Constraint-Aware Model Merging

Alessandro Rossini[1], Adrian Rutle[2], Florian Mantz[2], Yngve Lamo[2], Uwe Wolter[1]

[1]Department of Informatics, University of Bergen, Norway

[2]Faculty of Engineering, Bergen University College, Norway

11 November 2010
Nordic Workshop on Programming Theory, Turku, Finland

Outline

Model-driven engineering (MDE)

- *Model* in software engineering
  - abstract representation of a software system
  - typically graph-based structure, e.g. UML

Model-driven engineering (MDE)

- *Model* in software engineering
  - abstract representation of a software system
  - typically graph-based structure, e.g. UML
- Model-driven engineering (MDE)
  - models as primary artefacts of software development
  - generation of systems by model transformations

Model-driven engineering (MDE)

- *Model* in software engineering
  - abstract representation of a software system
  - typically graph-based structure, e.g. UML
- Model-driven engineering (MDE)
  - models as primary artefacts of software development
  - generation of systems by model transformations
- Complex evolution of models
  - need for techniques and tools to support version control

Optimistic version control

- A local copy of artefacts for each developer
  - independent and parallel modifications
- Merging of modifications
  - centralised (*copy-modify-merge*)
  - distributed

Three-way model merging

- Merging two versions of a model relying on their common ancestor

$$\begin{tikzcd}
 & V_1 \arrow[dl] \arrow[dr] & \\
V_1' \arrow[dr] & & V_1'' \arrow[dl] \\
 & V_2 &
\end{tikzcd}$$

Three-way model merging

- Merging two versions of a model relying on their common ancestor

$$\begin{tikzcd}
 & V_1 \arrow[dl] \arrow[dr] & \\
V_1' \arrow[dr] & & V_1'' \arrow[dl] \\
 & V_2 &
\end{tikzcd}$$

- Detection of conflicts
  - possibly resolution

Challenges

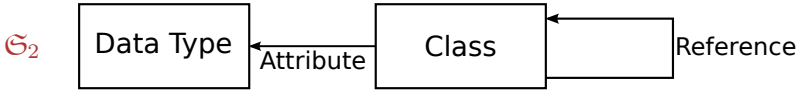
- Mainstream version control systems, e.g. Subversion
  - target text-based artefacts
  - not suitable for graph-based structures
- Prototype model version control systems, e.g. AMOR
  - lack of formal underpinning
- State-of-the-art of research
  - constraint on model elements not considered

Proposed solution

- Diagram Predicate Framework (DPF)
- Formal approach to MDE based on category theory
  - (meta)modelling
  - model transformation
  - model management

Outline

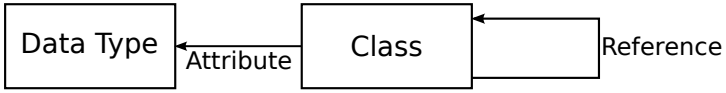
Sample Object-Oriented Modelling Hierarchy



- Modelling language: graph + ...

Sample Object-Oriented Modelling Hierarchy

$\mathfrak{S}_2$

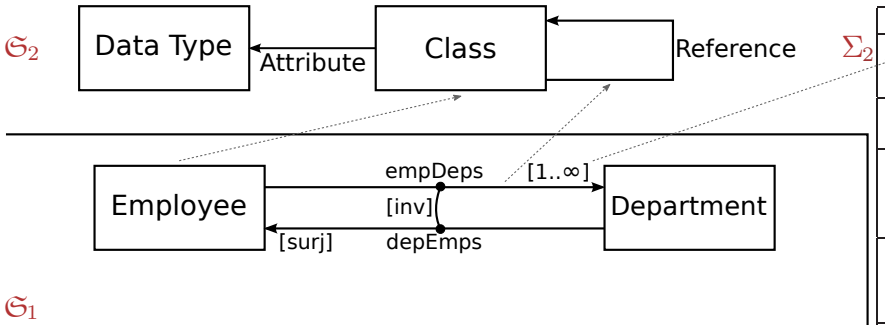


Σ_2

p	$\alpha^{\Sigma_2}(p)$	Proposed vis.	Intended semantics
`[mult(m,n)]`	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[m..n]}]{\text{f}}$ Y	$\forall x \in X : m \leq \lvert f(x) \rvert \leq n$, with $0 \leq m \leq n$ and $n \geq 1$
`[surjective]`	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[surj]}]{\text{f}}$ Y	$f(X) = Y$
`[inverse]`	$1 \underset{b}{\overset{a}{\rightleftarrows}} 2$	X $\overset{\text{f}}{\underset{\text{g}}{\rightleftarrows}}$ [inv] Y	$\forall x \in X, \forall y \in Y : y \in f(x)$ iff $x \in g(y)$
`[image-inclusion]`	$1 \underset{b}{\overset{a}{\rightrightarrows}} 2$	X $\overset{\text{f}}{\underset{\text{g}}{\rightrightarrows}}$ [⊑] Y	$\forall x \in X : f(x) \subseteq g(x)$
`[composition]`	$1 \xrightarrow{a} 2$, $2 \xrightarrow{b} 3$, $1 \xrightarrow{c} 3$	X $\xrightarrow{\text{f}}$ Y, Y $\xrightarrow{\text{g}}$ Z, X $\xrightarrow[\text{[comp]}]{\text{h}}$ Z	$\forall x \in X : h(x) = \{g(y) \mid y \in f(x)\}$

- Modelling language: graph + signature

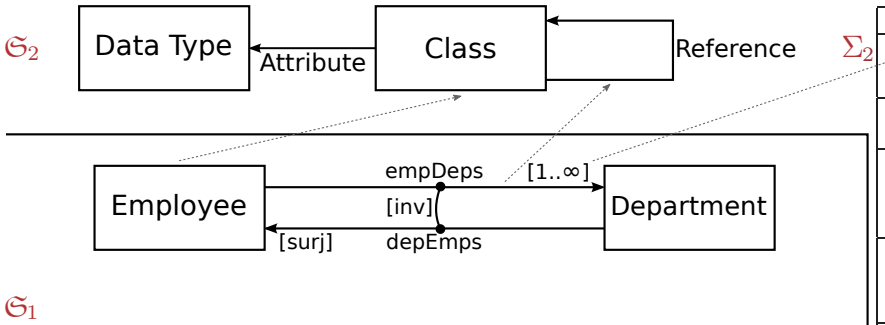
Sample Object-Oriented Modelling Hierarchy



p	$\alpha^{\Sigma_2}(p)$	Proposed vis.	Intended semantics
[mult (m,n)]	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[m..n]}]{f}$ Y	$\forall x \in X : m \leq \lvert f(x)\rvert \leq n$, with $0 \leq m \leq n$ and $n \geq 1$
[surjective]	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[surj]}]{f}$ Y	$f(X) = Y$
[inverse]	$1 \underset{b}{\overset{a}{\rightleftarrows}} 2$	X $\underset{g}{\overset{f}{\rightleftarrows}}$ Y [inv]	$\forall x \in X, \forall y \in Y : y \in f(x)$ iff $x \in g(y)$
[image-inclusion]	$1 \underset{b}{\overset{a}{\rightrightarrows}} 2$	X $\underset{g}{\overset{f}{\rightrightarrows}}$ Y [⊑]	$\forall x \in X : f(x) \subseteq g(x)$
[composition]	$1 \xrightarrow{a} 2 \xrightarrow{b} 3$, $1 \xrightarrow{c} 3$	X $\xrightarrow{f}$ Y $\xrightarrow{g}$ Z, X $\xrightarrow[\text{[comp]}]{h}$ Z	$\forall x \in X : h(x) = \{g(y) \mid y \in f(x)\}$

- Model (specification): graph + constraints
- Nodes interpreted as sets
- Arrows interpreted as multi-valued functions

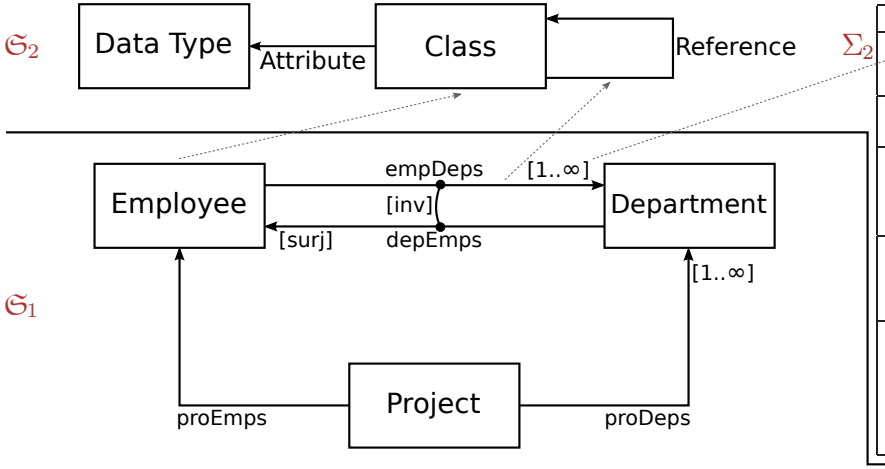
Sample Object-Oriented Modelling Hierarchy



p	$\alpha^{\Sigma_2}(p)$	Proposed vis.	Intended semantics
[mult (m,n)]	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[m..n]}]{f}$ Y	$\forall x \in X : m \leq \lvert f(x) \rvert \leq n$, with $0 \leq m \leq n$ and $n \geq 1$
[surjective]	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[surj]}]{f}$ Y	$f(X) = Y$
[inverse]	$1 \underset{b}{\overset{a}{\rightleftarrows}} 2$	X $\underset{g}{\overset{f}{\rightleftarrows}}$ Y [inv]	$\forall x \in X, \forall y \in Y : y \in f(x)$ iff $x \in g(y)$
[image-inclusion]	$1 \underset{b}{\overset{a}{\rightleftarrows}} 2$	X $\underset{g}{\overset{f}{\rightleftarrows}}$ Y [⊑]	$\forall x \in X : f(x) \subseteq g(x)$
[composition]	$1 \xrightarrow{a} 2$, $2 \xrightarrow{b} 3$, $1 \xrightarrow{c} 3$	X $\xrightarrow{f}$ Y, Y $\xrightarrow{g}$ Z, X $\xrightarrow[\text{[comp]}]{h}$ Z	$\forall x \in X : h(x) = \{g(y) \mid y \in f(x)\}$

- Req 1: "an employee must work for at least one department"
- Req 2: "a department may have none or many employees"

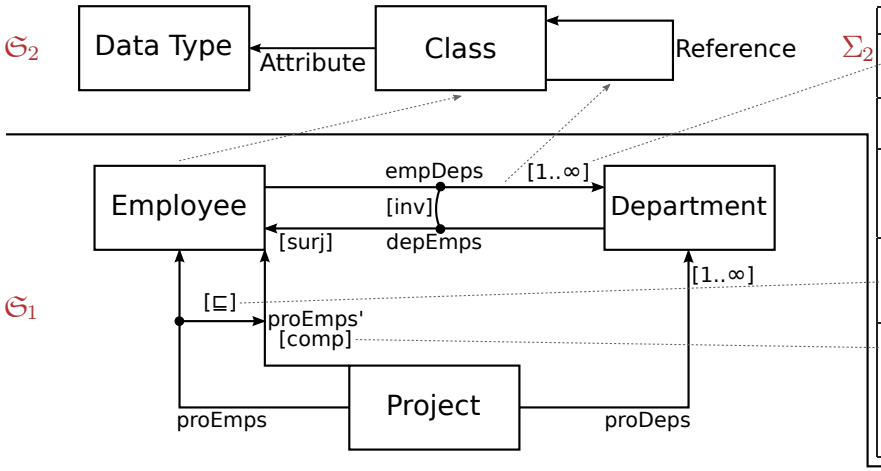
Sample Object-Oriented Modelling Hierarchy



p	$\alpha^{\Sigma_2}(p)$	Proposed vis.	Intended semantics
[mult (m,n)]	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[m..n]}]{f}$ Y	$\forall x \in X : m \leq \lvert f(x) \rvert \leq n$, with $0 \leq m \leq n$ and $n \geq 1$
[surjective]	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[surj]}]{f}$ Y	$f(X) = Y$
[inverse]	$1 \underset{b}{\overset{a}{\rightleftarrows}} 2$	X $\underset{g}{\overset{f}{\rightleftarrows}}$ Y [inv]	$\forall x \in X, \forall y \in Y : y \in f(x)$ iff $x \in g(y)$
[image-inclusion]	$1 \underset{b}{\overset{a}{\rightleftarrows}} 2$	X $\underset{g}{\overset{f}{\rightleftarrows}}$ Y [⊑]	$\forall x \in X : f(x) \subseteq g(x)$
[composition]	$1 \xrightarrow{a} 2$, $2 \xrightarrow{b} 3$, $1 \xrightarrow{c} 3$	X $\xrightarrow{f}$ Y, Y $\xrightarrow{g}$ Z, X $\xrightarrow[\text{[comp]}]{h}$ Z	$\forall x \in X : h(x) = \{g(y) \mid y \in f(x)\}$

- Req 3: "a project may involve none or many employees"
- Req 4: "a project must be controlled by at least one department"

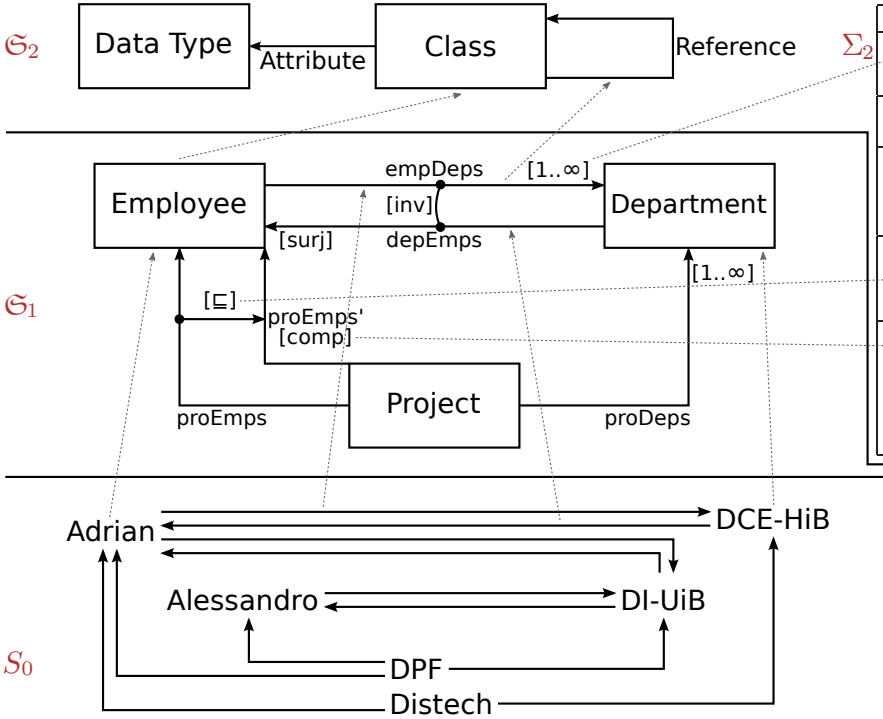
Sample Object-Oriented Modelling Hierarchy



p	$\alpha^{\Sigma_2}(p)$	Proposed vis.	Intended semantics
[mult (m,n)]	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[m..n]}]{f}$ Y	$\forall x \in X : m \leq \lvert f(x) \rvert \leq n$, with $0 \leq m \leq n$ and $n \geq 1$
[surjective]	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[surj]}]{f}$ Y	$f(X) = Y$
[inverse]	$1 \underset{b}{\overset{a}{\rightleftarrows}} 2$	X $\underset{g}{\overset{f}{\rightleftarrows}}$ Y [inv]	$\forall x \in X, \forall y \in Y : y \in f(x)$ iff $x \in g(y)$
[image-inclusion]	$1 \underset{b}{\overset{a}{\rightleftarrows}} 2$	X $\underset{g}{\overset{f}{\rightleftarrows}}$ Y [⊑]	$\forall x \in X : f(x) \subseteq g(x)$
[composition]	$1 \xrightarrow{a} 2$, $2 \xrightarrow{b} 3$, $1 \xrightarrow{c} 3$	X $\xrightarrow{f}$ Y, Y $\xrightarrow{g}$ Z, X $\xrightarrow[\text{[comp]}]{h}$ Z	$\forall x \in X : h(x) = \{g(y) \mid y \in f(x)\}$

- Req 5: "an employee involved in a project must work in the controlling department"

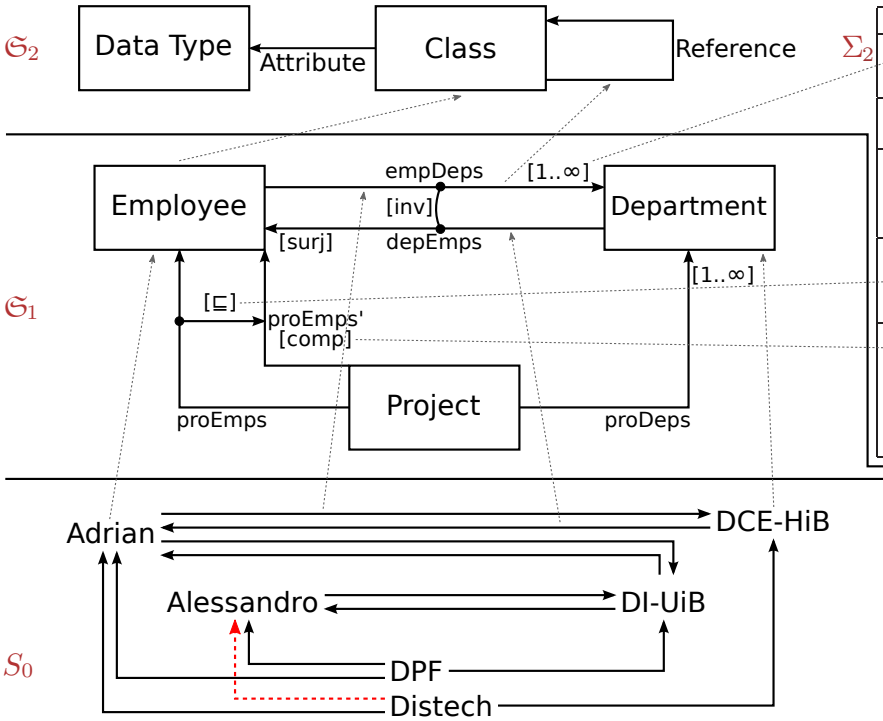
Sample Object-Oriented Modelling Hierarchy



p	$\alpha^{\Sigma_2}(p)$	Proposed vis.	Intended semantics
[mult (m,n)]	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[m..n]}]{f}$ Y	$\forall x \in X : m \leq \lvert f(x) \rvert \leq n$, with $0 \leq m \leq n$ and $n \geq 1$
[surjective]	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[surj]}]{f}$ Y	$f(X) = Y$
[inverse]	$1 \underset{b}{\overset{a}{\rightleftarrows}} 2$	X $\underset{g}{\overset{f}{\underset{\text{[inv]}}{\rightleftarrows}}}$ Y	$\forall x \in X, \forall y \in Y : y \in f(x)$ iff $x \in g(y)$
[image-inclusion]	$1 \underset{b}{\overset{a}{\rightleftarrows}} 2$	X $\underset{g}{\overset{f}{\underset{\text{[⊑]}}{\rightleftarrows}}}$ Y	$\forall x \in X : f(x) \subseteq g(x)$
[composition]	$1 \xrightarrow{a} 2$, $2 \xrightarrow{b} 3$, $1 \xrightarrow{c} 3$	X $\xrightarrow{f}$ Y, Y $\xrightarrow{g}$ Z, X $\xrightarrow[\text{[comp]}]{h}$ Z	$\forall x \in X : h(x) = \{g(y) \mid y \in f(x)\}$

- Instance

Sample Object-Oriented Modelling Hierarchy



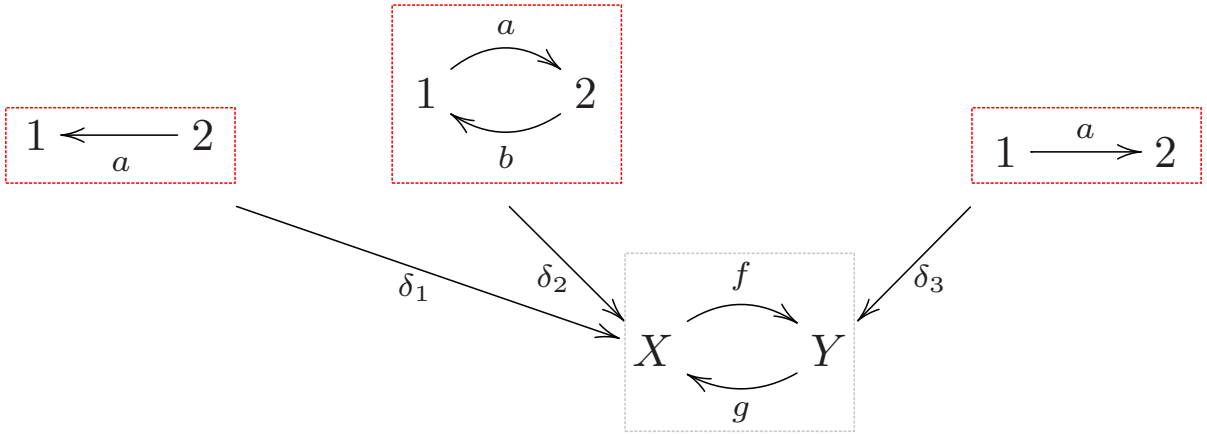
p	$\alpha^{\Sigma_2}(p)$	Proposed vis.	Intended semantics
[mult (m,n)]	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[m..n]}]{f}$ Y	$\forall x \in X : m \leq \lvert f(x) \rvert \leq n$, with $0 \leq m \leq n$ and $n \geq 1$
[surjective]	$1 \xrightarrow{a} 2$	X $\xrightarrow[\text{[surj]}]{f}$ Y	$f(X) = Y$
[inverse]	$1 \underset{b}{\overset{a}{\rightleftarrows}} 2$	X $\underset{g}{\overset{f}{\underset{\text{[inv]}}{\rightleftarrows}}}$ Y	$\forall x \in X, \forall y \in Y : y \in f(x)$ iff $x \in g(y)$
[image-inclusion]	$1 \underset{b}{\overset{a}{\rightleftarrows}} 2$	X $\underset{g}{\overset{f}{\underset{\text{[⊑]}}{\rightleftarrows}}}$ Y	$\forall x \in X : f(x) \subseteq g(x)$
[composition]	$\begin{tikzcd} 1 \arrow[r, "a"] \arrow[dr, "c"'] & 2 \arrow[d, "b"] \\ & 3 \end{tikzcd}$	$\begin{tikzcd} X \arrow[r, "f"] \arrow[dr, "h"', "\text{[comp]}"] & Y \arrow[d, "g"] \\ & Z \end{tikzcd}$	$\forall x \in X : h(x) = \{g(y) \mid y \in f(x)\}$

- Invalid instance

Specification entailment

$$\mathfrak{L} \quad \vdash \quad \mathfrak{R}$$

$$\alpha(\texttt{[surjective]})\ \alpha(\texttt{[inverse]}) \quad \vdash \quad \alpha(\texttt{[mult(1,}\infty\texttt{)]})$$



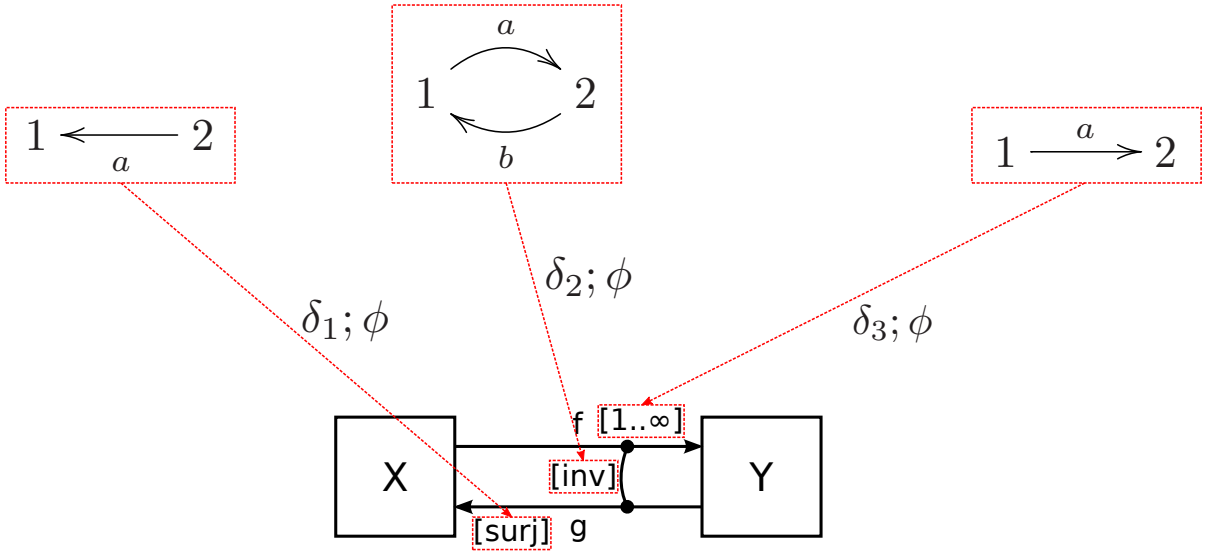
Context $L{=}R$

$$\begin{array}{lll}
 & \forall x \in X \text{ and } \forall y \in Y : x \in g(y) \text{ iff } y \in f(x) & f, g \text{ inverse} \\
 & \forall x \in X : \bigcup\{g(y)\} = X & g \text{ surjective} \\
\Rightarrow & \forall x \in X\ \exists y \in Y : x \in g(y) & \\
\Rightarrow & \forall x \in X\ \exists y \in Y : y \in f(x) & \\
\Rightarrow & \forall x \in X : |f(x)| \geq 1 & f \text{ total}
\end{array}$$

Embedding of specification entailment

$$\mathfrak{S} \quad \vdash \quad \mathfrak{S} \cup \phi(\mathfrak{R})$$

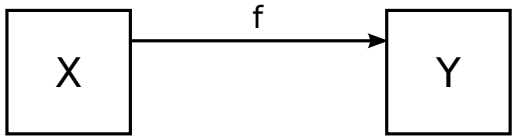
$$\alpha(\texttt{[surjective]})\ \alpha(\texttt{[inverse]}) \quad \vdash \quad \alpha(\texttt{[mult(1,}\infty\texttt{)]})$$



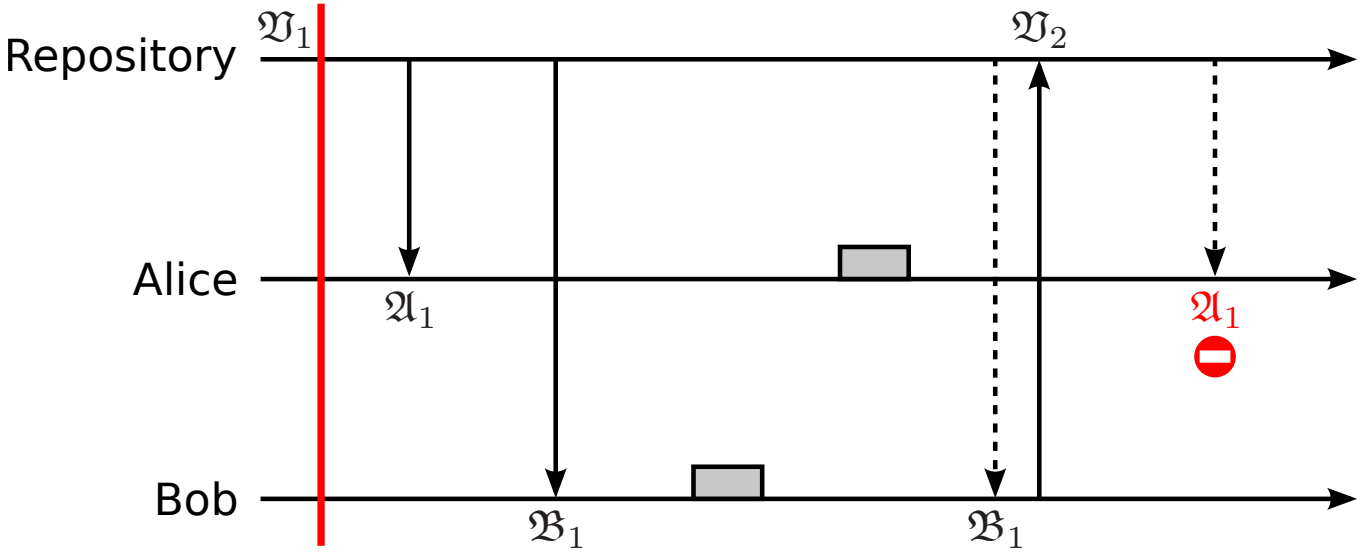
Outline

Example 1: multiplicity

Repository Alice's copy Bob's copy

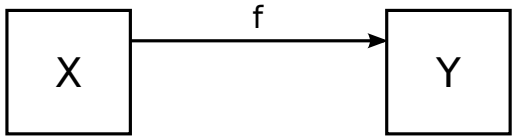


Timeline

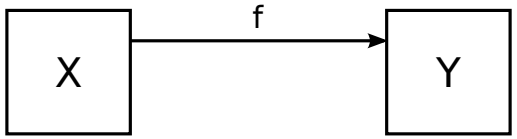


Example 1: multiplicity

Repository

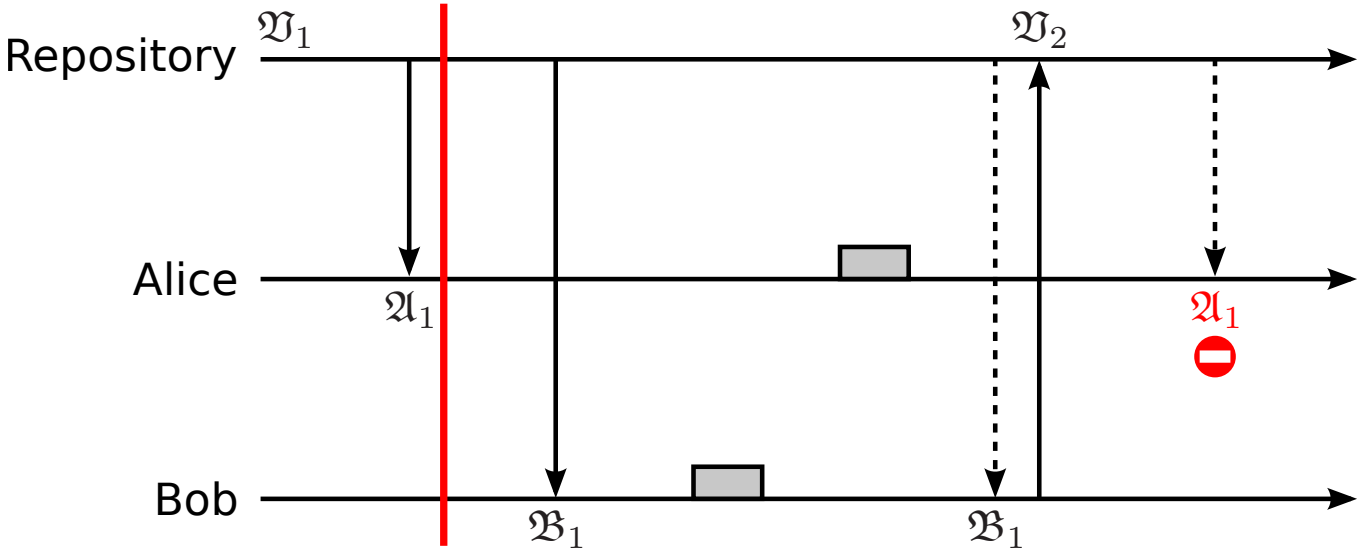


Alice's copy



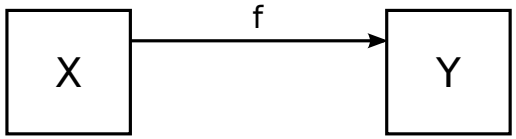
Bob's copy

Timeline

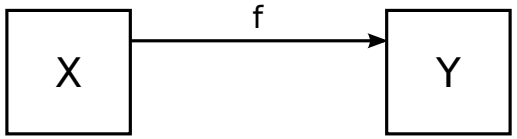


Example 1: multiplicity

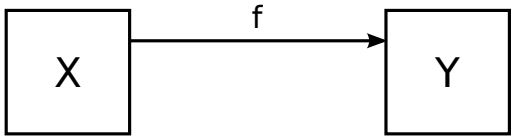
Repository



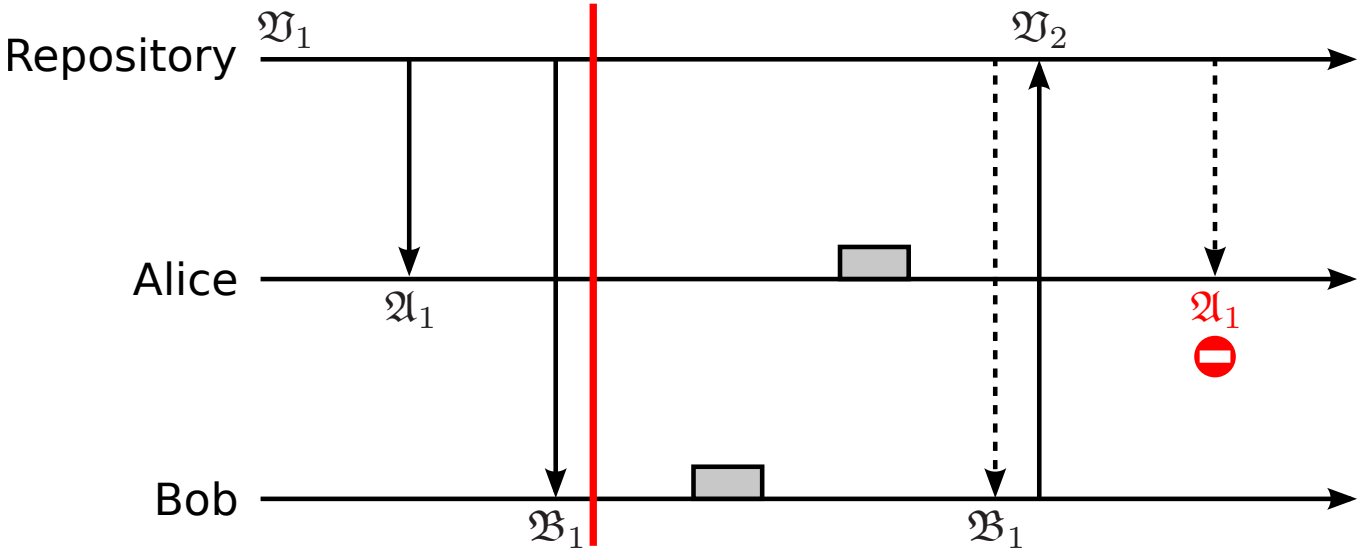
Alice's copy



Bob's copy

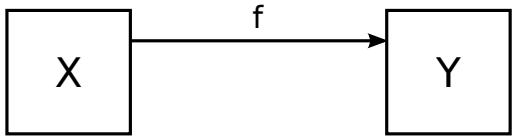


Timeline

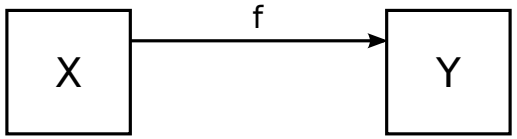


Example 1: multiplicity

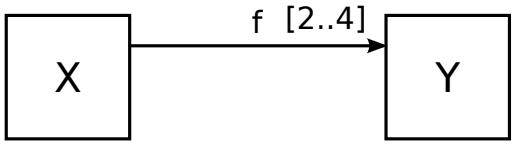
Repository



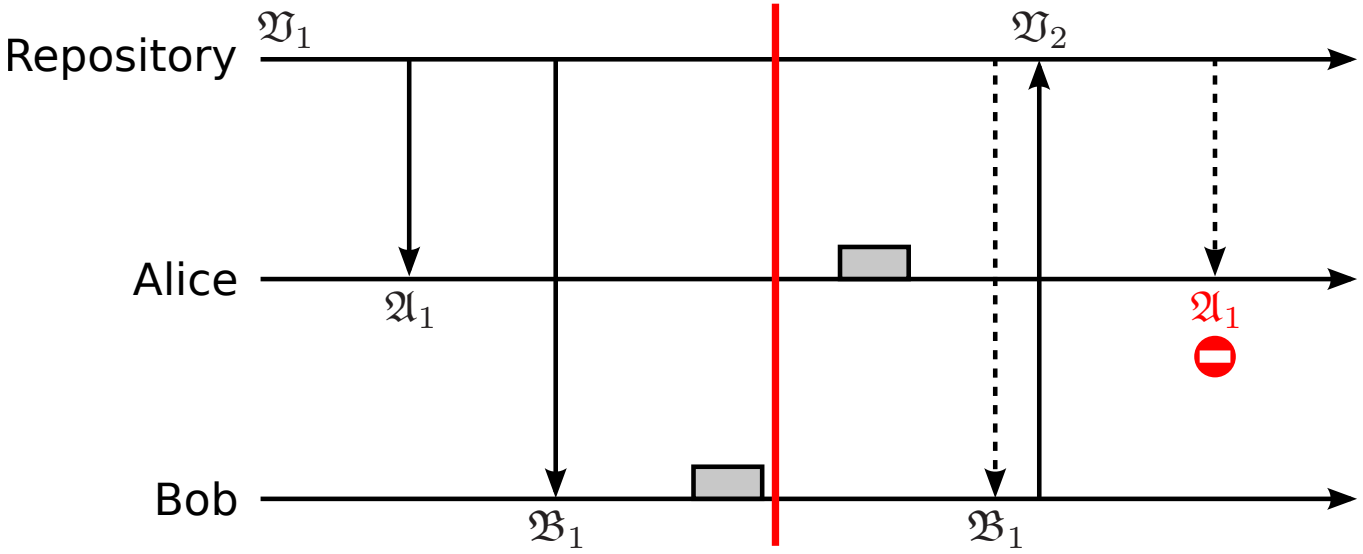
Alice's copy



Bob's copy

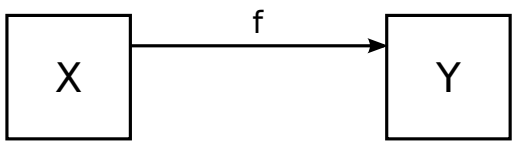


Timeline

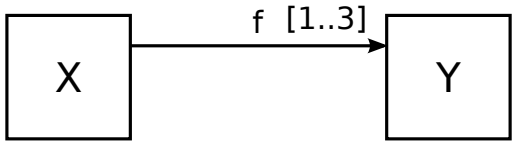


Example 1: multiplicity

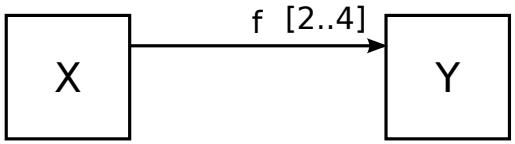
Repository



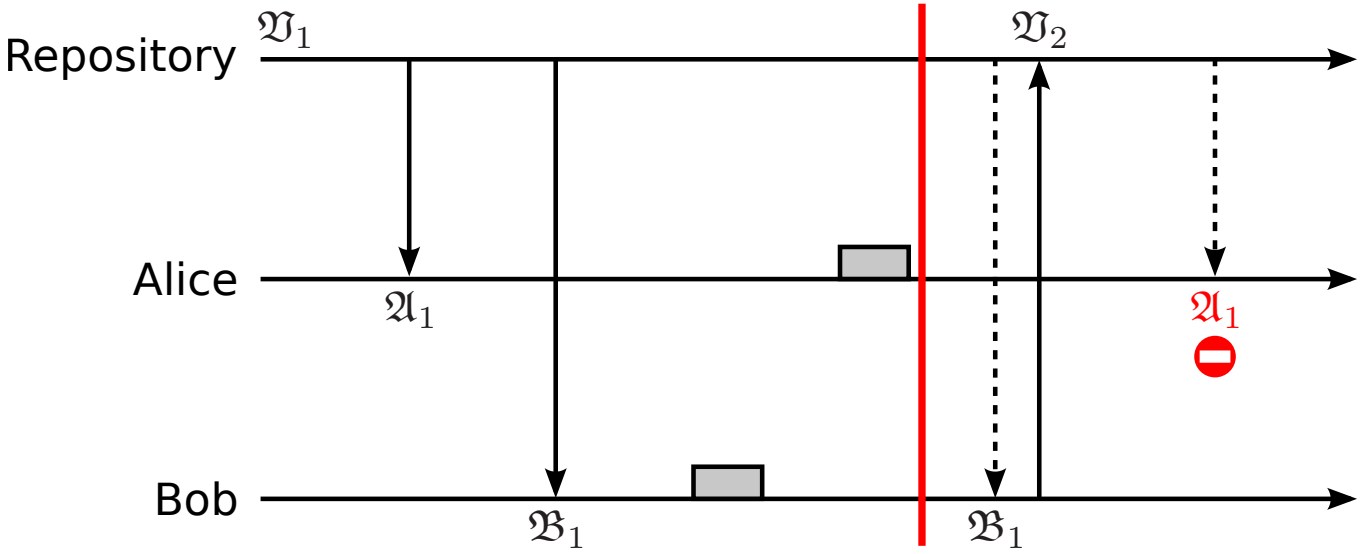
Alice's copy



Bob's copy

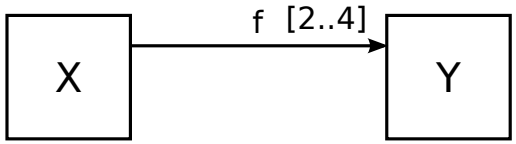


Timeline

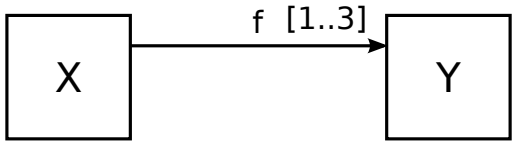


Example 1: multiplicity

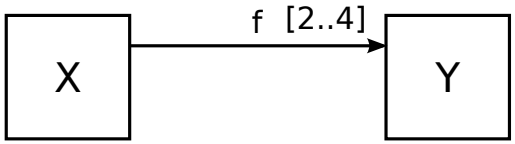
Repository



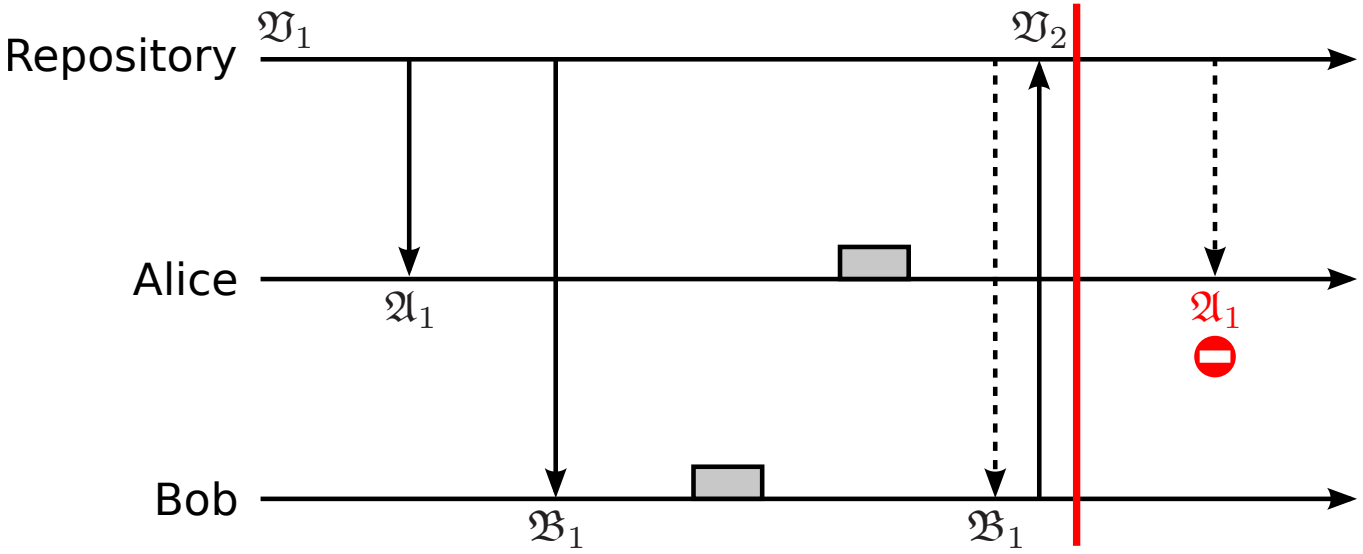
Alice's copy



Bob's copy

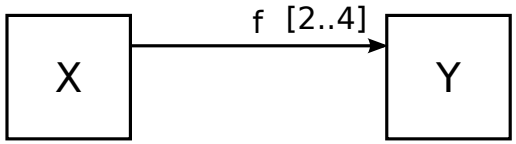


Timeline

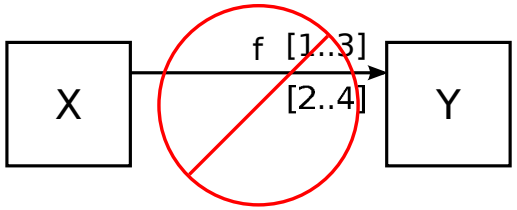


Example 1: multiplicity

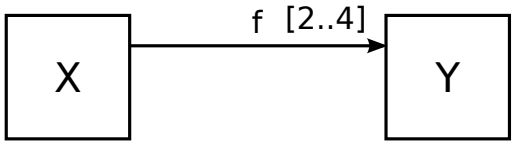
Repository



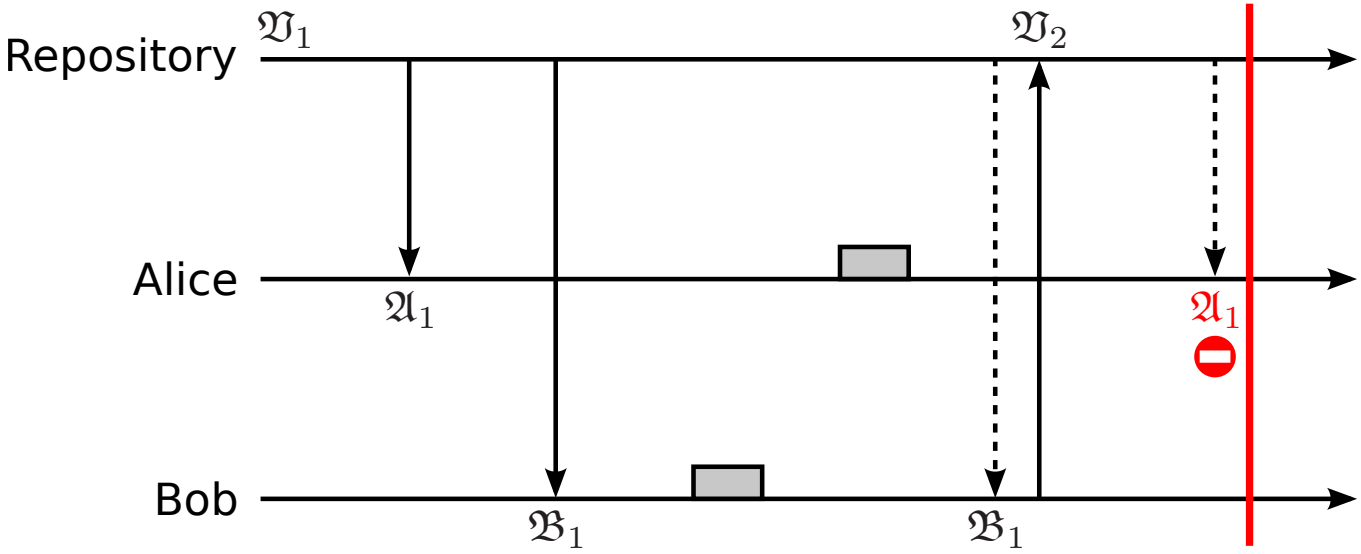
Alice's copy



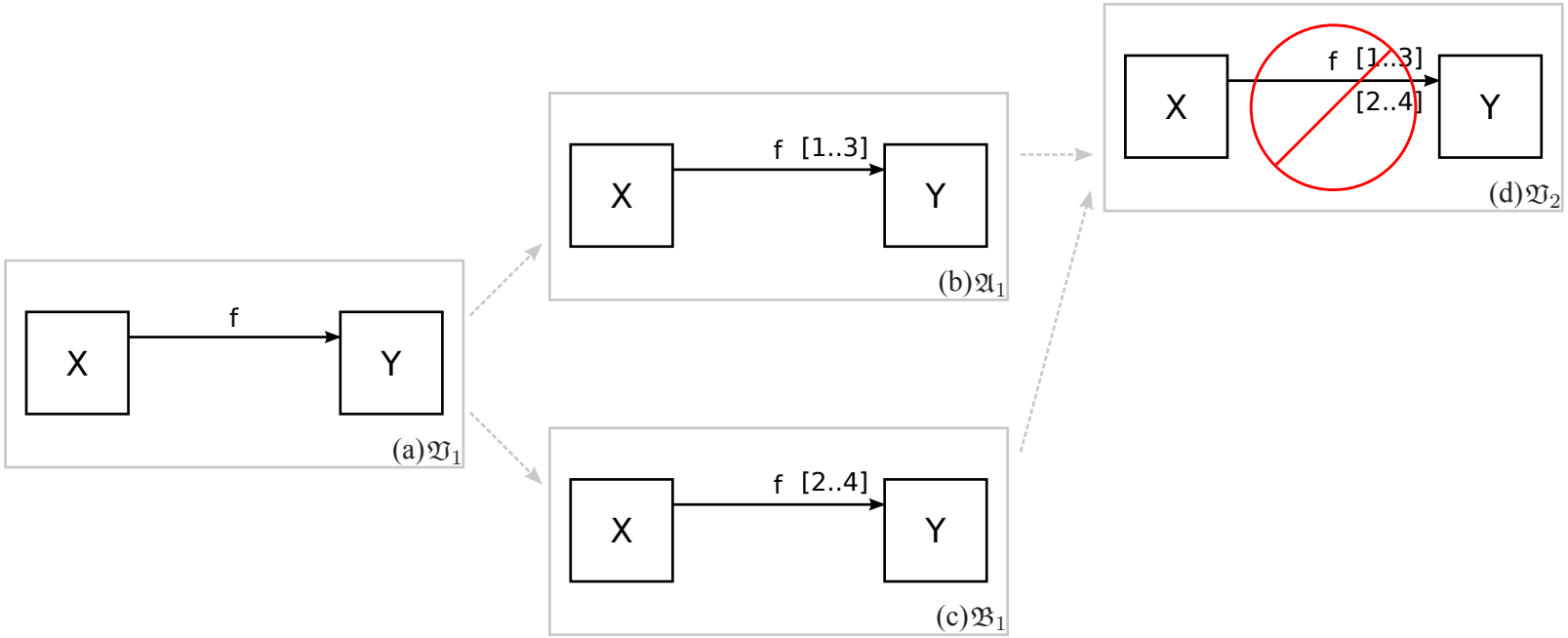
Bob's copy



Timeline

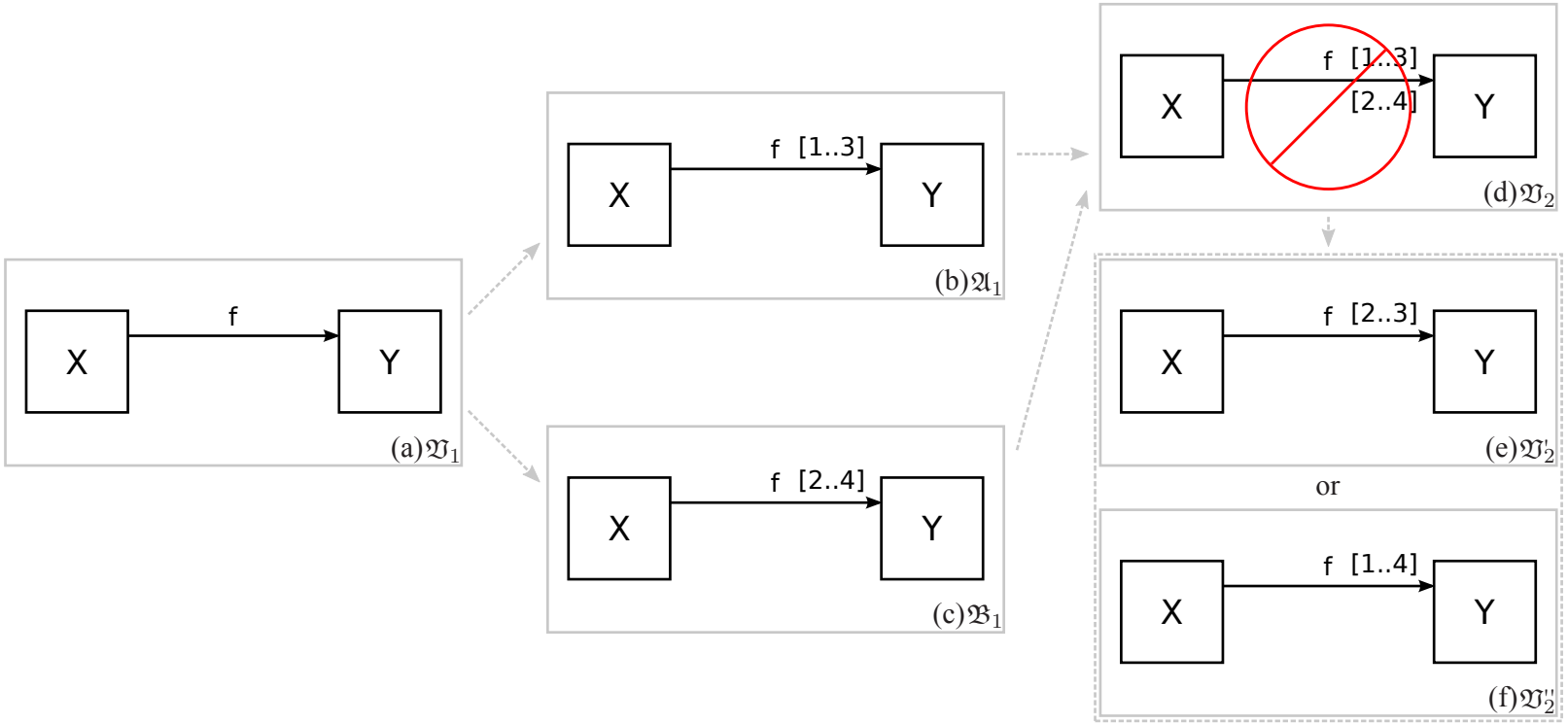


Example 1: multiplicity



- Model merging
  - push out [Rossini et al., 2010]
- Conflict detection

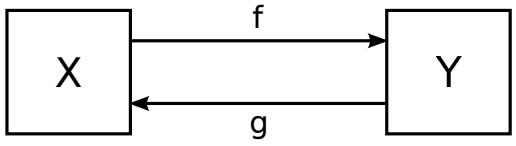
Example 1: multiplicity



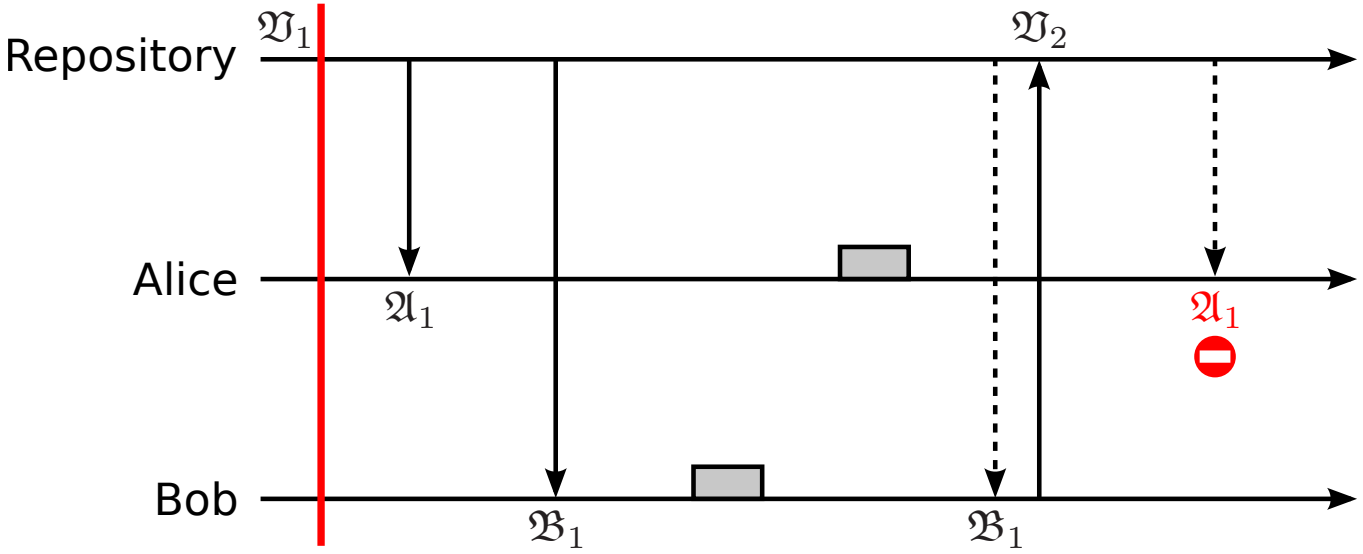
- Model merging
  - push out [Rossini et al., 2010]
- Conflict detection
- Conflict resolution

Example 2: multiplicity and surjectivity

Repository Alice's copy Bob's copy

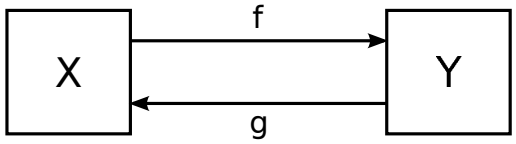


Timeline

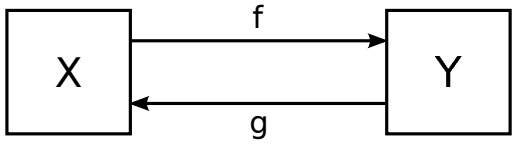


Example 2: multiplicity and surjectivity

Repository

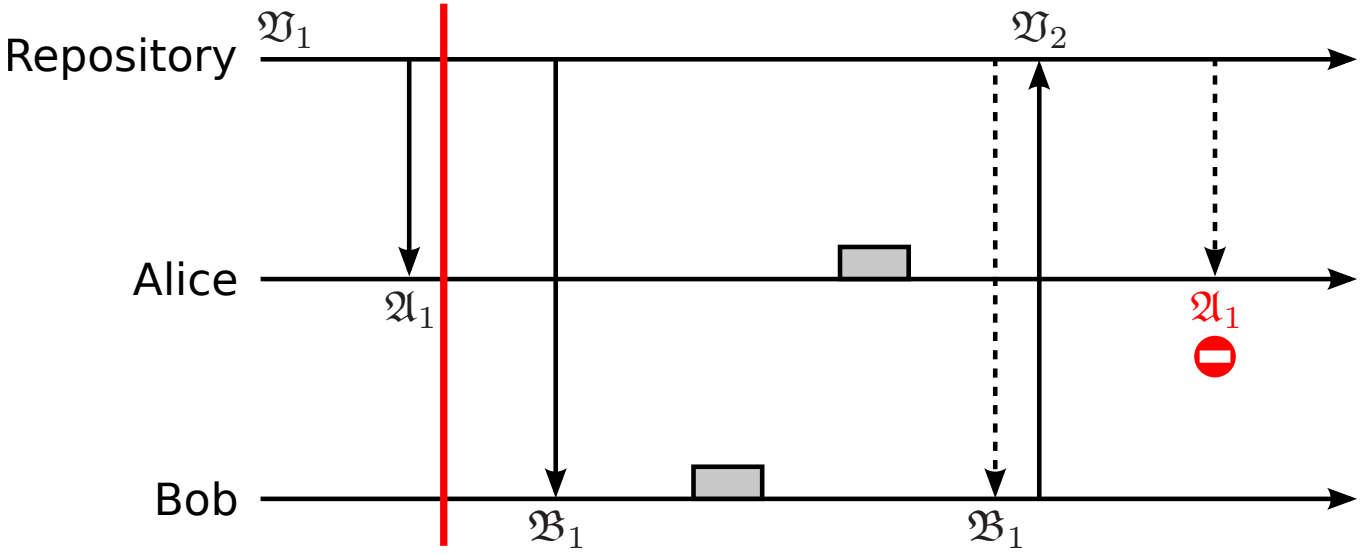


Alice's copy



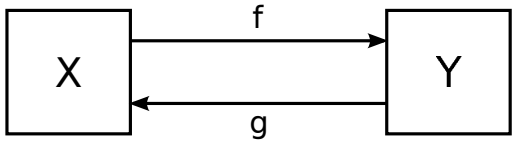
Bob's copy

Timeline

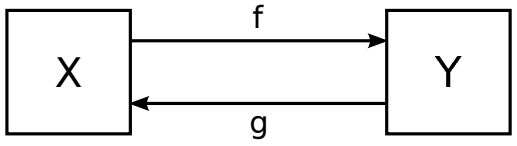


Example 2: multiplicity and surjectivity

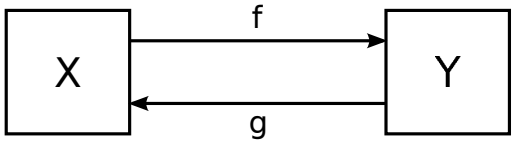
Repository



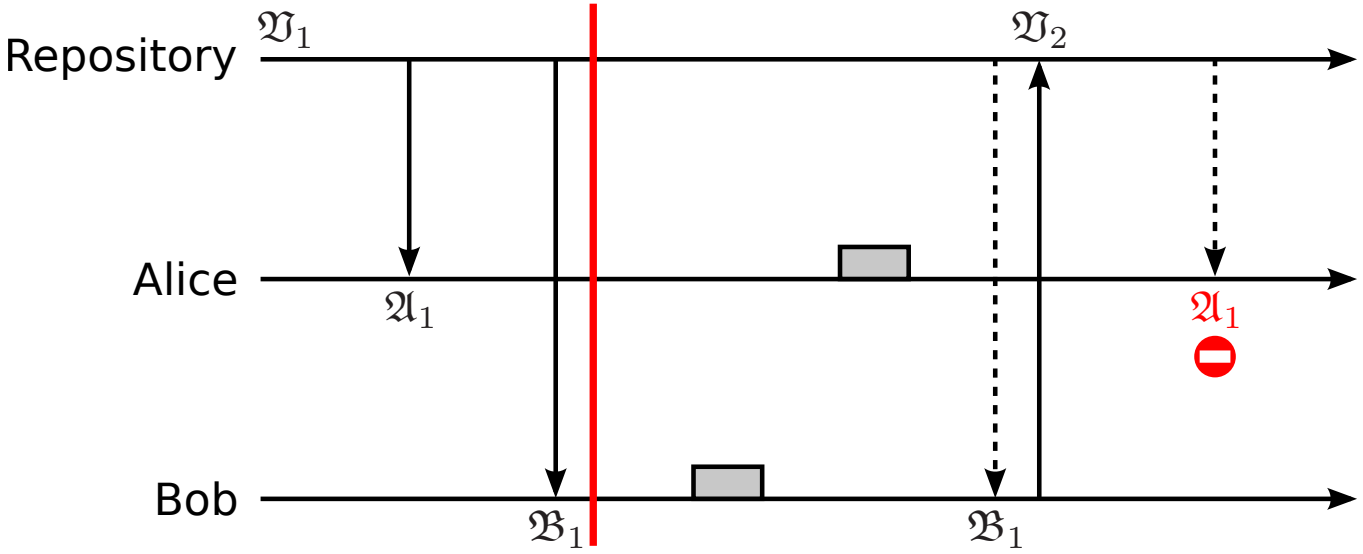
Alice's copy



Bob's copy

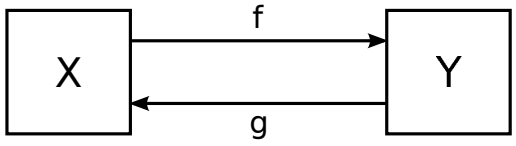


Timeline

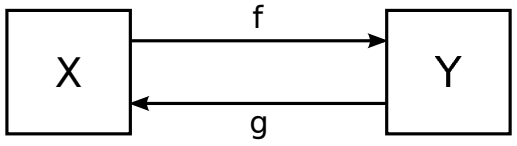


Example 2: multiplicity and surjectivity

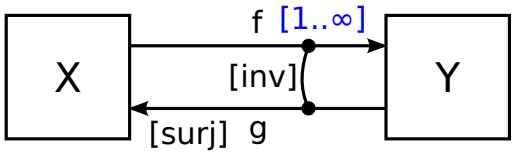
Repository



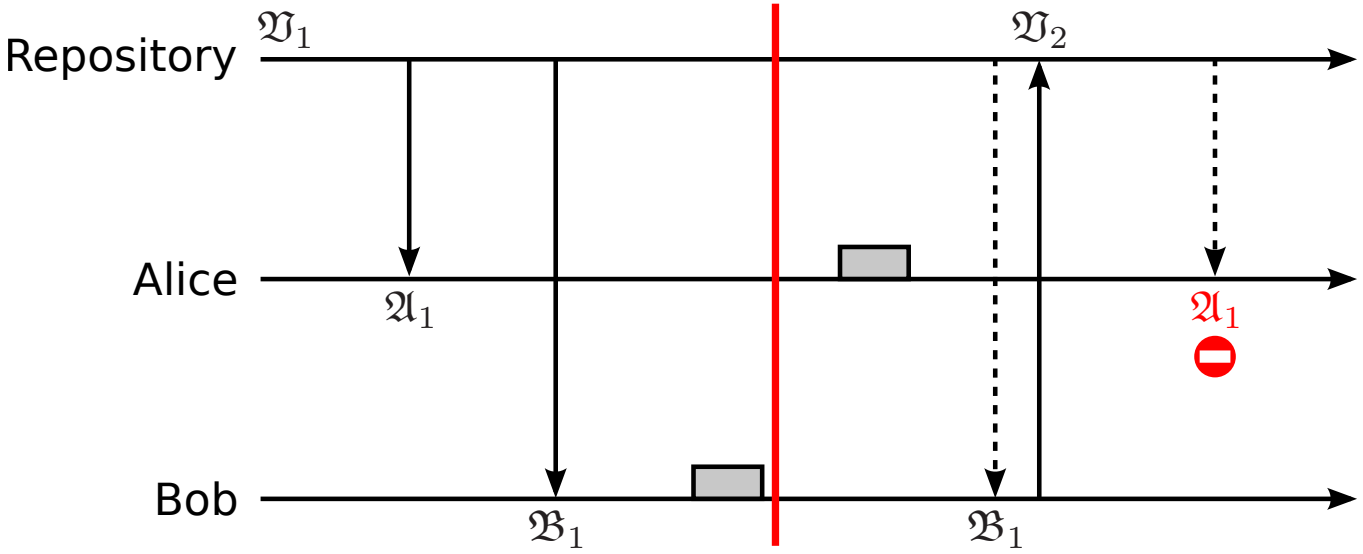
Alice's copy



Bob's copy

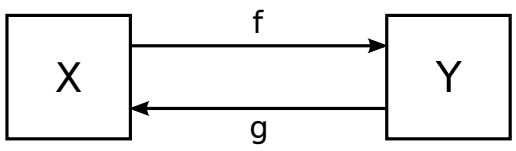


Timeline

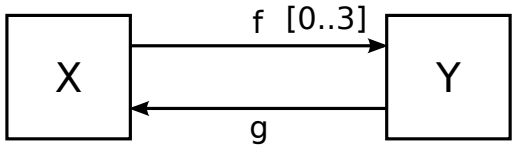


Example 2: multiplicity and surjectivity

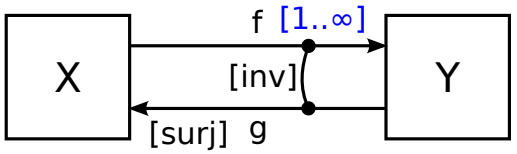
Repository



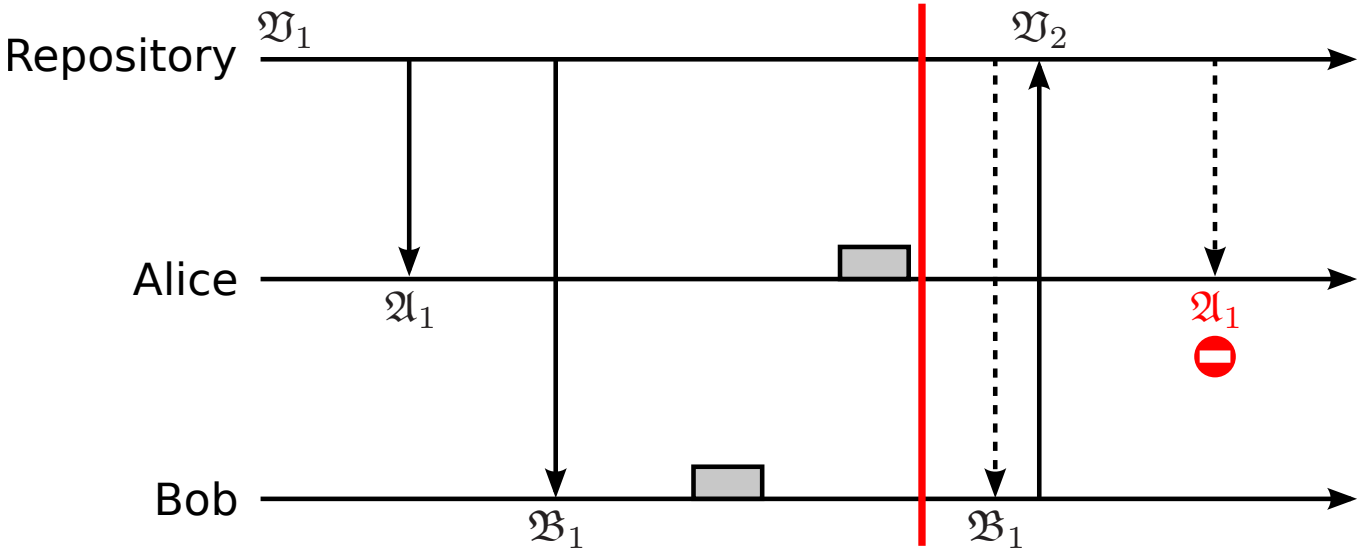
Alice's copy



Bob's copy

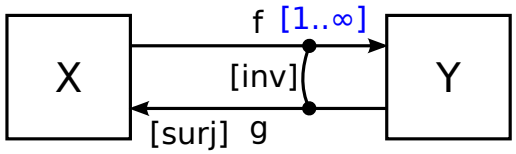


Timeline

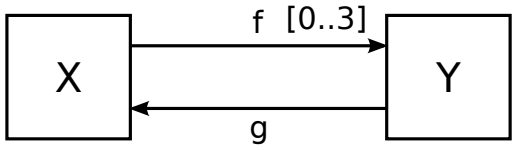


Example 2: multiplicity and surjectivity

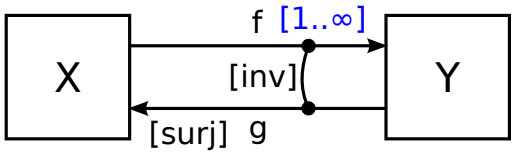
Repository



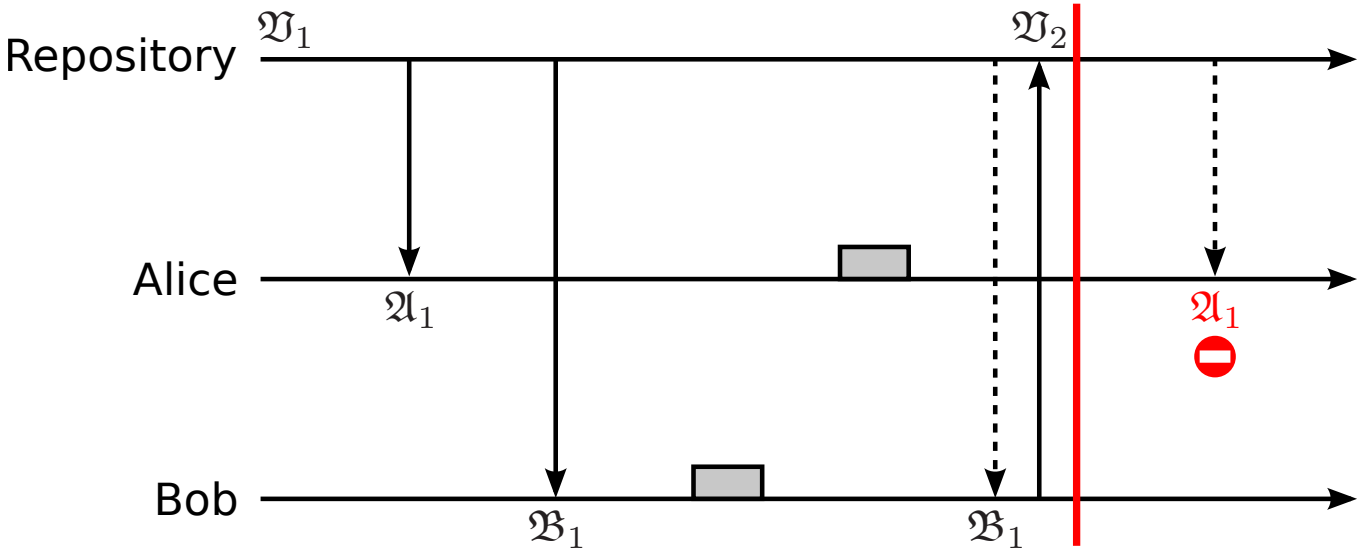
Alice's copy



Bob's copy

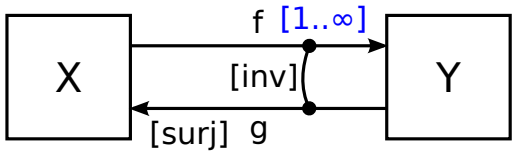


Timeline

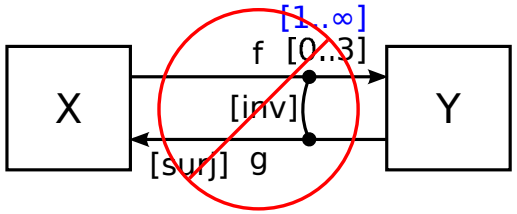


Example 2: multiplicity and surjectivity

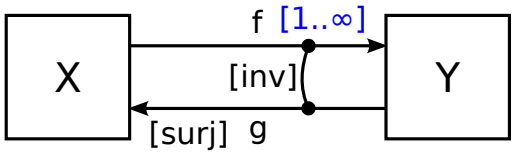
Repository



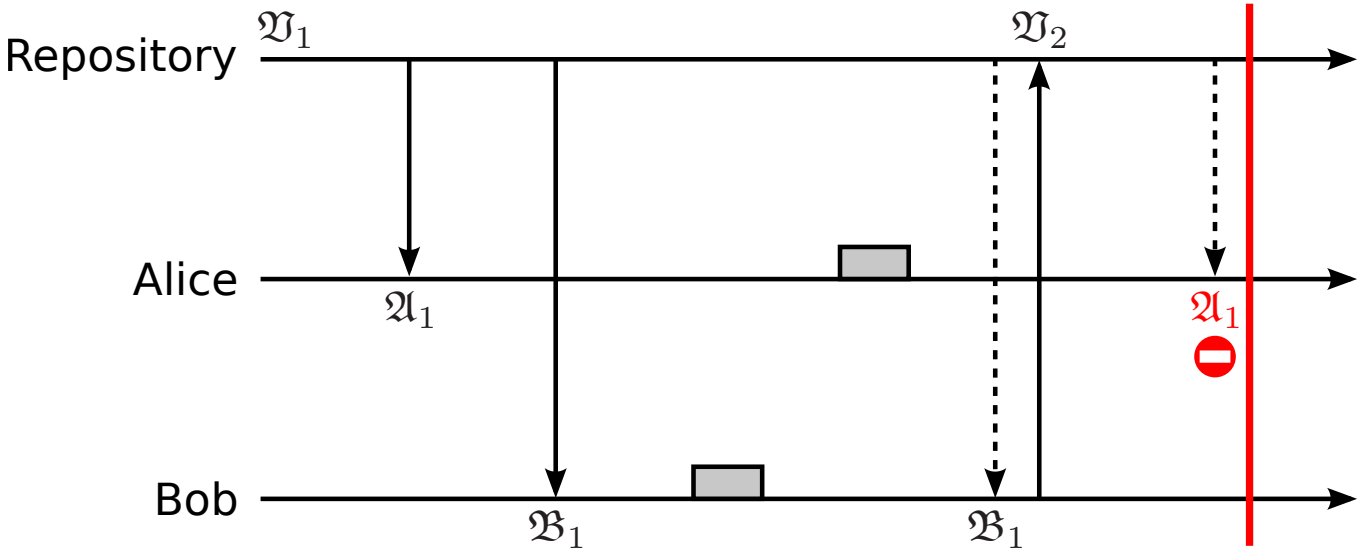
Alice's copy



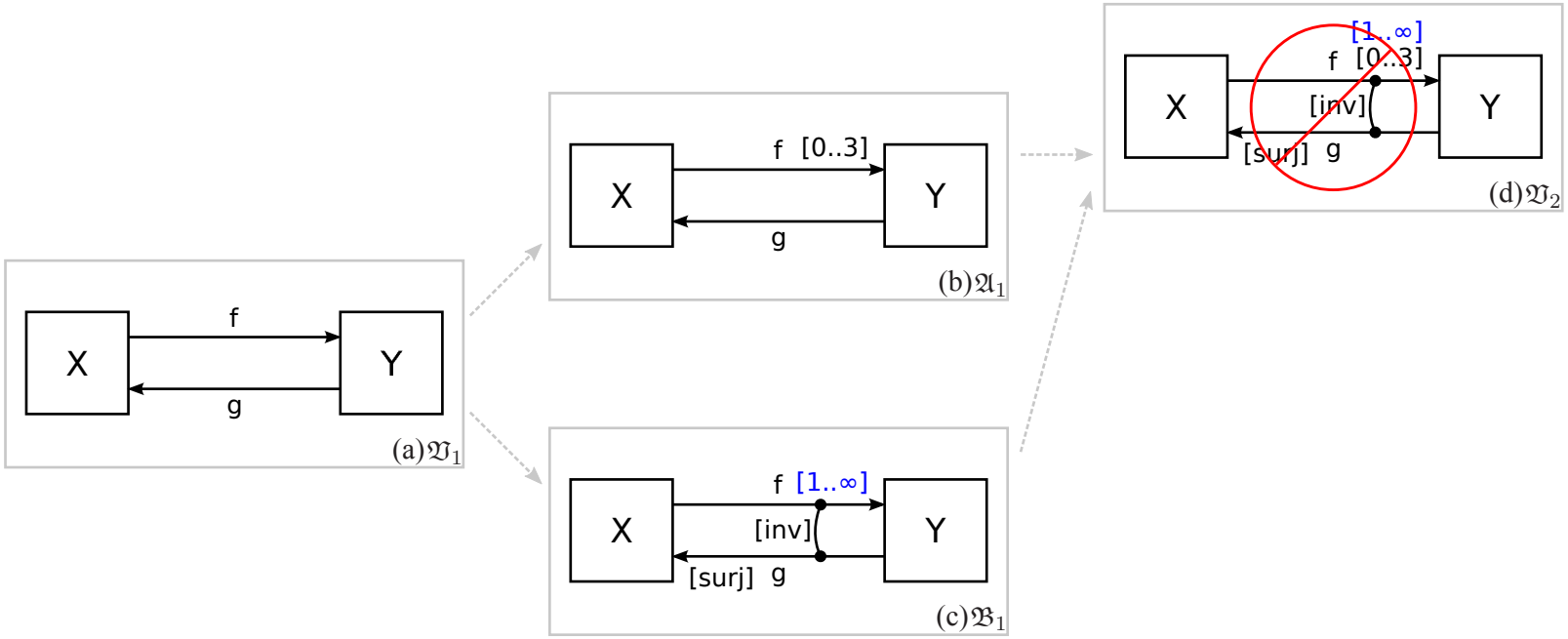
Bob's copy



Timeline

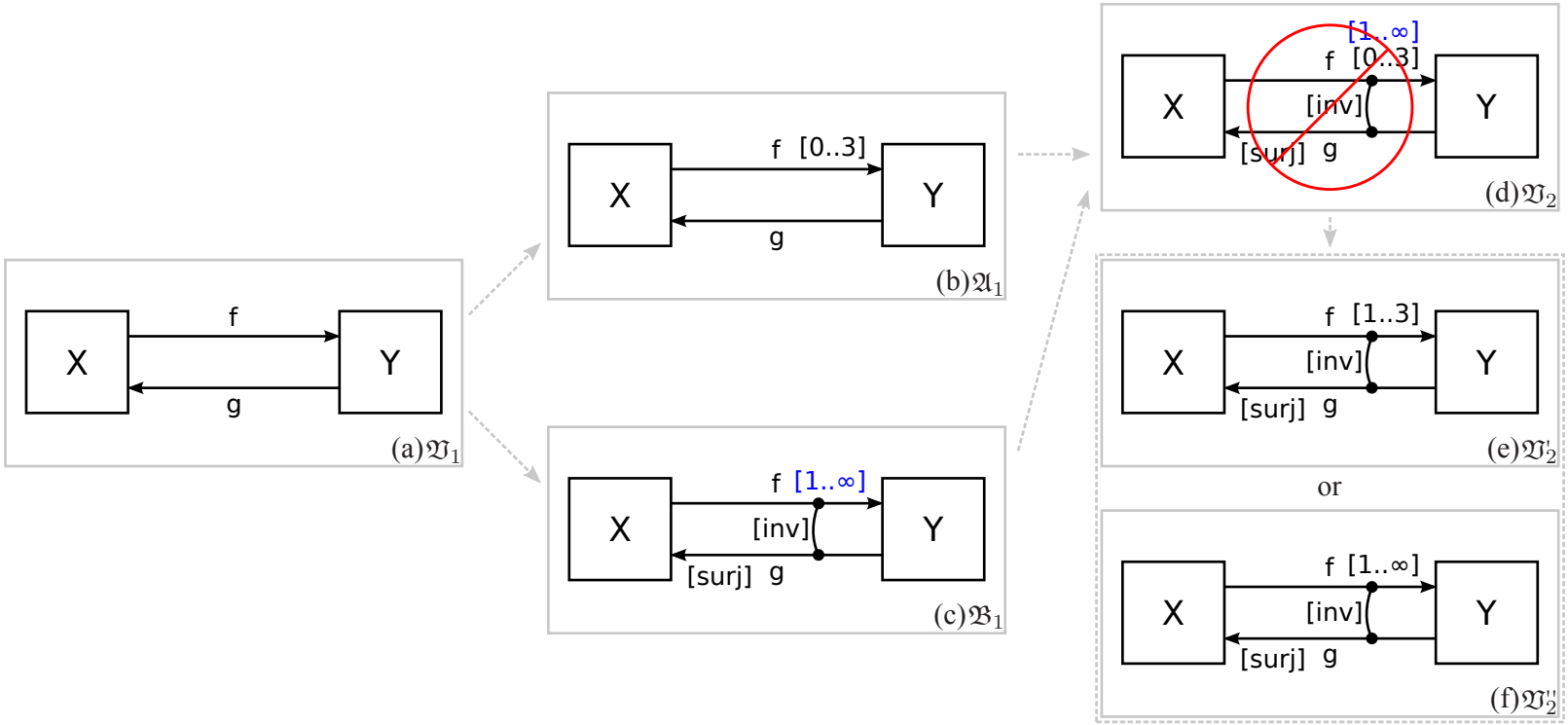


Example 2: multiplicity and surjectivity



- Model merging
- Conflict detection

Example 2: multiplicity and surjectivity



- Model merging
- Conflict detection
- Conflict resolution

Summary

- Extension of DPF
  - specification entailments
- Constraint-aware model merging
  - detection of semantic conflicts
  - resolution of conflicts

Future work

- Fully-fledge logic for predicates
  - Deduction calculus
- Prototype tool
- Real-size case study

Related work

- [Westfechtel, 2010] Three-way merging of Ecore models based on set theory and predicate logic
- [Taentzer et al., 2010] Merging of typed attributed graphs based on graph transformations and category theory
- [da Silva et al., 2010] Automatic generation of repair plans for a given inconsistent model

Related work

- [Westfechtel, 2010] Three-way merging of Ecore models based on set theory and predicate logic
- [Taentzer et al., 2010] Merging of typed attributed graphs based on graph transformations and category theory
- [da Silva et al., 2010] Automatic generation of repair plans for a given inconsistent model
- Constraints on model elements not considered

Thank you!

Questions?

For more information and list of publications visit:
`http://dpf.hib.no/`

Resolution pattern

$$
\begin{aligned}
& ([\texttt{mult}(1,3)], \delta_3) \vee ([\texttt{mult}(2,4)], \delta_3) \\
\equiv \quad & ([\texttt{mult}(1,4)], \delta_3)
\end{aligned}
$$

Resolution pattern

$$
\begin{aligned}
&\big(\texttt{[mult}(m_1, n_1)\texttt{]}, \delta\big) \vee \big(\texttt{[mult}(m_2, n_2)\texttt{]}, \delta\big) \\
\equiv &\begin{cases}
\big(\texttt{[mult}(m_1, n_1)\texttt{]}, \delta\big) & \text{if} \quad m_1 \leq m_2 \leq n_2 \leq n_1 \\
\big(\texttt{[mult}(m_2, n_2)\texttt{]}, \delta\big) & \text{if} \quad m_2 \leq m_1 \leq n_1 \leq n_2 \\
\big(\texttt{[mult}(m_1, n_2)\texttt{]}, \delta\big) & \text{if} \quad m_1 \leq m_2 \leq n_1 \leq n_2 \\
\big(\texttt{[mult}(m_2, n_1)\texttt{]}, \delta\big) & \text{if} \quad m_2 \leq m_1 \leq n_2 \leq n_1
\end{cases}
\end{aligned}
$$