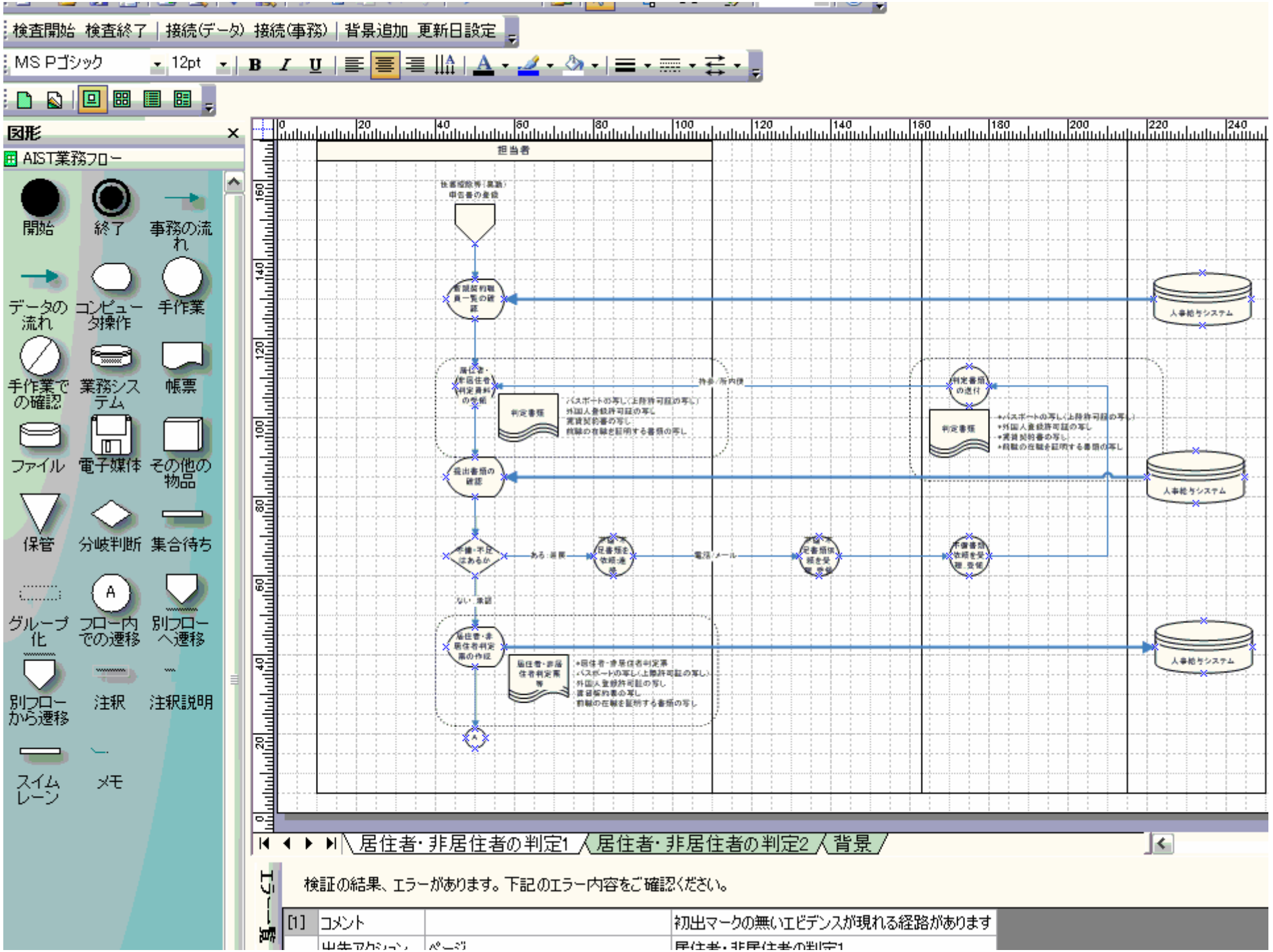
業務フロー図の検証

高木理，　清野貴博，　竹内泉，
和泉憲明，　高橋孝一

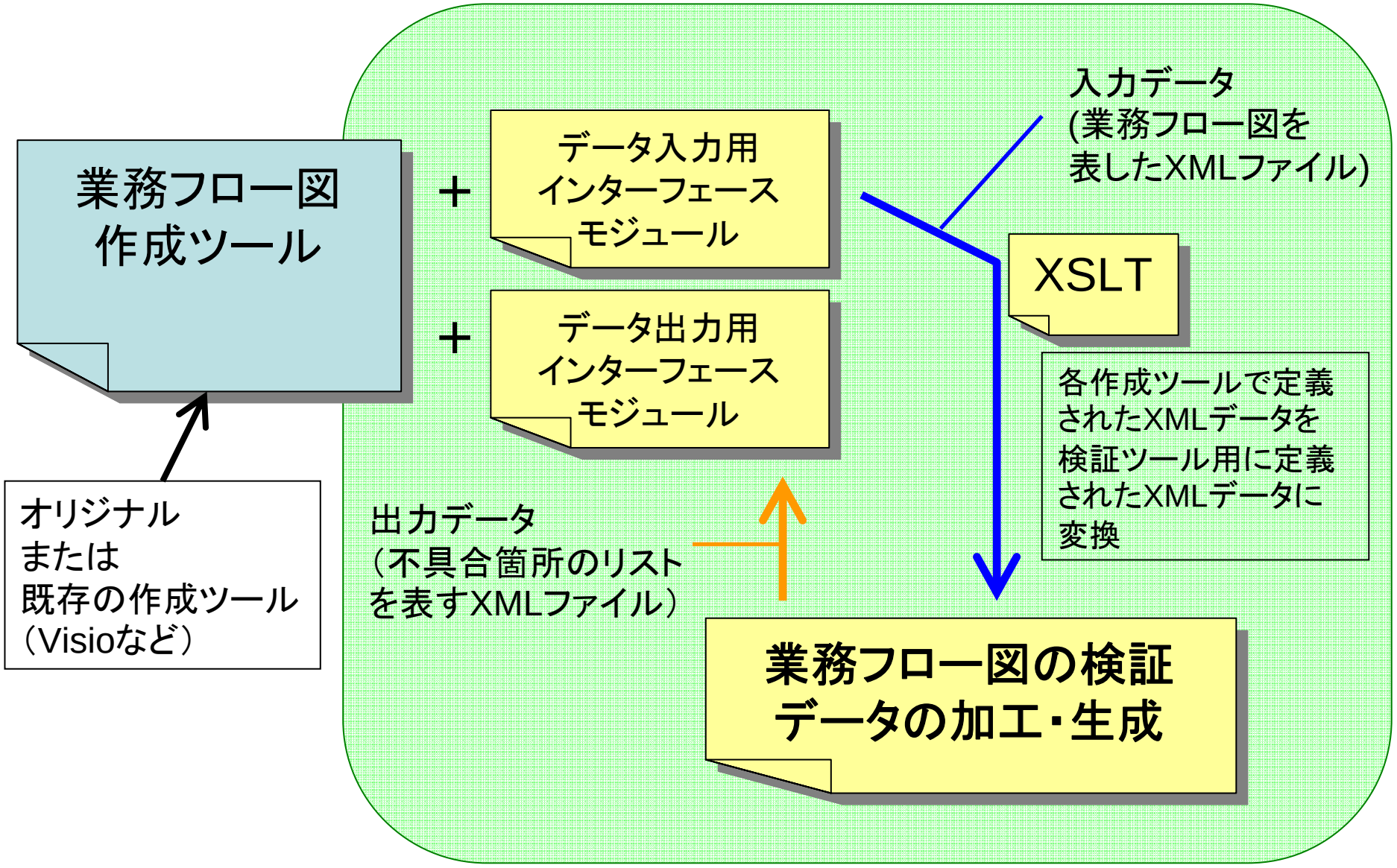
独立行政法人 産業技術総合研究所
（システム検証研究センター ＆ 情報技術研究部門）

背景

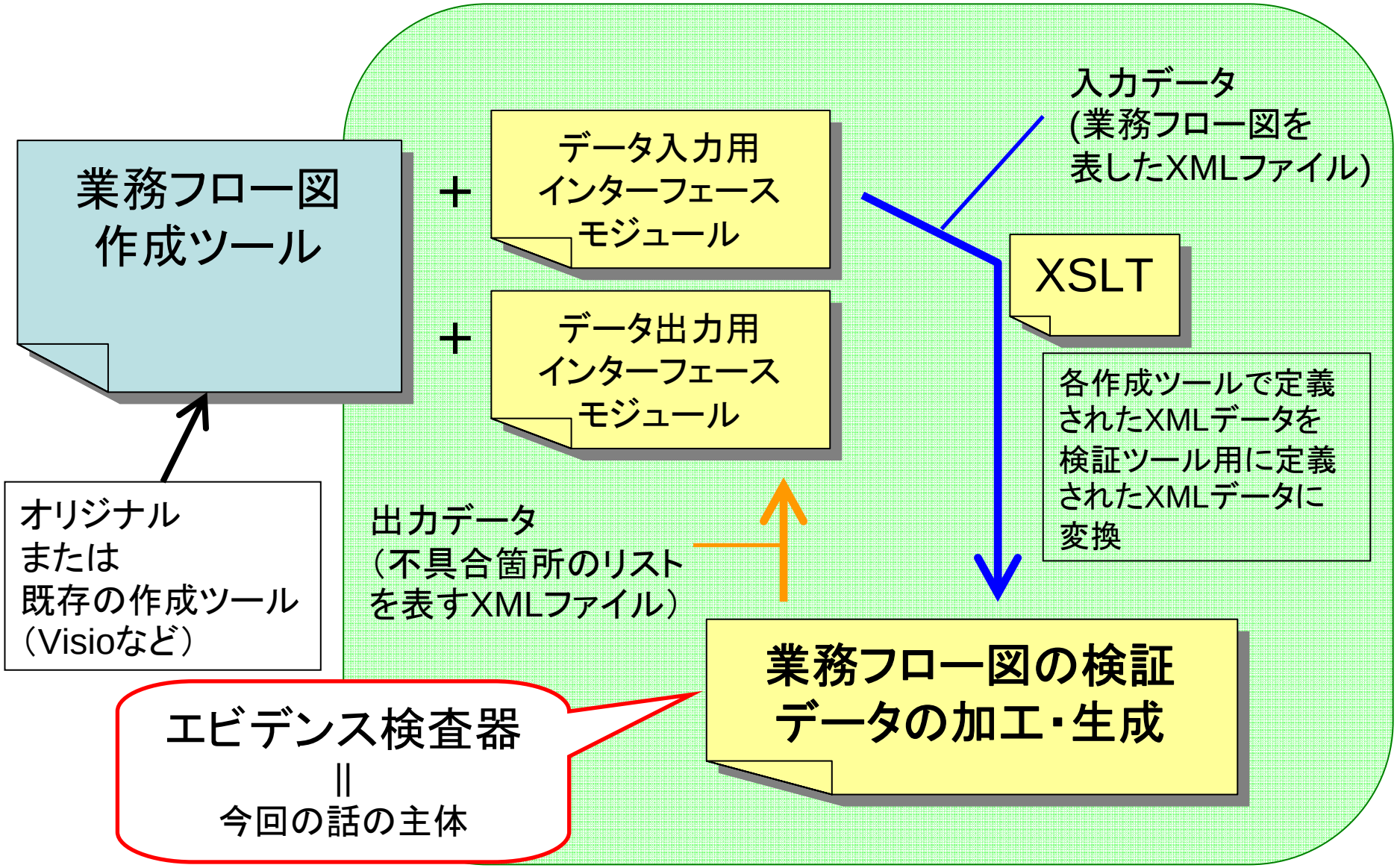
- EAI2: 産総研の大規模情報システム
  - 開発ベンダーに依存しない包括フレームワーク
  - 逐次的・全体的な最適化
- 本研究グループの役割
  - EAI2で用いられるダイアグラム（業務フロー図・画面遷移図・データモデル図等）の検証手法の開発
- AIST workflow verifier
  - EAI2の開発グループの協力のもと，本グループが開発を行ってきた業務フロー図の検証システム
  - 主な機能の一つとして，エビデンス・ライフサイクルの検証機能が上げられる.


検査開始 検査終了 | 接続(データ) 接続(事務) | 背景追加 更新日設定
MS Pゴシック
12pt
図形
AIST業務フロー
開始
終了
事務の流れ
データの流れ
コンピュータ操作
手作業
手作業での確認
業務システム
帳票
ファイル
電子媒体
その他の物品
保管
分岐判断
集合待ち
グループ化
フロー内での遷移
別フローへ遷移
別フローから遷移
注釈
注釈説明
スイムレーン
メモ
担当者
人事給与システム
判定書類
提出書類の確認
ある：差戻
電話/メール
ない：承認
居住者・非居住者判定票
パスポートの写し(上陸許可証の写し)
外国人登録許可証の写し
雇用契約書の写し
前職の在職を証明する書類の写し
居住者・非居住者の判定1
居住者・非居住者の判定2
背景
検証の結果、エラーがあります。下記のエラー内容をご確認ください。
[1] コメント
初出マークの無いエビデンスが現れる経路があります
居住者・非居住者の判定1

AIST Workflow Verifier



AIST Workflow Verifier



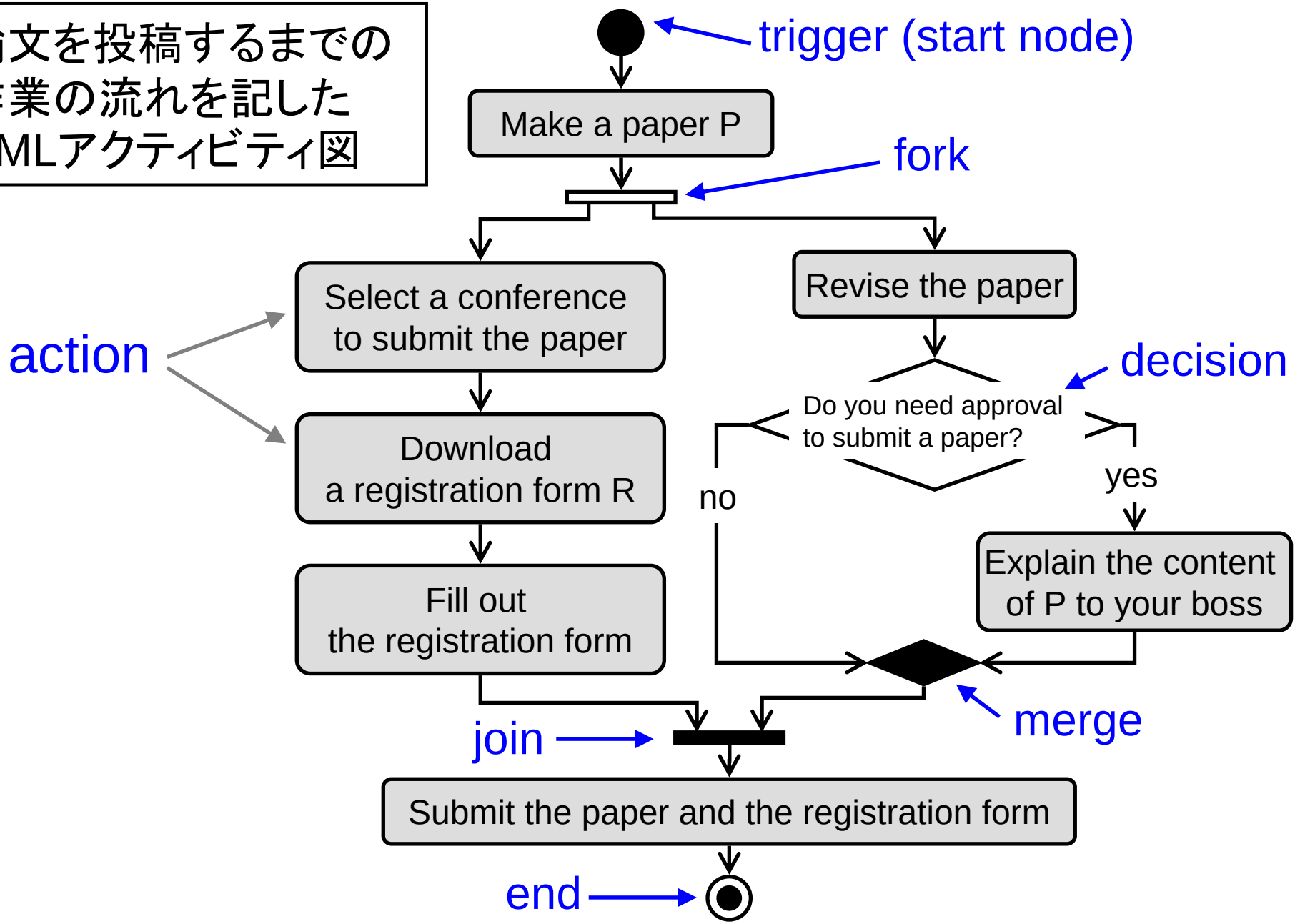
- これ以降においてお話しする内容

  - 背景
    - なぜエビデンス･ライフサイクルの整合性を検証するのか？
    - ELAD
  - エビデンス検査器の構成
  - 適用実験

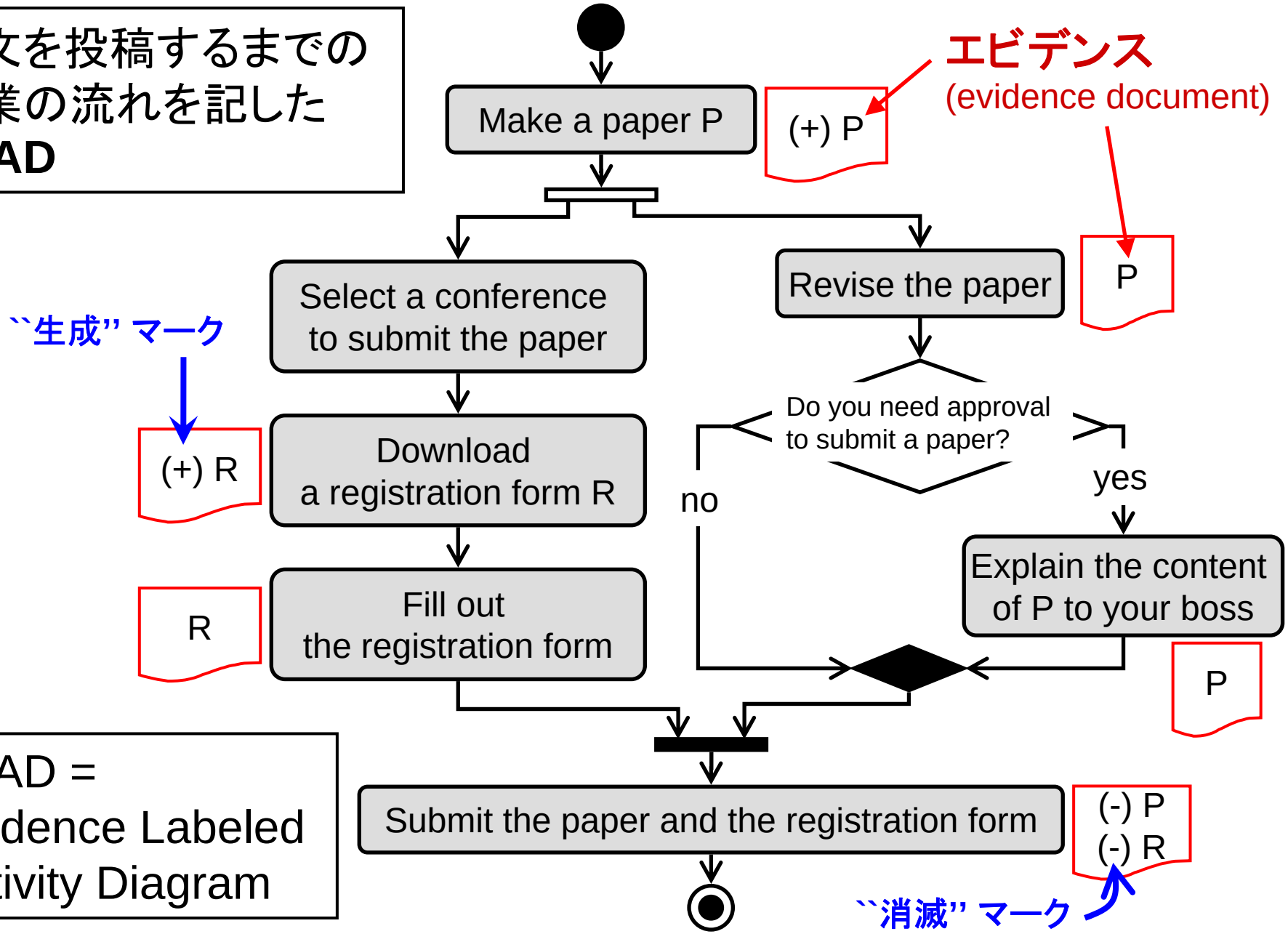
- これ以降においてお話しする内容

  - **背景**
    - **なぜエビデンス･ライフサイクルの整合性を検証するのか？**
    - **ELAD**
  - エビデンス検査器の構成
  - 適用実験

論文を投稿するまでの
作業の流れを記した
UMLアクティビティ図

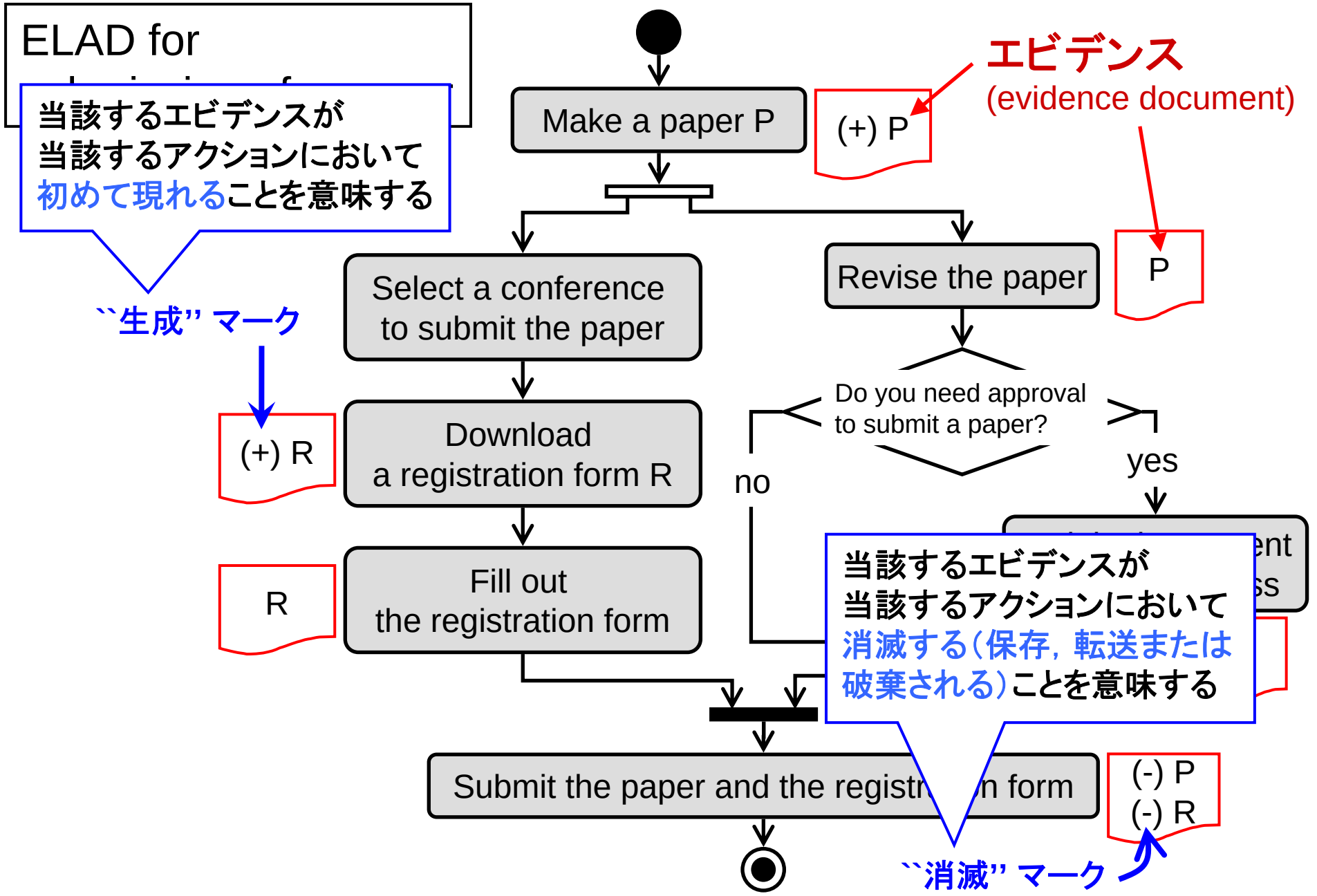


論文を投稿するまでの作業の流れを記した **ELAD**



ELAD =
Evidence Labeled
Activity Diagram

ELAD for


当該するエビデンスが
当該するアクションにおいて
初めて現れることを意味する
``生成'' マーク
Make a paper P
(+) P
エビデンス
(evidence document)
Select a conference
to submit the paper
Revise the paper
P
(+) R
Download
a registration form R
Do you need approval
to submit a paper?
no
yes
R
Fill out
the registration form
当該するエビデンスが
当該するアクションにおいて
消滅する(保存，転送または
破棄される)ことを意味する
Submit the paper and the regist n form
(-) P
(-) R
``消滅'' マーク

- ここでは　ELAD ≒ 業務フロー図　として議論を進める.

- 簡単のため，今回の講演では，ELADにおけるアクターやデータフローに関する記述は省略する.

なぜエビデンス検証か？（1）

Fact 1

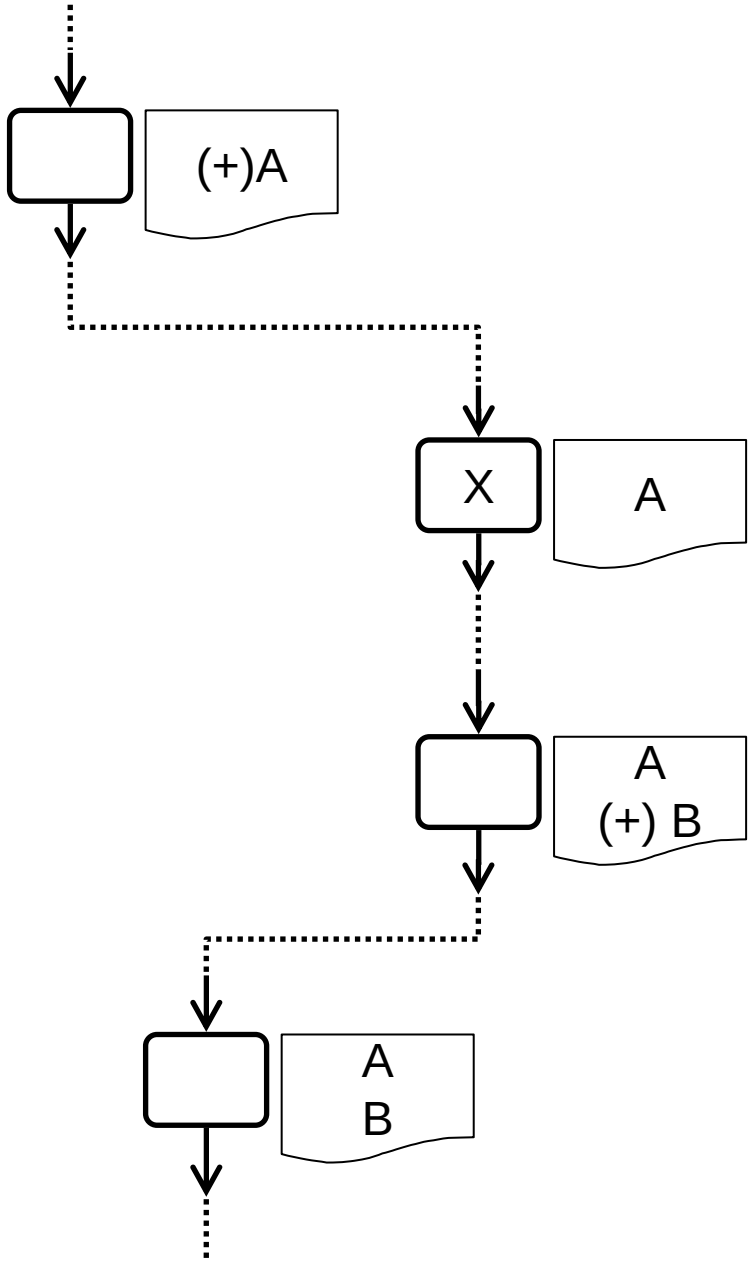
基幹業務系システム開発で用いられる業務フロー図において記述される作業のほとんどは，何らかのエビデンスまたはデータベース上のデータを加工する一連の作業に外ならない．

なぜエビデンス検証か？(2)

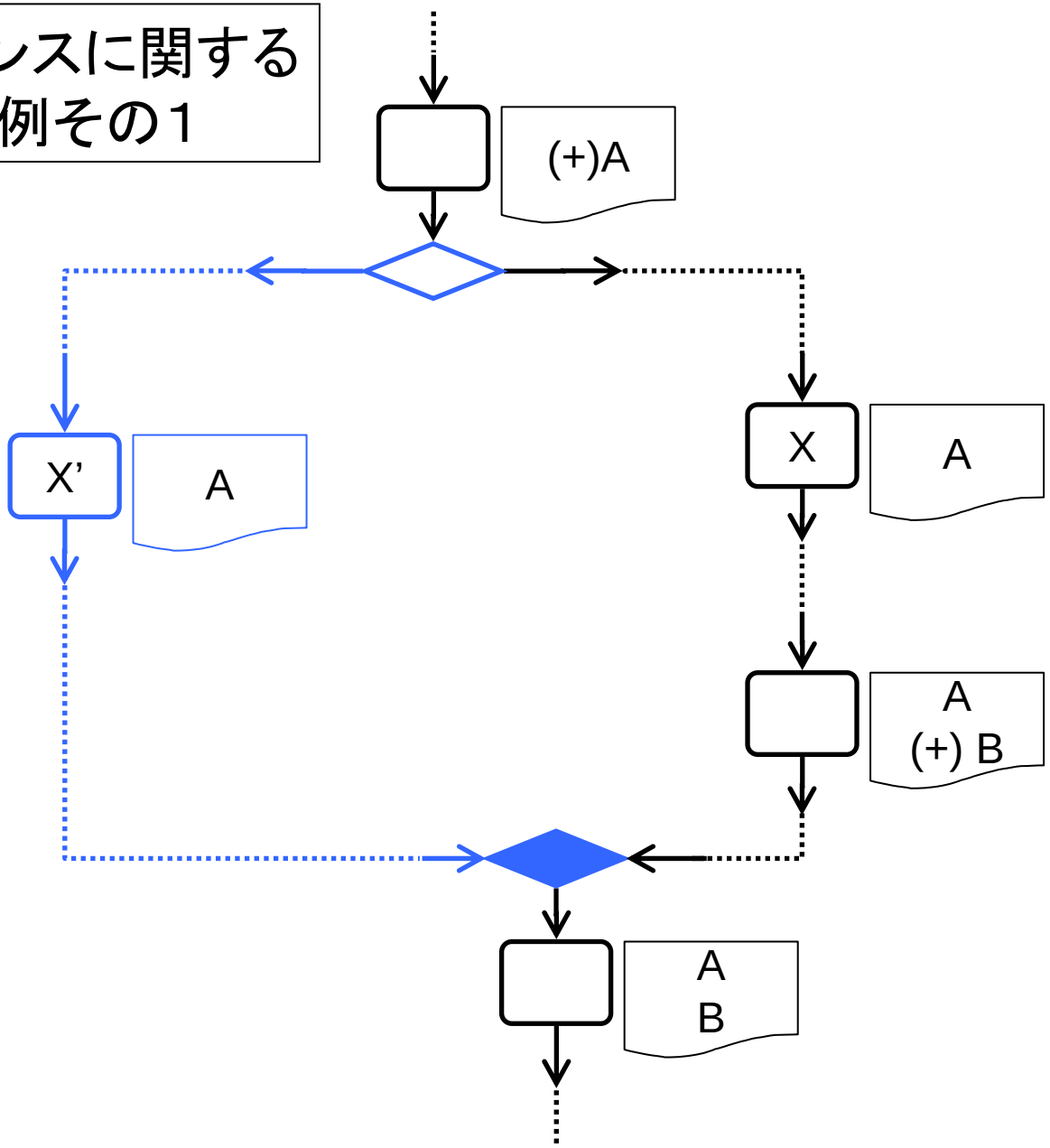
Fact 2

実際の基幹業務系システム開発において作成される業務フロー図の中には，エビデンスに関して曖昧な，あるいは整合的なない記述が多く見られる.

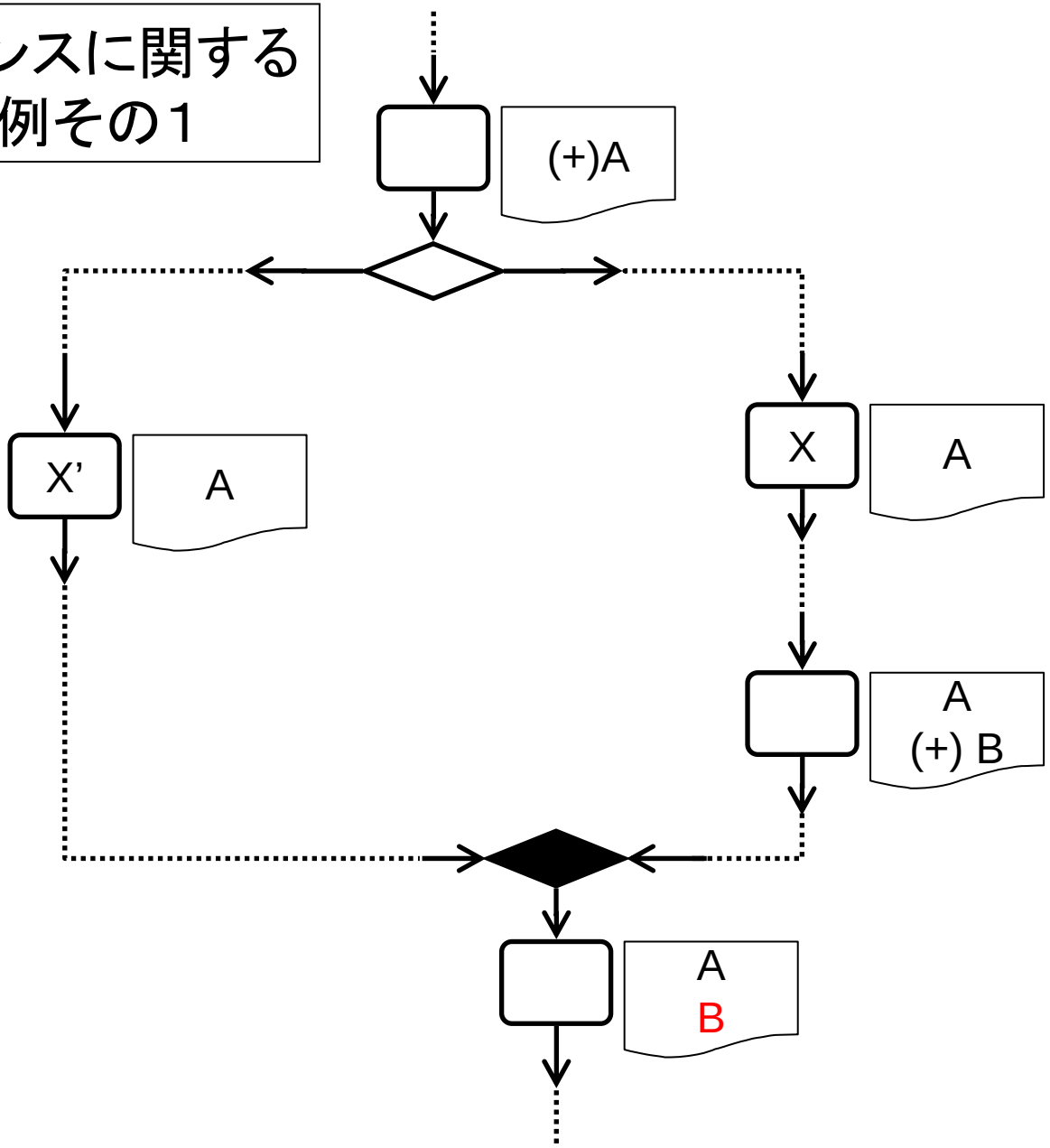
エビデンスに関するバグの例その1



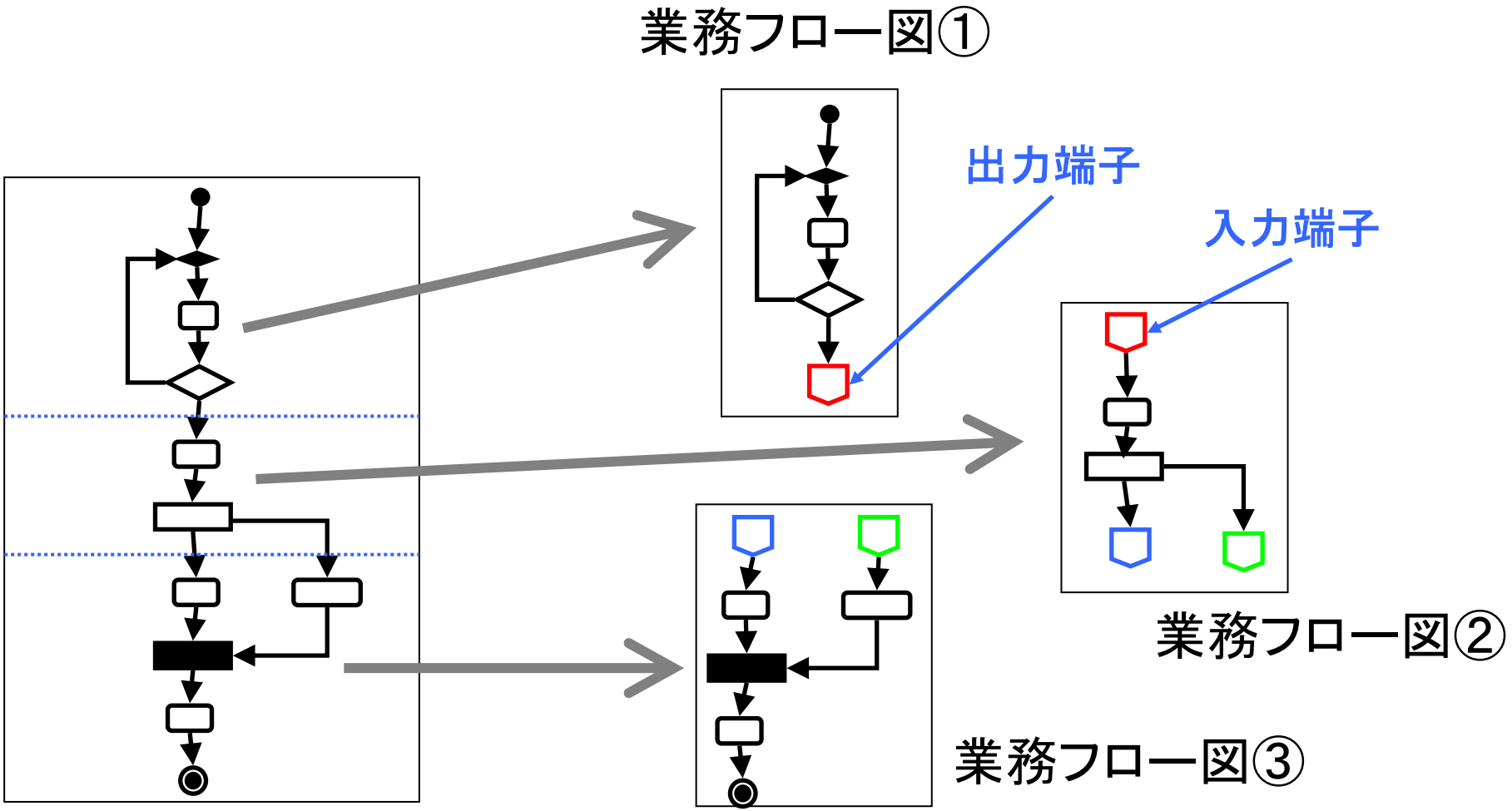
エビデンスに関するバグの例その1



エビデンスに関するバグの例その1

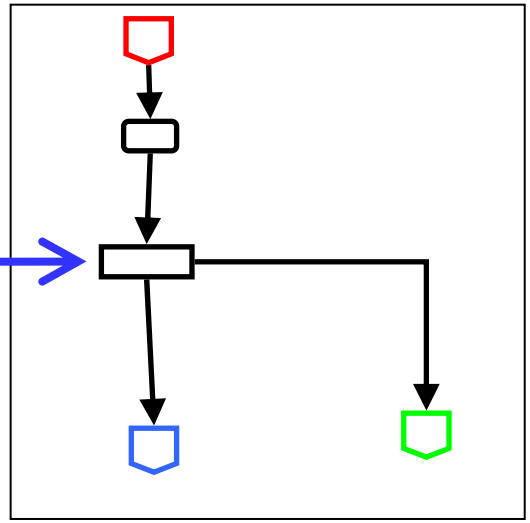


エビデンスに関するバグの例その2



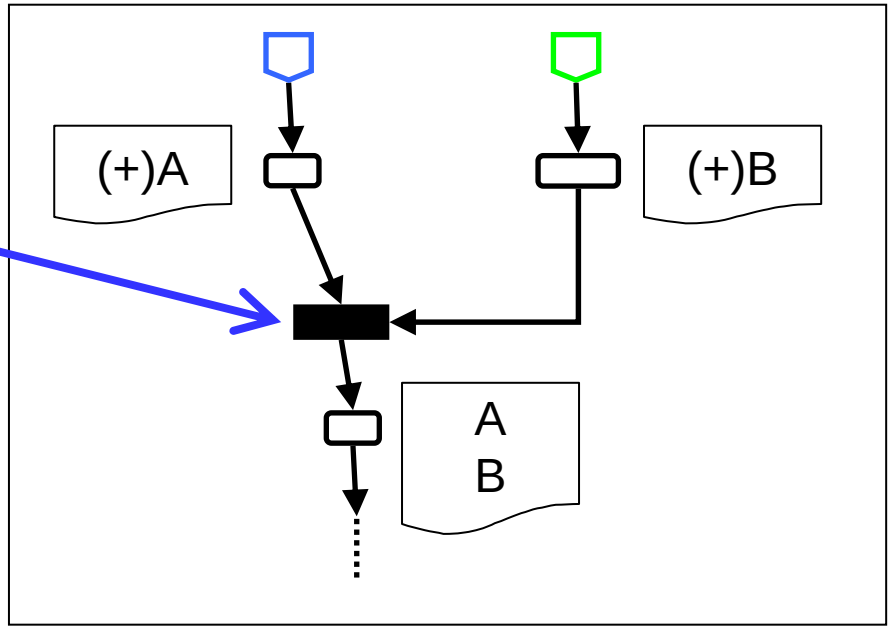
エビデンスに関するバグの例その2

フォーク（作業の並行分岐，分担）



業務フロー図②

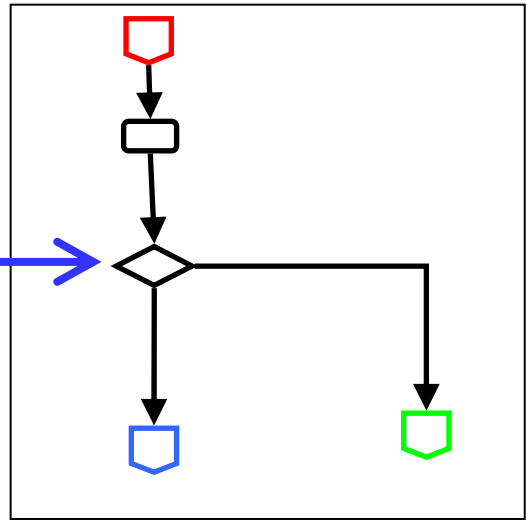
ジョイン（作業の待ち合わせ，合流）



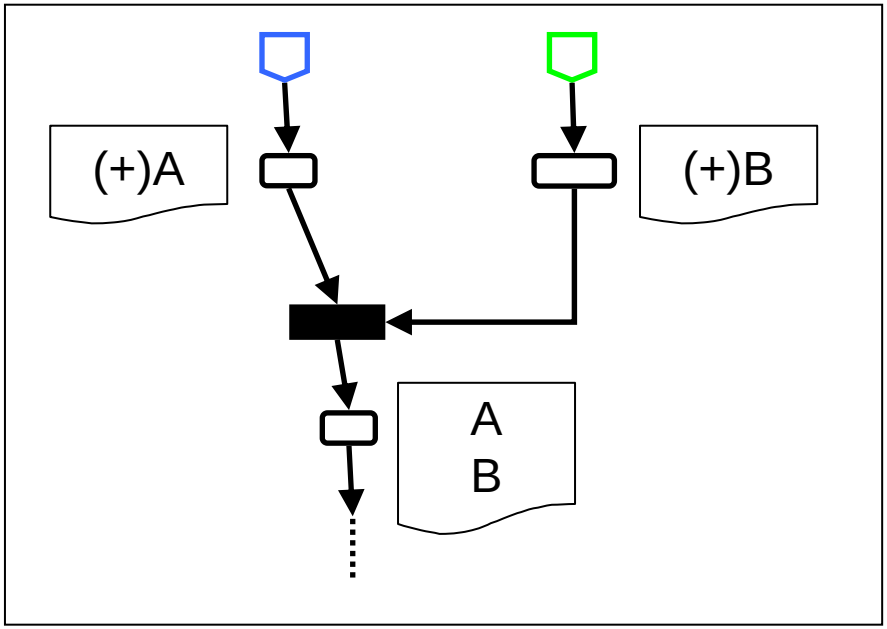
業務フロー図③

エビデンスに関するバグの例その2

ここをデシジョン（判断分岐）に変更すると…



業務フロー図②



業務フロー図③

なぜエビデンス検証か？(3)

観察

- 600以上の``リアルな''ワークフロー図を調査
  - 数多くのバグを発見
  - その内のいくつかは手作業では発見しにくい
  - バグの原因は以下の3種類に分類できる
    - ① エビデンスの記述そのものに関する単純なミス
    - ② ワークフロー図の構造が複雑化したため
    - ③ ワークフロー図の構造そのものの不整合
- ②や③を原因とするバグを見直すことによって，ワークフロー図そのものの不具合を訂正することが出来た．

- これ以降においてお話しする内容

  - 背景
    - なぜエビデンス･ライフサイクルの整合性を検証するのか？
    - ELAD
  - **エビデンス検査器の構成**
  - 適用実験

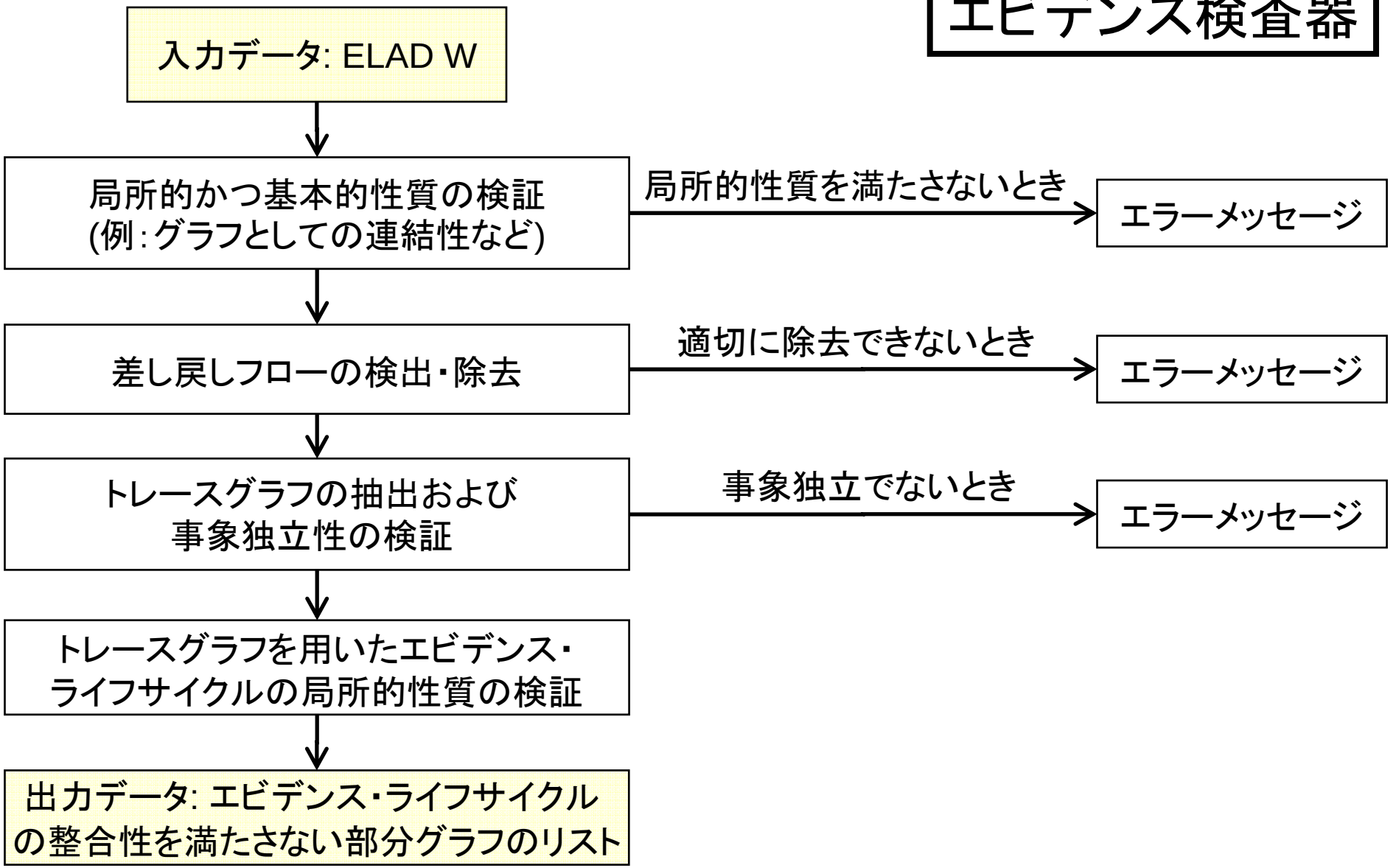
エビデンス検査器

入力データ: ELAD W

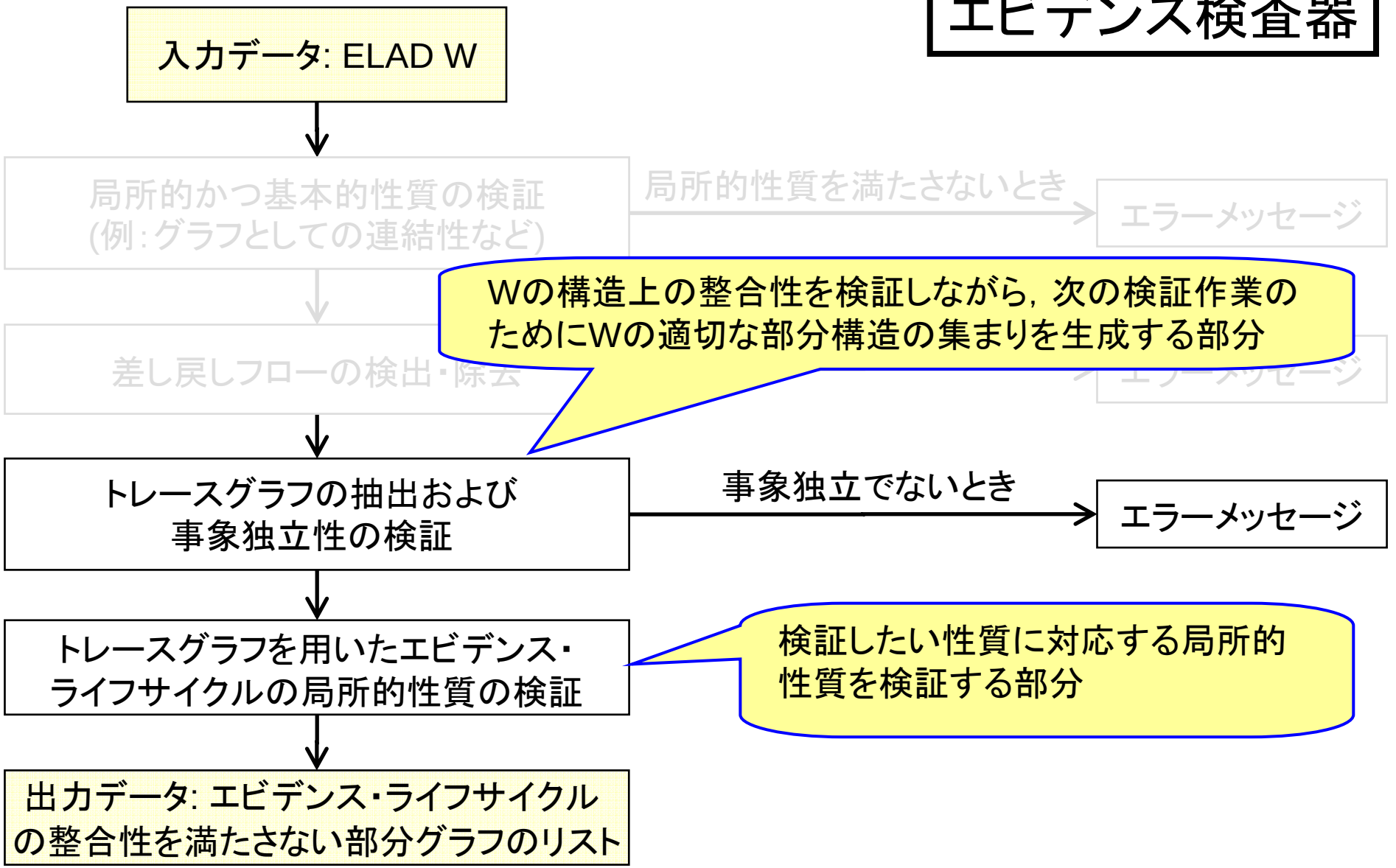
Wに含まれるすべてのエビデンスのライフサイクルの整合性を検証

出力データ: エビデンス・ライフサイクルの整合性を満たさない部分グラフのリスト

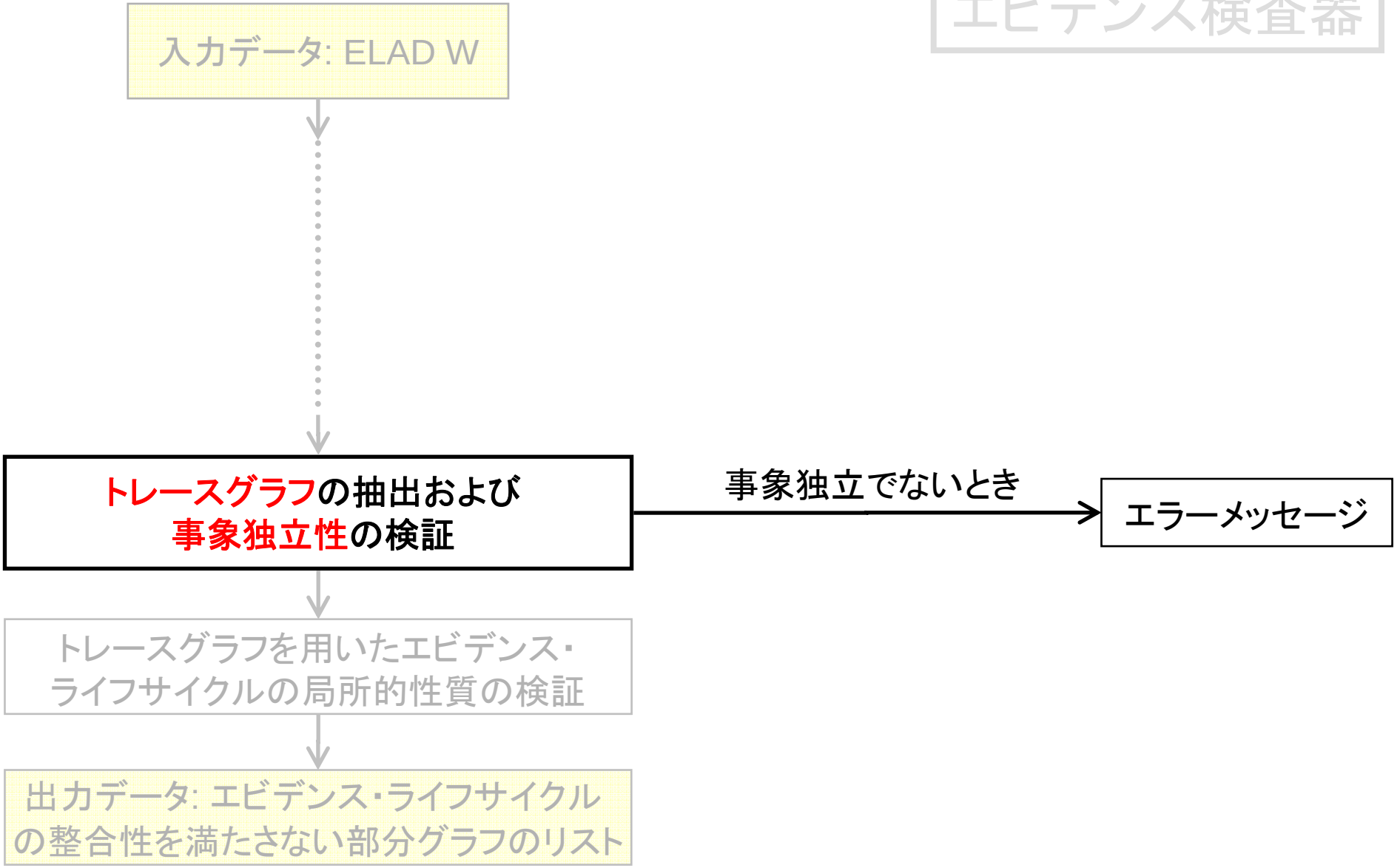
エビデンス検査器


入力データ: ELAD W
局所的かつ基本的性質の検証
(例:グラフとしての連結性など)
局所的性質を満たさないとき
エラーメッセージ
差し戻しフローの検出・除去
適切に除去できないとき
エラーメッセージ
トレースグラフの抽出および
事象独立性の検証
事象独立でないとき
エラーメッセージ
トレースグラフを用いたエビデンス・
ライフサイクルの局所的性質の検証
出力データ: エビデンス・ライフサイクル
の整合性を満たさない部分グラフのリスト

エビデンス検査器


入力データ: ELAD W
局所的かつ基本的性質の検証
(例:グラフとしての連結性など)
局所的性質を満たさないとき
エラーメッセージ
差し戻しフローの検出・除去
Wの構造上の整合性を検証しながら，次の検証作業のためにWの適切な部分構造の集まりを生成する部分
トレースグラフの抽出および
事象独立性の検証
事象独立でないとき
エラーメッセージ
トレースグラフを用いたエビデンス・
ライフサイクルの局所的性質の検証
検証したい性質に対応する局所的性質を検証する部分
出力データ: エビデンス・ライフサイクルの整合性を満たさない部分グラフのリスト

エビデンス検査器


入力データ: ELAD W
トレースグラフの抽出および
事象独立性の検証
事象独立でないとき
エラーメッセージ
トレースグラフを用いたエビデンス・
ライフサイクルの局所的性質の検証
出力データ: エビデンス・ライフサイクル
の整合性を満たさない部分グラフのリスト

注意：これ以降，簡単のため，ELADはすべて差し戻しフローを持たない（≒非循環）とする．

定義

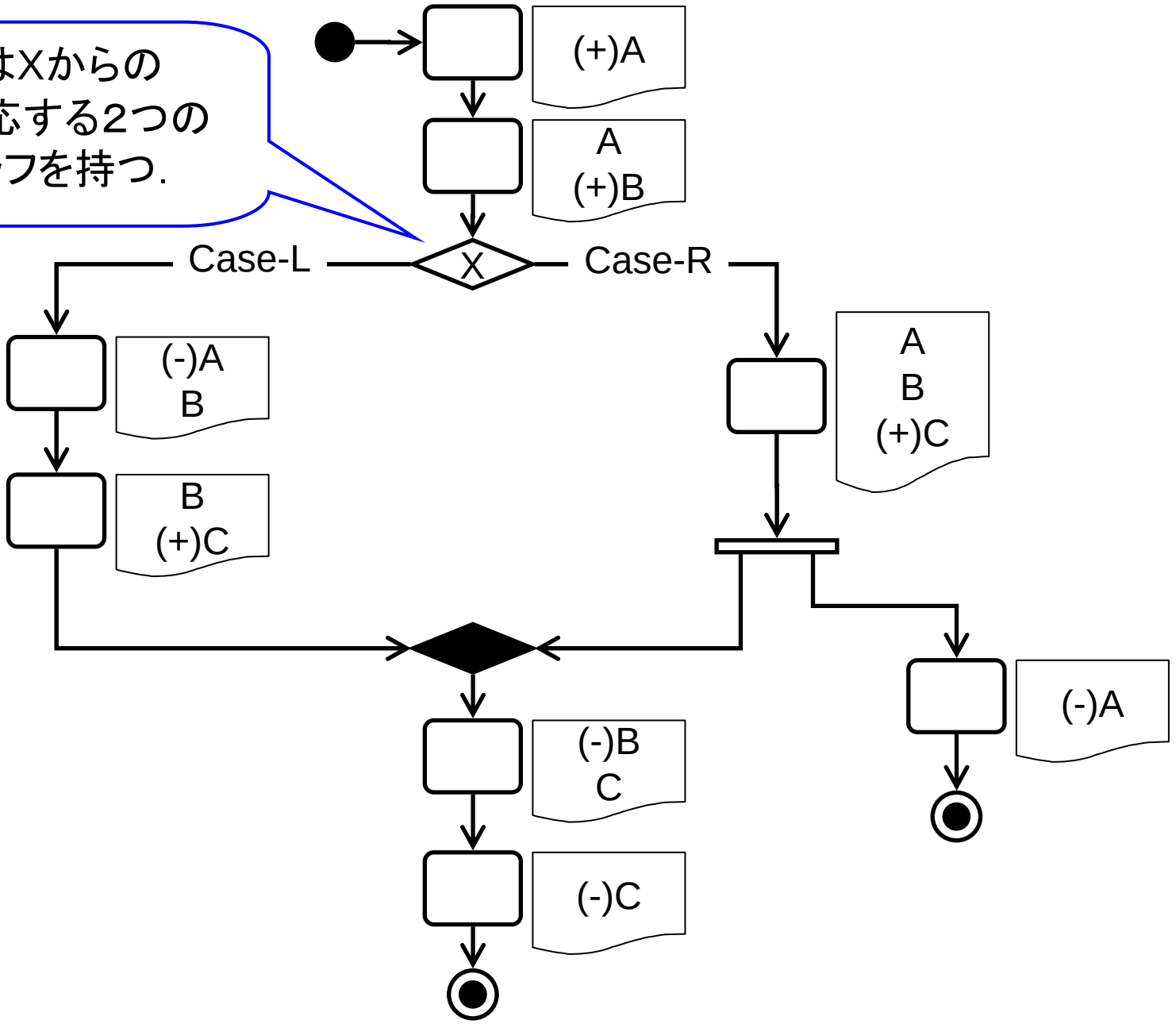
任意のELAD W について，以下の性質を満たす W の（空でない）部分グラフ V を W のトレースグラフと呼ぶ．

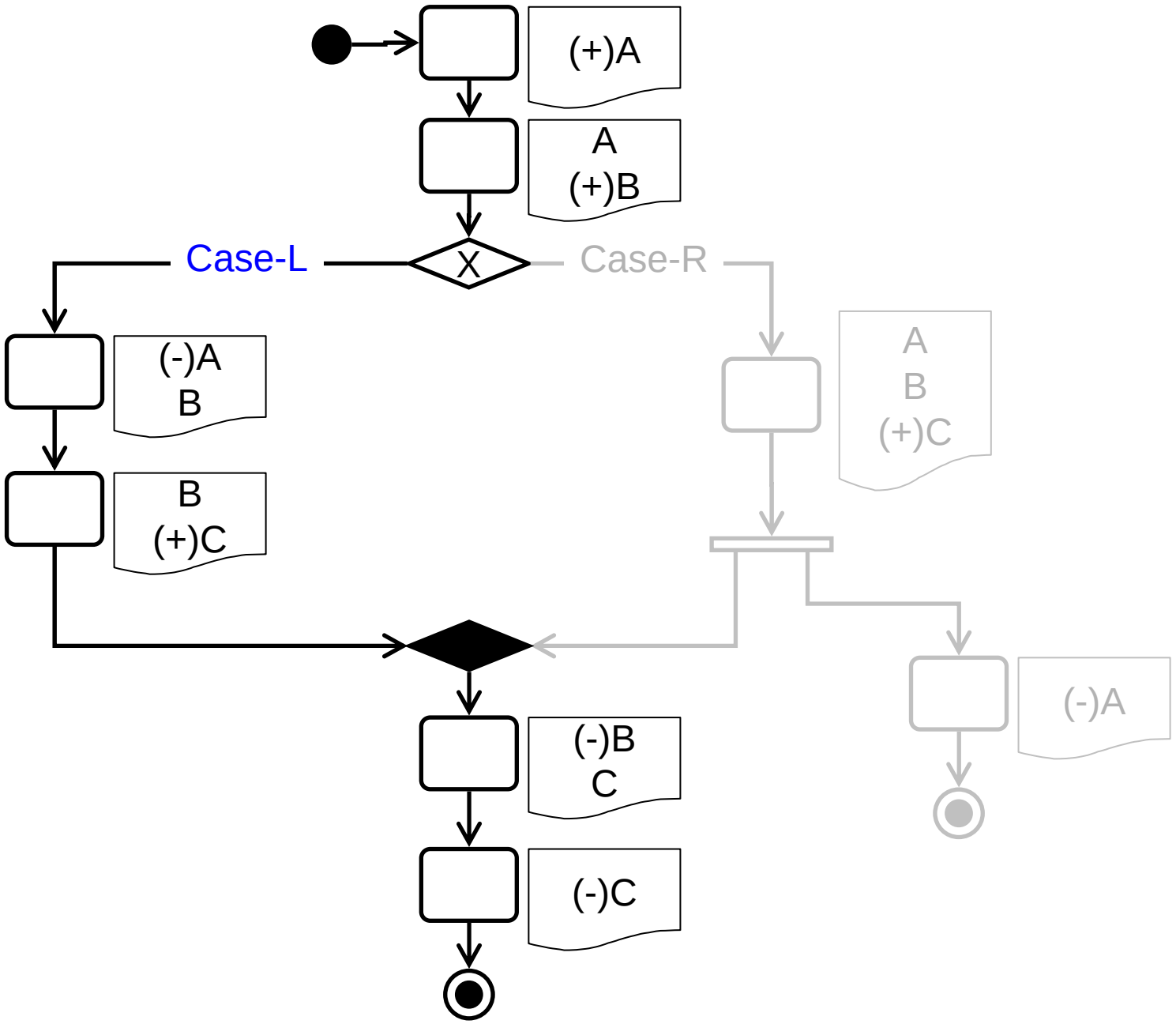
1. Vがデシジョンノード Dを含むならば，VはDからのフローを唯1つ含み，Dへのフローをすべて含む．
2. Vがマージノード Mを含むならば，VはMからのフローをすべて含み，Mへのフローを唯1つ含む．
3. Vが上記以外のノード Nを含むならば，VはNからのフローとNへのフローをすべて含む．

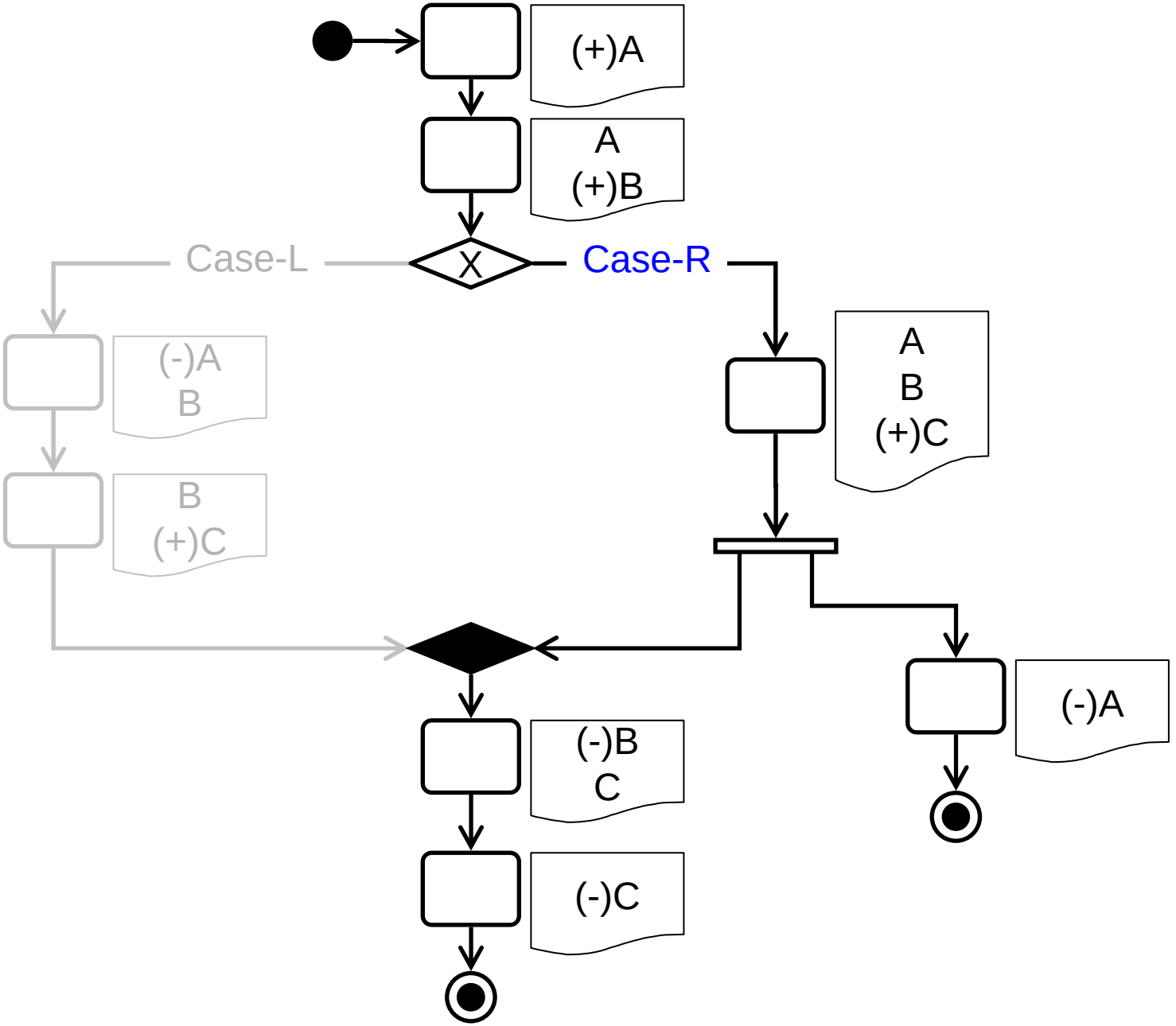
定義

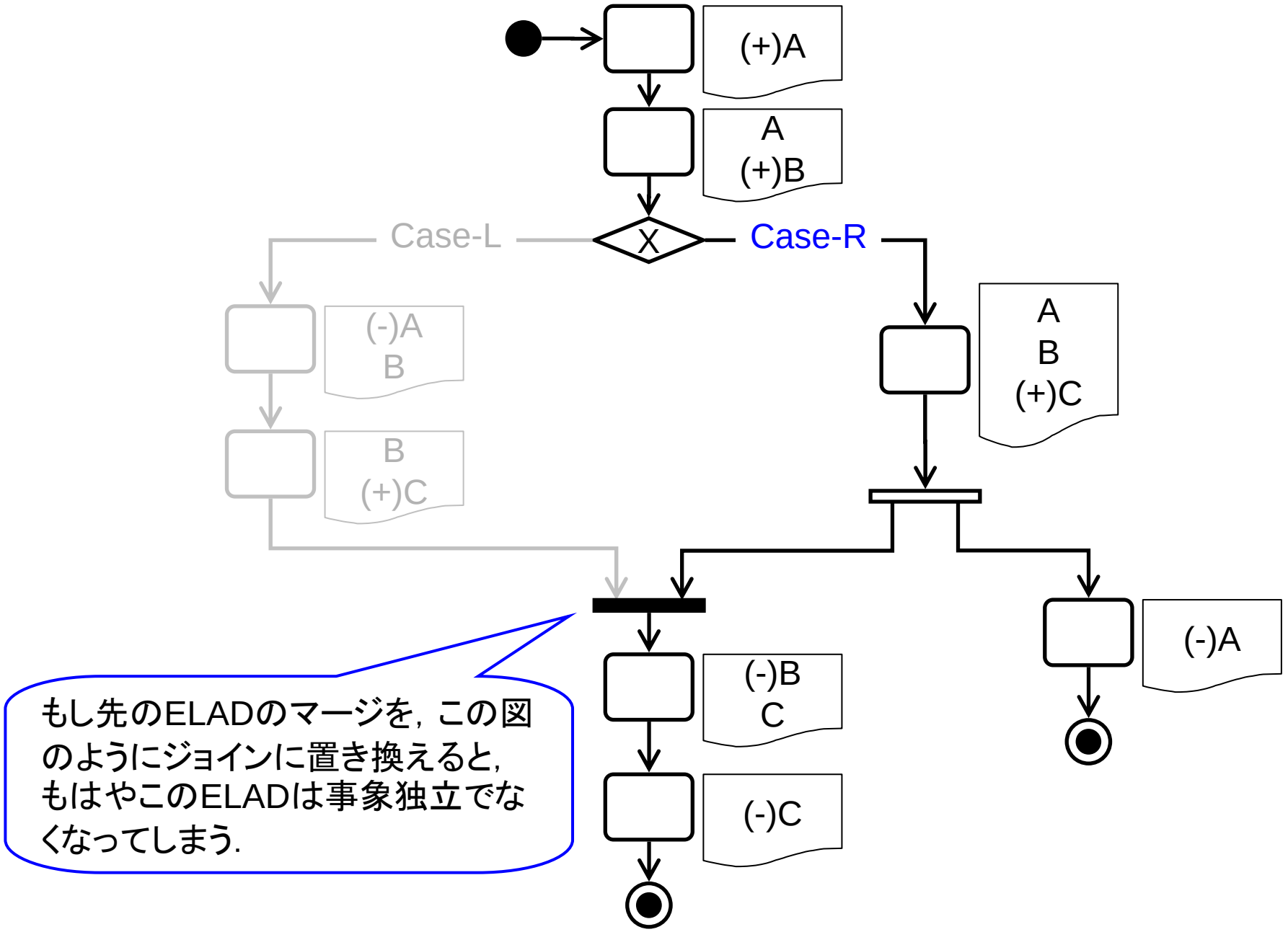
ELAD W が*事象独立*とは，W の任意のフロー f に対して，W のトレースグラフ V が存在して V が f を含むことを言う．

このELADはXからの
フローに対応する2つの
トレースグラフを持つ.

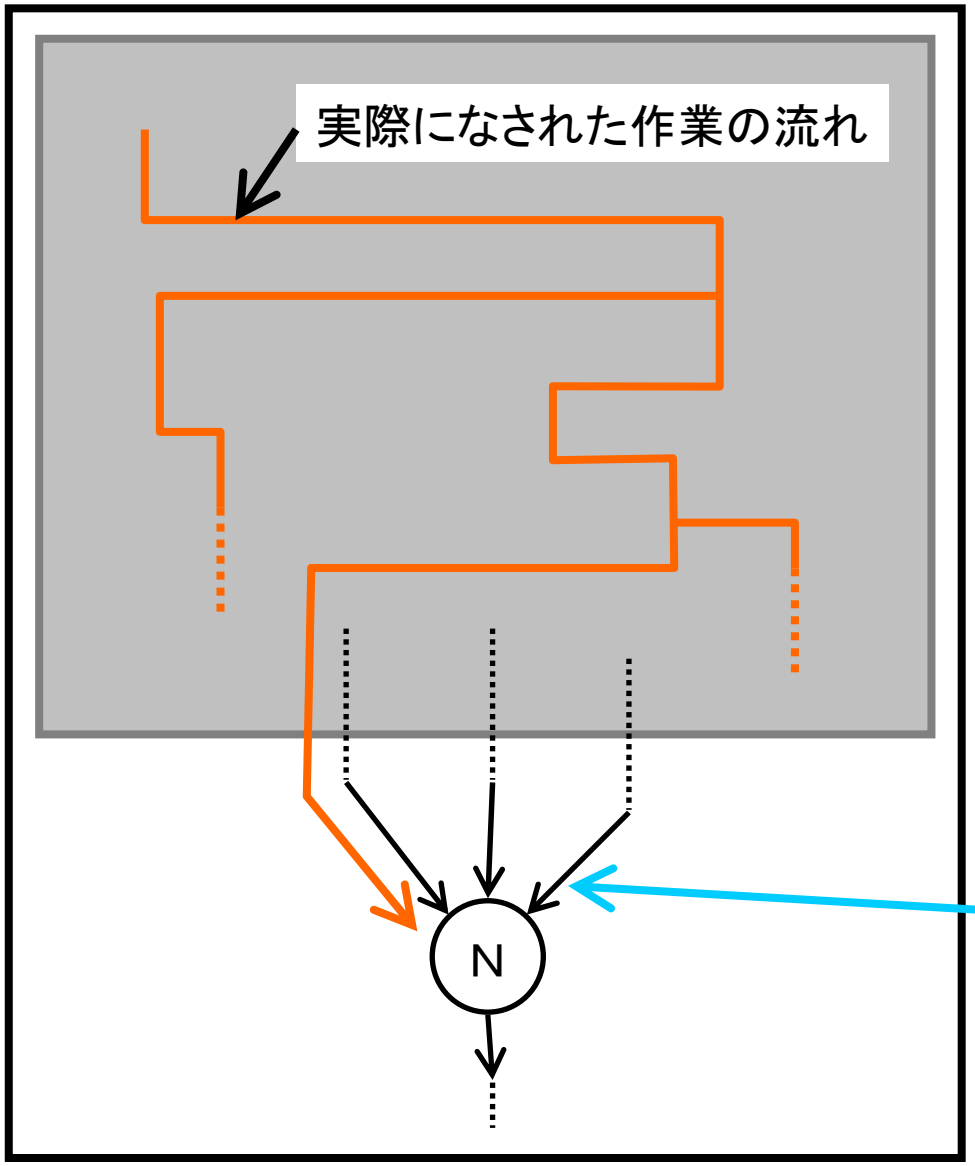



(+)A
A
(+)B
Case-L
X
Case-R
(-)A
B
B
(+)C
A
B
(+)C
(-)A
(-)B
C
(-)C


(+)A
A
(+)B
Case-L
X
Case-R
(-)A
B
B
(+)C
A
B
(+)C
(-)B
C
(-)C
(-)A


(+)A
A
(+)B
Case-L
X
Case-R
(-)A
B
B
(+)C
A
B
(+)C
(-)B
C
(-)C
(-)A
もし先のELADのマージを，この図のようにジョインに置き換えると，もはやこのELADは事象独立でなくなってしまう.

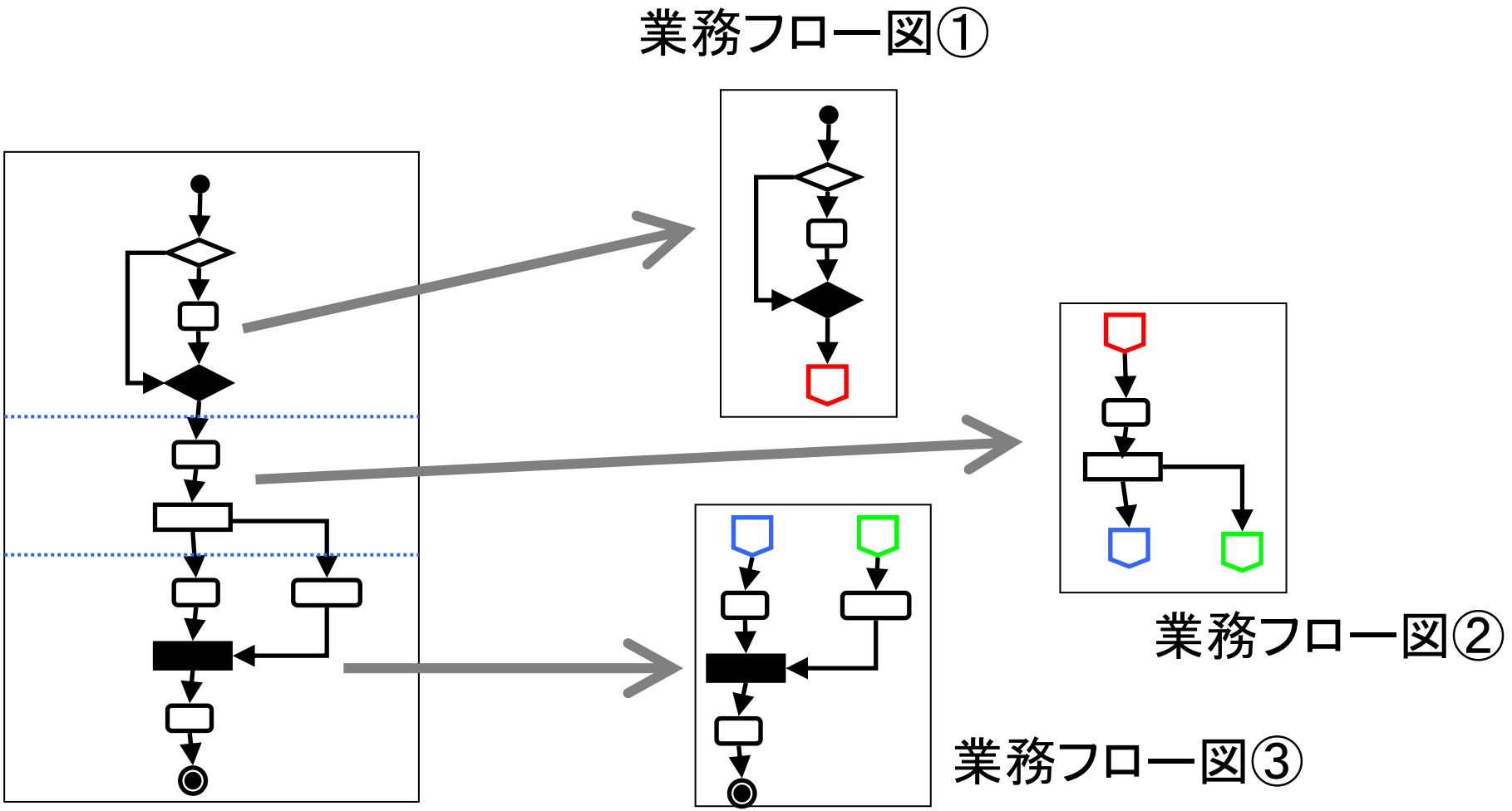
ELAD W



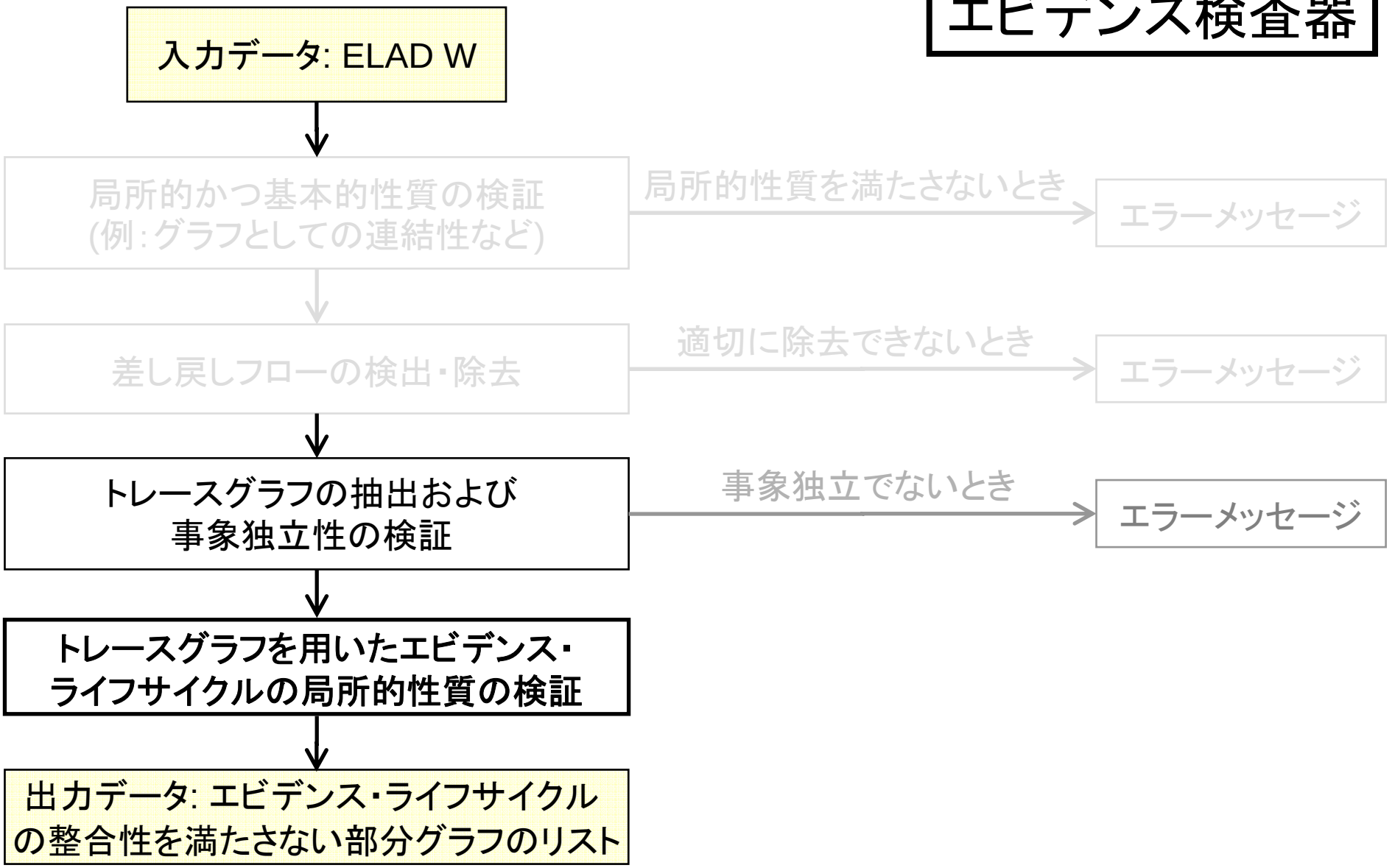
- ELAD W が事象独立のとき，各ノードNに対して，Nに入ってくる作業の流れの数が，Nに至るまでの作業の状態に依存しない．

Nに入ってくる作業の流れの数は一定（1本だけ，または3本すべて）であり，その数はNがジョインかどうかのみによって決まる．

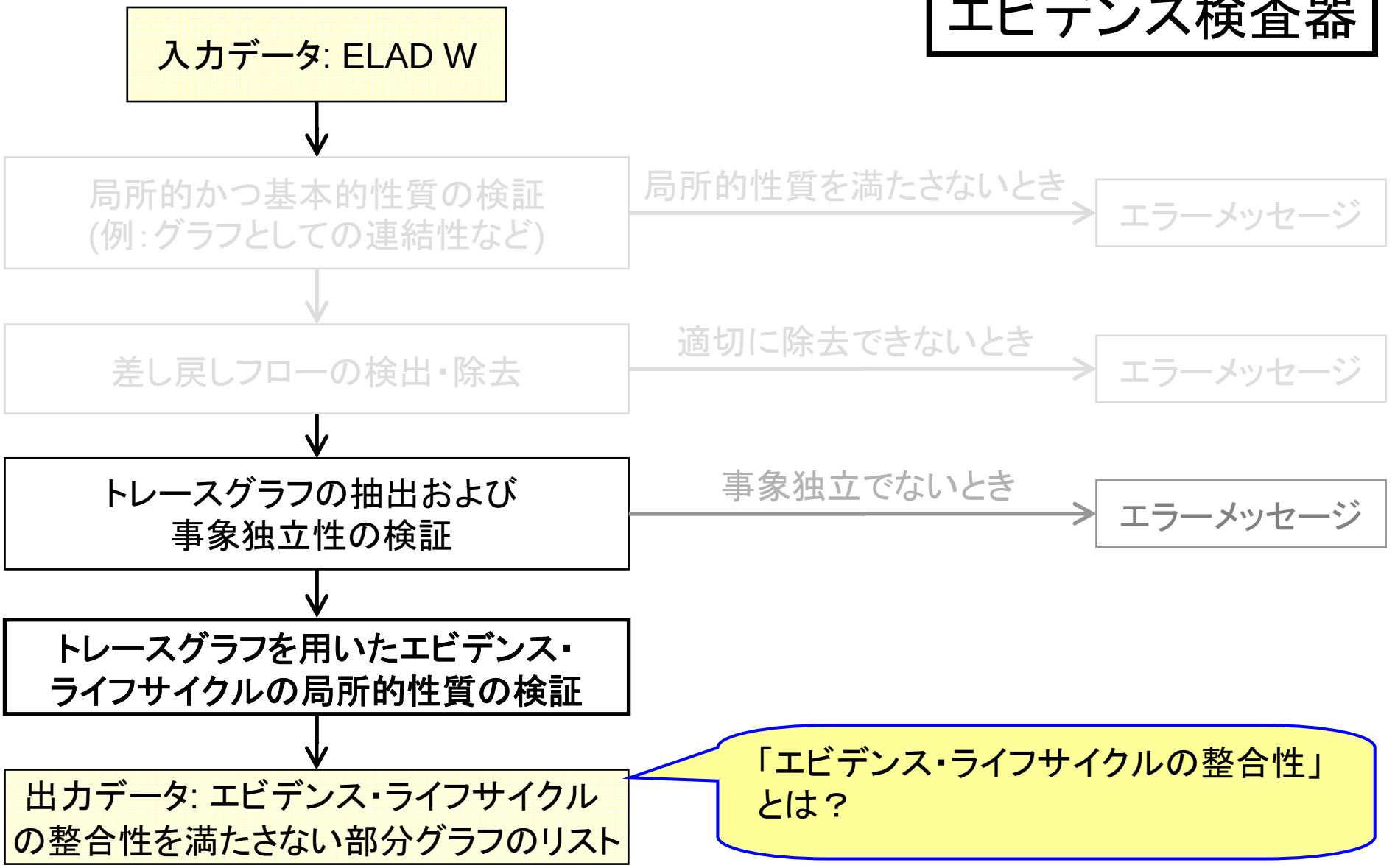
業務フロー図はトリガー部分から作成されるとは限らない.



エビデンス検査器


入力データ: ELAD W
局所的かつ基本的性質の検証
(例:グラフとしての連結性など)
局所的性質を満たさないとき
エラーメッセージ
差し戻しフローの検出・除去
適切に除去できないとき
エラーメッセージ
トレースグラフの抽出および
事象独立性の検証
事象独立でないとき
エラーメッセージ
トレースグラフを用いたエビデンス・
ライフサイクルの局所的性質の検証
出力データ: エビデンス・ライフサイクル
の整合性を満たさない部分グラフのリスト

エビデンス検査器


入力データ: ELAD W
局所的かつ基本的性質の検証
(例:グラフとしての連結性など)
局所的性質を満たさないとき
エラーメッセージ
差し戻しフローの検出・除去
適切に除去できないとき
エラーメッセージ
トレースグラフの抽出および
事象独立性の検証
事象独立でないとき
エラーメッセージ
トレースグラフを用いたエビデンス・
ライフサイクルの局所的性質の検証
出力データ: エビデンス・ライフサイクル
の整合性を満たさない部分グラフのリスト
「エビデンス・ライフサイクルの整合性」
とは？

定義　ELAD Wがエビデンス・ライフサイクルについて整合的であるとは，

1. Wの任意のトレースグラフV
2. V上の任意のアクションノードA
3. A上の任意のエビデンスeについて，

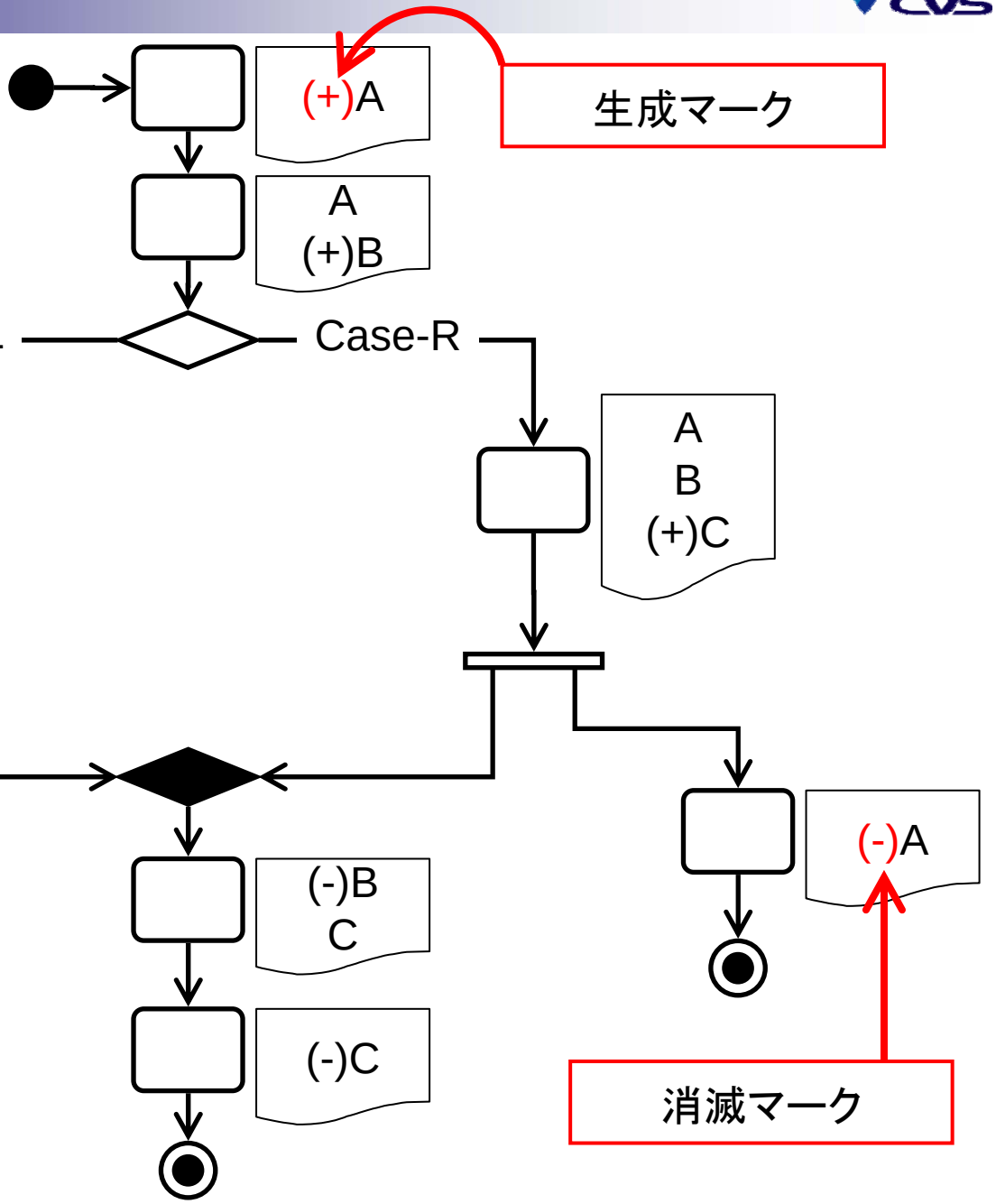
V上のフローの連なり

L := (A1 －f1→ A2 －f2→ ... －f(n-1)→ An)

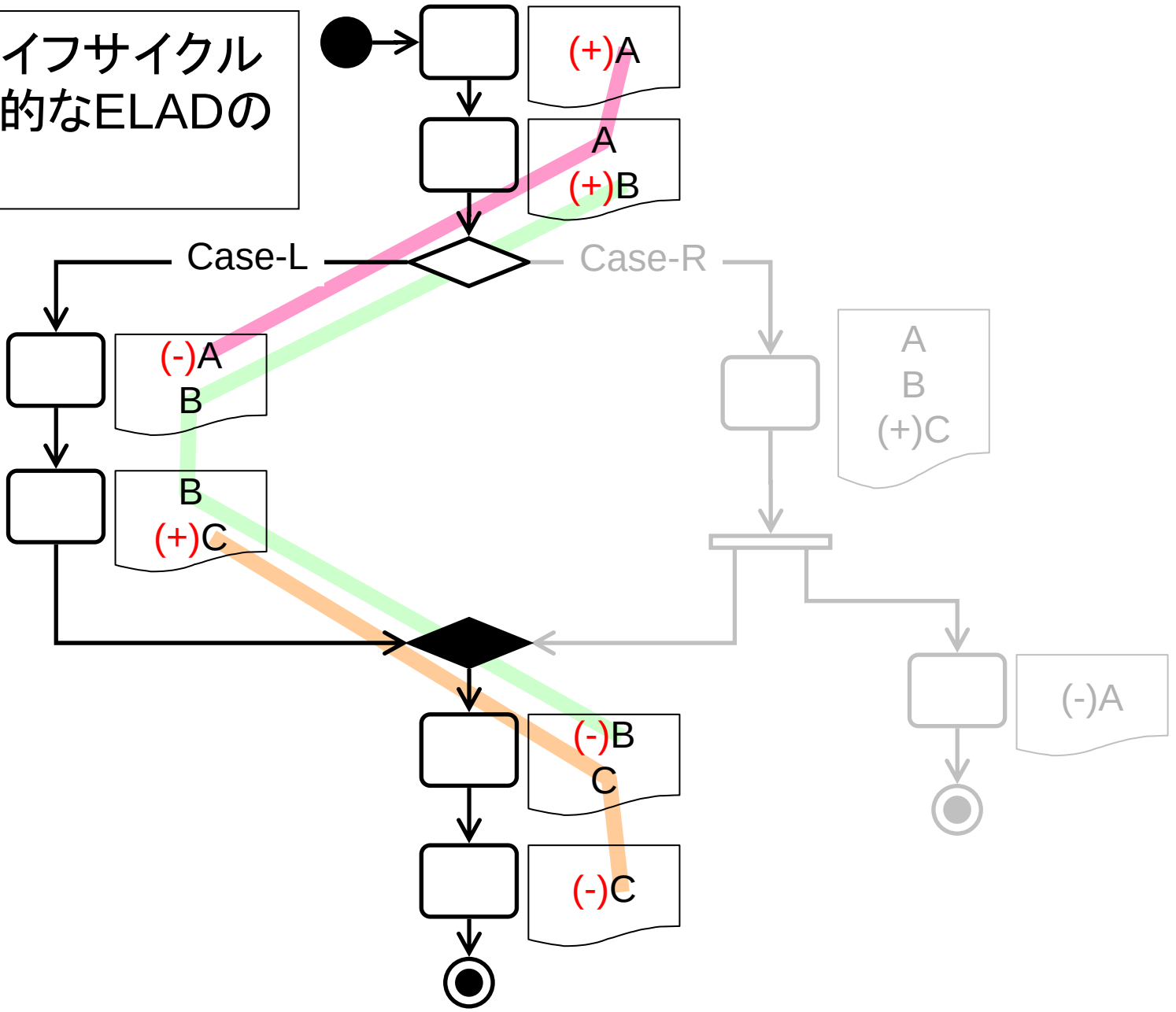
が唯1つ存在して次を満たすことを言う.

(i)　L上のすべてのアクションノードはeを含む.
(ii)　A1のみ生成マーク付のエビデンス「(+)e」を持つ.
(iii)　Anのみ消滅マーク付のエビデンス「(-)e」を持つ.
(iii)　LはAを含む.

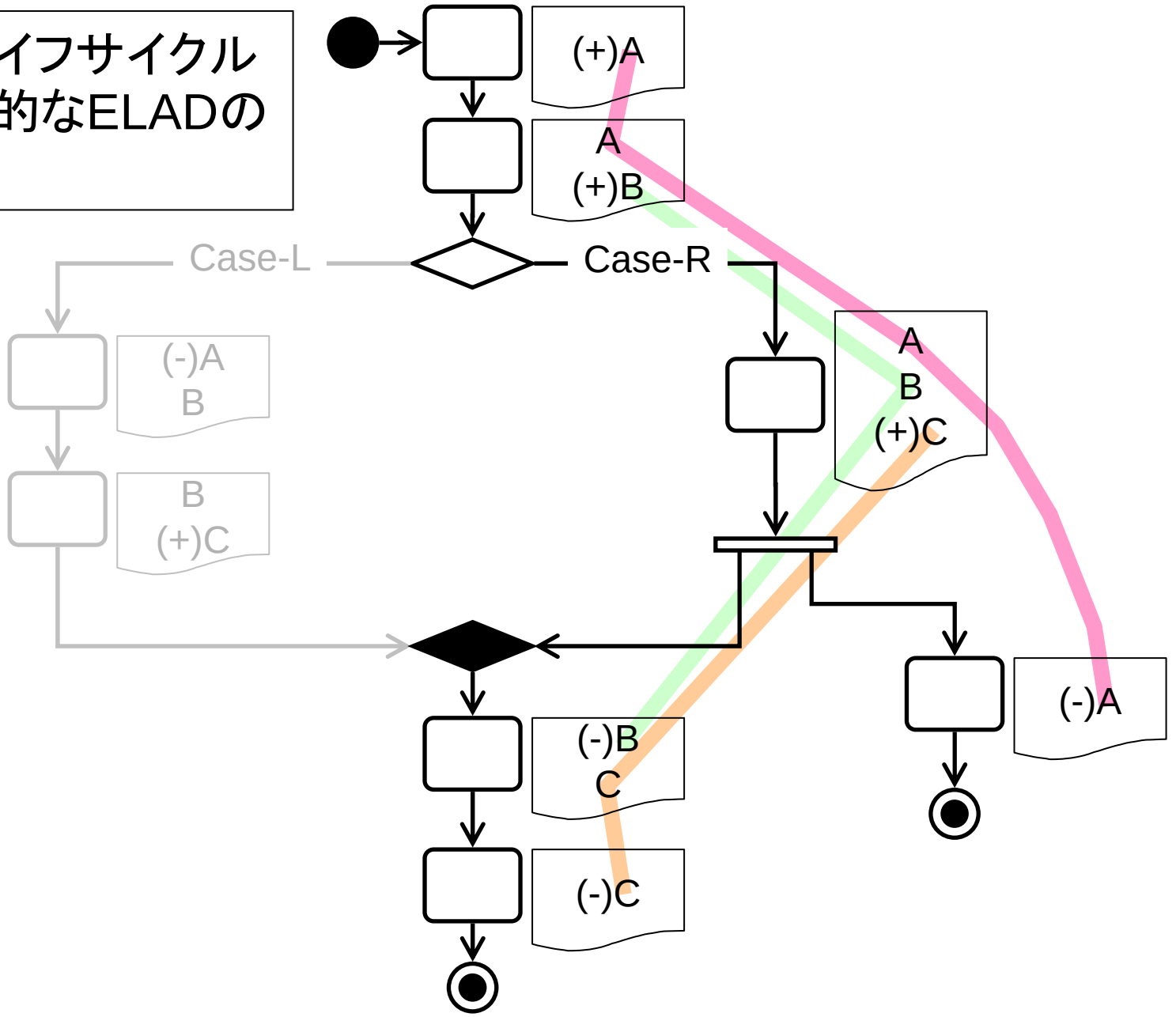
エビデンス・ライフサイクルについて整合的なELADの例



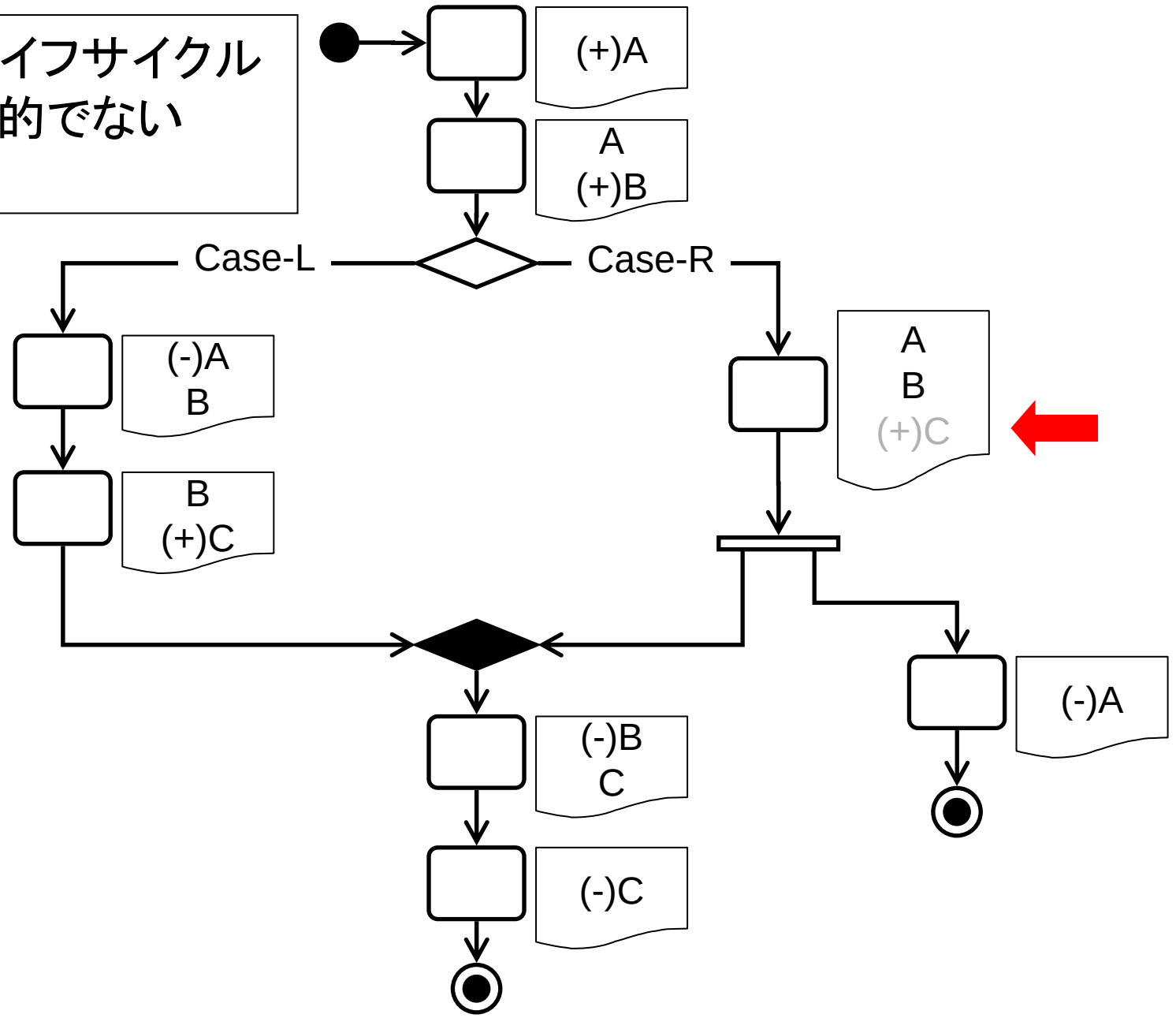
エビデンス・ライフサイクルについて整合的なELADの例



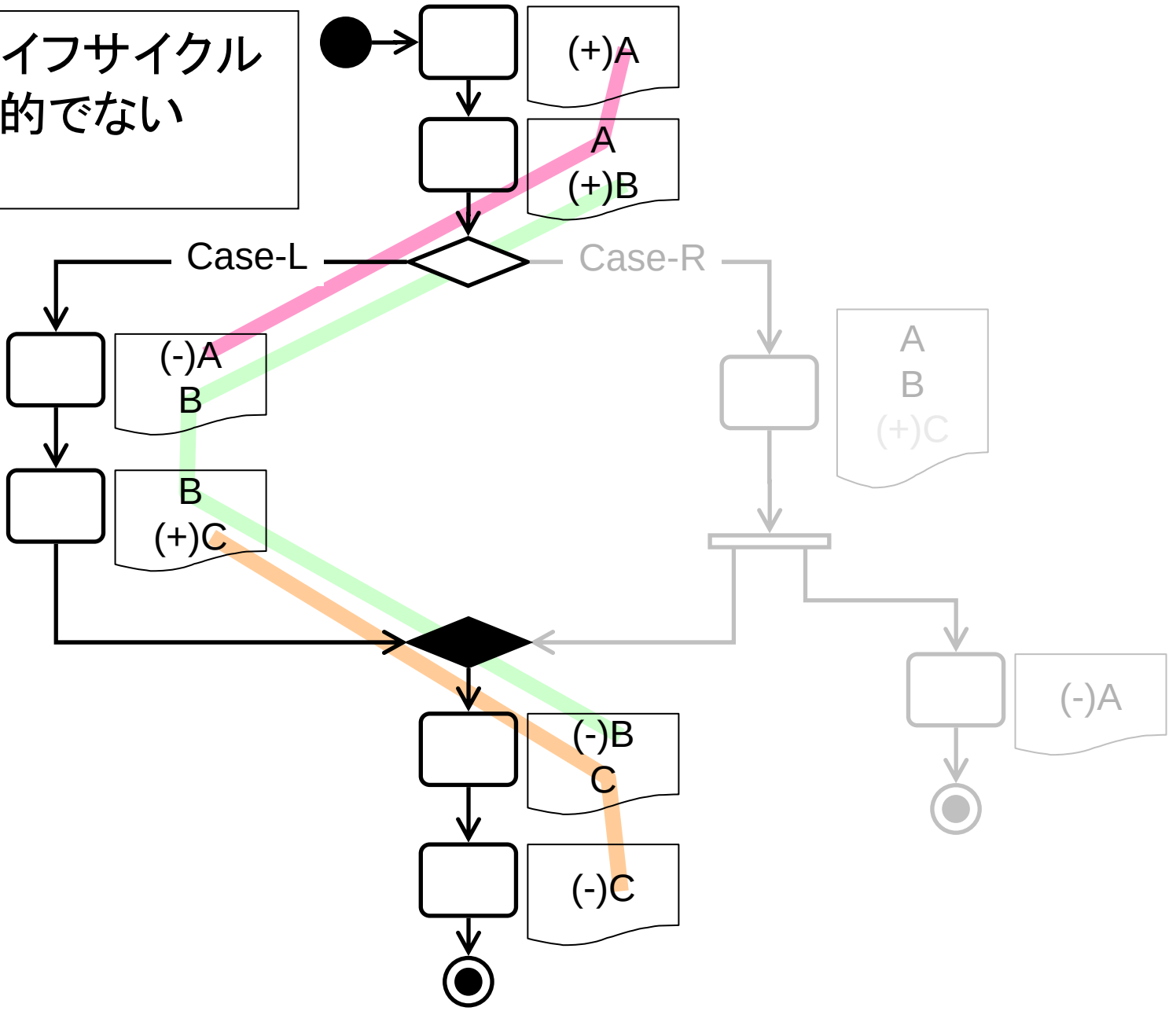
エビデンス・ライフサイクルについて整合的なELADの例



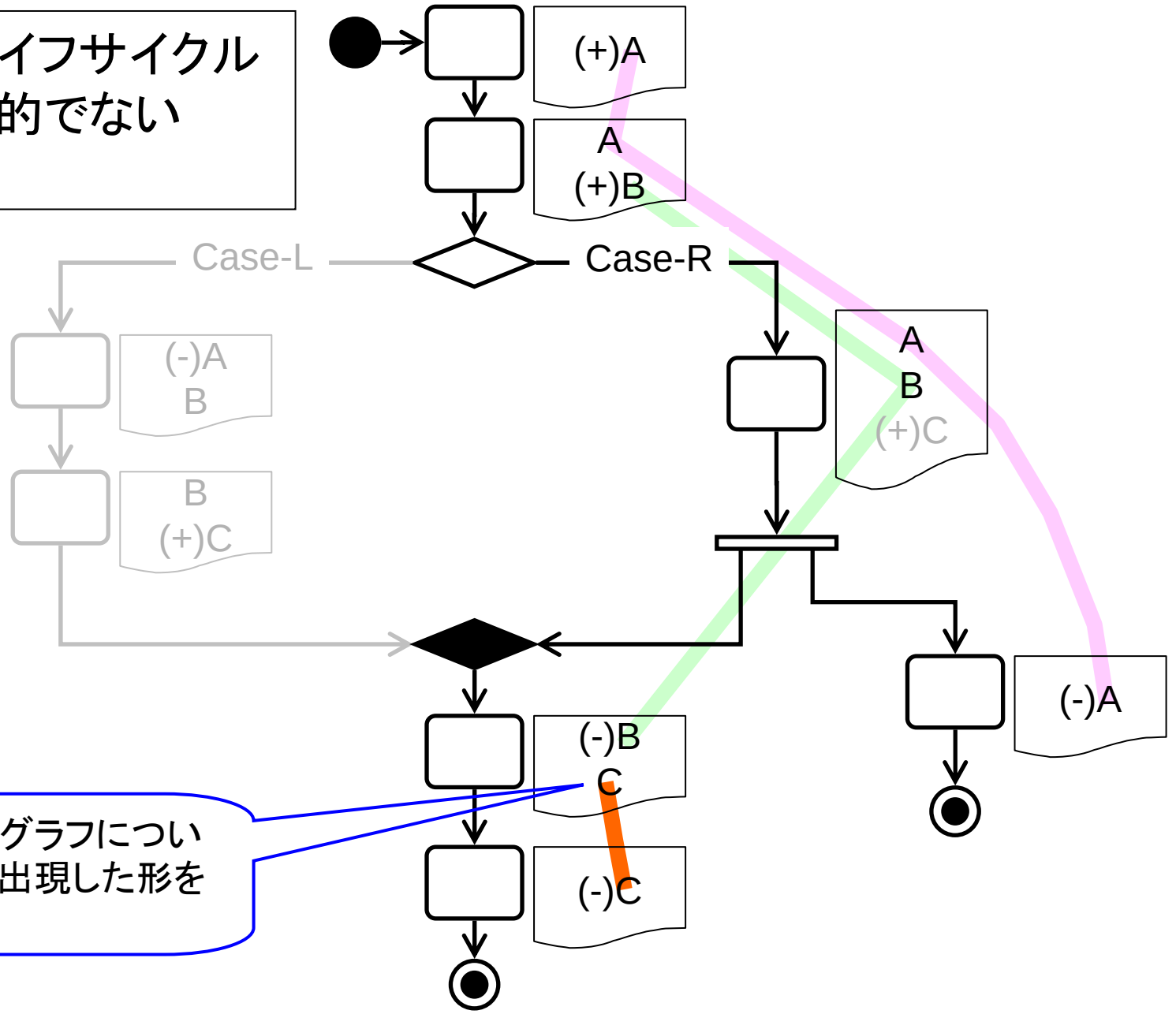
エビデンス・ライフサイクル
について整合的でない
ELADの例



エビデンス・ライフサイクル
について整合的でない
ELADの例



エビデンス・ライフサイクルについて整合的でないELADの例

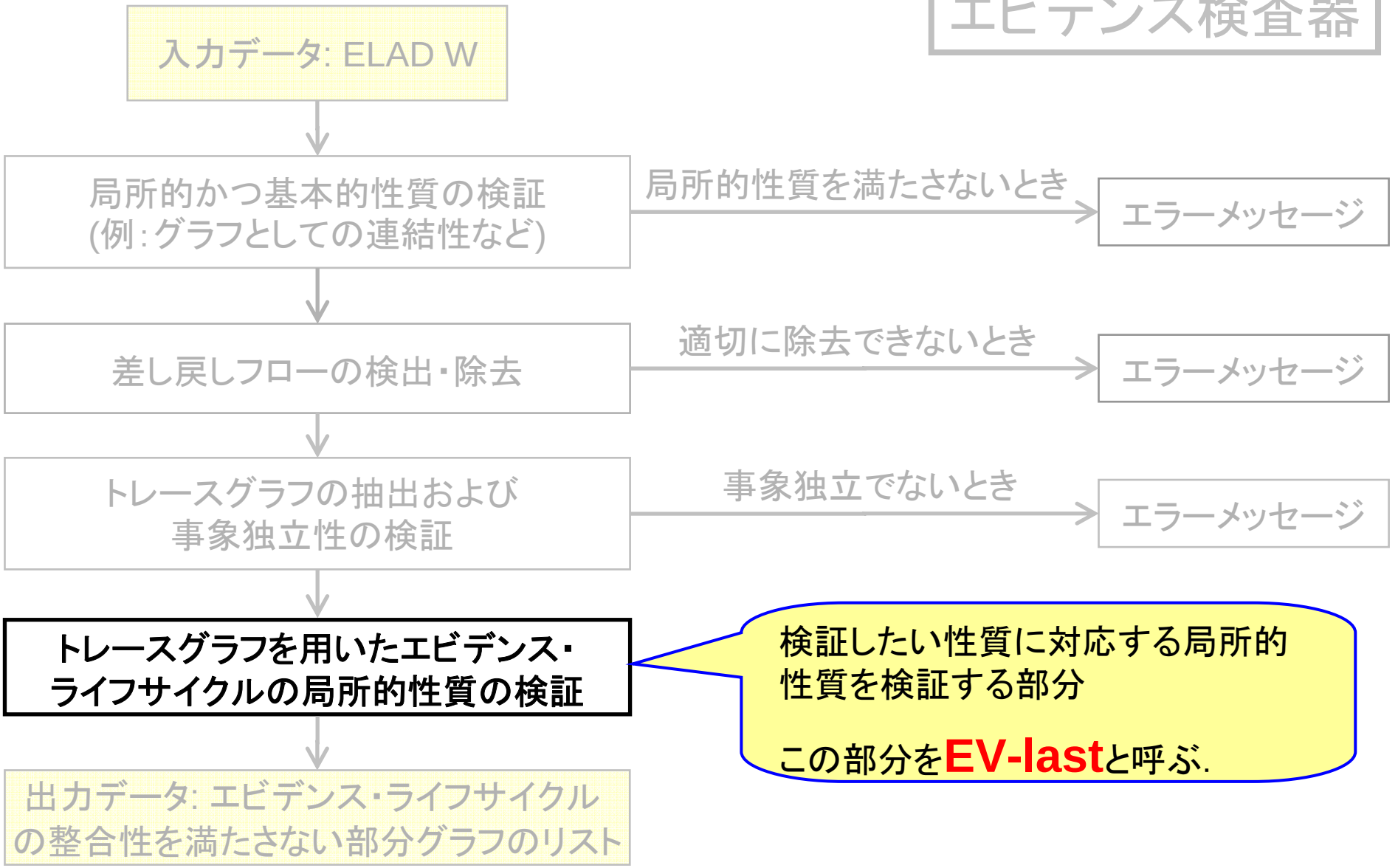


右側のトレースグラフについては，Cは突然出現した形をとっている．

- エビデンス検査器は，与えられたELAD Wに対して，W上のエビデンス・ライフサイクルの整合性を阻害するものだけをすべて検出する.

- 特に，Wがエビデンス・ライフサイクルについて整合的であるとき，かつそのときに限り，エビデンス検査器は空リストを出力する.

エビデンス検査器


入力データ: ELAD W
局所的かつ基本的性質の検証
(例:グラフとしての連結性など)
局所的性質を満たさないとき
エラーメッセージ
差し戻しフローの検出・除去
適切に除去できないとき
エラーメッセージ
トレースグラフの抽出および
事象独立性の検証
事象独立でないとき
エラーメッセージ
トレースグラフを用いたエビデンス・
ライフサイクルの局所的性質の検証
検証したい性質に対応する局所的
性質を検証する部分
この部分をEV-lastと呼ぶ.
出力データ: エビデンス・ライフサイクル
の整合性を満たさない部分グラフのリスト

定理

任意の事象独立なELAD Wについて, 以下の3つの性質はそれぞれ同値である.

1. Wはエビデンス・ライフサイクルについて整合的.
2. Wはエビデンス・ライフサイクルに関する局所的な検証項目をすべて満たす.
3. EV-LastによるWに対する出力値が空である.

EV-lastの振る舞い

- 与えられた事象独立なELAD Wに対して,

1. W上のすべてのラインデータを抽出

2. ラインデータに基づくエビデンス・ライフサイクルに関する局所的な検証項目をチェック

3. 局所的性質を満たさないラインデータを組み合わせて, エビデンス・ライフサイクルの整合性を阻害する箇所を出力データとして構築.

定義

与えられたELAD Wに対して，以下の性質(i)～(iii)を満たすW上のフローの連なり

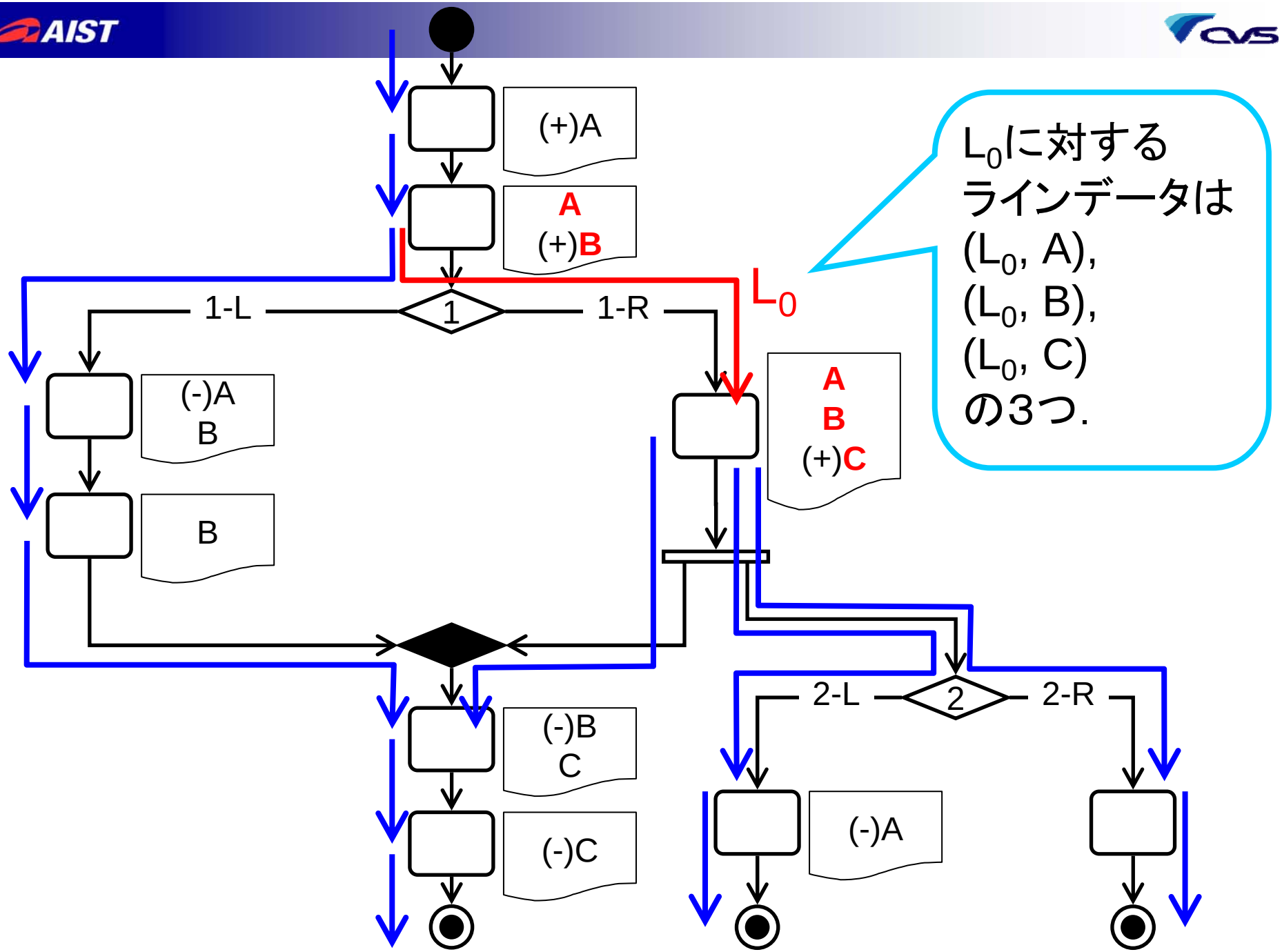
L := (A1 －f1→ A2 －f2→ ... －f(n-1)→ An)

をW上のラインと呼ぶ.

1. A1はアクションノードまたはトリガーである.
2. Anはアクションノードまたはエンドノードである.
3. A1とAn以外のノードはアクションノードではない.

定義

W上のラインLおよびLに含まれるエビデンスeの組(L,e)をW上のラインデータと呼ぶ.


(+)A
A
(+)B
1-L
1
1-R
(-)A
B
B
A
B
(+)C
L_0
L_0に対する
ラインデータは
(L_0, A),
(L_0, B),
(L_0, C)
の3つ.
(-)B
C
(-)C
2-L
2
2-R
(-)A

エビデンス・ライフサイクルに関する局所的検証項目

1. エビデンスの不適切に出現しないかどうか

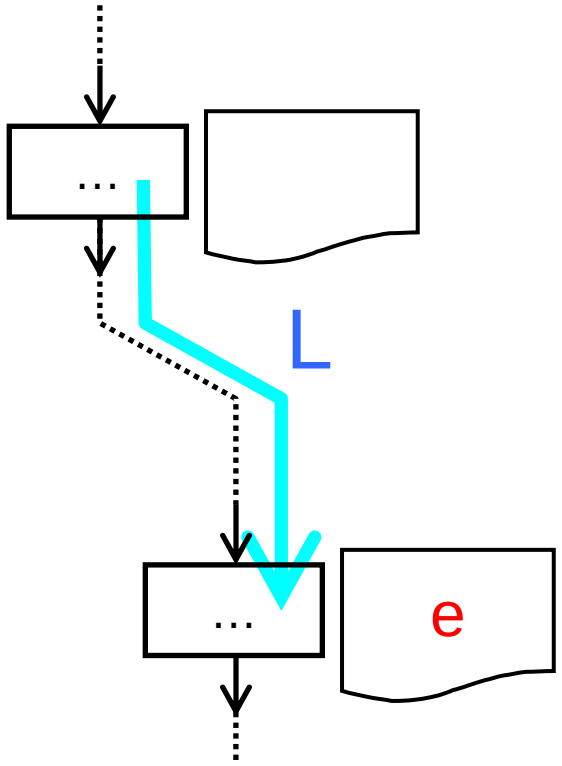
2. エビデンスの不適切に消滅しないかどうか

3. エビデンスが増加しないかどうか

4. エビデンスが減少しないかどうか

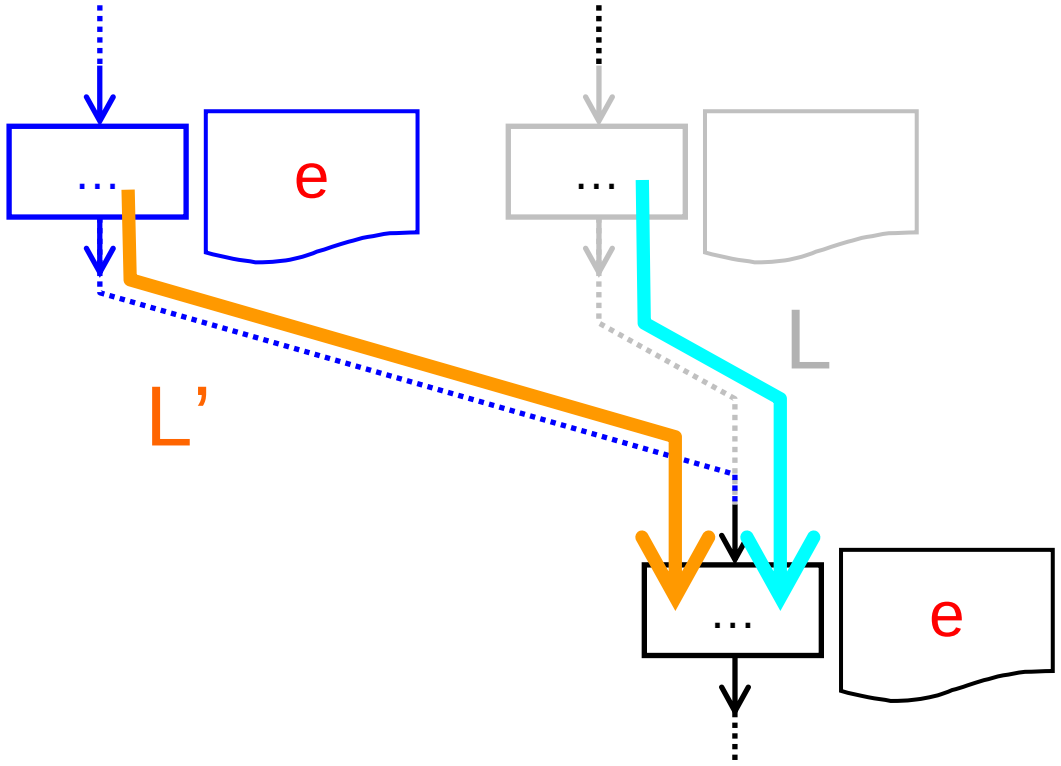
エビデンスの不適切な出現の有無

- 任意のラインLに対して, eがLに沿って``突然''出現したとき,



エビデンスの不適切な出現の有無

- 任意のラインLに対して, eがLに沿って``突然''出現したとき, Lを含むすべてのトレースグラフVに対して, V上のラインL'が存在して, L'とLは同じ到着点を持ち, しかも, L'の出発点において``消滅しない''eが存在する.



- これ以降においてお話しする内容
  - 背景
    - なぜエビデンス･ライフサイクルの整合性を検証するのか？
    - ELAD
  - エビデンス検査器の構成
  - 適用実験

エビデンス検査器の適用実験

- 60個の``リアルな''業務フロー図
- 完成品（手作業によるチェックは既に済ませてある）
- エビデンス検査器のユーザがすべきことは，基本的には，``ボタンをワンクリックする''のみ.
- 各業務フロー図（10～60ノード程度）に対する実行時間はすべて0.5秒未満.
  - 実験機：Intel Pentium M 1200MHz + 1.49GB RAM
- 今回の実験で用いられた業務フロー図では，エビデンスの消滅マークが設定されていなかったため，消滅マークに関する検証を除外して行った.

実験結果

エビデンス検査器が指摘した不具合箇所を，不具合の原因によって分類した.

	不具合の原因	不具合の数
1	業務フロー図の構造が不整合なため	8
2	業務フロー図の構造が複雑化したため	6
3	エビデンスの書き忘れ	11
4	(+)マークの付け忘れ	3
5	(+)マークの消し忘れ	10
6	エビデンスの名前の書き間違い	2
7	不適切な指摘	10

まとめ

- エビデンス検査器の開発
  - トレースグラフを抽出しながら事象独立性を検証する機能
  - エビデンス・ライフサイクルに関する局所的な検証項目をチェックする機能
- 60個の実際の業務フロー図をエビデンス検査器で検証した結果，適切と見なせる40の指摘と不適切と考えれる10の指摘がなされた
- その内で業務フロー図の構造そのものに欠陥があったために不具合が起きた箇所が8個指摘され，それを調べることによって，3箇所の構造上の欠陥を修正することが出来た.

展望

- 複数の開発環境にある業務フロー図を一括して検証・管理する機能の開発
- エビデンス以外のライフサイクルを検証する機能の開発
  - データベース上のデータのライフサイクル
  - アクションノードによって記述される一連の作業のライフサイクル

参考文献・特許出願記事

1. Verification Algorithm of Evidence Life Cycles in Extended UML Activity Diagrams, O.Takaki, T.Seino, I.Takeuti, N.Izumi, K.Takahashi, ICSEA2007 Proceedings, IEEE Press.

2. UML アクティビティ図に対する事象部分グラフ抽出および事象独立性検証アルゴリズム. 高木理, 清野貴博, 竹内泉, 和泉憲明, 高橋孝一 (著), ソフトウェアエンジニアリング最前線 2007, 近代科学社. pp. 153-164. 2007.

3. 特願2007-170616 （名称：検証システム）

4. 特願2007-218850 （名称：検証システム）