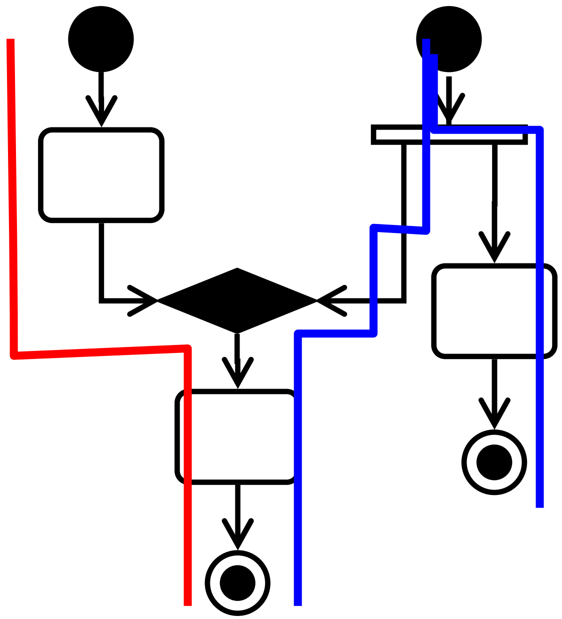
Definition 0.8 Let W be a workflow and S a set of starts $s_1, \ldots, s_n$ of W. Then, S is called an import of W if, for every set consisting of $U_i \in \mathbf{IG}(W, s_i)$ $(i = 1, \ldots, n)$ that is not conflict on any XOR-split in W, there exists a summation $\mathbb{V}$ with $\mathbb{V} = U_1 \cup \cdots \cup U_n$.

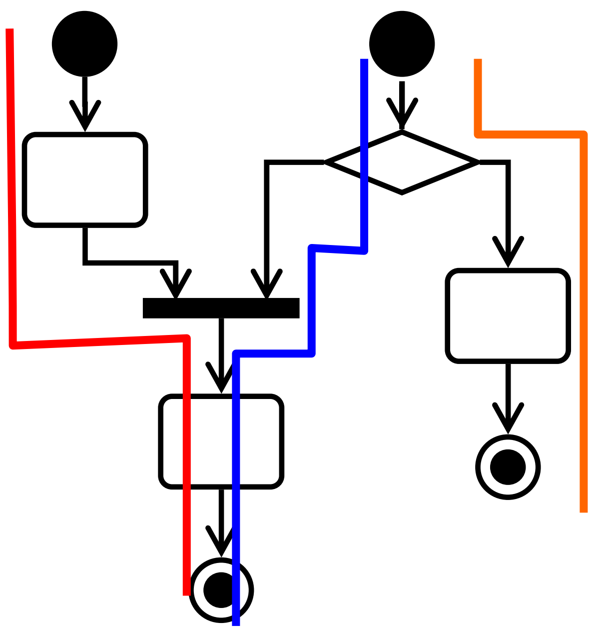
Definition 0.9 A workflow W is said to be generally correct if W has a set $\mathbb{I}$ of imports of W with $\bigcup_{I \in \mathbb{I}} I = \mathbf{start}(W)$.

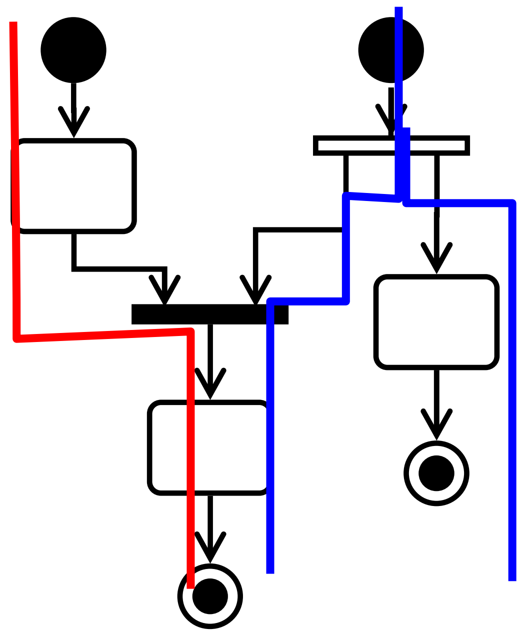
Definition 0.10 Let W be a workflow.
(1) For a summation $\mathbb{V}$ of trace graphs $V_1, \ldots, V_n$ of W, the export $\mathbf{ex}(\mathbb{V})$ of $\mathbb{V}$ denotes $\mathbf{end}(V_1) \cup \cdots \cup \mathbf{end}(V_n)$.
(2) For an import I of W, the set $\{\mathbf{ex}(\mathbb{V}) | \mathbb{V} \in \mathbf{TG}_s(W, I)\}$ is called by the export family of I and denoted by $\mathbb{E}_s(W, I)$.
(3) For an import family $\mathbb{I}$ of W, the set $\bigcup_{I \in \mathbb{I}} \mathbb{E}_s(W, I)$ is called by the export family of $\mathbb{I}$ and denoted by $\mathbb{E}^*_s(W, \mathbb{I})$.

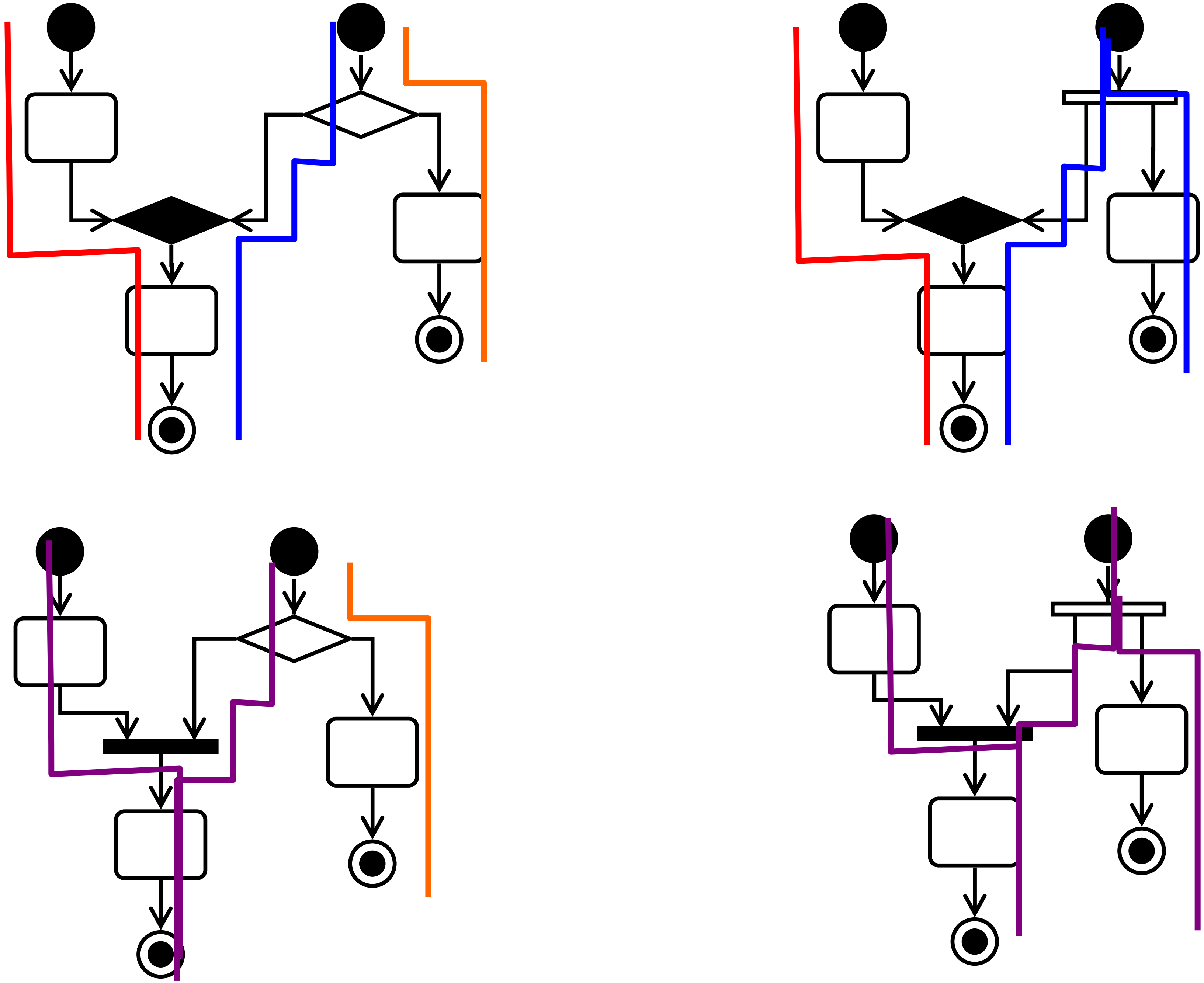





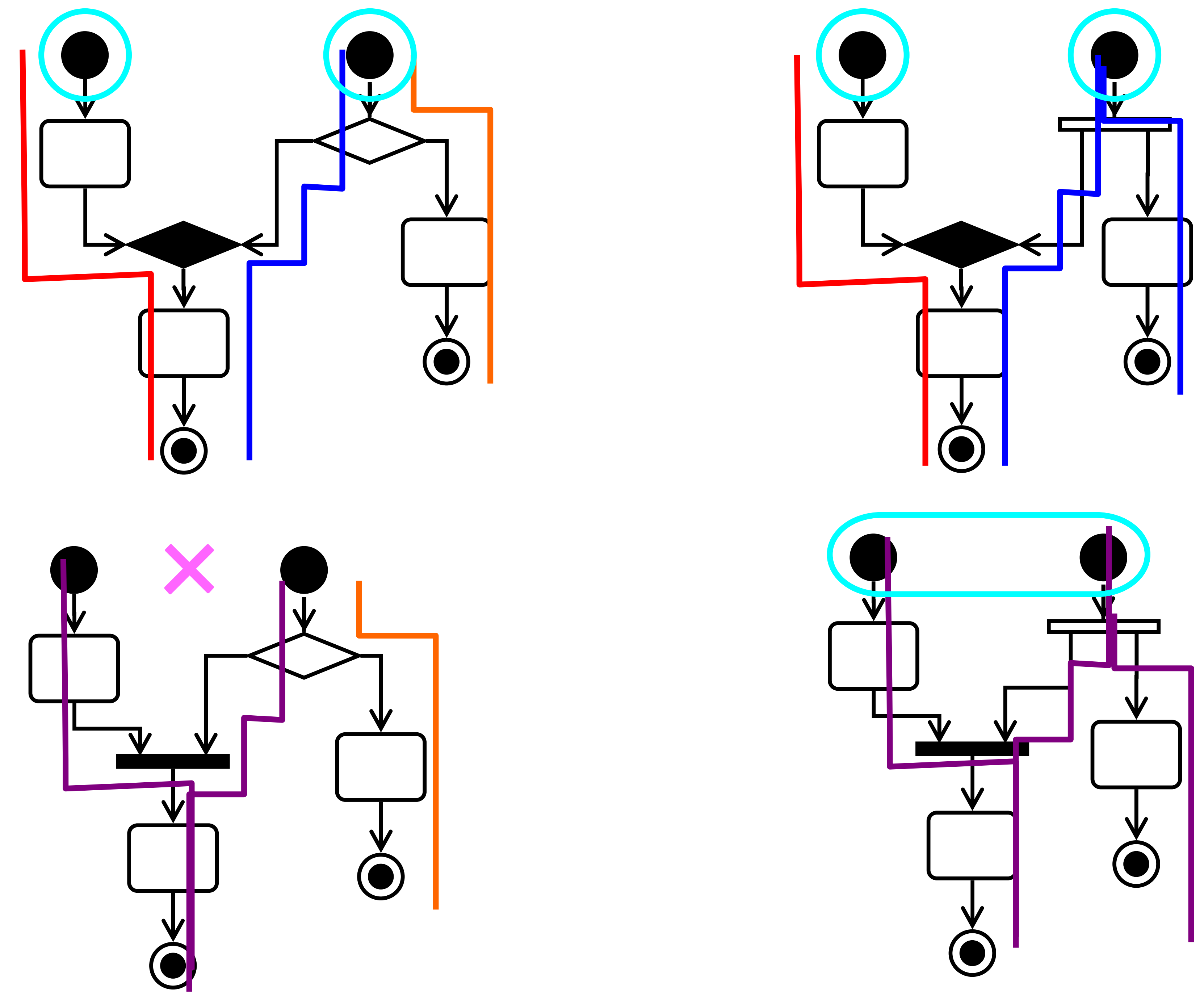


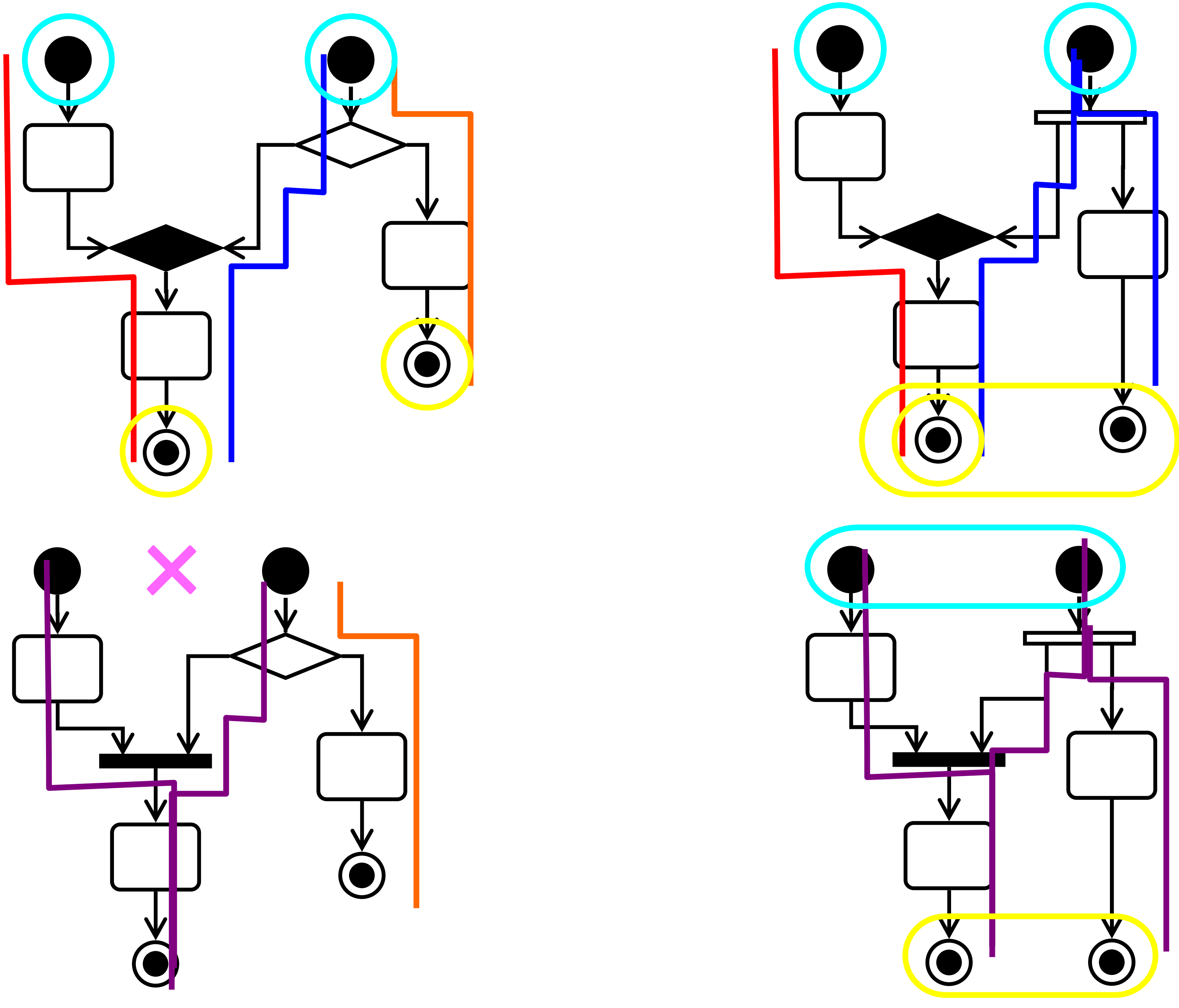












- $\mathbf{WF}(n, m)$: the set of workflows with n starts and m ends
- $\mathbf{WF} := \bigcup_{n,m} \mathbf{WF}(n, m)$.

Definition 0.14 Let $W_1 \in \mathbf{WF}(n, m)$, $W_2 \in \mathbf{WF}(m, l)$ and f a bijection from $\mathbf{end}(W_1)$ to $\mathbf{start}(W_2)$. Then, $W_1 *_f W_2$ denotes the workflow obtained from W_1 and W_2 by executing the following procedures.

1. Remove all ends of W_1 and their in-degrees.
2. Remove all starts in W_1 and their out-degrees.
3. For the source n of the in-degree of each end e in W_1 and the target n' of the out-degree of each start $f(e)$ in W_2, add the arc from n to n'.

In the remainder of this paper, we omit "f" in $W_1 *_f W_2$ and identify each $e \in \mathbf{end}(W_1)$ with $f(e) \in \mathbf{start}(W_2)$.

Theorem 0.15 Let $W_1 \in \mathbf{WF}(n, m)$ and $W_2 \in \mathbf{WF}(m, l)$. Then, $W_1 * W_2$ is generally correct for an import family $\mathbb{I}$ if and only if

- $W_1 \in \mathbf{WF}(n, m)$ is generally correct for $\mathbb{I}$, and
- $W_2 \in \mathbf{WF}(m, l)$ is generally correct for $\mathbb{E}^*_s(W, \mathbb{I})$.

Definition 0.16 For a workflow $W \in \mathbf{WF}(n,m)$, W is said to be extendible if there exists a workflow $W_0 \in \mathbf{WF}(1,n)$ such that $W_0 * W$ is correct.

Lemma 0.17 For every finite set S with $\sharp S = n > 0$ and every subset $\mathbb{S}$ of the power set of S, there exists a (generally) correct workflow $W \in WF(1,n)$ with $\mathbb{S} = \mathbb{E}_s^*(W, \{\mathbf{start}(W)\})$.

Corollary 0.18 Let $W \in \mathbf{WF}(n,m)$ be a generally correct workflow. Then, W is extendible if and only if W is generally correct.

- Conclusion
  - We extend the concept of correctness over workflows with one start and end to that over workflows with multiple starts and/or end.

1. General correctness is a natural extension of correctness, thar is, for a workflow with one start, general correctness is the same as original one. (Theorem 0.13)
2. General correctness is preserved by the operation of connection and/or division of workflows. (Theorem 0.15)
3. General correctness assures the possibility for a workflow to be completed to a correct workflow. (Corollary 0.18)