# On the Essence and Initiality of Conflicts

Guilherme Grochau Azzi[1], *Andrea Corradini*[2]
and Leila Ribeiro[1]

[1]Instituto de Informática
Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil

[2]Dipartimento di Informatica
Università di Pisa, Pisa, Italy

IFIP WG1.3 Foundations of System Specification
Prague, April 6-7, 2019

*based on a paper presented at the 11th International Conference on Graph Transformation, June 2018*

# Contribution to System Specification or Modeling

- The work contributes to the theory of **Graph Transformation Systems** (GTS).

# Contribution to System Specification or Modeling

- The work contributes to the theory of **Graph Transformation Systems** (GTS).
- GTS provide formal foundations of declarative techniques for specification, modeling and analysis of systems, preferably when:

# Contribution to System Specification or Modeling

- The work contributes to the theory of **Graph Transformation Systems** (GTS).
- GTS provide formal foundations of declarative techniques for specification, modeling and analysis of systems, preferably when:
  - The state is logically and/or physically distributed: it can be abstracted to a graph.

# Contribution to System Specification or Modeling

- The work contributes to the theory of **Graph Transformation Systems** (GTS).
- GTS provide formal foundations of declarative techniques for specification, modeling and analysis of systems, preferably when:
  - The state is logically and/or physically distributed: it can be abstracted to a graph.
  - The dynamics is determined by local changes: they can be described declaratively as rules.

# Contribution to System Specification or Modeling (2)

- Locality of transformations, described by rules, enables parallelism.

# Contribution to System Specification or Modeling (2)

- Locality of transformations, described by rules, enables parallelism.
- Rule-based specifications introduce various levels of non-determinism in the system's behaviour:

# Contribution to System Specification or Modeling (2)

- Locality of transformations, described by rules, enables parallelism.
- Rule-based specifications introduce various levels of non-determinism in the system's behaviour:
  - selection of a rule

# Contribution to System Specification or Modeling (2)

- Locality of transformations, described by rules, enables parallelism.
- Rule-based specifications introduce various levels of non-determinism in the system's behaviour:
  - selection of a rule
  - selection of a match where to apply the rule

# Contribution to System Specification or Modeling (2)

- Locality of transformations, described by rules, enables parallelism.
- Rule-based specifications introduce various levels of non-determinism in the system's behaviour:
  - selection of a rule
  - selection of a match where to apply the rule
- Intrinsic non-determinism of the modeled system could be mixed with the one arising from the rule-based specification of functional transformations.

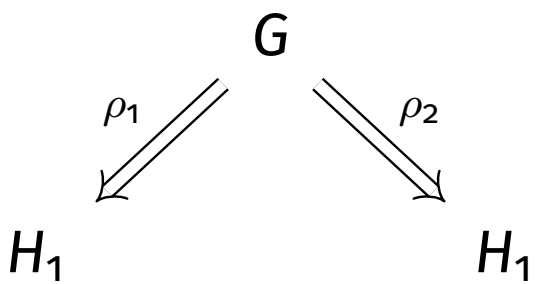
# Contribution to System Specification or Modeling (2)

- Locality of transformations, described by rules, enables parallelism.
- Rule-based specifications introduce various levels of non-determinism in the system's behaviour:
  - selection of a rule
  - selection of a match where to apply the rule
- Intrinsic non-determinism of the modeled system could be mixed with the one arising from the rule-based specification of functional transformations.
- Analysis of conditions for independence and for potential conflicts among transformations becomes fundamental for the analysis of such systems.

# Parallel independence vs. conflicts

Since (linear) graph transformation is resource-conscious, “confluence” is “strict confluence”.

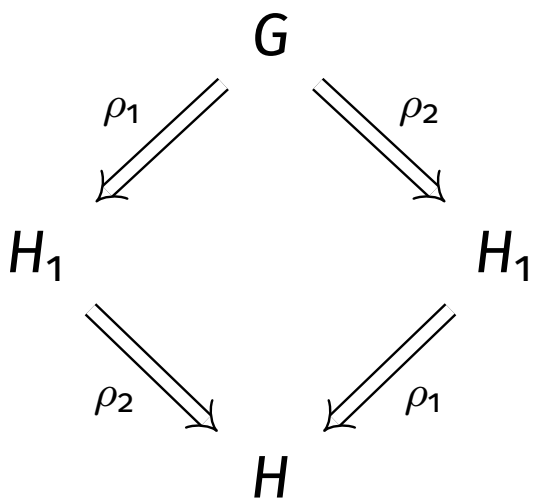
## Parallel Independence
### of Transformations



# Parallel independence vs. conflicts

Since (linear) graph transformation is resource-conscious, “confluence” is “strict confluence”.

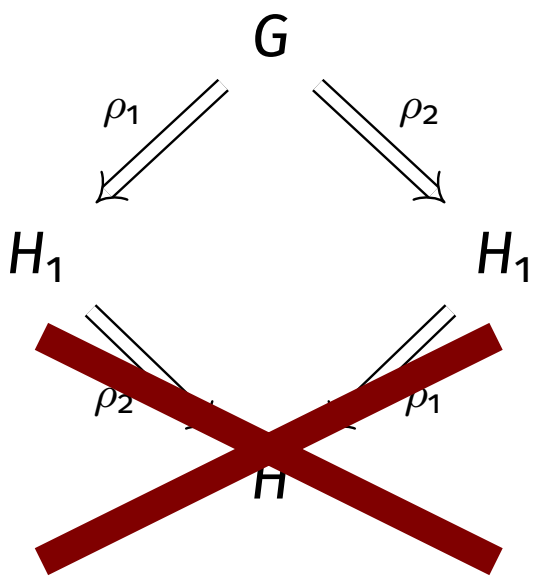
## Parallel Independence
### of Transformations



# Parallel independence vs. conflicts

Since (linear) graph transformation is resource-conscious, "confluence" is "strict confluence".

## Parallel Independence
### of Transformations




Conflict

# Motivations of the present talk

- Conflicts capture important information about behaviour

# Motivations of the present talk

- Conflicts capture important information about behaviour
- **Critical Pair Analysis** (CPA) can be used to verify confluence of (sub)sets of rules

# Motivations of the present talk

- Conflicts capture important information about behaviour
- **Critical Pair Analysis** (CPA) can be used to verify confluence of (sub)sets of rules
- CPA requires enumerating all “potential conflicts”

# Motivations of the present talk

- Conflicts capture important information about behaviour
- **Critical Pair Analysis** (CPA) can be used to verify confluence of (sub)sets of rules
- CPA requires enumerating all “potential conflicts”
- Two main issues:

# Motivations of the present talk

- Conflicts capture important information about behaviour
- **Critical Pair Analysis** (CPA) can be used to verify confluence of (sub)sets of rules
- CPA requires enumerating all “potential conflicts”
- Two main issues:
  - Reducing as much as possible the set of critical pairs to be analyzed

# Motivations of the present talk

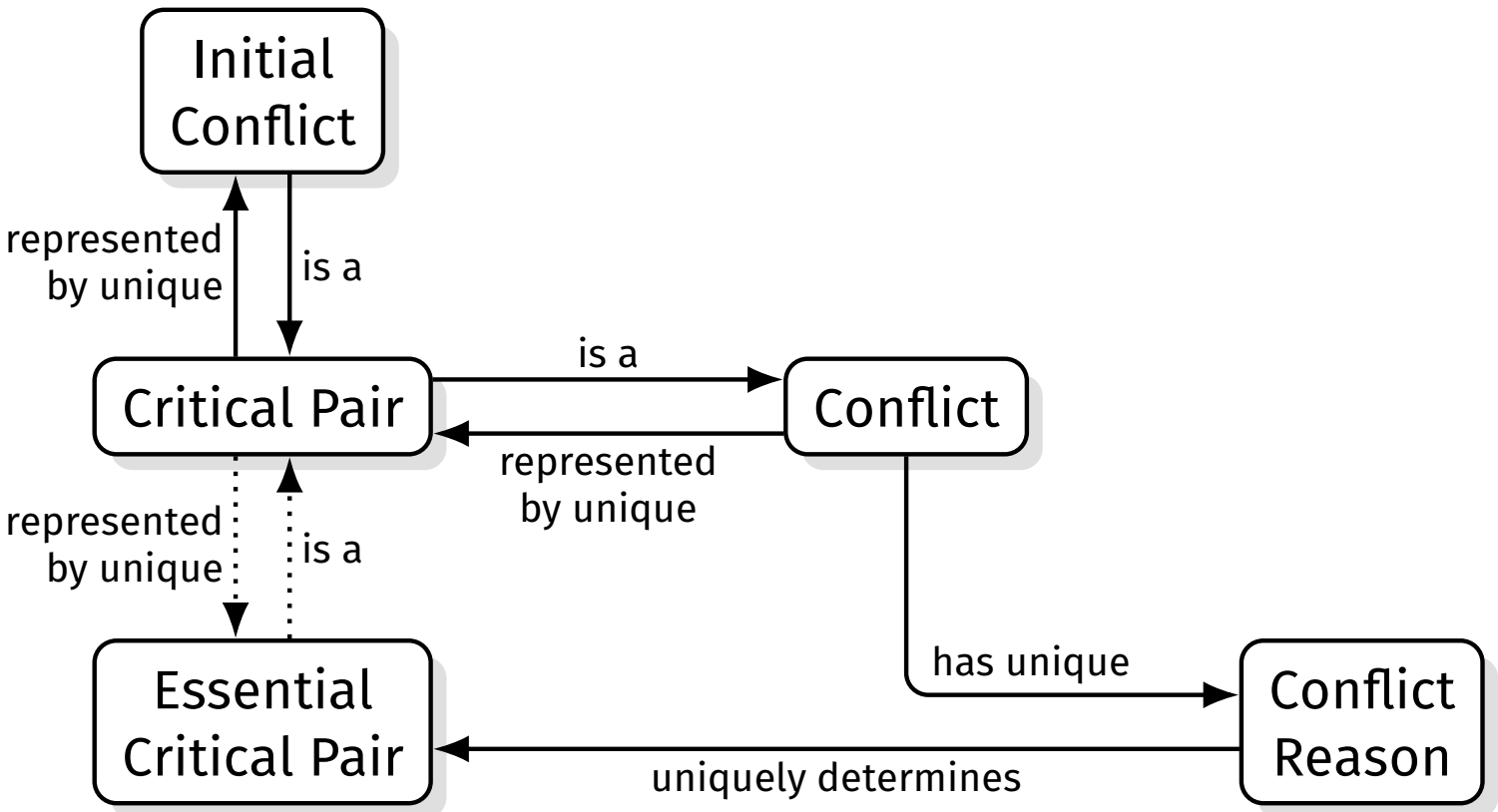
- Conflicts capture important information about behaviour
- **Critical Pair Analysis** (CPA) can be used to verify confluence of (sub)sets of rules
- CPA requires enumerating all “potential conflicts”
- Two main issues:
  - Reducing as much as possible the set of critical pairs to be analyzed
  - To identify the “root causes” of conflicts in a precise way

# Motivations of the present talk

- Conflicts capture important information about behaviour
- **Critical Pair Analysis** (CPA) can be used to verify confluence of (sub)sets of rules
- CPA requires enumerating all “potential conflicts”
- Two main issues:
  - Reducing as much as possible the set of critical pairs to be analyzed
  - To identify the “root causes” of conflicts in a precise way
- “Root causes” are the resources for which the transformation compete

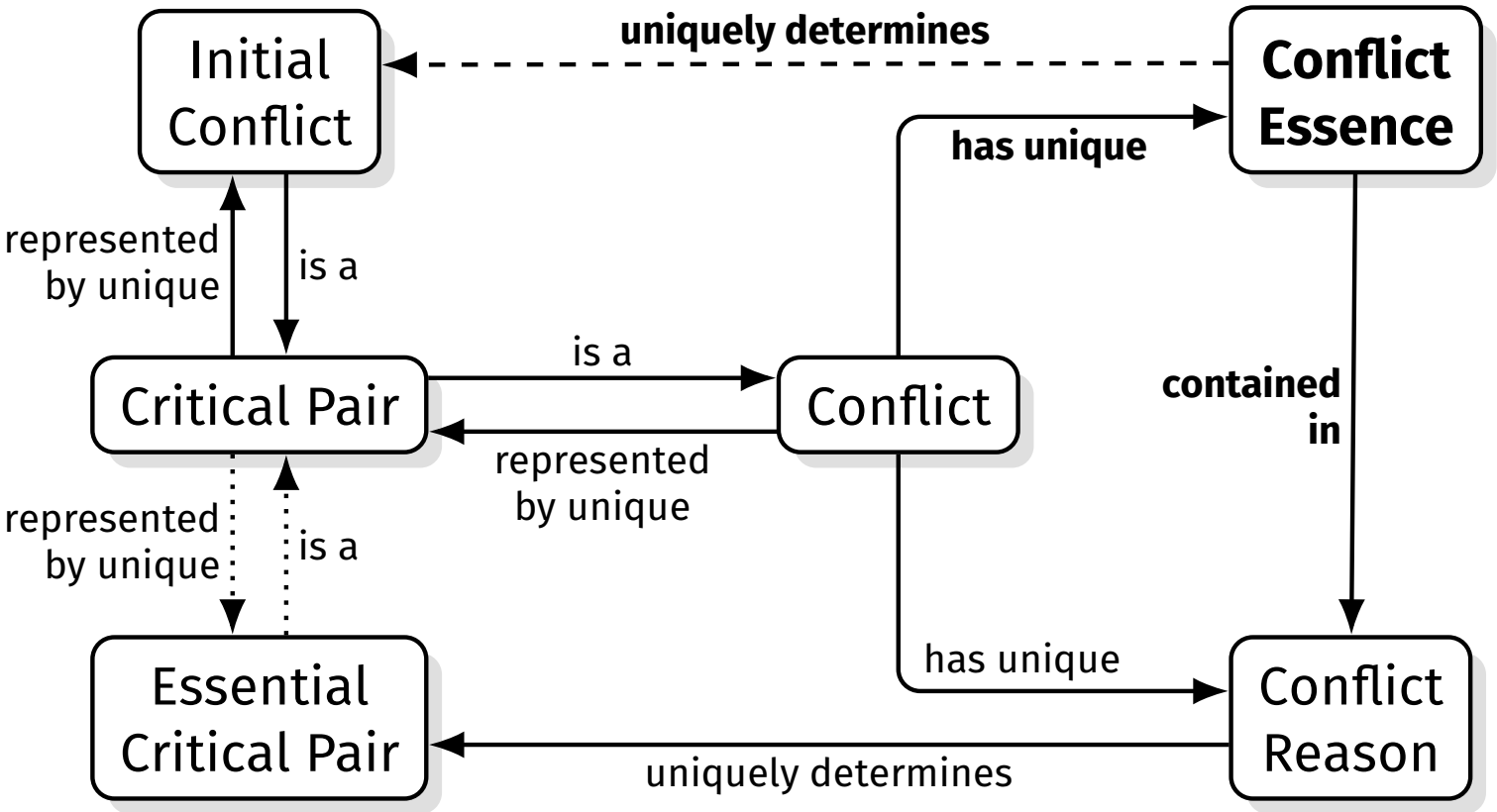
# Overview of previous results

Available for: —— Adhesive Categories - - - - $\mathbb{S}\mathrm{et}^{\mathbb{S}}$ ····· $\mathbb{G}\mathrm{raph}_T$



# Overview of our results in context

Available for: —— Adhesive Categories - - - - $\mathbb{S}\mathrm{et}^{\mathbb{S}}$ ····· $\mathbb{G}\mathrm{raph}_T$



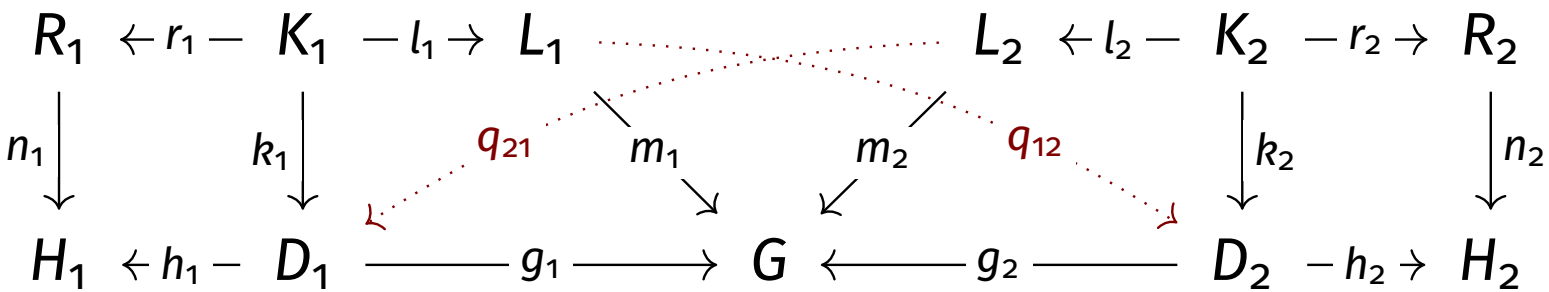
# Background: The DPO Approach

$$\begin{aligned} \text{Rule:} \quad & \rho = L \overset{l}{\rightarrowtail} K \overset{r}{\rightarrowtail} R \\ \text{Match:} \quad & m : L \rightarrowtail G \\ \text{Transformation:} \quad & G \overset{\rho,m}{\Longrightarrow} H \end{aligned}$$

$$\begin{array}{ccccc} L & \xleftarrow{\ l\ } & K & \xrightarrow{\ r\ } & R \\ \downarrow{\scriptstyle m} & \text{PO} & \downarrow{\scriptstyle k} & \text{PO} & \downarrow{\scriptstyle n} \\ G & \xleftarrow[\ g\ ]{} & D & \xrightarrow[\ h\ ]{} & H \end{array}$$

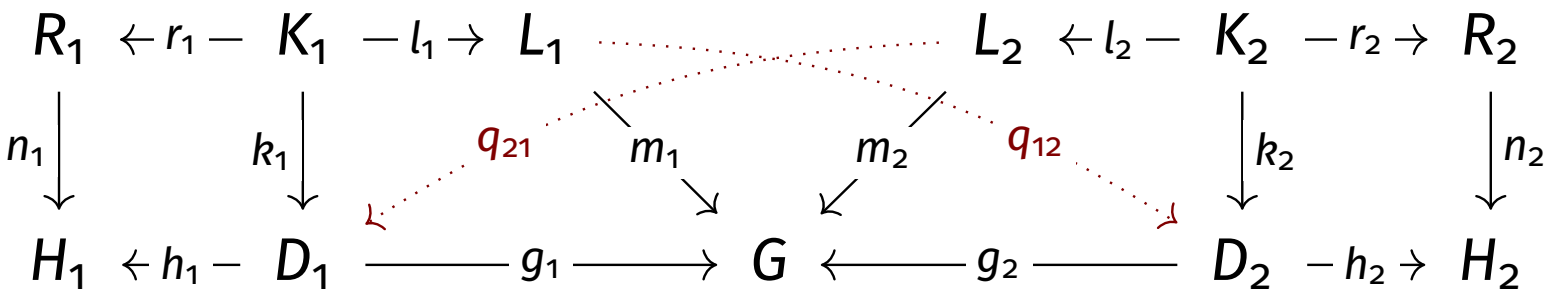
# New Perspective on Parallel Independence

- Previous work based on the **standard condition** for parallel independence



# New Perspective on Parallel Independence

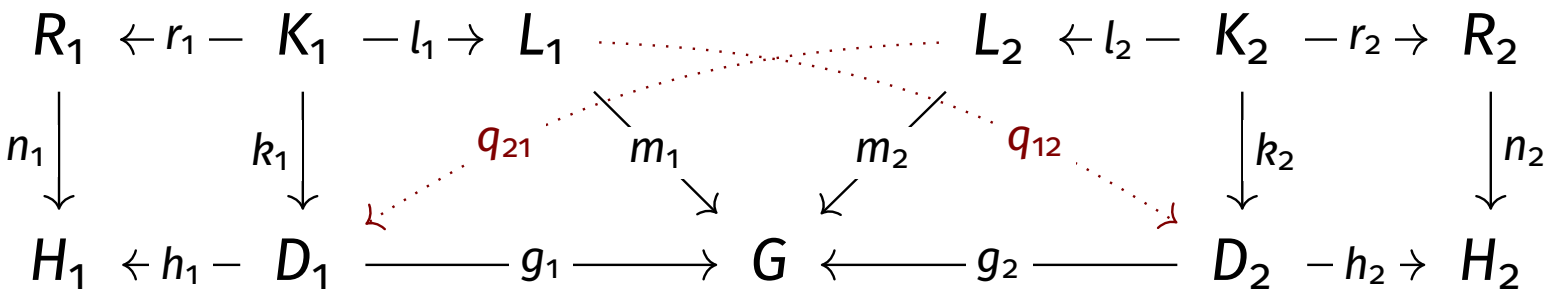
- Previous work based on the **standard condition** for parallel independence




- Recently: **essential condition** for parallel independence (Corradini et al. 2018)
- Equivalent to standard condition

# New Perspective on Parallel Independence

- Previous work based on the **standard condition** for parallel independence

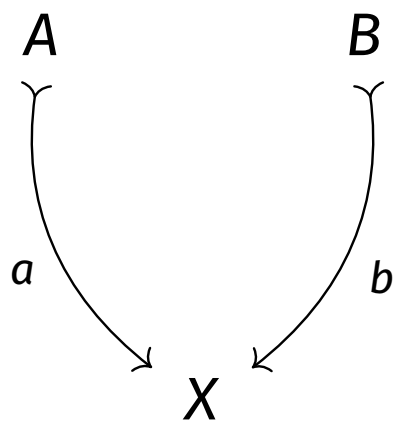



- Recently: **essential condition** for parallel independence (Corradini et al. 2018)
- Equivalent to standard condition

- **Goal:** review characterization of conflicts under new light

# Background: Adhesive Categories

Subobjects behave like subsets

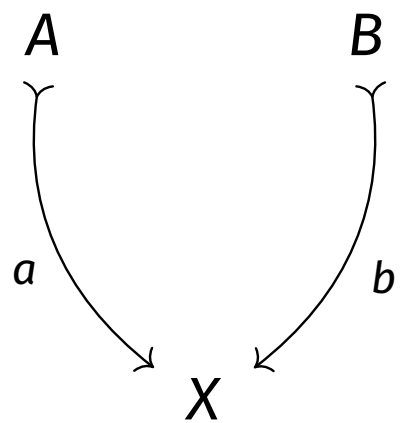


# Background: Adhesive Categories

Subobjects behave like subsets

**Lemma (Lack and Sobocinski 2005)**

*In adhesive categories,* **Sub**$(X)$ *is distributive lattice*



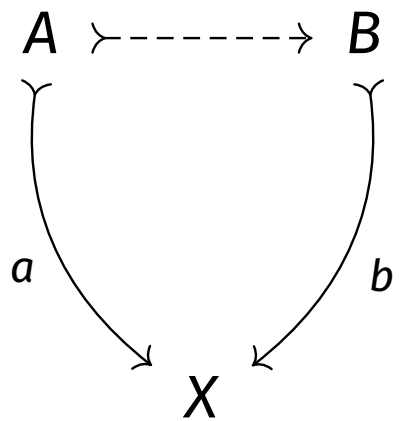
# Background: Adhesive Categories

Subobjects behave like subsets

**Lemma (Lack and Sobocinski 2005)**

*In adhesive categories,* **Sub**$(X)$ *is distributive lattice*

Containment existence of mono



# Background: Adhesive Categories

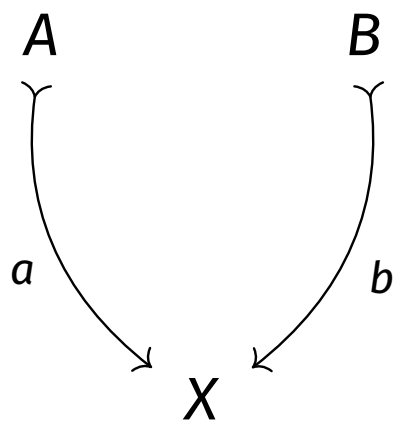
Subobjects behave like subsets

**Lemma (Lack and Sobocinski 2005)**

*In adhesive categories,* **Sub**$(X)$ *is distributive lattice*

Containment existence of mono

Intersection pullback



# Background: Adhesive Categories

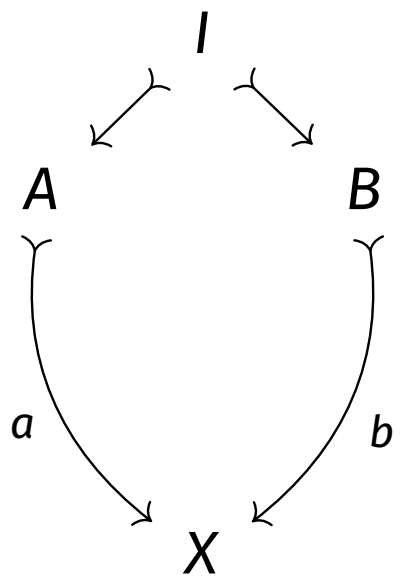
Subobjects behave like subsets

**Lemma (Lack and Sobocinski 2005)**

*In adhesive categories,* **Sub**$(X)$ *is distributive lattice*

Containment existence of mono

Intersection pullback



# Background: Adhesive Categories

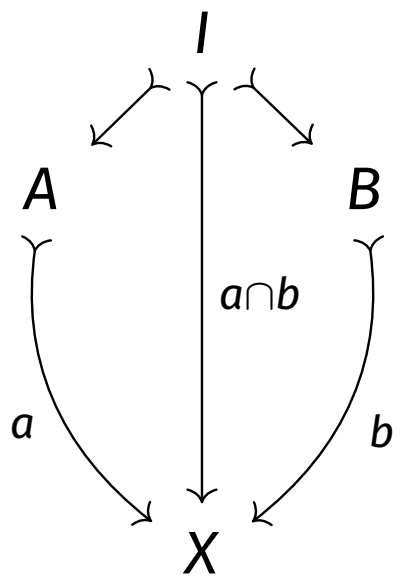
Subobjects behave like subsets

**Lemma (Lack and Sobocinski 2005)**

*In adhesive categories,* **Sub**$(X)$ *is distributive lattice*

Containment existence of mono

Intersection pullback



# Background: Adhesive Categories

Subobjects behave like subsets

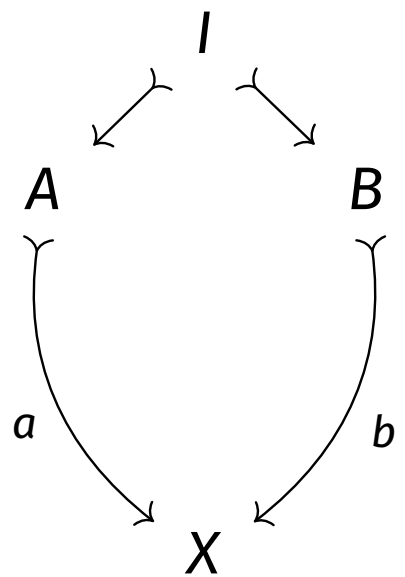
**Lemma (Lack and Sobocinski 2005)**

*In adhesive categories,* **Sub**$(X)$ *is distributive lattice*

Containment existence of mono

Intersection pullback

Union pushout over intersection



# Background: Adhesive Categories

Subobjects behave like subsets

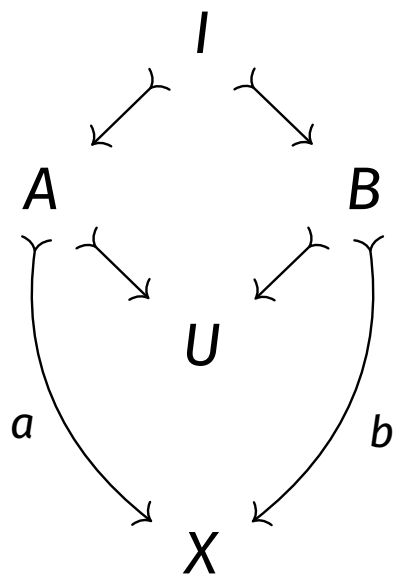
**Lemma (Lack and Sobocinski 2005)**

*In adhesive categories,* **Sub**$(X)$ *is distributive lattice*

**Containment** existence of mono

**Intersection** pullback

**Union** pushout over intersection



# Background: Adhesive Categories

Subobjects behave like subsets

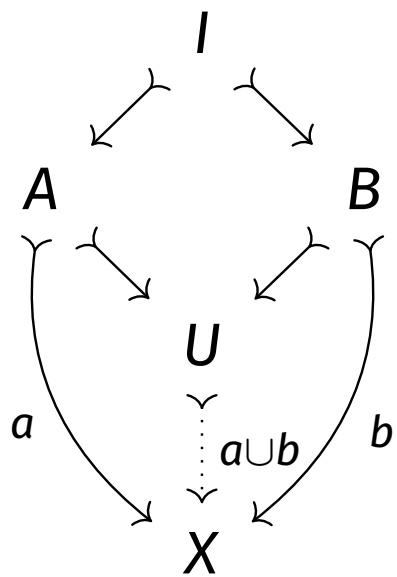
**Lemma (Lack and Sobocinski 2005)**

*In adhesive categories,* **Sub**$(X)$ *is distributive lattice*

Containment existence of mono

Intersection pullback

Union pushout over intersection



# Background: Adhesive Categories

Subobjects behave like subsets

**Lemma (Lack and Sobocinski 2005)**

*In adhesive categories,* $\mathbf{Sub}(X)$ *is distributive lattice*

| | |
|---|---|
| Containment | existence of mono |
| Intersection | pullback |
| Union | pushout over intersection |
| Top | is $X$ |

# Background: Adhesive Categories

Subobjects behave like subsets

**Lemma (Lack and Sobocinski 2005)**

*In adhesive categories,* **Sub**$(X)$ *is distributive lattice*

| | |
|---|---|
| Containment | existence of mono |
| Intersection | pullback |
| Union | pushout over intersection |
| Top | is $X$ |
| Bottom | usually “empty”, if exists |

# Background: Set-Valued Functor Categories

- Some results not proven for all adhesive categories

# Background: Set-Valued Functor Categories

- Some results not proven for all adhesive categories
- We use categories $\mathbb{S}\mathrm{et}^{\mathbb{S}}$ of functors $\mathbb{S} \to \mathbb{S}\mathrm{et}$ with natural transformations as arrows (essentially presheaves)

# Background: Set-Valued Functor Categories

- Some results not proven for all adhesive categories
- We use categories $\mathbb{S}\mathrm{et}^{\mathbb{S}}$ of functors $\mathbb{S} \to \mathbb{S}\mathrm{et}$ with natural transformations as arrows (essentially presheaves)
- Generalizes graphs and graph structures

$$\mathbb{G}\mathrm{raph} = \mathbb{S}\mathrm{et}^{\mathbb{G}} \qquad \mathbb{G} = \quad V \underset{t}{\overset{s}{\rightrightarrows}} E$$

# Background: Set-Valued Functor Categories

- Some results not proven for all adhesive categories
- We use categories $\mathbb{S}\mathrm{et}^{\mathbb{S}}$ of functors $\mathbb{S} \to \mathbb{S}\mathrm{et}$ with natural transformations as arrows (essentially presheaves)
- Generalizes graphs and graph structures

$$\mathbb{G}\mathrm{raph} = \mathbb{S}\mathrm{et}^{\mathbb{G}} \qquad \mathbb{G} = \quad V \underset{t}{\overset{s}{\rightrightarrows}} E$$

- Limits, colimits, monos and epis are pointwise

# Background: Set-Valued Functor Categories

- Some results not proven for all adhesive categories
- We use categories $\mathbb{S}\mathrm{et}^{\mathbb{S}}$ of functors $\mathbb{S} \to \mathbb{S}\mathrm{et}$ with natural transformations as arrows (essentially presheaves)
- Generalizes graphs and graph structures

$$\mathbb{G}\mathrm{raph} = \mathbb{S}\mathrm{et}^{\mathbb{G}} \qquad \mathbb{G} = \quad V \underset{t}{\overset{s}{\rightrightarrows}} E$$

- Limits, colimits, monos and epis are pointwise
- Always adhesive

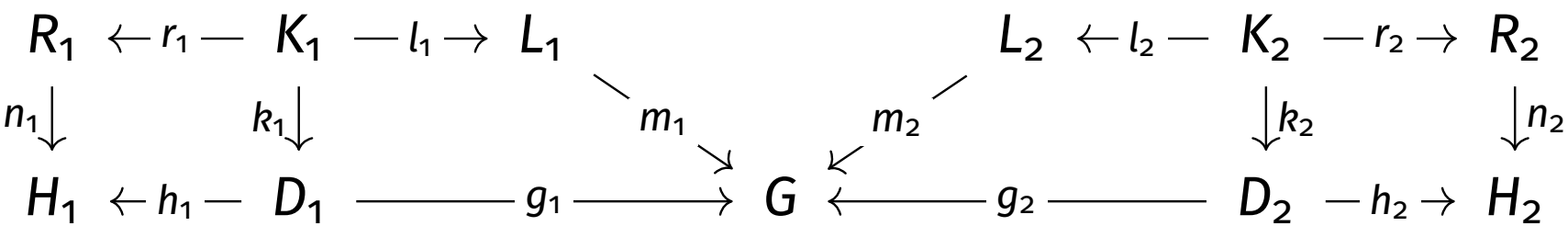
# Outline

1. Characterize conflict between transformations
2. Useful properties of the characterization
3. Compare with conflict reasons of Lambers, Ehrig, and Orejas (2008)
4. Relate to initial conflicts

# Essential Condition of Parallel Independence

Corradini et al. (2018)

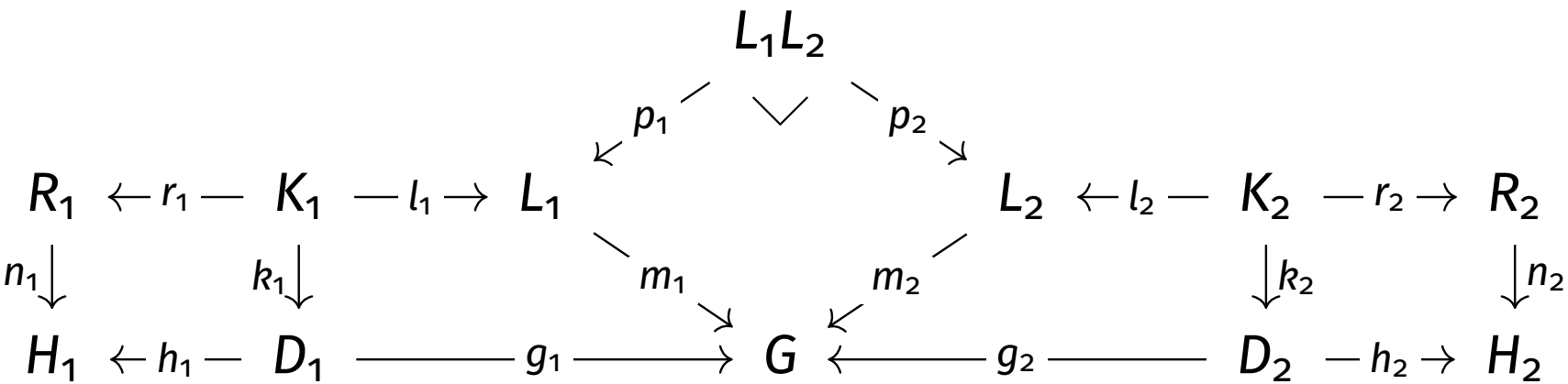
$$H_1 \overset{t_1}{\Longleftarrow} G \overset{t_2}{\Longrightarrow} H_2$$



# Essential Condition of Parallel Independence

Corradini et al. (2018)

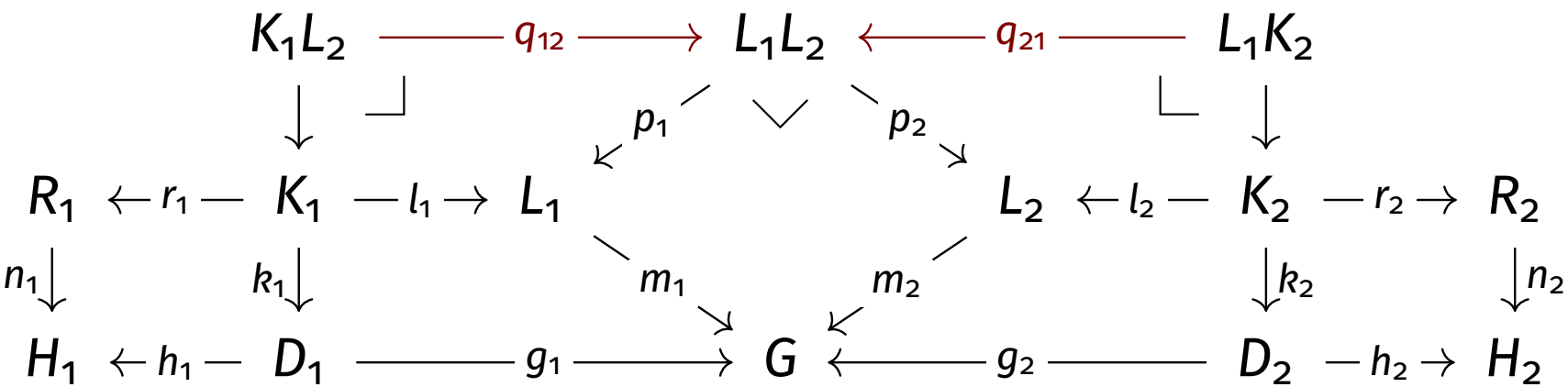
$$H_1 \overset{t_1}{\Longleftarrow} G \overset{t_2}{\Longrightarrow} H_2$$



# Essential Condition of Parallel Independence

Corradini et al. (2018)

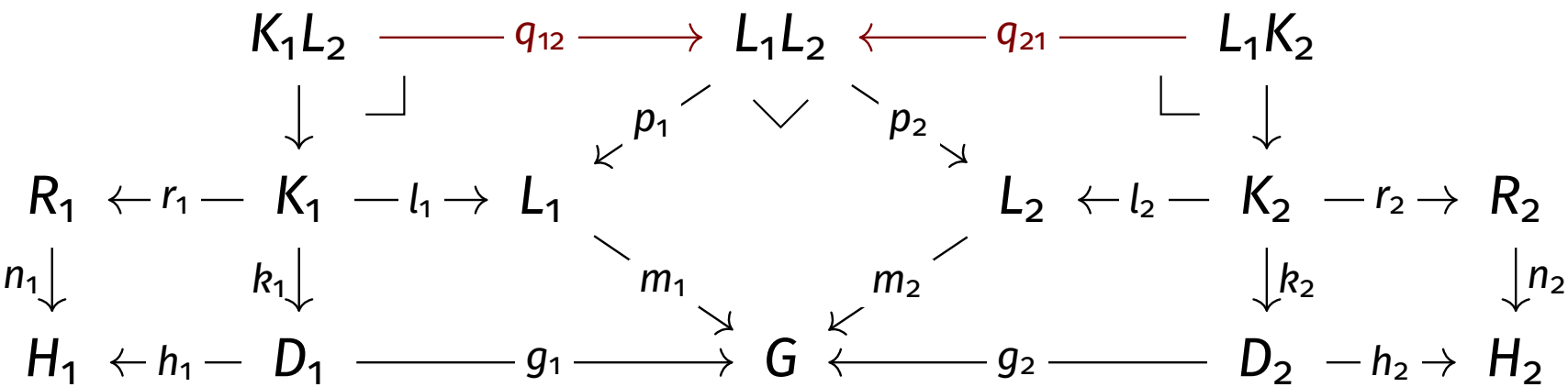
$$H_1 \overset{t_1}{\Longleftarrow} G \overset{t_2}{\Longrightarrow} H_2$$



# Essential Condition of Parallel Independence

Corradini et al. (2018)

$$H_1 \overset{t_1}{\Longleftarrow} G \overset{t_2}{\Longrightarrow} H_2$$

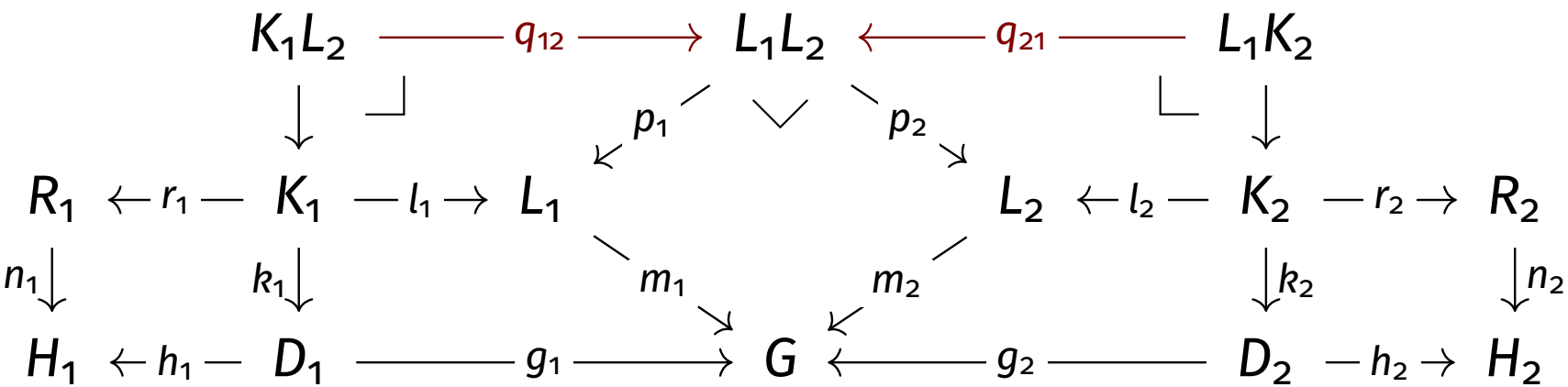



- Both morphisms iso $\Rightarrow$ parallel independence

# Essential Condition of Parallel Independence

Corradini et al. (2018)

$$H_1 \overset{t_1}{\Longleftarrow} G \overset{t_2}{\Longrightarrow} H_2$$

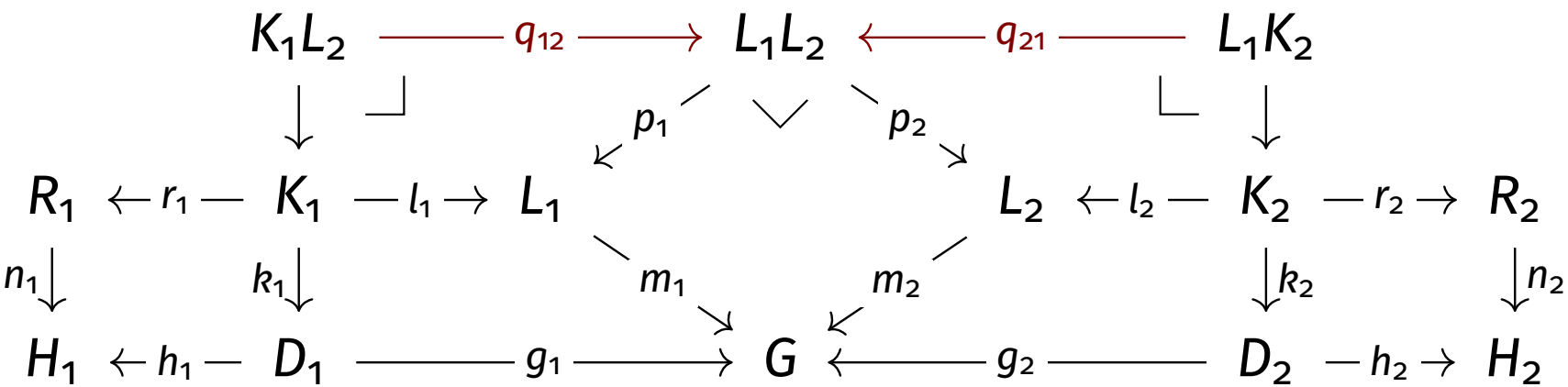



- Both morphisms iso $\Rightarrow$ parallel independence
- Either morphism not iso $\Rightarrow$ conflict

# Essential Condition of Parallel Independence

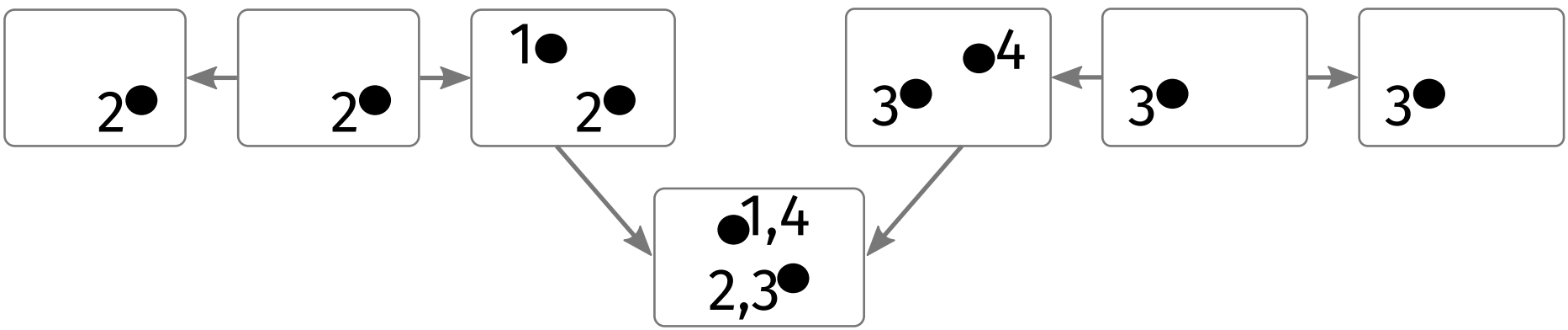
Corradini et al. (2018)

$$H_1 \overset{t_1}{\Longleftarrow} G \overset{t_2}{\Longrightarrow} H_2$$

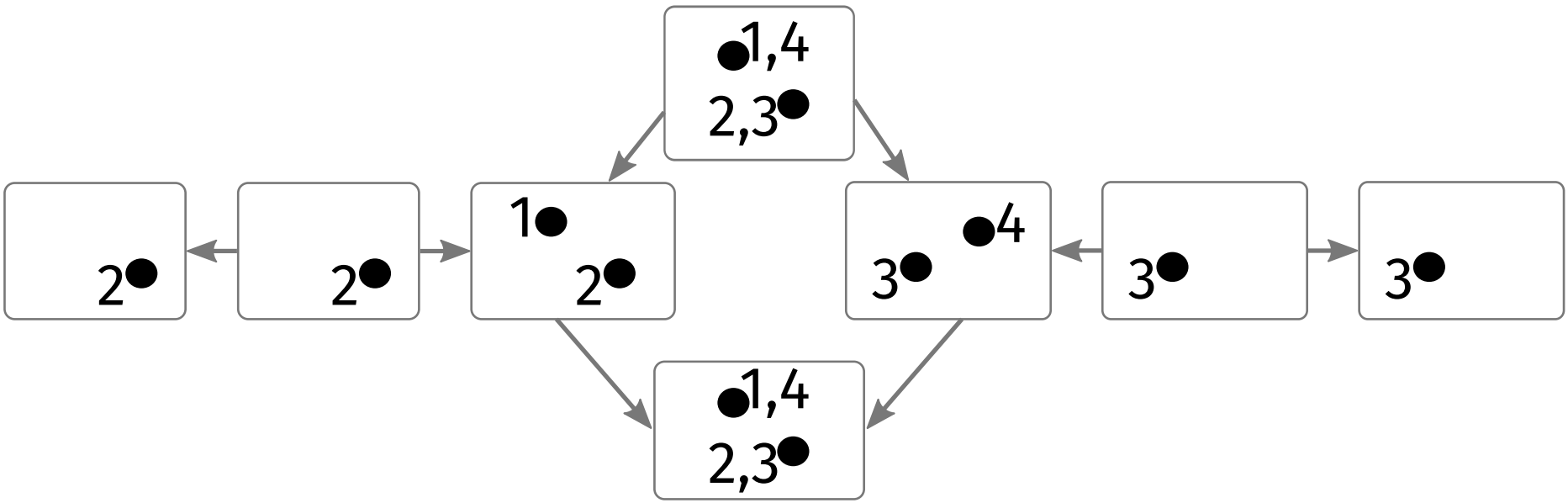



- Both morphisms iso $\Rightarrow$ parallel independence
- Either morphism not iso $\Rightarrow$ conflict
- $K_1L_2 \to L_1L_2$ not iso $\Rightarrow$ $t_1$ disables $t_2$

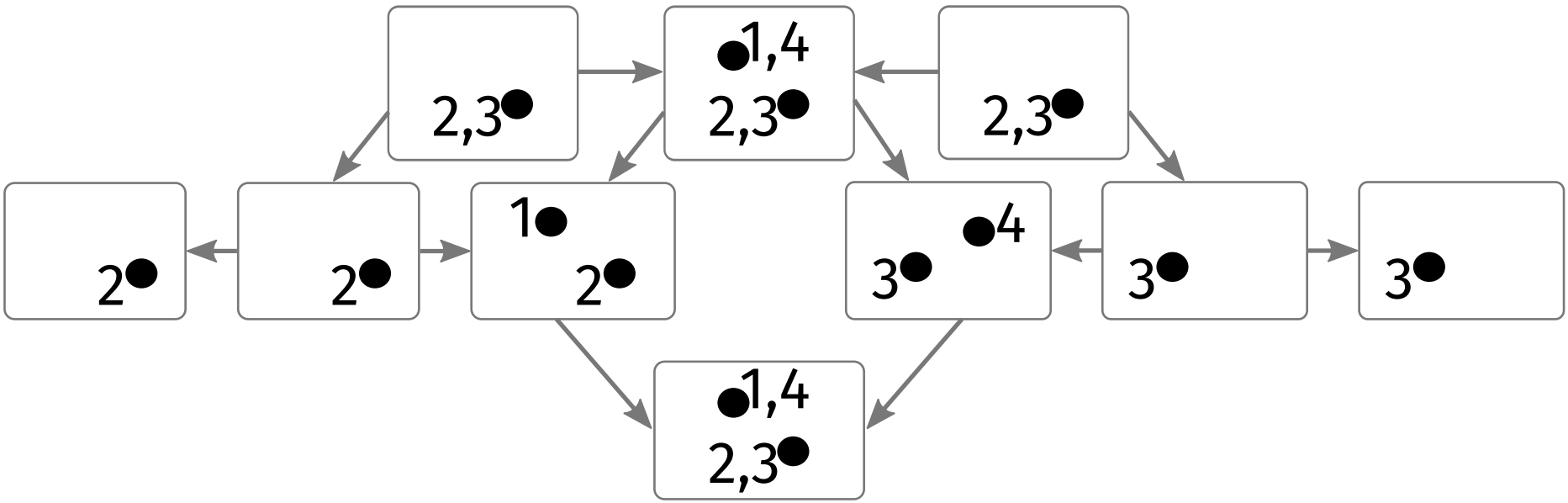
# Example: Conflict



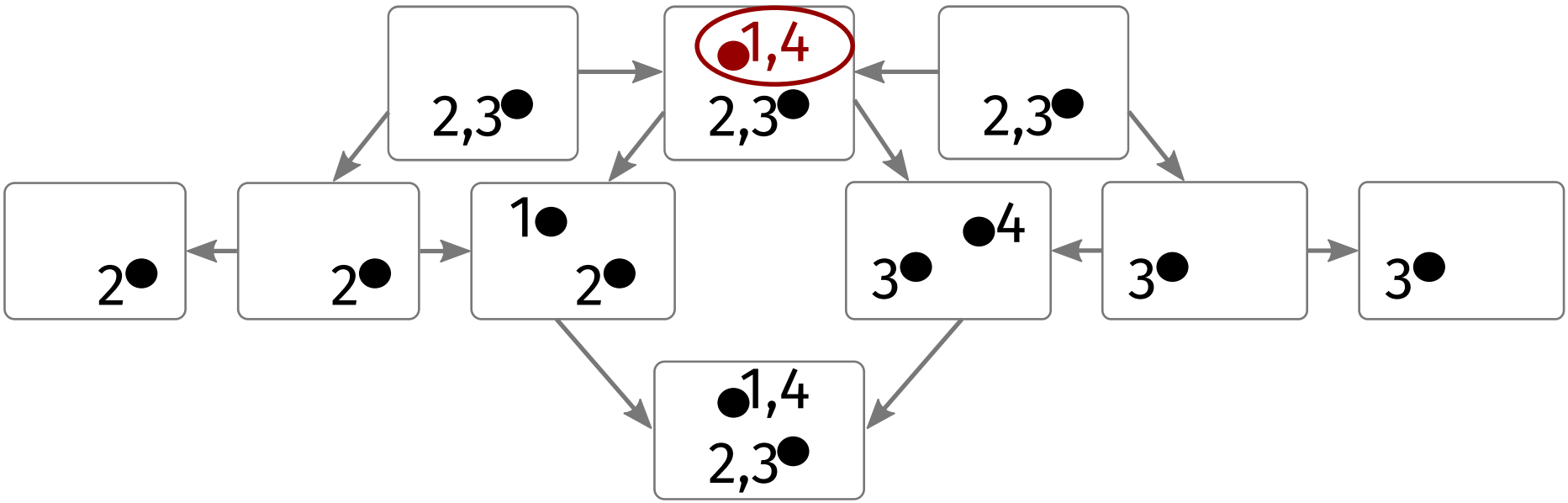
# Example: Conflict



# Example: Conflict



# Example: Conflict



# Determining the Root Cause

- Useful concept: initial pushout over $f : X \to Y$

$$\begin{array}{ccc} B & \rightarrowtail b \to & X \\ \bar{f}\downarrow & & \downarrow f \\ C & \rightarrowtail c \to & Y \end{array}$$

- "Categorical diff" for a morphism

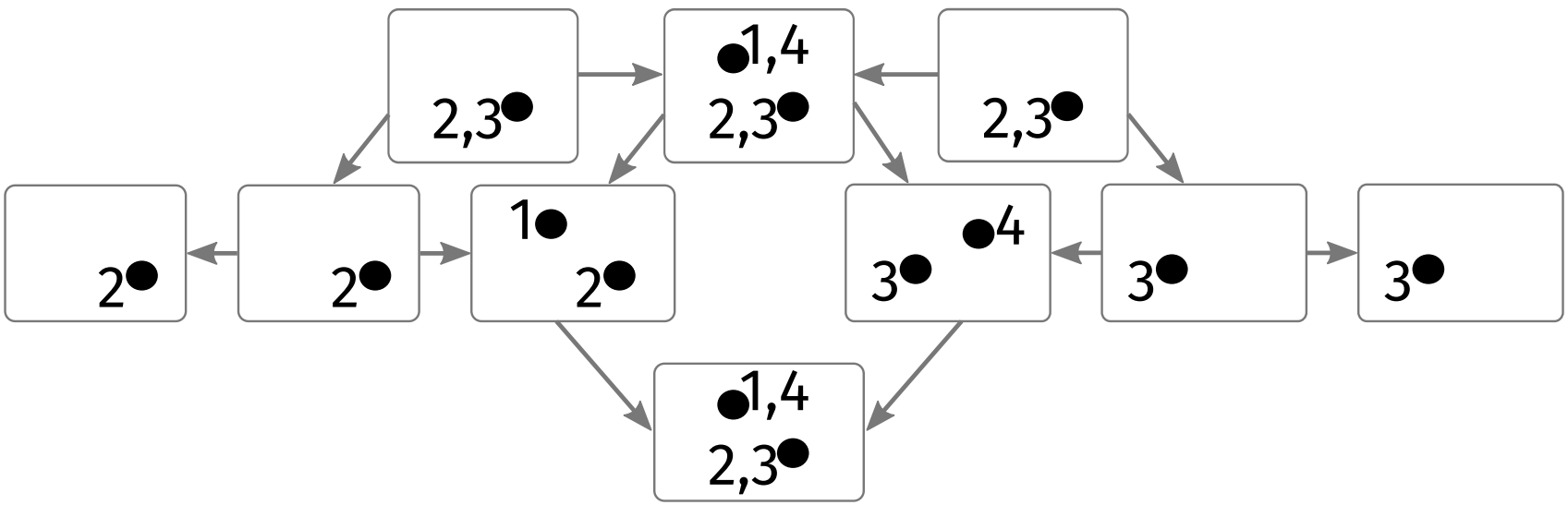
# Determining the Root Cause

- Useful concept: initial pushout over $f : X \to Y$

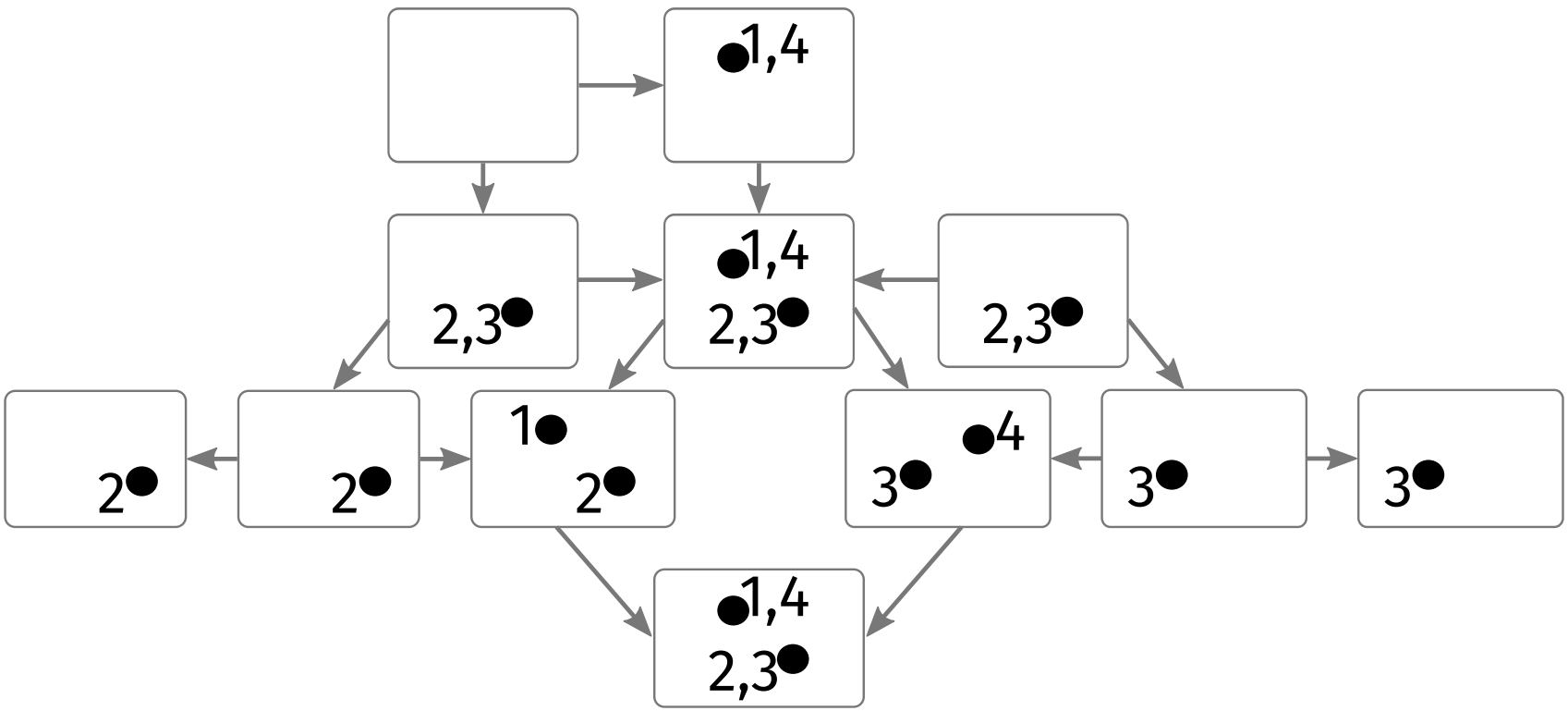
$$\begin{array}{ccc} B & \rightarrowtail b \to & X \\ {\scriptstyle \bar{f}}\downarrow & & \downarrow{\scriptstyle f} \\ C & \rightarrowtail c \to & Y \end{array}$$

- "Categorical diff" for a morphism
- Context $c : C \rightarrowtail Y$ contains "modified stuff"
- Boundary $b : B \rightarrowtail C$ contains "points of contact"

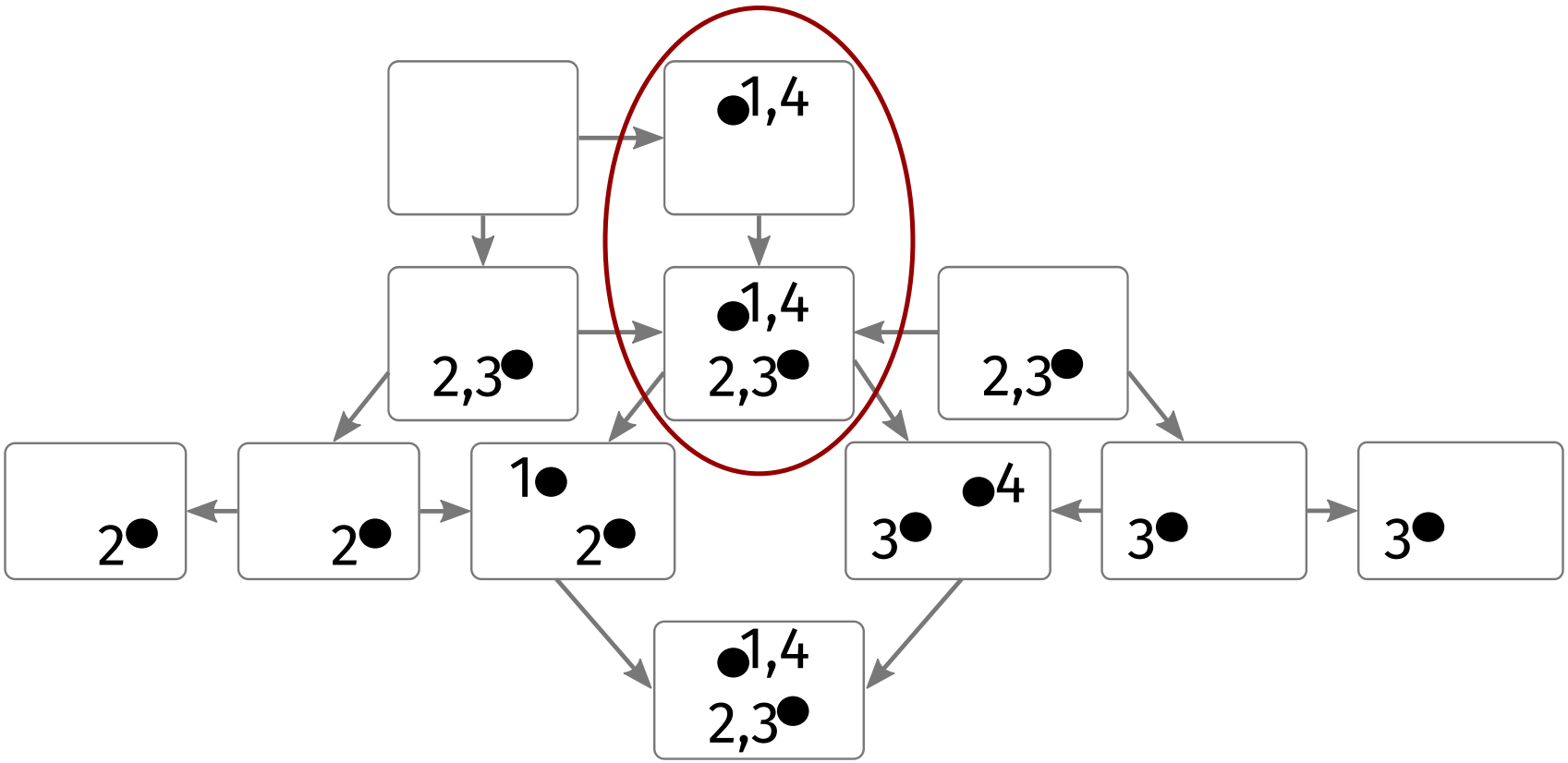
# Example: Initial Pushout



# Example: Initial Pushout



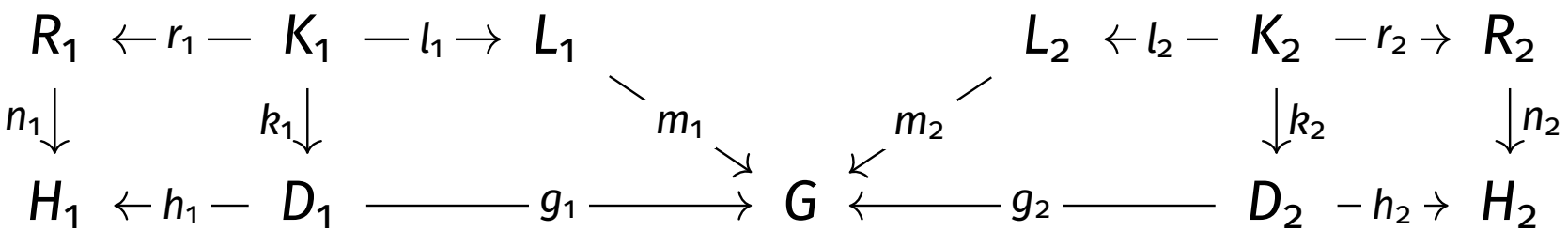
# Example: Initial Pushout



# Conflict and Disabling Essences

## Definition

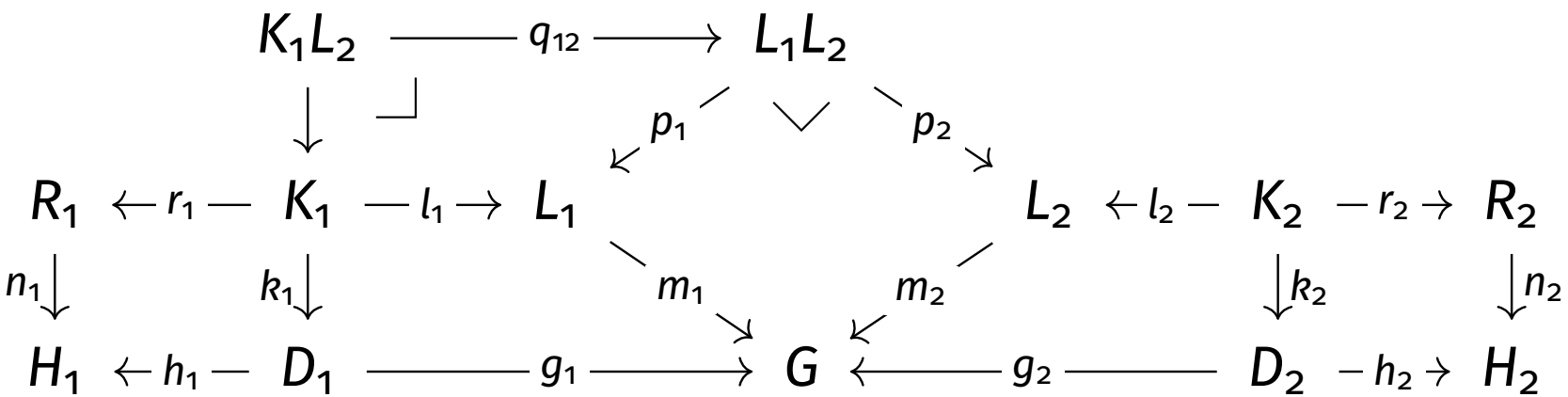
Given transformations $(t_1, t_2) : H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$:



# Conflict and Disabling Essences

## Definition

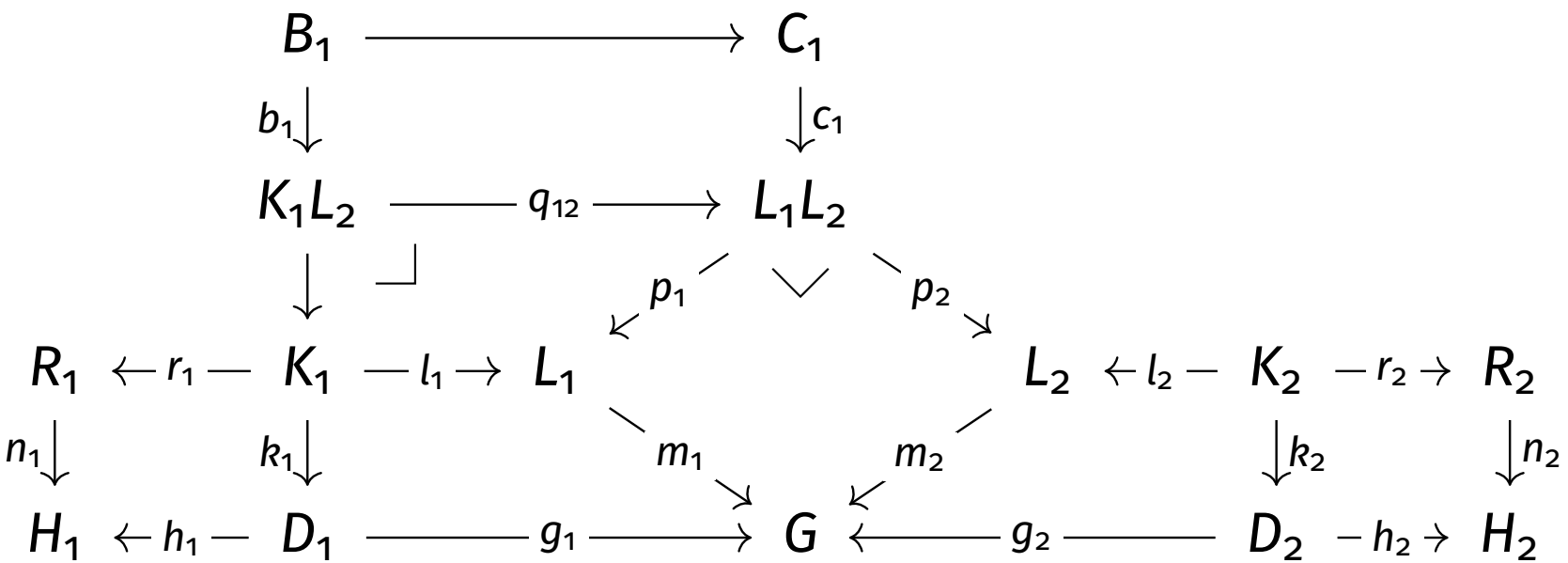
Given transformations $(t_1, t_2) : H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$:



# Conflict and Disabling Essences

## Definition

Given transformations $(t_1, t_2) : H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$:

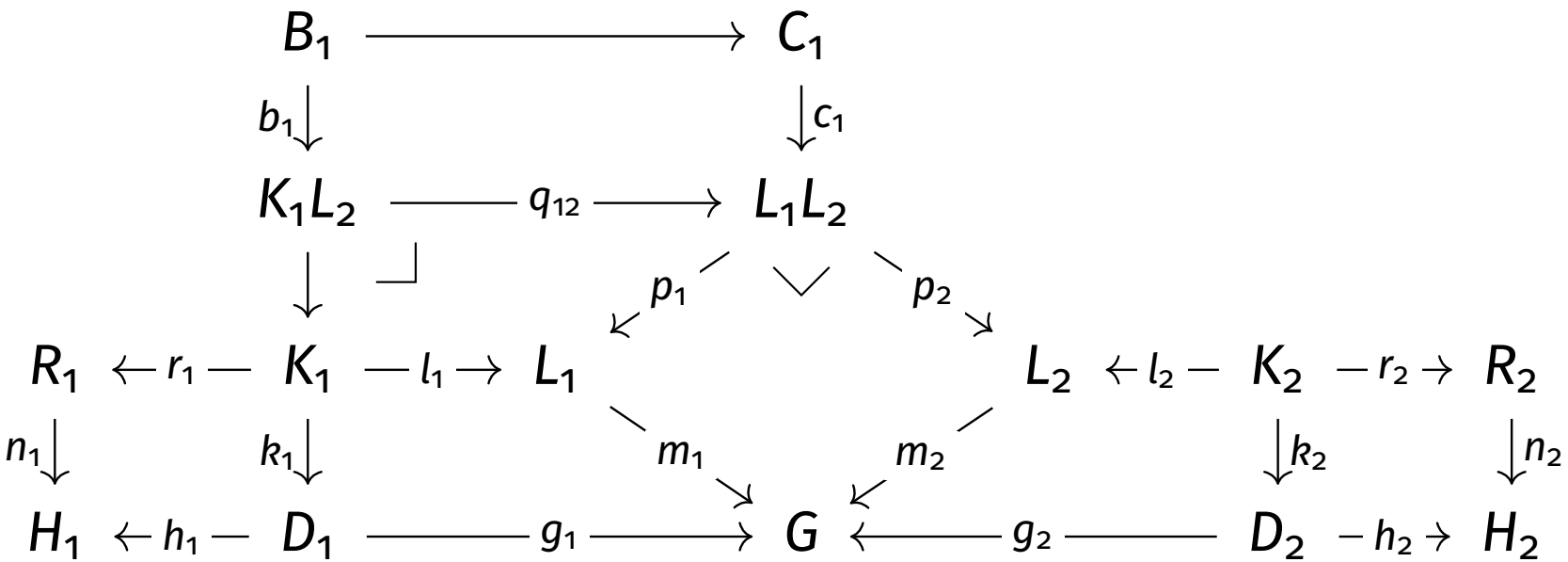


# Conflict and Disabling Essences

## Definition

Given transformations $(t_1, t_2) : H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$:

- Disabling essence for $(t_1, t_2)$ is $c_1 \in \mathbf{Sub}(L_1 L_2)$

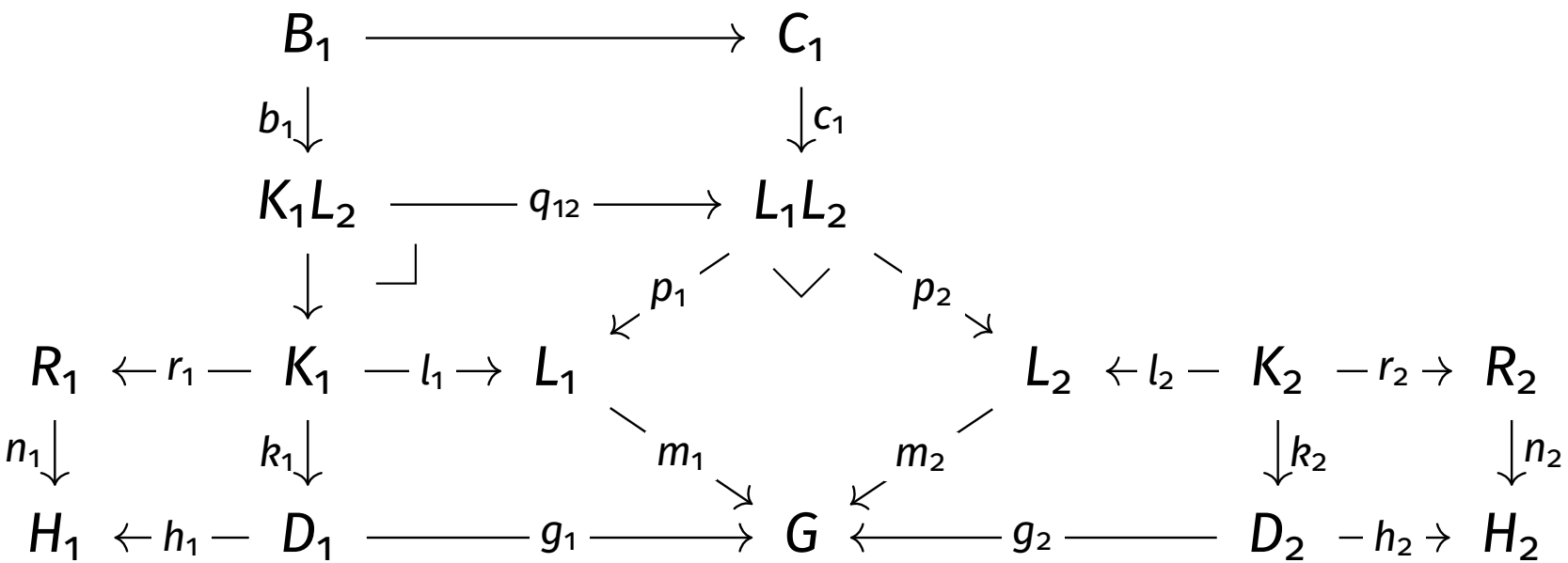


# Conflict and Disabling Essences

## Definition

Given transformations $(t_1, t_2) : H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$:

- Disabling essence for $(t_1, t_2)$ is $c_1 \in \mathbf{Sub}(L_1 L_2)$
- Disabling essence for $(t_2, t_1)$ is $c_2 \in \mathbf{Sub}(L_1 L_2)$

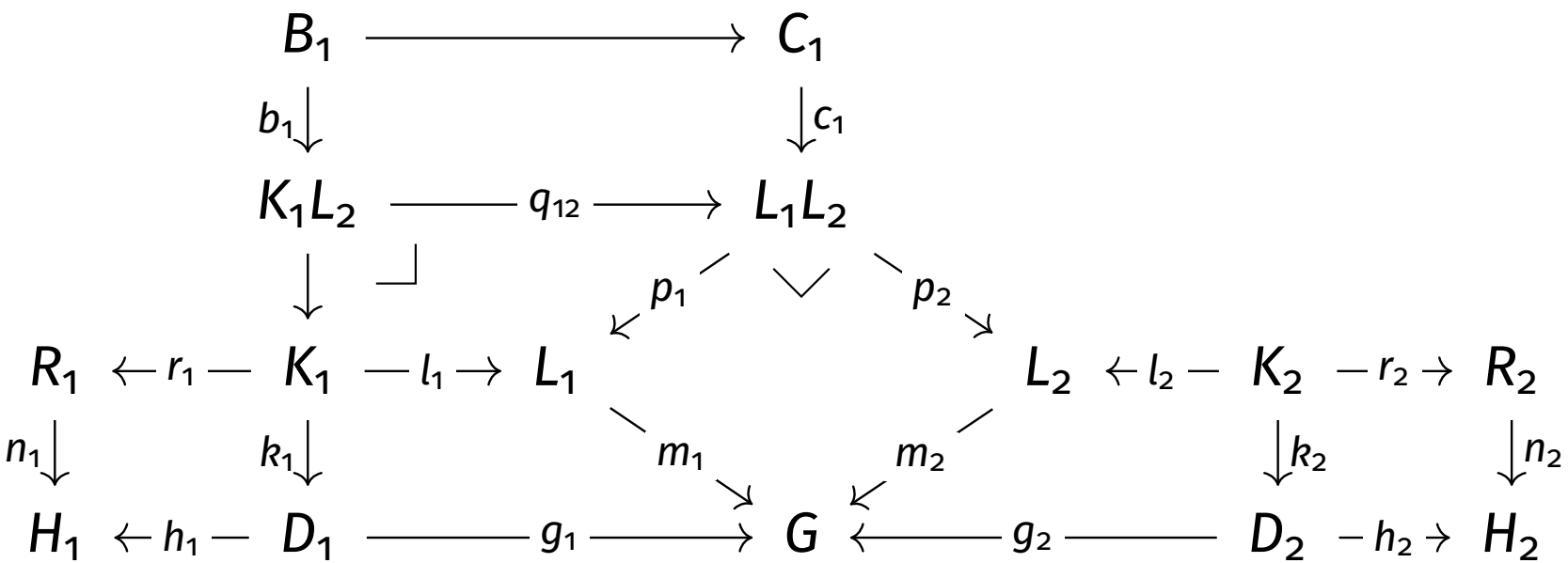


# Conflict and Disabling Essences

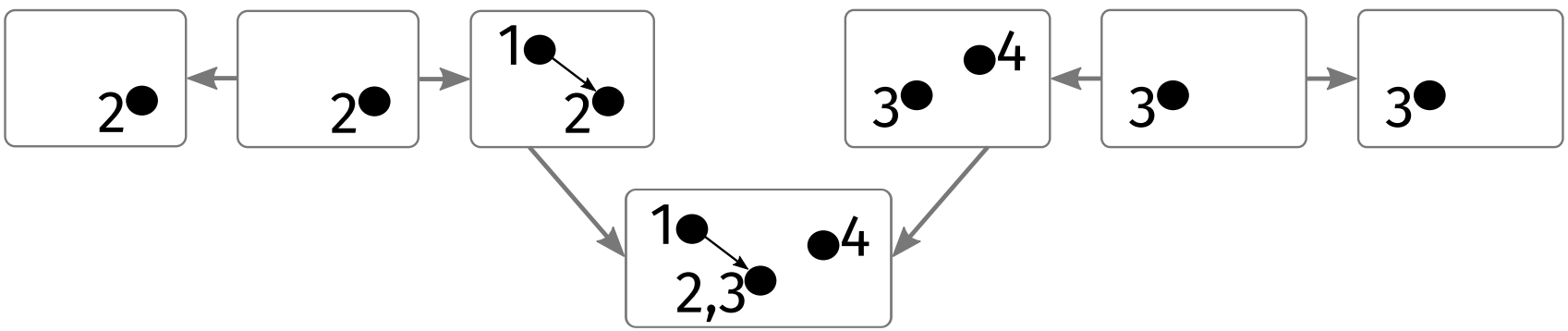
## Definition

Given transformations $(t_1, t_2) : H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$:

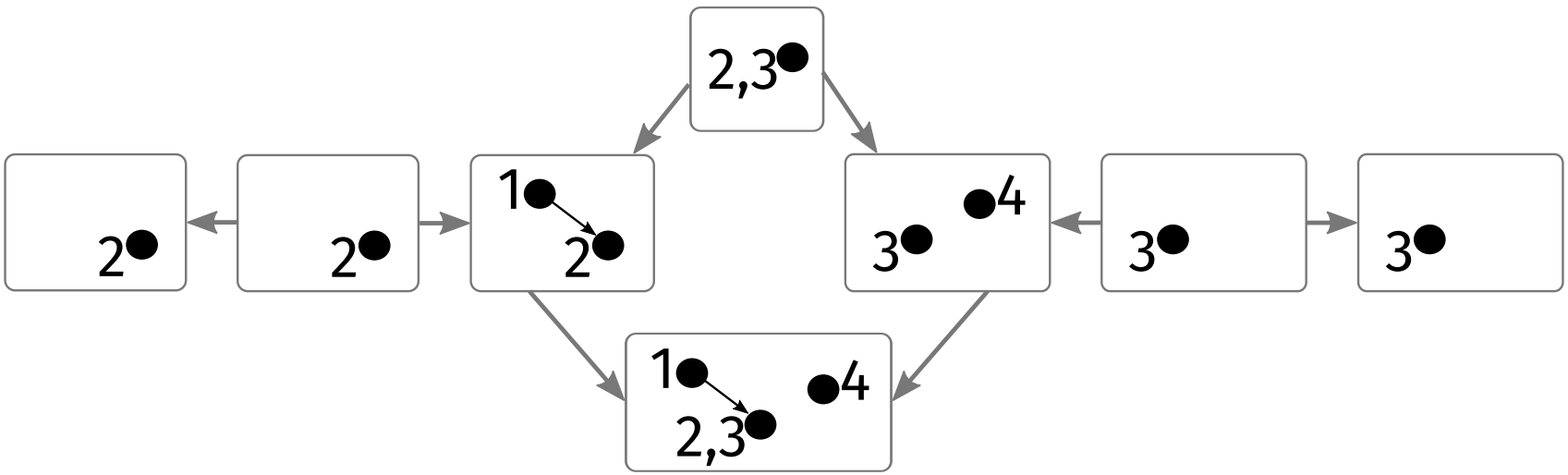
- Disabling essence for $(t_1, t_2)$ is $c_1 \in \mathbf{Sub}(L_1 L_2)$
- Disabling essence for $(t_2, t_1)$ is $c_2 \in \mathbf{Sub}(L_1 L_2)$
- Conflict essence for $(t_1, t_2)$ is $c = c_1 \cup c_2$



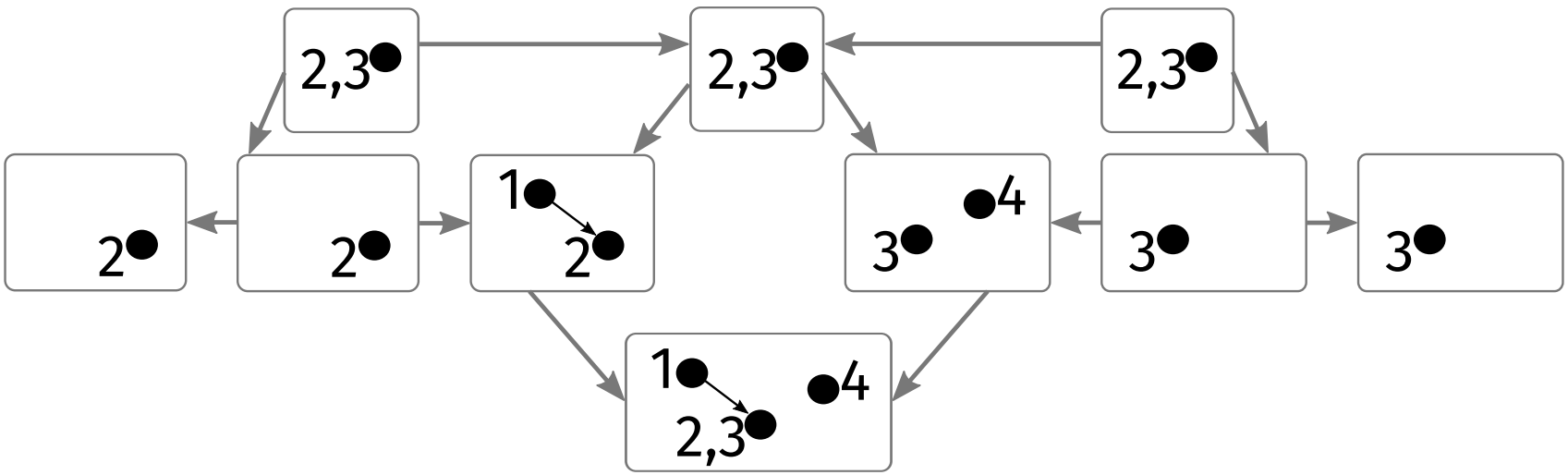
# Example: Parallel Independence



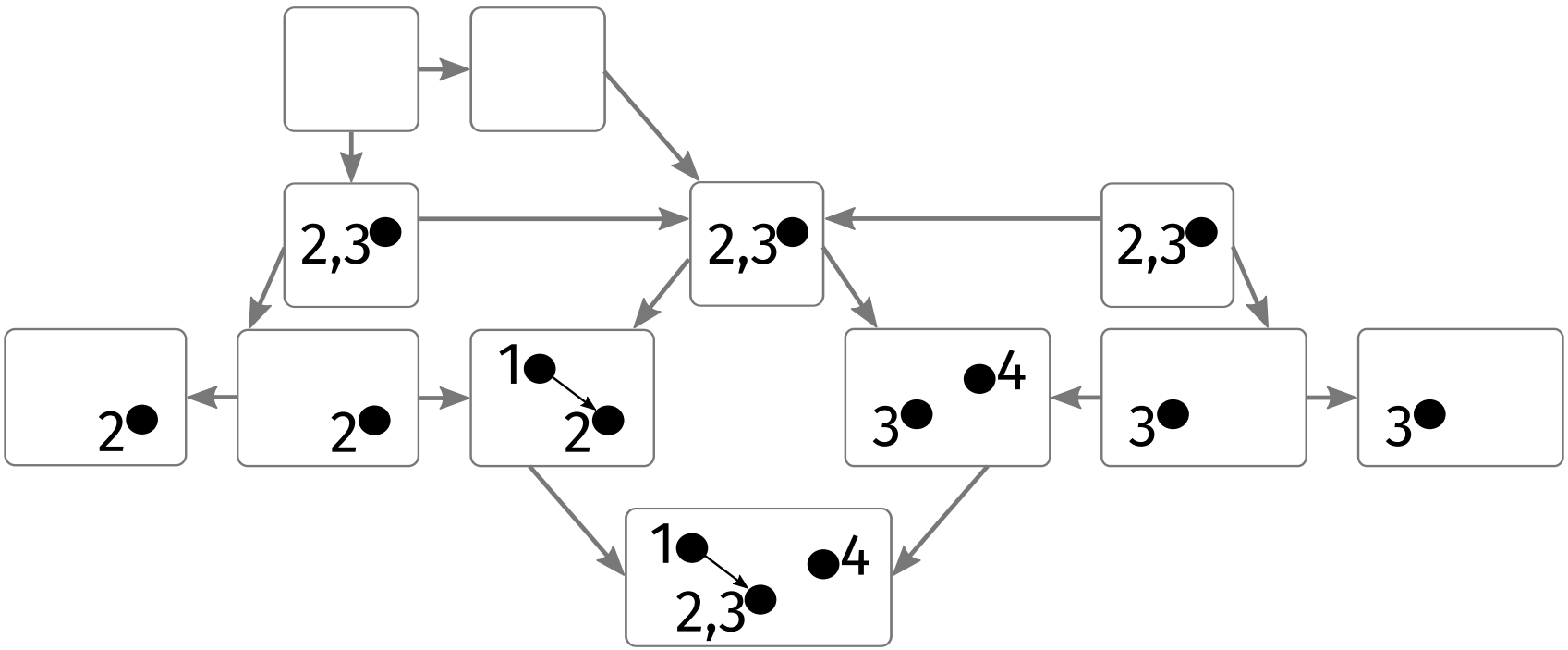
# Example: Parallel Independence



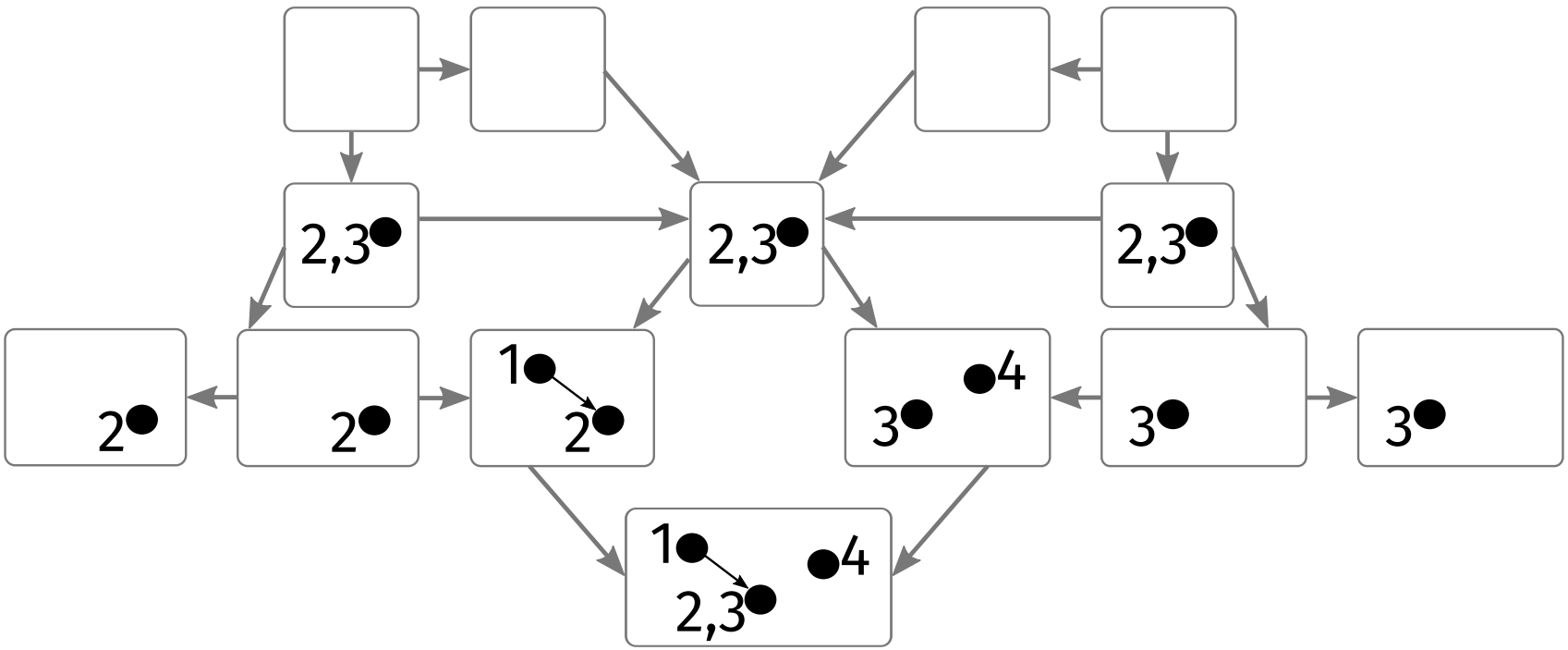
# Example: Parallel Independence



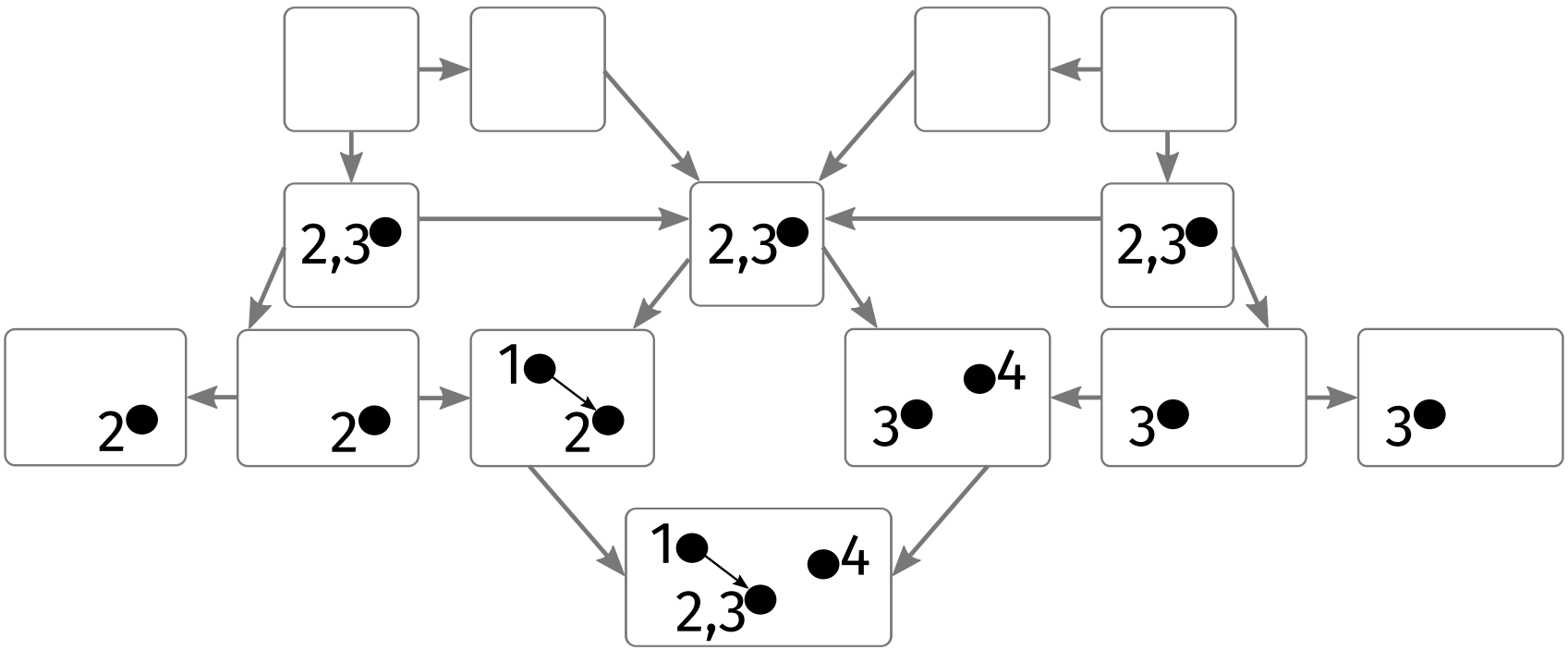
# Example: Parallel Independence



# Example: Parallel Independence



# Example: Parallel Independence




No conflict $\Longrightarrow$ no element caused a conflict

# Empty Essences

Recall: bottom subobject generalizes “emptiness”

# Empty Essences

Recall: bottom subobject generalizes "emptiness"

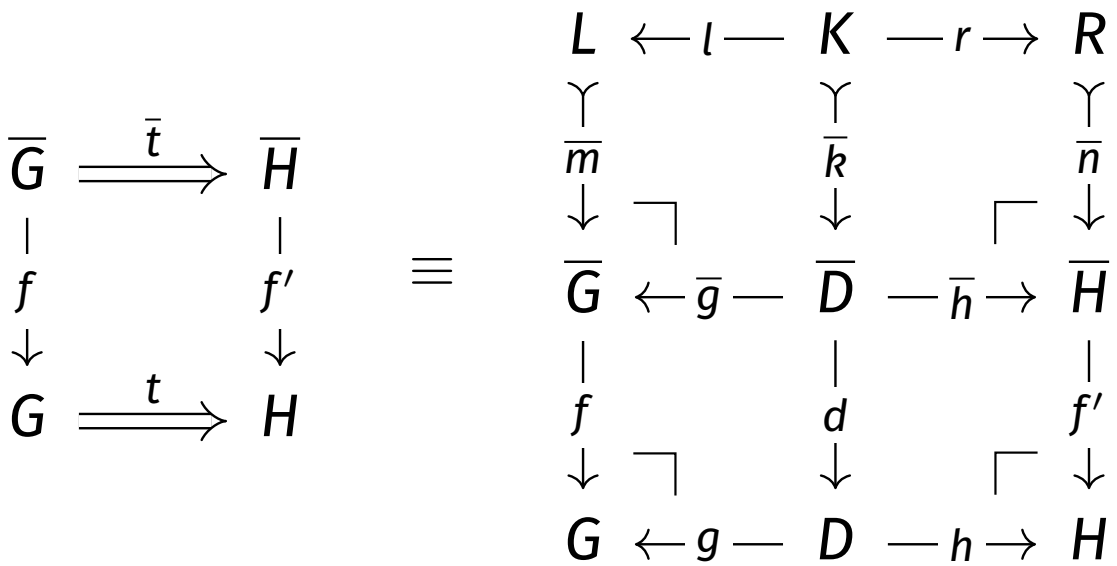
Consider $(t_1, t_2) : H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$

**Theorem**

*The conflict essence for* $(t_1, t_2)$ *is* $\bot \in \mathbf{Sub}(L_1 L_2)$
*if and only if*
$t_1$ *and* $t_2$ *are parallel independent.*

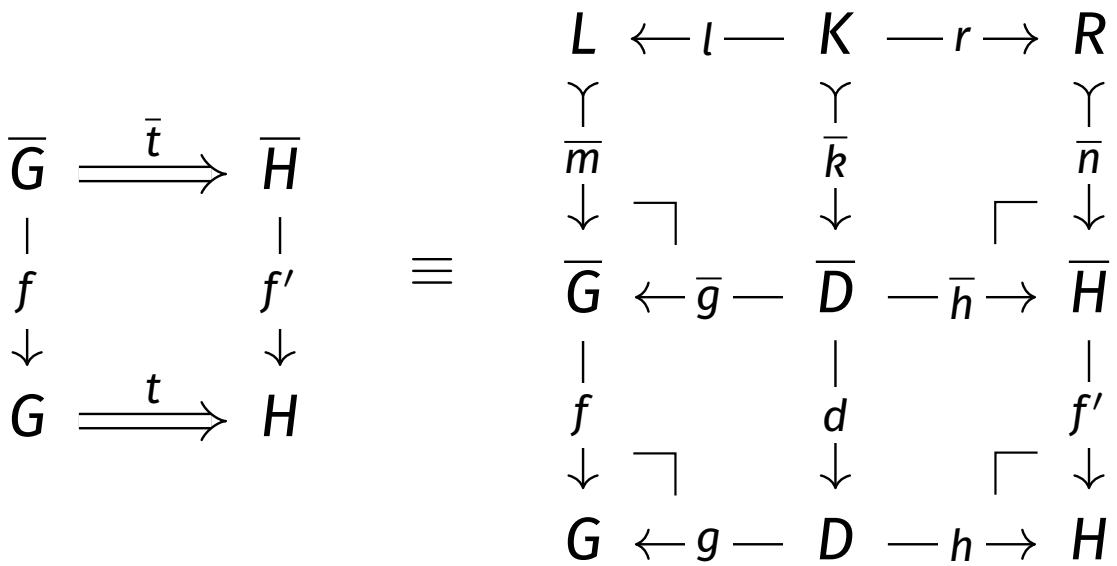
# Extension

- Same transformation in “larger context”



# Extension

- Same transformation in “larger context”




- Lower pushouts ensure $t$ behaves like $\overline{t}$

# Essence Inheritance

**Theorem**

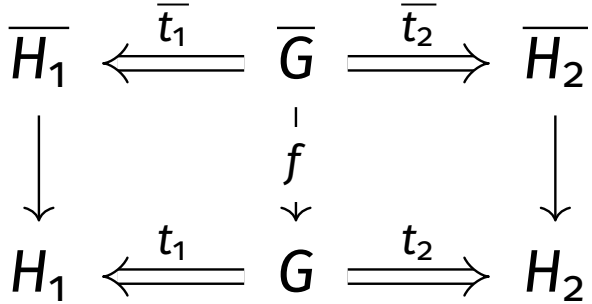
*If extension diagrams below exist,* $(t_1, t_2)$ *and* $(\overline{t_1}, \overline{t_2})$ *have the same disabling and conflict essences.*

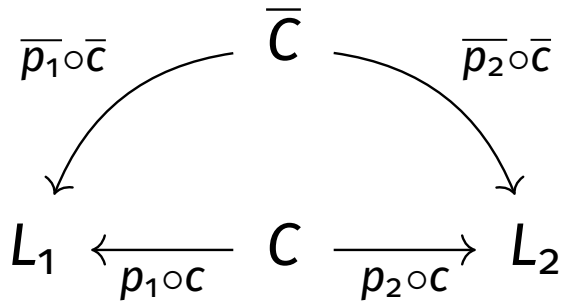
$$\begin{array}{ccccc} \overline{H_1} & \xleftarrow{\overline{t_1}} & \overline{G} & \xrightarrow{\overline{t_2}} & \overline{H_2} \\ \downarrow & & \downarrow f & & \downarrow \\ H_1 & \xleftarrow{t_1} & G & \xrightarrow{t_2} & H_2 \end{array}$$

# Essence Inheritance

**Theorem**

*If extension diagrams below exist,* $(t_1, t_2)$ *and* $(\overline{t_1}, \overline{t_2})$ *have the same disabling and conflict essences.*

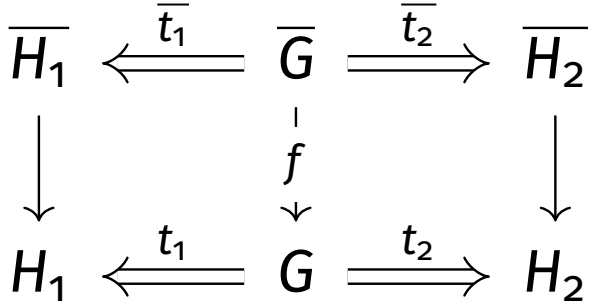


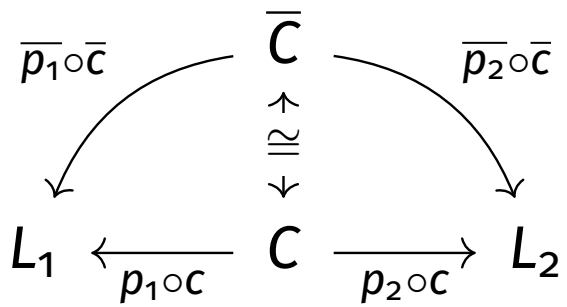



# Essence Inheritance

**Theorem**

*If extension diagrams below exist, $(t_1, t_2)$ and $(\overline{t_1}, \overline{t_2})$ have the same disabling and conflict essences.*



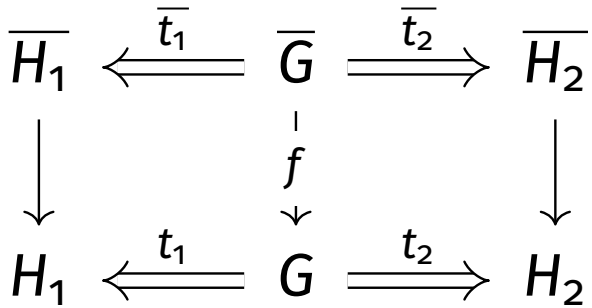



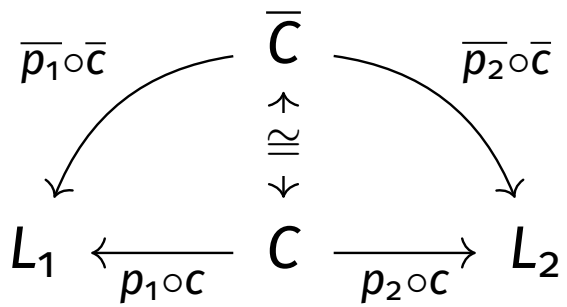
# Essence Inheritance

In categories of set-valued functors
(also graphs, typed graphs...)

**Theorem**

*If extension diagrams below exist,* $(t_1, t_2)$ *and* $(\overline{t_1}, \overline{t_2})$ *have the same disabling and conflict essences.*



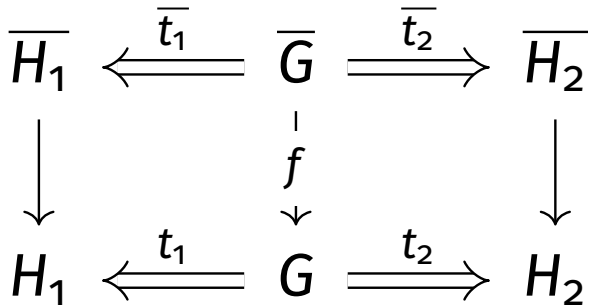



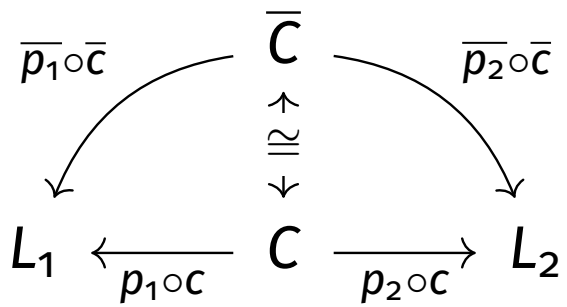
# Essence Inheritance

In categories of set-valued functors
(also graphs, typed graphs...)

**Theorem**

*If extension diagrams below exist,* $(t_1, t_2)$ *and* $(\overline{t_1}, \overline{t_2})$ *have the same disabling and conflict essences.*




Conflicts are preserved and reflected by extension.

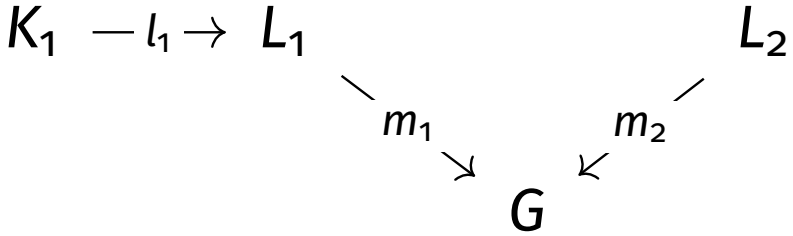
# Disabling Reasons

Essences are not the first proposed characterization

Given transformations $(t_1, t_2) : H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$

**Definition (Lambers, Ehrig, and Orejas 2008)**

The **disabling reason** $L_1 \leftarrow S_1 \rightarrow L_2$ for $(t_1, t_2)$



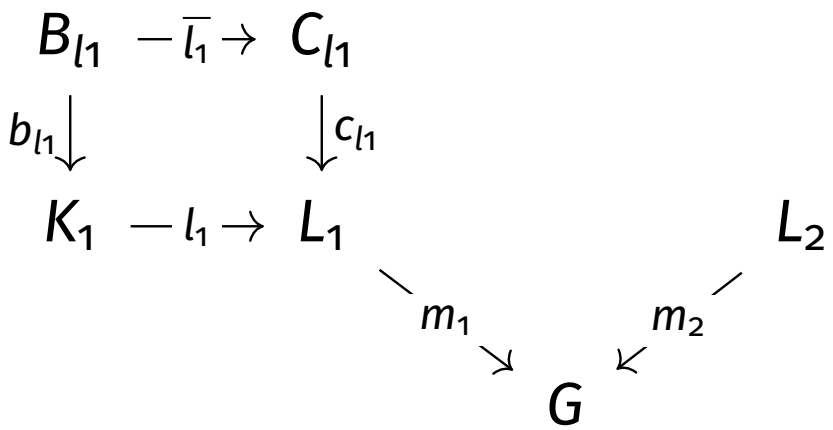
# Disabling Reasons

Essences are not the first proposed characterization

Given transformations $(t_1, t_2) : H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$

**Definition (Lambers, Ehrig, and Orejas 2008)**

The **disabling reason** $L_1 \leftarrow S_1 \rightarrow L_2$ for $(t_1, t_2)$ is obtained from the initial pushout over $l_1$,



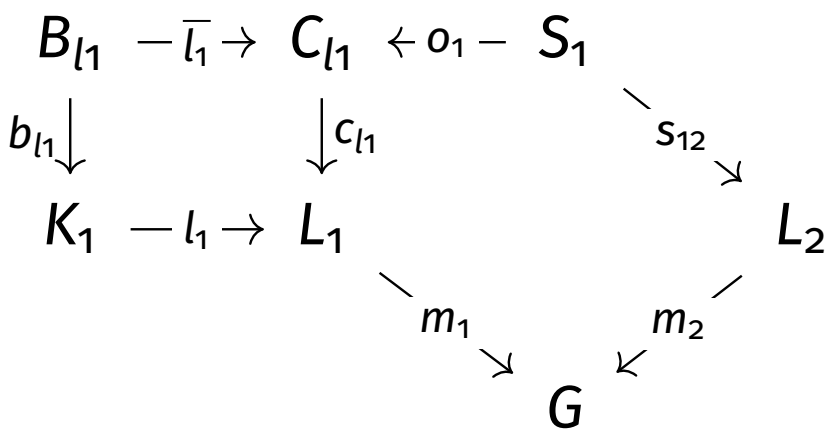
# Disabling Reasons

Essences are not the first proposed characterization

Given transformations $(t_1, t_2) : H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$

**Definition (Lambers, Ehrig, and Orejas 2008)**

The **disabling reason** $L_1 \leftarrow S_1 \rightarrow L_2$ for $(t_1, t_2)$ is obtained from the initial pushout over $l_1$, then pullback of $(m_1 \circ c_{l1}, m_2)$.



# Disabling Reasons

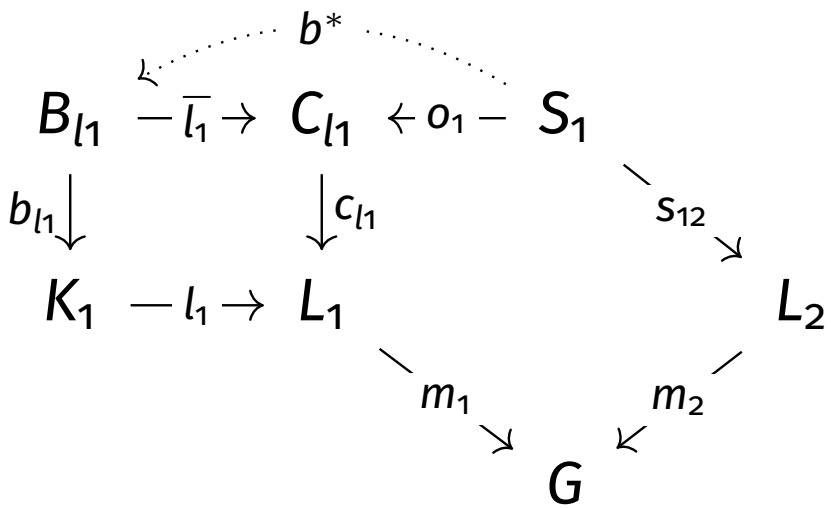
Essences are not the first proposed characterization

Given transformations $(t_1, t_2) : H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$

**Definition (Lambers, Ehrig, and Orejas 2008)**

The **disabling reason** $L_1 \leftarrow S_1 \rightarrow L_2$ for $(t_1, t_2)$ is obtained from the initial pushout over $l_1$, then pullback of $(m_1 \circ c_{l1}, m_2)$.

**Conflict condition:**
There is no $b^*$ making diagram commute.



# Disabling Reasons

Essences are not the first proposed characterization

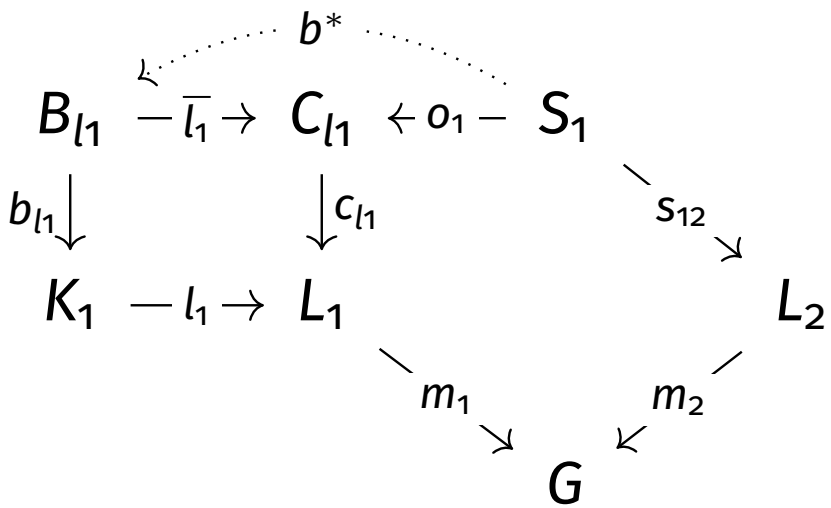
Given transformations $(t_1, t_2) : H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$

**Definition (Lambers, Ehrig, and Orejas 2008)**

The **disabling reason** $L_1 \leftarrow S_1 \rightarrow L_2$ for $(t_1, t_2)$ is obtained from the initial pushout over $l_1$, then pullback of $(m_1 \circ c_{l1}, m_2)$.

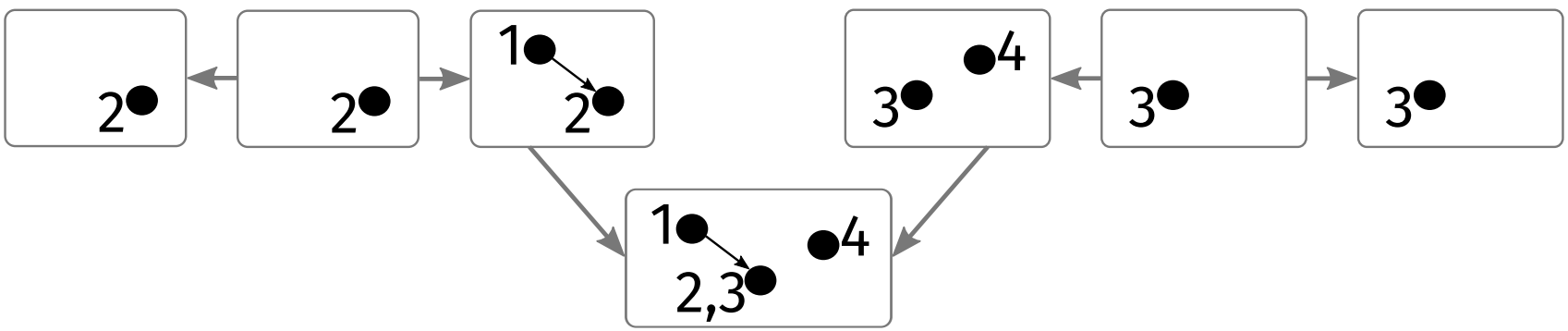
**Conflict condition:**
There is no $b^*$ making diagram commute.

**Conflict reason** is union of *relevant* disabling reasons.



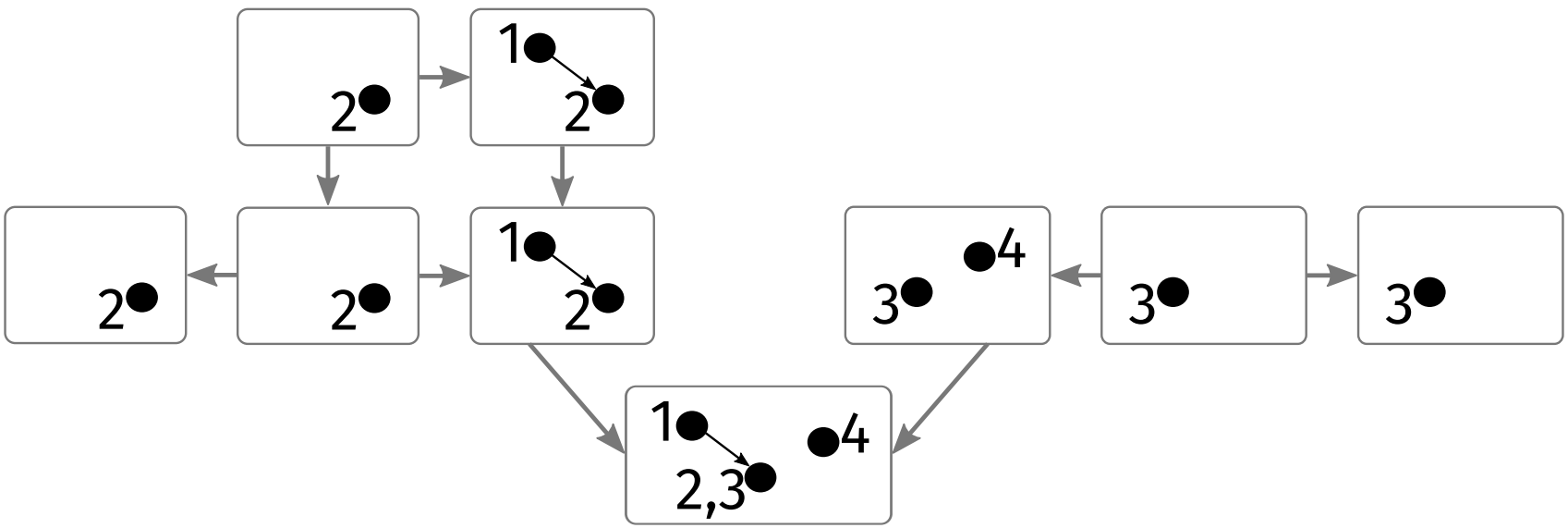
# Comparing Reasons and Essences

- Non-empty reasons exist even with parallel independence



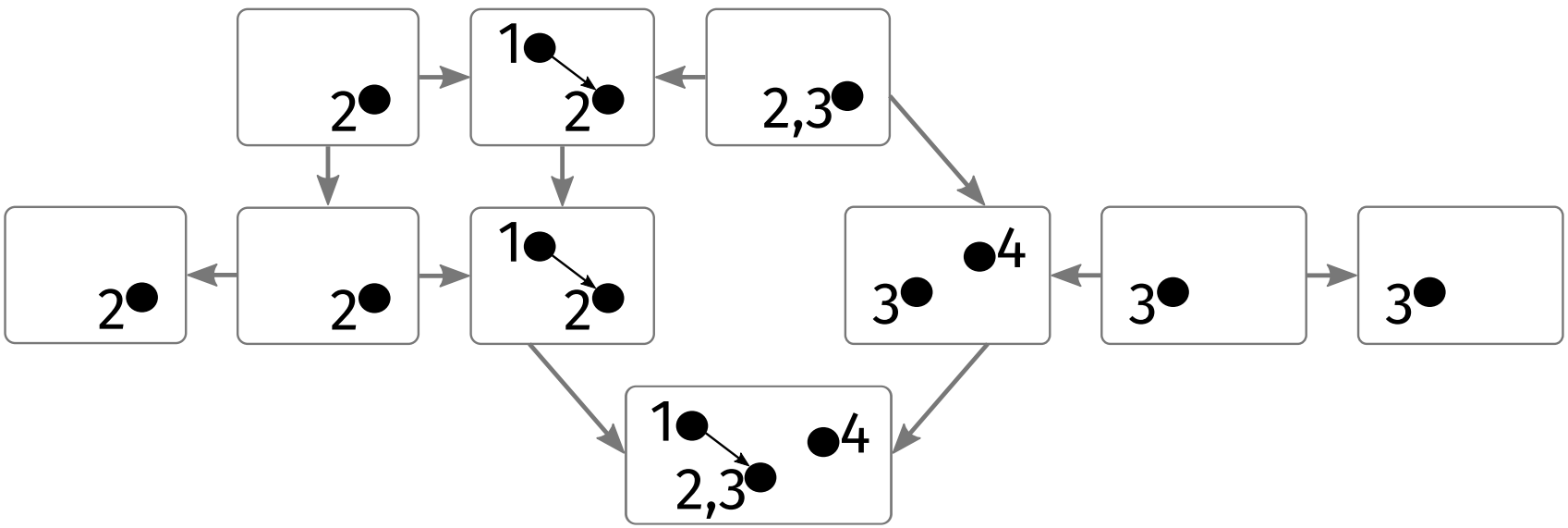
# Comparing Reasons and Essences

- Non-empty reasons exist even with parallel independence



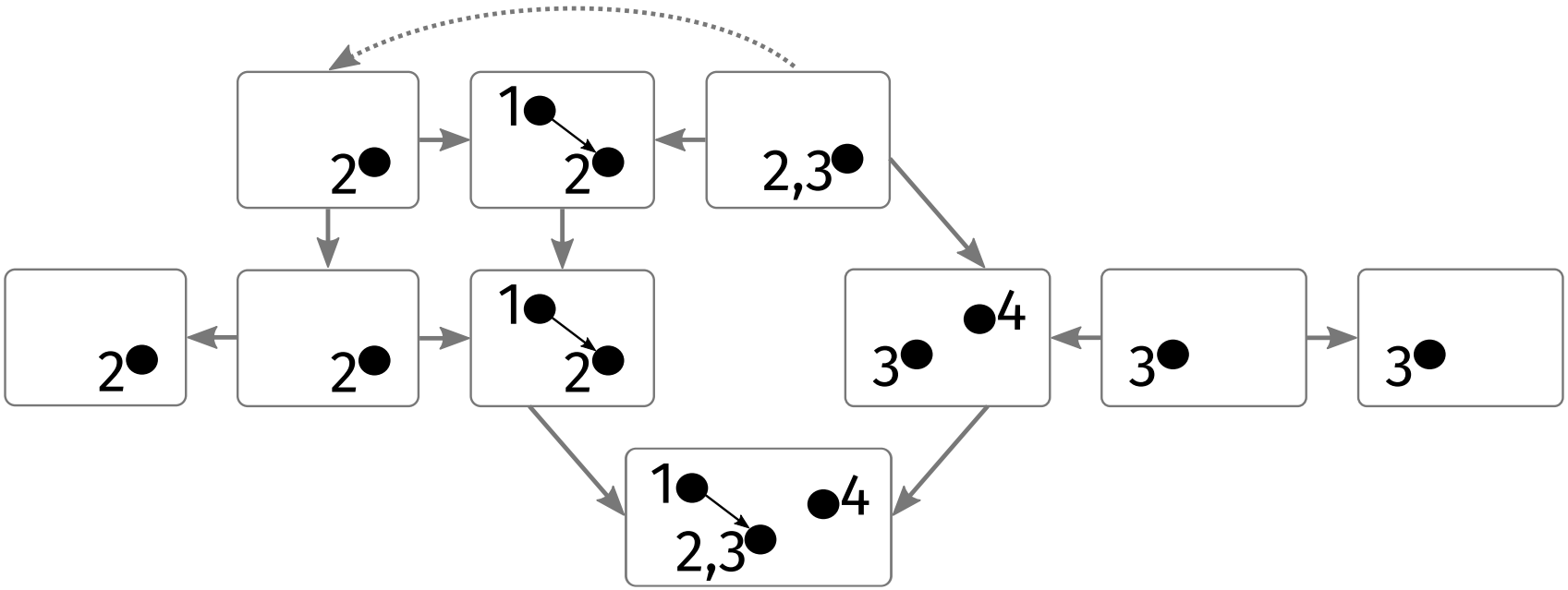
# Comparing Reasons and Essences

- Non-empty reasons exist even with parallel independence



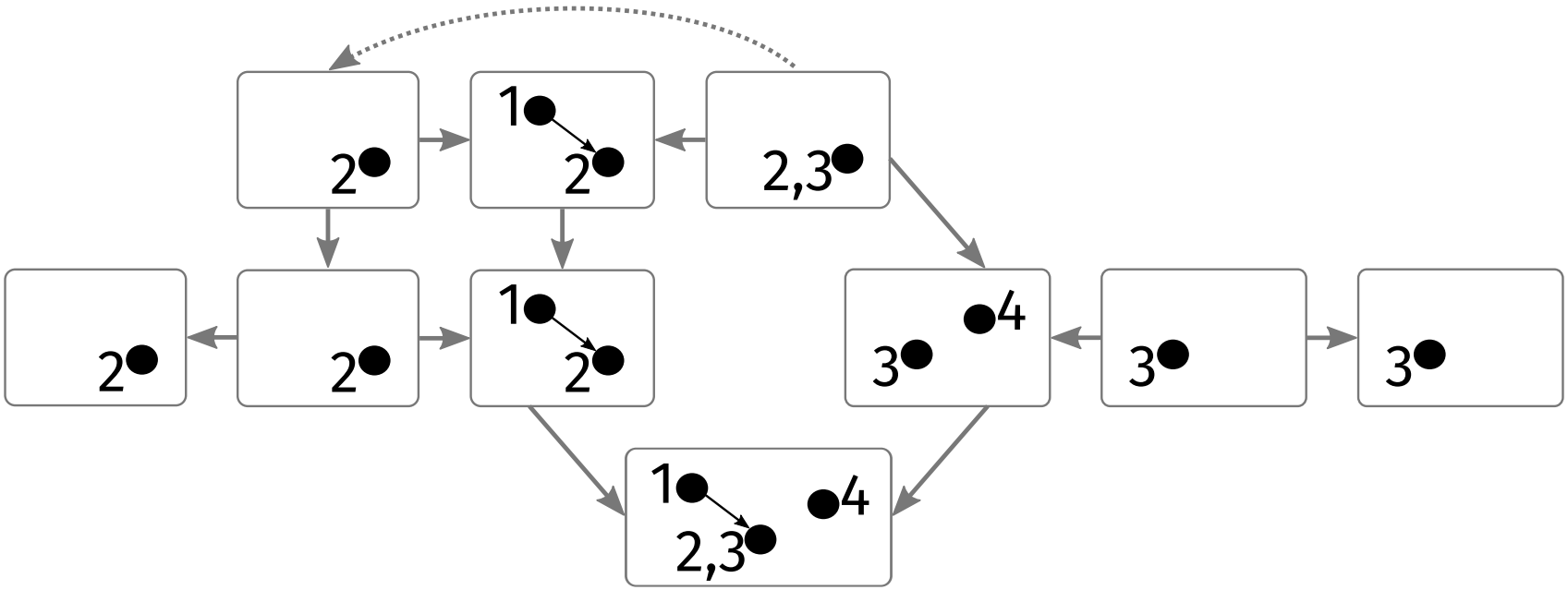
# Comparing Reasons and Essences

- Non-empty reasons exist even with parallel independence



# Comparing Reasons and Essences

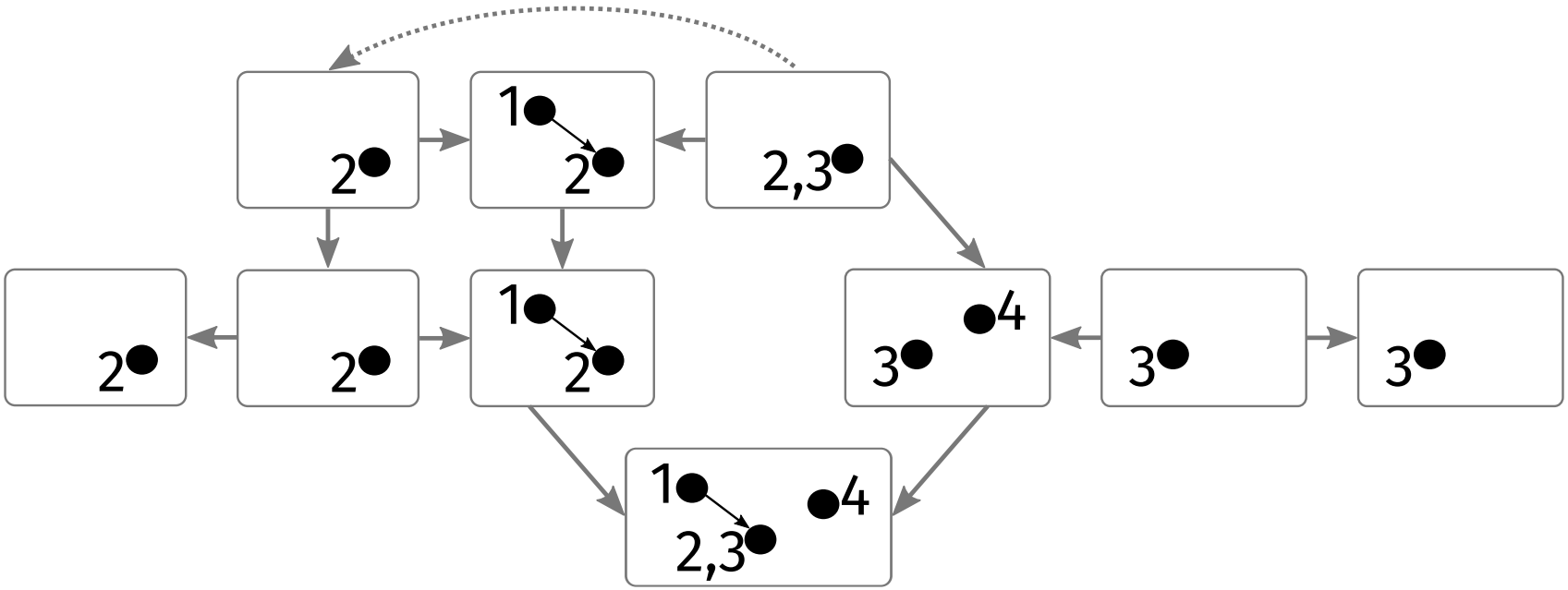
- Non-empty reasons exist even with parallel independence




- Isolated boundary nodes (Lambers, Born, et al. 2018)

# Comparing Reasons and Essences

- Non-empty reasons exist even with parallel independence



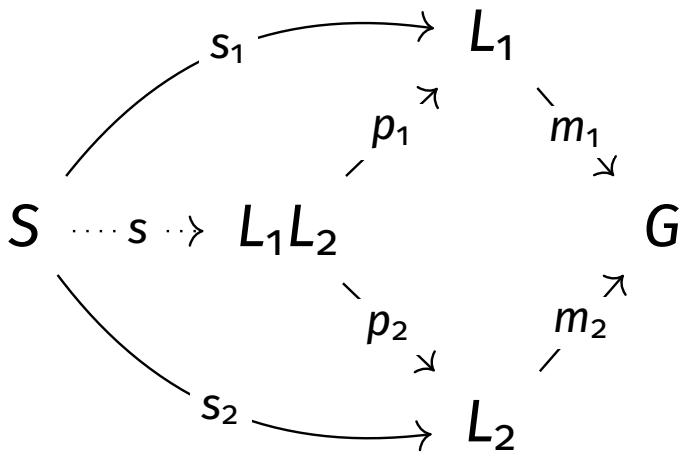

- Isolated boundary nodes (Lambers, Born, et al. 2018)
- Inheritance also doesn't hold

# Essence $\subseteq$ Reason

# Essence $\subseteq$ Reason

## Remark

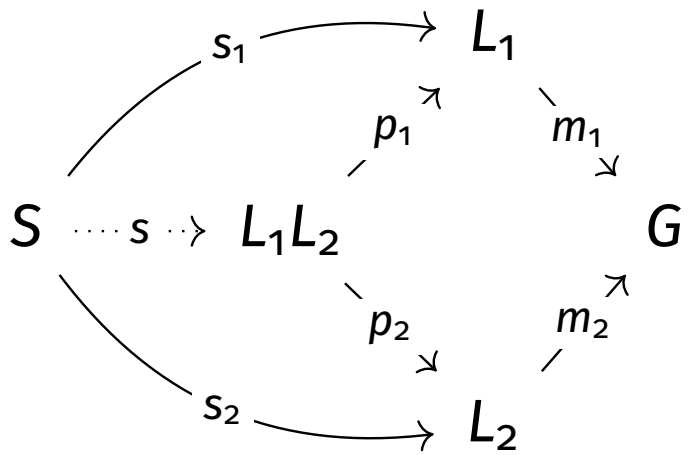
Conflict reason determines $s \in \mathbf{Sub}(L_1 L_2)$.



# Essence $\subseteq$ Reason

## Remark

Conflict reason determines $s \in \mathbf{Sub}(L_1 L_2)$.




## Theorem

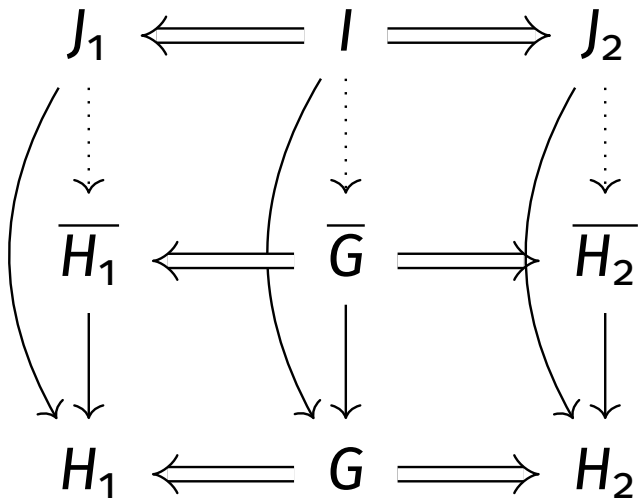
*If $c \in \mathbf{Sub}(L_1 L_2)$ is disabling essence and $s \in \mathbf{Sub}(L_1 L_2)$ disabling reason, then $c \subseteq s$.*
*The same holds if $c$ is conflict essence and $s$ conflict reason.*

# Initial Conflicts

- We now understand **individual** conflicting transformations
- We want overview of **potential** conflicts for rules

# Initial Conflicts

- We now understand **individual** conflicting transformations
- We want overview of **potential** conflicts for rules
- Lambers, Born, et al. (2018) proposed **initial conflicts** (w.r.t extension)



# Initial Conflicts

- Initial conflicts are subset of critical pairs, often much smaller!

# Initial Conflicts

- Initial conflicts are subset of critical pairs, often much smaller!
- Initial conflicts capture all conflicts $\Longleftrightarrow$ every transformation pair is extension of some initial transformation pair

# Initial Conflicts

- Initial conflicts are subset of critical pairs, often much smaller!
- Initial conflicts capture all conflicts $\Longleftrightarrow$ every transformation pair is extension of some initial transformation pair
- **But:** no categorical construction yet

# Constructing Initial Transformation Pairs

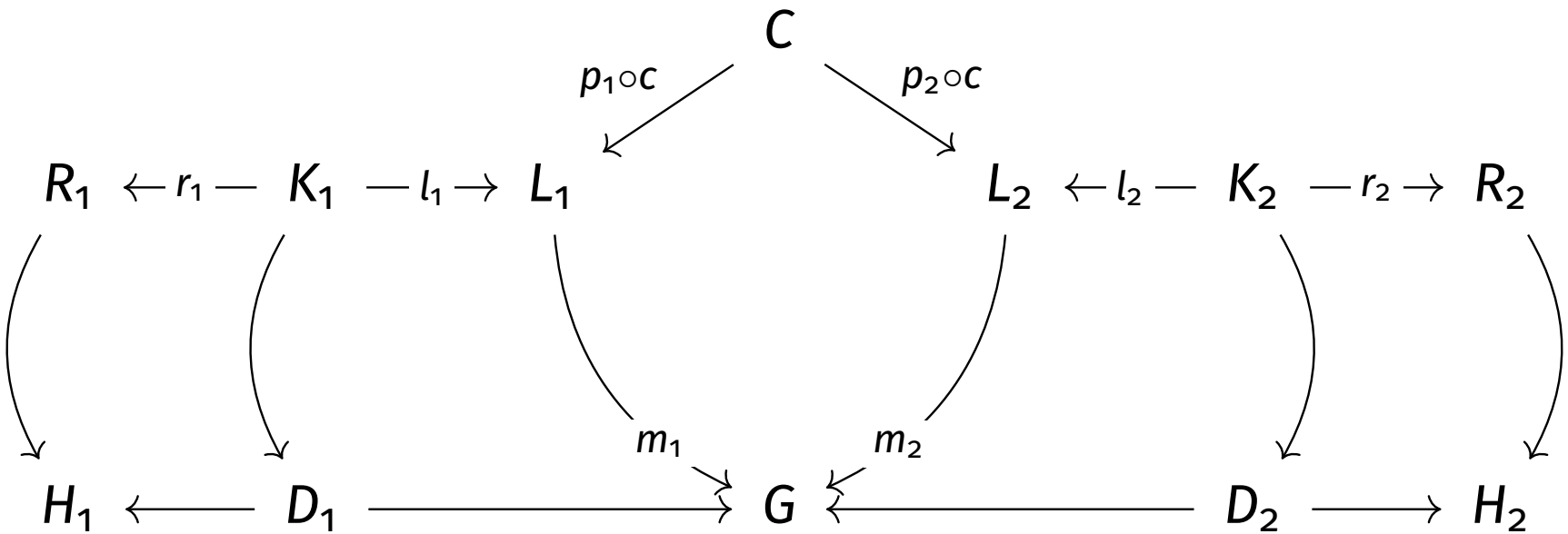
Conflict essences and initial transformation pairs
are closely related (in categories of set-valued functors)

# Constructing Initial Transformation Pairs

Conflict essences and initial transformation pairs
are closely related (in categories of set-valued functors)

**Theorem**

*Given $H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$, the pushout of its conflict essence determines its initial transformation pair.*

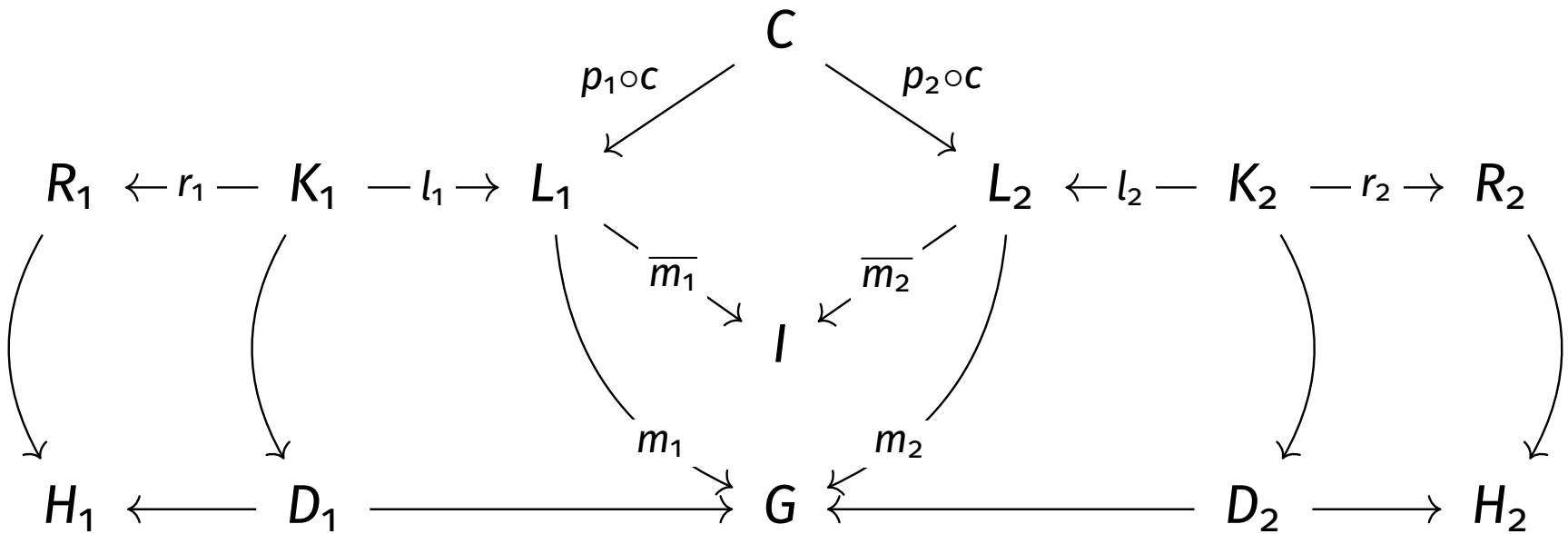


# Constructing Initial Transformation Pairs

Conflict essences and initial transformation pairs
are closely related (in categories of set-valued functors)

**Theorem**

*Given $H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$, the pushout of its conflict essence determines its initial transformation pair.*

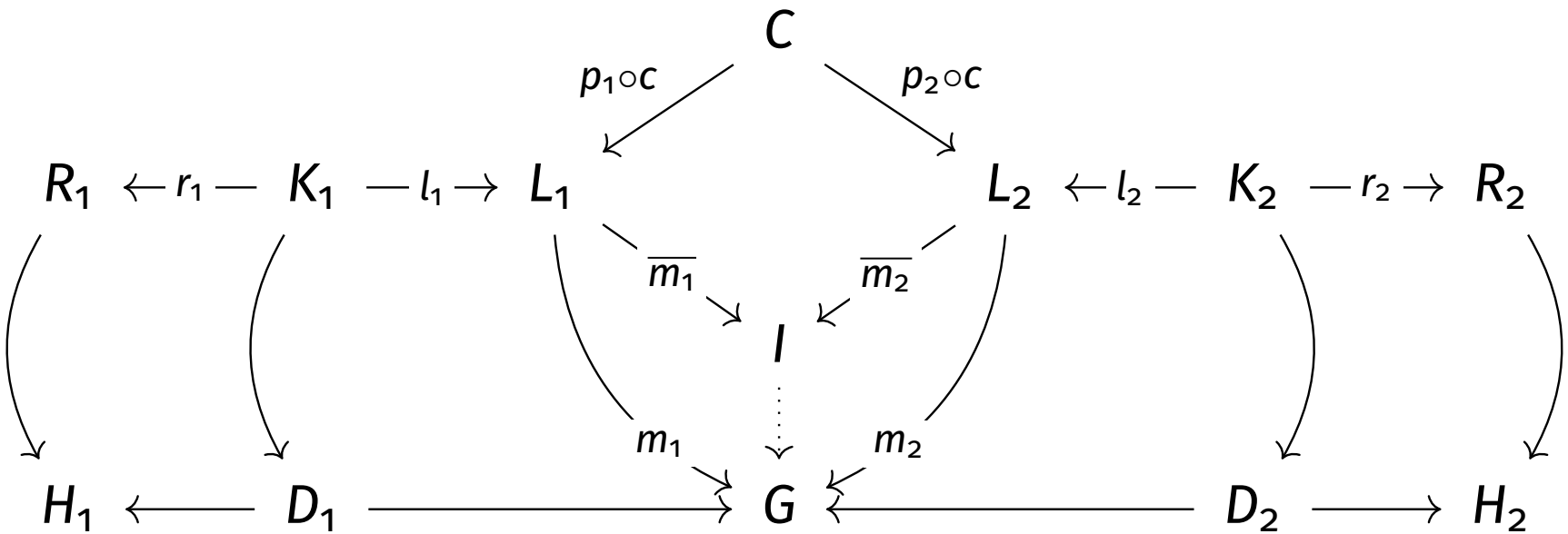


# Constructing Initial Transformation Pairs

Conflict essences and initial transformation pairs
are closely related (in categories of set-valued functors)

**Theorem**

*Given $H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$, the pushout of its conflict essence determines its initial transformation pair.*

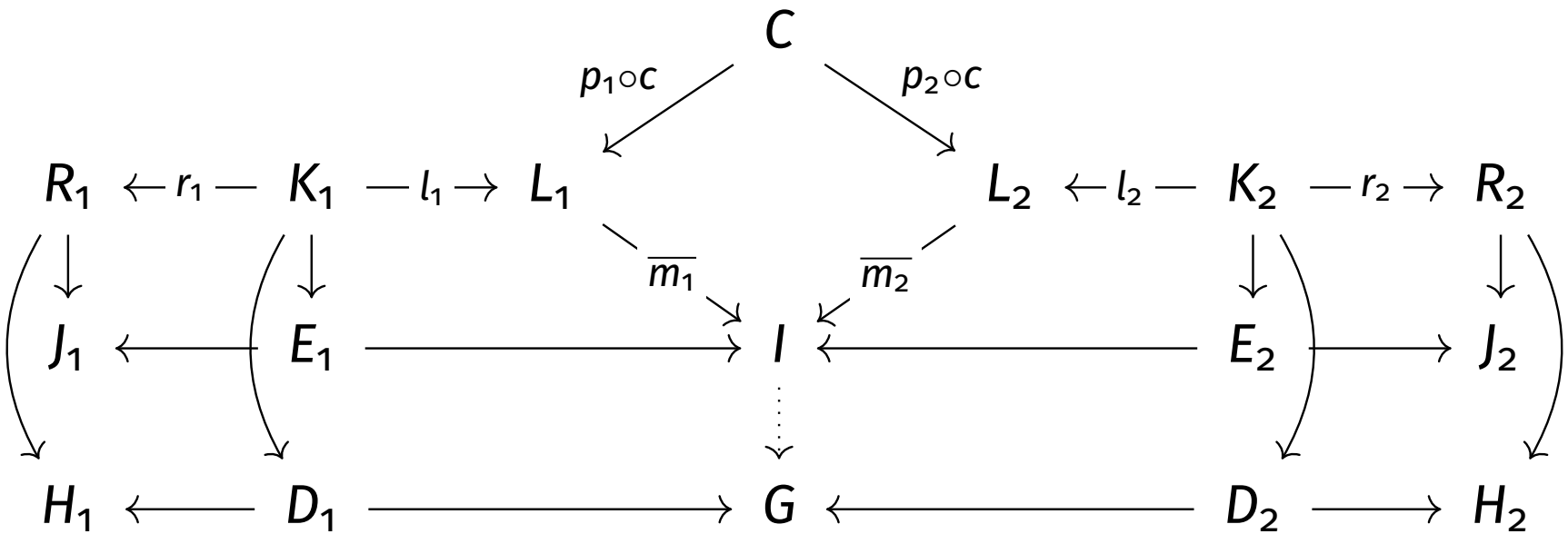


# Constructing Initial Transformation Pairs

Conflict essences and initial transformation pairs
are closely related (in categories of set-valued functors)

**Theorem**

*Given $H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$, the pushout of its conflict essence determines its initial transformation pair.*

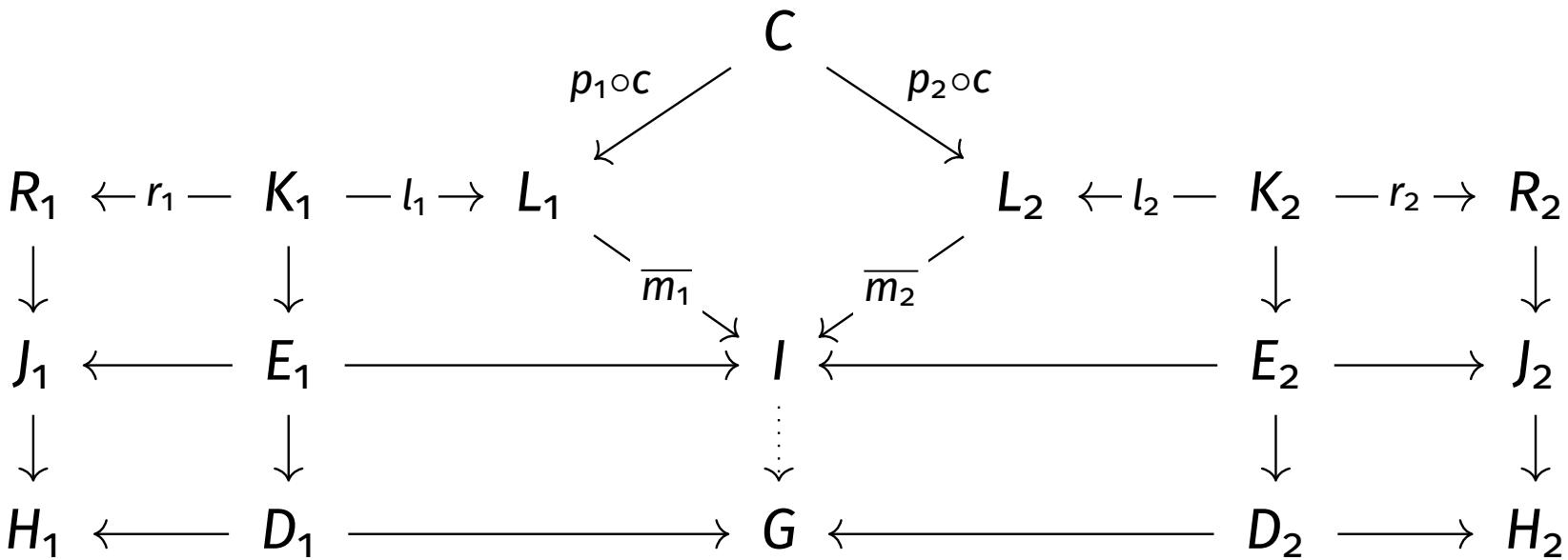


# Constructing Initial Transformation Pairs

Conflict essences and initial transformation pairs
are closely related (in categories of set-valued functors)

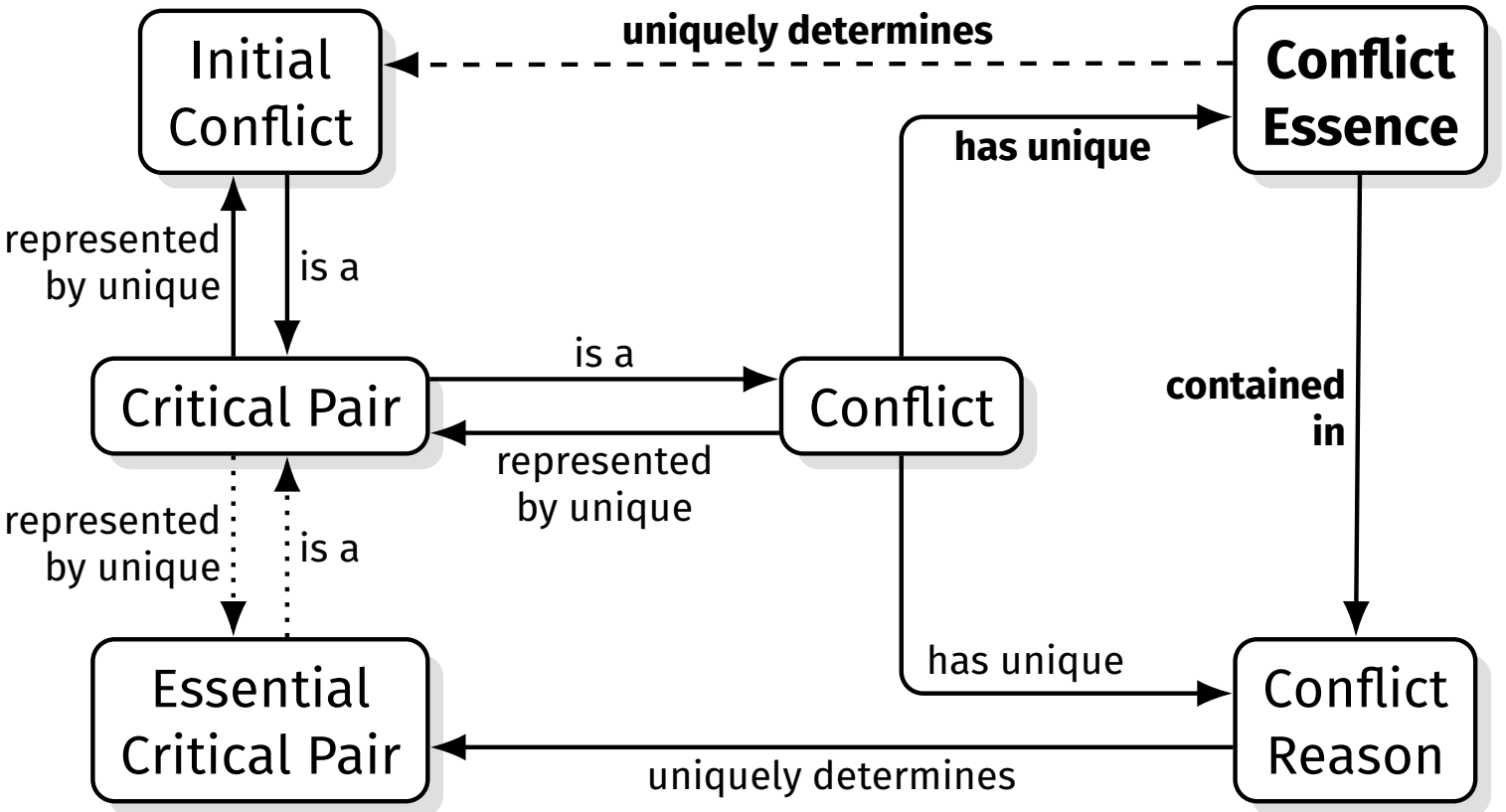
**Theorem**

*Given $H_1 \overset{\rho_1, m_1}{\Longleftarrow} G \overset{\rho_2, m_2}{\Longrightarrow} H_2$, the pushout of its conflict essence determines its initial transformation pair.*



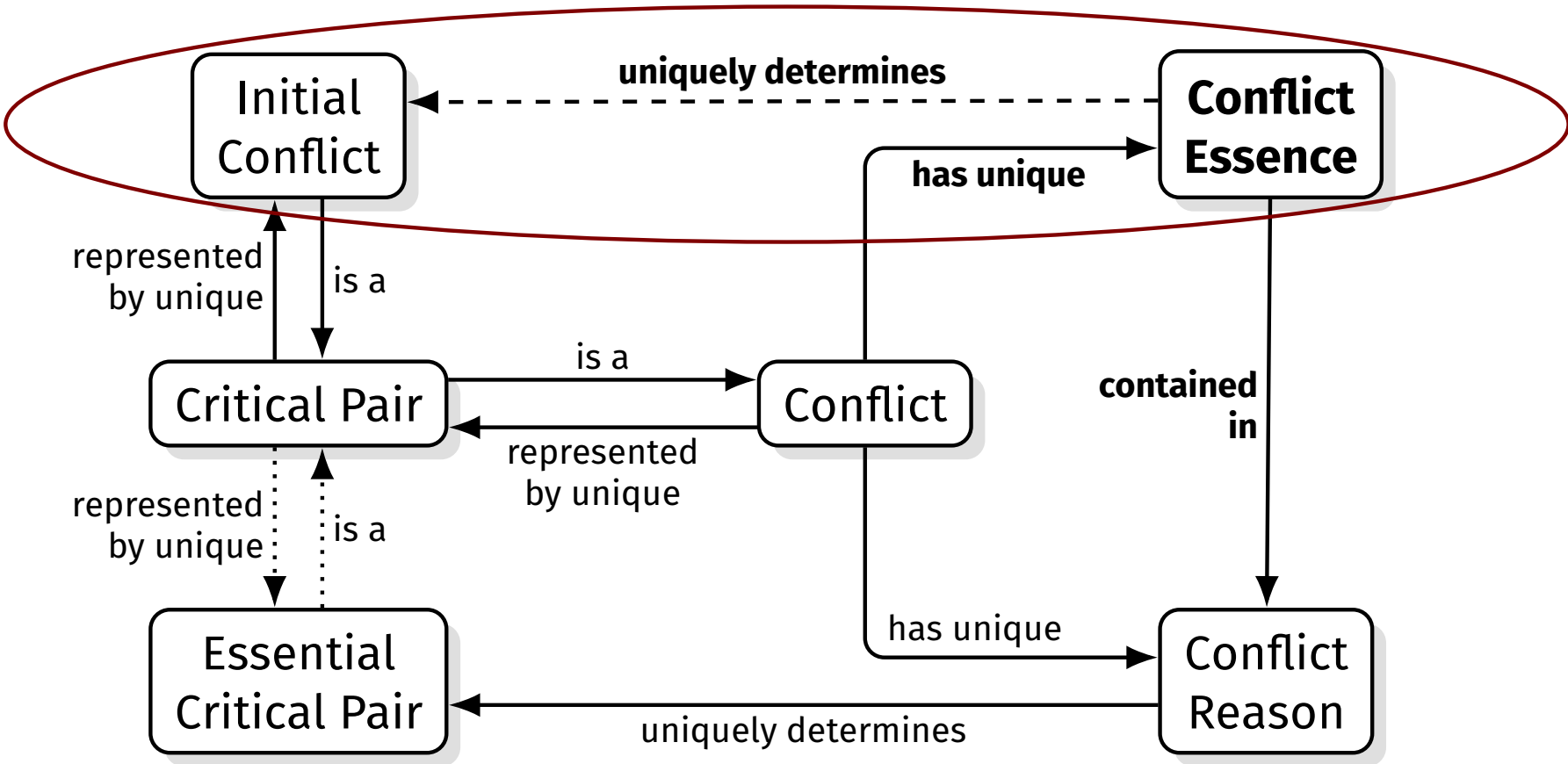
# Overview

Available for: —— Adhesive Categories - - - - $\mathbb{S}\mathrm{et}^{\mathbb{S}}$ ····· $\mathbb{G}\mathrm{raph}_T$



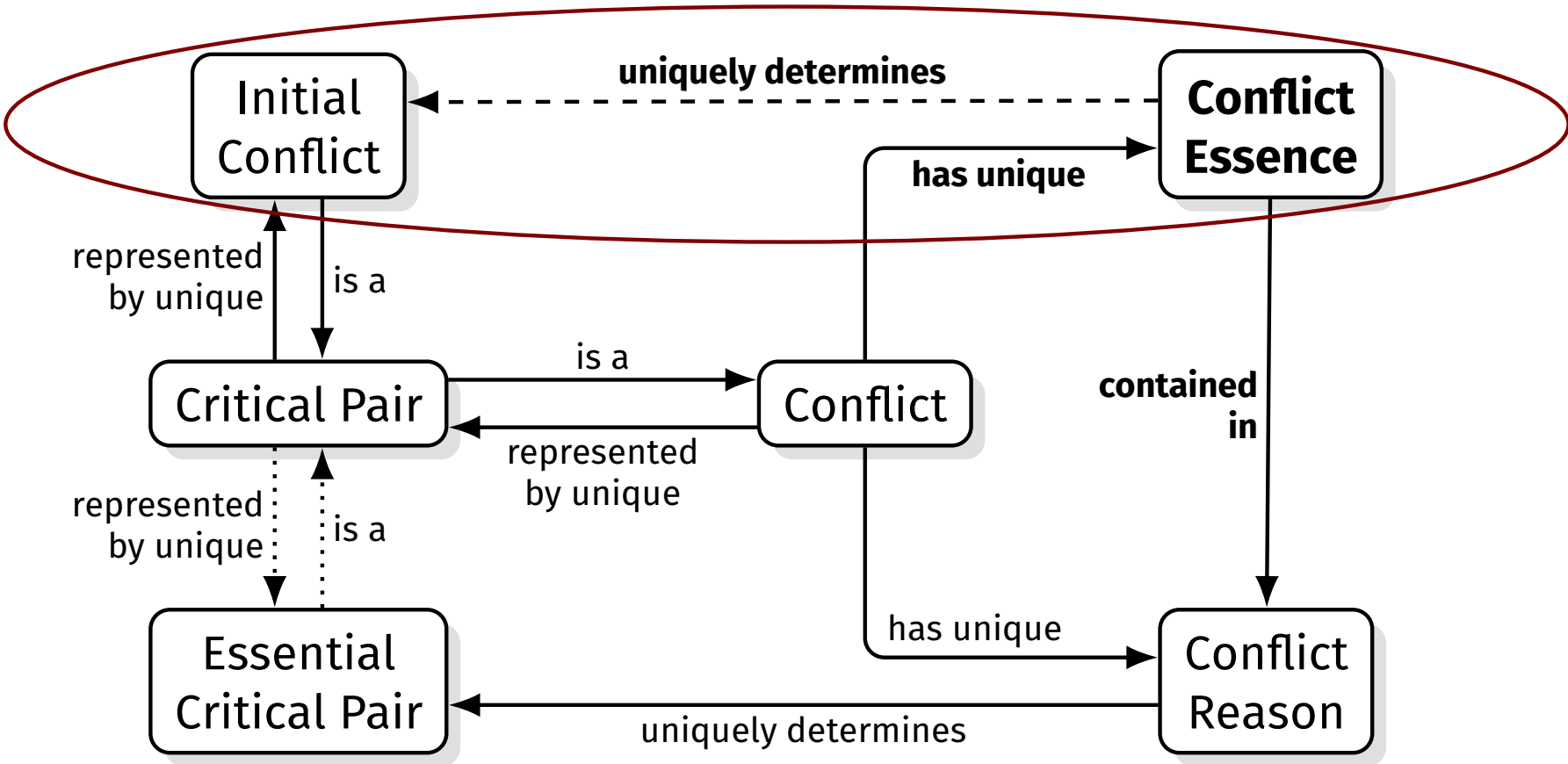
# Overview

Available for: —— Adhesive Categories - - - - $\mathbb{S}\mathrm{et}^{\mathbb{S}}$ ····· $\mathbb{G}\mathrm{raph}_T$



# Overview

Available for: —— Adhesive Categories - - - - $\mathbb{S}\mathrm{et}^{\mathbb{S}}$ ····· $\mathbb{G}\mathrm{raph}_T$




**Open Problem:** in all adhesive categories?

# Conclusions

- Essential condition allowed powerful characterization for root causes of conflicts
- Lots of future work!
  - Constraints and application conditions
  - Compare with notions of granularity (Born et al. 2017)
  - Attributed graphs and other adhesive categories
  - Sesqui-Pushout and AGREE

# Thank you!
## Questions?

# References I



Born, Kristopher et al. (2017). "Granularity of Conflicts and Dependencies in Graph Transformation Systems". In: *ICGT*. Vol. 10373. LNCS. Springer, pp. 125–141. DOI: 10.1007/978-3-319-61470-0_8. URL: https://doi.org/10.1007/978-3-319-61470-0_8.


Corradini, Andrea et al. (2018). "On the Essence of Parallel Independence for the Double-Pushout and Sesqui-Pushout Approaches". In: *Graph Transformation, Specifications, and Nets*. Vol. 10800. LNCS. Springer, pp. 1–18. DOI: 10.1007/978-3-319-75396-6_1. URL: https://doi.org/10.1007/978-3-319-75396-6_1.


Lack, Stephen and Pawel Sobocinski (2005). "Adhesive and quasiadhesive categories". In: *ITA* 39.3, pp. 511–545. DOI: 10.1051/ita:2005028. URL: https://doi.org/10.1051/ita:2005028.

# References II


Lambers, Leen, Kristopher Born, et al. (2018). "Initial Conflicts and Dependencies: Critical Pairs Revisited". In: *Graph Transformation, Specifications, and Nets*. Vol. 10800. LNCS. Springer, pp. 105–123. DOI: 10.1007/978-3-319-75396-6_6. URL: https://doi.org/10.1007/978-3-319-75396-6_6.

Lambers, Leen, Hartmut Ehrig, and Fernando Orejas (2008). "Efficient Conflict Detection in Graph Transformation Systems by Essential Critical Pairs". In: *ENTCS* 211, pp. 17–26. DOI: 10.1016/j.entcs.2008.04.026. URL: https://doi.org/10.1016/j.entcs.2008.04.026.