Categorical Semantics of Reversible Pattern-Matching

Louis Lemonnier, Benoît Valiron, Kostia Chardonnet

September 1st, 2021
Reversible programming
Reversibility

- Reversible computation: only apply bijections.

---

Reversibility

- Reversible computation: only apply bijections.
- Application in low-consumption hardware \(^1\).

Reversibility

- Reversible computation: only apply bijections.
- Application in low-consumption hardware \(^1\).
- Application in quantum computing, where operations (except measurement) are reversible.

Reversibility

- Reversible computation: only apply bijections.
- Application in low-consumption hardware \(^1\).
- Application in quantum computing, where operations (except measurement) are reversible.

What we are focusing on today:

Reversibility

- Reversible computation: only apply bijections.
- Application in low-consumption hardware \(^1\).
- Application in quantum computing, where operations (except measurement) are reversible.

What we are focusing on today:

- Reversible programming language

Reversibility

- Reversible computation: only apply bijections.
- Application in low-consumption hardware \(^1\).
- Application in quantum computing, where operations (except measurement) are reversible.

What we are focusing on today:

- Reversible programming language
- Detailed denotational semantics

---

Terms and types

(Value types) \( a, b \ ::= \alpha \mid a \oplus b \mid a \otimes b \)
(Iso types) \( T ::= a \leftrightarrow b \)
(Values) \( v ::= c_\alpha \mid x \mid \text{inj}_l v \mid \text{inj}_r v \mid \langle v_1, v_2 \rangle \)
(Functions) \( \omega ::= \{ \mid v_1 \leftrightarrow v'_1 \mid \ldots \mid v_n \leftrightarrow v'_n \} \)
(Terms) \( t ::= v \mid \omega t \)
(Value types) \[a, b ::= \alpha \mid a \oplus b \mid a \otimes b\]
(Iso types) \[T ::= a \leftrightarrow b\]
(Values) \[v ::= c_\alpha \mid x \mid \text{inj}_l v \mid \text{inj}_r v \mid \langle v_1, v_2 \rangle\]
(Function) \[\omega ::= \{ \mid v_1 \leftrightarrow v'_1 \mid \ldots \mid v_n \leftrightarrow v'_n \}\]
(Terms) \[t ::= v \mid \omega t\]

\[
\{ \mid \text{inj}_l x \leftrightarrow \text{inj}_r x \mid \text{inj}_r x \leftrightarrow \text{inj}_l x \}\]
Terms and types

(Value types)  \( a, b ::= \alpha \mid a \oplus b \mid a \otimes b \)

(Iso types)  \( T ::= a \leftrightarrow b \)

(Values)  \( v ::= c_\alpha \mid x \mid \text{inj}_l v \mid \text{inj}_r v \mid \langle v_1, v_2 \rangle \)

(Functions)  \( \omega ::= \{ \mid v_1 \leftrightarrow v'_1 \mid \ldots \mid v_n \leftrightarrow v'_n \} \)

(Terms)  \( t ::= v \mid \omega t \)

\[
\{ \mid \text{inj}_l x \leftrightarrow \text{inj}_r x \mid \text{inj}_r x \leftrightarrow \text{inj}_l x \}
\]

- Two partial morphisms joined
Terms and types

(Value types) \[ a, b ::= \alpha \mid a \oplus b \mid a \otimes b \]

(Iso types) \[ T ::= a \leftrightarrow b \]

(Values) \[ v ::= c_\alpha \mid x \mid \text{inj}_l v \mid \text{inj}_r v \mid \langle v_1, v_2 \rangle \]

(Functions) \[ \omega ::= \{ \mid v_1 \leftrightarrow v'_1 \mid \ldots \mid v_n \leftrightarrow v'_n \} \]

(Terms) \[ t ::= v \mid \omega t \]

\[
\{ \mid \text{inj}_l x \leftrightarrow \text{inj}_r x \mid \text{inj}_r x \leftrightarrow \text{inj}_l x \}
\]

- Two partial morphisms joined
- With compatible domains
Terms and types

(Value types) \( a, b ::= \alpha \mid a \oplus b \mid a \otimes b \)

(Iso types) \( T ::= a \leftrightarrow b \)

(Values) \( v ::= c_\alpha \mid x \mid \text{inj}_l v \mid \text{inj}_r v \mid \langle v_1, v_2 \rangle \)

_Functions_ \( \omega ::= \{ \mid v_1 \leftrightarrow v'_1 \mid \ldots \mid v_n \leftrightarrow v'_n \} \)

(Terms) \( t ::= v \mid \omega t \)

\[
\{ \mid \text{inj}_l x \leftrightarrow \text{inj}_r x \mid \text{inj}_r x \leftrightarrow \text{inj}_l x \}
\]

- Two partial morphisms _joined_
- With compatible domains
- Which inverses have compatible domains
Terms and types

(Value types) \( a, b ::= \alpha \mid a \oplus b \mid a \otimes b \)

(Iso types) \( T ::= a \leftrightarrow b \)

(Values) \( v ::= c_\alpha \mid x \mid \text{inj}_l v \mid \text{inj}_r v \mid \langle v_1, v_2 \rangle \)

(Functions) \( \omega ::= \{ \mid v_1 \leftrightarrow v'_1 \mid \ldots \mid v_n \leftrightarrow v'_n \} \)

(Terms) \( t ::= v \mid \omega t \)

\[
\{ \mid \text{inj}_l x \leftrightarrow \text{inj}_r x \mid \text{inj}_r x \leftrightarrow \text{inj}_l x \}
\]

- Two partial morphisms joined
- With compatible domains
- Which inverses have compatible domains

Operational semantics:

\[
\{ \mid \text{inj}_l x \leftrightarrow \text{inj}_r x \mid \text{inj}_r x \leftrightarrow \text{inj}_l x \}(\text{inj}_r \star) \longrightarrow \text{inj}_l \star
\]
Categorical model
Restriction and inverse category

Domain and partiality

Partial inverse

Compatibility

Union of partial isos
Restriction and inverse category

Domain and partiality

Restriction category: \( f \mapsto \bar{f} \).

Partial inverse

Compatibility

Union of partial isos
Restriction and inverse category

Domain and partiality

Restriction category: \( f \mapsto \bar{f} \).

\[ f \circ \bar{f} = f, \quad \bar{f} \circ \bar{g} = \bar{g} \circ \bar{f}, \quad \bar{f} \circ \bar{g} = \bar{f} \circ \bar{g}, \quad \bar{h} \circ f = f \circ \bar{h} \circ f. \]

Partial inverse

Compatibility

Union of partial isos
Restriction and inverse category

Domain and partiality

Restriction category: \( f \mapsto \tilde{f} \).

\[
\begin{align*}
    f \circ \tilde{f} &= f, \\
    \tilde{f} \circ \tilde{g} &= g \circ \tilde{f}, \\
    f \circ \tilde{g} &= \tilde{f} \circ g, \\
    \tilde{h} \circ f &= f \circ \tilde{h} \circ f.
\end{align*}
\]

Partial inverse

Compatibility

Union of partial isos
Restriction and inverse category

Domain and partiality

Restriction category: \( f \mapsto \overline{f} \).
\[
f \circ \overline{f} = f, \quad \overline{f} \circ \overline{g} = \overline{g} \circ \overline{f}, \quad \overline{f \circ g} = \overline{f} \circ \overline{g}, \quad \overline{h \circ f} = f \circ \overline{h \circ f}.
\]

Partial inverse

Inverse category: \( f \mapsto \overline{f} \). \( \overline{f} \circ f = \overline{f} \) and \( f \circ \overline{f} = \overline{f} \).

Compatibility

Union of partial isos
Restriction and inverse category

Domain and partiality

Restriction category: $f \mapsto \overline{f}$.

$$f \circ \overline{f} = f, \quad \overline{f} \circ \overline{g} = \overline{g \circ f}, \quad \overline{f \circ g} = \overline{f} \circ \overline{g}, \quad \overline{h \circ f} = \overline{f \circ h \circ f}.$$ 

Partial inverse

Inverse category: $f \mapsto f^\circ$. 

$$f^\circ \circ f = \overline{f} \quad \text{and} \quad f \circ f^\circ = \overline{f}.$$ 

Compatibility

Union of partial isos
Restriction and inverse category

Domain and partiality

Restriction category: $f \mapsto \overline{f}$.

$$f \circ \overline{f} = f, \quad \overline{f} \circ \overline{g} = \overline{g} \circ \overline{f}, \quad f \circ \overline{g} = \overline{f} \circ \overline{g}, \quad \overline{h \circ f} = \overline{f} \circ \overline{h \circ f}.$$ 

Partial inverse

Inverse category: $f \mapsto f^\circ$. $f^\circ \circ f = \overline{f}$ and $f \circ f^\circ = \overline{f}$.

Compatibility

Restriction compatible:

$$f \sim g : f \overline{g} = g \overline{f}, \quad f \sim g : f \sim g \text{ and } f^\circ \sim g^\circ.$$ 

Union of partial isos
Restriction and inverse category

Domain and partiality

Restriction category: \( f \mapsto \overline{f} \).

\[
\begin{align*}
f \circ \overline{f} &= f, & \overline{f} \circ g &= \overline{g} \circ f, & f \circ \overline{g} &= \overline{f} \circ g, & \overline{h} \circ f &= f \circ \overline{h} \circ f.
\end{align*}
\]

Partial inverse

Inverse category: \( f \mapsto f^\circ \). \( f^\circ \circ f = \overline{f} \) and \( f \circ f^\circ = \overline{f} \).

Compatibility

Restriction compatible:

\[
\begin{align*}
f \sim g : f \overline{g} &= g \overline{f}, & f \sim g : f \sim g \text{ and } f^\circ \sim g^\circ.
\end{align*}
\]

Union of partial isos
Restriction and inverse category

Domain and partiality

Restriction category: \( f \mapsto \bar{f} \).

\[ f \circ \bar{f} = f, \quad \bar{f} \circ \bar{g} = \bar{g} \circ \bar{f}, \quad \bar{f} \circ \bar{g} = \bar{f} \circ \bar{g}, \quad \bar{h} \circ f = f \circ \bar{h} \circ f. \]

Partial inverse

Inverse category: \( f \mapsto f^\circ \). \( f^\circ \circ f = \bar{f} \) and \( f \circ f^\circ = \bar{f} \).

Compatibility

Restriction compatible:

\[ f \bowtie g : f \bar{g} = g \bar{f}, \quad f \bowtie g : f \bowtie g \quad \text{and} \quad f^\circ \bowtie g^\circ. \]

Union of partial isos

Partial order: \( f \leq g : g \bar{f} = f \).
Restriction and inverse category

Domain and partiality

Restriction category: \( f \mapsto \tilde{f} \).

\[
\begin{align*}
f \circ \tilde{f} &= f, \\
\tilde{f} \circ g &= g \circ \tilde{f}, \\
\tilde{f} \circ \tilde{g} &= \tilde{f} \circ g, \\
h \circ \tilde{f} &= \tilde{f} \circ h \circ f.
\end{align*}
\]

Partial inverse

Inverse category: \( f \mapsto f^\circ \). \( f^\circ \circ f = \tilde{f} \) and \( f \circ f^\circ = \tilde{f} \).

Compatibility

Restriction compatible:

\[
\begin{align*}
f \sim g : fg &= g\tilde{f}, \\
\tilde{f} \sim g : f \sim g \text{ and } f^\circ \sim g^\circ.
\end{align*}
\]

Union of partial isos

Partial order: \( f \leq g : g\tilde{f} = f \).
Restriction and inverse category

**Domain and partiality**

Restriction category: \( f \mapsto \bar{f} \).

\[
\begin{align*}
\bar{f} \circ f &= f, & \bar{f} \circ \bar{g} &= \bar{g} \circ \bar{f}, & \bar{f} \circ \bar{g} &= \bar{f} \circ \bar{g}, & \bar{h} \circ f &= f \circ \bar{h} \circ f.
\end{align*}
\]

\[
\begin{array}{ccc}
\text{h:} & a & \rightarrow & a \\
& b & \rightarrow & b \\
& c & \rightarrow & c \\
\end{array}
\quad
\begin{array}{ccc}
\text{\bar{h}:} & a & \rightarrow & a \\
& b & \rightarrow & b \\
& c & \rightarrow & c \\
\end{array}
\]

**Partial inverse**

Inverse category: \( f \mapsto f^\circ \). \( f^\circ \circ f = \bar{f} \) and \( f \circ f^\circ = \bar{f} \).

\[
\begin{array}{ccc}
\text{h^\circ:} & a & \rightarrow & a \\
& b & \rightarrow & b \\
& c & \rightarrow & c \\
\end{array}
\]

**Compatibility**

Restriction compatible:

\[
\begin{align*}
f \sim g : f \bar{g} &= g \bar{f}, & f \sim g & : f \sim g \quad \text{and} \quad f^\circ \sim g^\circ.
\end{align*}
\]

\[
\begin{array}{ccc}
\text{f:} & a & \rightarrow & a \\
& b & \rightarrow & b \\
& c & \rightarrow & c \\
\end{array}
\quad
\begin{array}{ccc}
\text{g:} & a & \rightarrow & a \\
& b & \rightarrow & b \\
& c & \rightarrow & c \\
\end{array}
\]

**Union of partial isos**

Partial order: \( f \leq g : g \bar{f} = f \).

\[
\begin{array}{ccc}
\text{k:} & a & \rightarrow & a \\
& b & \rightarrow & b \\
& c & \rightarrow & c \\
\end{array}
\]

**Join**

\[
\bigvee_{s \in S} s.
\]

if \( s \leq t \), then \( s \leq \bigvee_{s \in S} s \), \( \bigvee_{s \in S} s \leq t \), \( \bigvee_{s \in S} s = \bigvee_{s \in S} \bar{s} \),
Restriction and inverse category

**Domain and partiality**

Restriction category: $f \mapsto \overline{f}$.

$$f \circ \overline{f} = f, \quad \overline{f} \circ g = \overline{g} \circ \overline{f}, \quad \overline{f} \circ \overline{g} = \overline{f \circ g}, \quad \overline{h \circ f} = f \circ h \circ \overline{f}.$$ 

**Partial inverse**

Inverse category: $f \mapsto \overline{f}$. $f \circ \overline{f} = \overline{f}$ and $f \circ \overline{f} = \overline{f}$.

**Compatibility**

Restriction compatible:

$$f \sim g : f \overline{g} = \overline{g} f, \quad f \sim g : f \sim g \quad \text{and} \quad f \circ \overline{g} \sim g \circ \overline{f}.$$ 

**Union of partial isos**

Partial order: $f \leq g : g \overline{f} = f$.

Join: $\bigvee_{s \in S} s$. 

if $s \leq t$, then $s \leq \bigvee_{s \in S} s$, $\bigvee_{s \in S} s \leq t$, $\overline{\bigvee_{s \in S} s} = \bigvee_{s \in S} \overline{s}$, $f \circ \left( \bigvee_{s \in S} s \right) = \bigvee_{s \in S} f s$, $\overline{\left( \bigvee_{s \in S} s \right)} \circ g = \bigvee_{s \in S} s g$.

$\overline{\overline{f}} = f$.
Restriction and inverse category

Domain and partiality

Restriction category: $f \mapsto \tilde{f}$.

\[ f \circ \tilde{f} = f, \quad \tilde{f} \circ g = g \circ \tilde{f}, \quad \tilde{f} \circ \tilde{g} = \tilde{f} \circ \tilde{g}, \quad \tilde{h} \circ f = f \circ h \circ f. \]

Partial inverse

Inverse category: $f \mapsto f^\circ$. $f^\circ \circ f = \tilde{f}$ and $f \circ f^\circ = \tilde{f}$.

Compatibility

Restriction compatible:

$f \bowtie g : fg = g\tilde{f}$, $f \bowtie g : f \bowtie g$ and $f^\circ \bowtie g^\circ$.

Union of partial isos

Partial order: $f \leq g : g\tilde{f} = f$.

Join: $\bigvee_{s \in S} s$.

if $s \leq t$, then $s \leq \bigvee_{s \in S} s$, $\bigvee_{s \in S} s \leq t$, $\overline{\bigvee_{s \in S} s} = \bigvee_{s \in S} \overline{s}$,

$\bigvee_{s \in S} f \left( \bigvee_{s \in S} s \right) = \bigvee_{s \in S} fs$, $\left( \bigvee_{s \in S} s \right) \circ g = \bigvee_{s \in S} sg$.
Example

S a set. \( \text{PlId}_S \) category with one object \( * \) and morphisms \( Y \subseteq S \).

- Composition: intersection
- Identity: \( S \)
- Restriction of \( Y \): \( Y \)
- Partial inverse of \( Y \): \( Y \)
- Join: union.
Denotational semantics
\[ [\Delta] = [a_1] \otimes \cdots \otimes [a_n] \] whenever \( \Delta \vdash x_1 : a_1, \ldots, x_n : a_n \).

\[ [\Delta \vdash \text{inj} \_ v : a \oplus b] \vdash \iota_l \circ f, \] whenever \( f = [\Delta \vdash v : a] \), and similarly for the right-projection.

If \( f = [\Delta_1 \vdash v_1 : a_1] \) and \( g = [\Delta_2 \vdash v_2 : a_2] \),
\[ [\Delta_1, \Delta_2 \vdash \langle v_1, v_2 \rangle : a_1 \otimes a_2] \vdash f \otimes g. \]
An iso: \( \{ v_1 \leftrightarrow v'_1 \mid \ldots \mid v_n \leftrightarrow v'_n \} : a \leftrightarrow b. \)
An iso: \( \{ v_1 \leftrightarrow v'_1 \mid \ldots \mid v_n \leftrightarrow v'_n \} : a \leftrightarrow b. \)

The situation:

\[
\begin{array}{c}
\Delta_i \\
\downarrow \omega \\
\leftarrow v_i \\
\downarrow \\
\leftarrow v'_i
\end{array}
\]

\[
\begin{array}{c}
\rightarrow a \\
\leftarrow b
\end{array}
\]
An iso: $\{ v_1 \leftrightarrow v'_1 | \ldots | v_n \leftrightarrow v'_n \} : a \leftrightarrow b$.

The situation:

![Diagram](image)

What we'd like to build:

![Diagram](image)

Careful: this diagram does not commute!
**Lemma**

If $v_1 \perp v_2$ and $v_1' \perp v_2'$, then $[v_1'] \circ [v_1] \prec [v_2'] \circ [v_2]'$. 
Lemma

If \( v_1 \perp v_2 \) and \( v'_1 \perp v'_2 \), then \([v'_1] \circ [v_1]^\circ \simeq [v'_2] \circ [v_2]^\circ\).

The definition of isos in the language involves these orthogonalities, thus all \([v'_i] \circ [v_i]^\circ\) form a compatible set.
Lemma

If $v_1 \perp v_2$ and $v'_1 \perp v'_2$, then $[v'_1] \circ [v_1]^\circ \simeq [v'_2] \circ [v_2]^\circ$.

The definition of isos in the language involves these orthogonalities, thus all $[v'_i] \circ [v_i]^\circ$ form a compatible set.

Definition

$$\left[\vdash \omega \{ \mid v_1 \leftrightarrow v'_1 \mid v_2 \leftrightarrow v'_2 \ldots \} : a \leftrightarrow b\right] = \bigvee_{i} [v'_i] \circ [v_i]^\circ : [a] \rightarrow [b].$$
What we would like to have thus

The language is based is pattern-matching:

$$\vdash_\omega \{ \quad v_1 \leftrightarrow v'_1 \mid v_2 \leftrightarrow v'_2 \quad \ldots \}$$

decidable which pattern a term $u$ fits.

It should also be in the denotational semantics:

$$([[v'_1] \circ [v_1]^{\circ}) \lor ([v'_2] \circ [v_2]^{\circ})) \circ [u] = [v'_i] \circ [v_i]^{\circ} \circ [u]$$

for some $i$. 
The language is based is pattern-matching:

\[ \vdash \omega \{ \mid v_1 \leftrightarrow v'_1 \mid v_2 \leftrightarrow v'_2 \ldots \} \]

decidable which pattern a term \( u \) fits.

It should also be in the denotational semantics:

\[
\left(\left(\left[ v'_1 \right] \circ \left[ v_1 \right]^\circ \right) \lor \left(\left[ v'_2 \right] \circ \left[ v_2 \right]^\circ \right)\right) \circ \left[ u \right] = \left[ v'_i \right] \circ \left[ v_i \right]^\circ \circ \left[ u \right]
\]

for some \( i \).

Usually, it relies on disjointness (\( \oplus \)).
Pattern-matching and consistency
In \((f \lor g)h\), we want \(h\) to choose:

\[
fh \lor gh = fh \text{ or } fh \lor gh = gh
\]

A join inverse category with this property will be called \textit{pattern-matching} category.
The origin of consistency

In \((f \lor g)h\), we want \(h\) to choose:

\[
fh \lor gh = fh \text{ or } fh \lor gh = gh
\]

A join inverse category with this property will be called pattern-matching category. With disjointness \((\oplus)\), rather \(fh = 0\) or \(gh = 0\).
The origin of consistency

In \((f \lor g)h\), we want \(h\) to choose:

\[ fh \lor gh = fh \text{ or } fh \lor gh = gh \]

A join inverse category with this property will be called *pattern-matching* category. With disjointness \((\oplus)\), rather \(fh = 0\) or \(gh = 0\).

**The idea:** translate it as a notion of *non decomposability*, to have \(fh \lor gh\) not decomposable.

For this, we need \(h\) to be *not decomposable* either.
The origin of consistency

In \((f \vee g)h\), we want \(h\) to choose:

\[
fh \vee gh = fh \quad \text{or} \quad fh \vee gh = gh
\]

A join inverse category with this property will be called \textit{pattern-matching} category. With disjointness \((\oplus)\), rather \(fh = 0\) or \(gh = 0\).

\textbf{The idea:} translate it as a notion of \textit{non decomposability}, to have \(fh \vee gh\) not decomposable.

For this, we need \(h\) to be \textit{not decomposable} either.

\textbf{Consistency:} for any morphism \(k\), if \(h\) is \textit{non decomposable}, so is \(kh\).

A join inverse category with this property will be called \textit{consistent} category.
In \((f \lor g)h\), we want \(h\) to choose:

\[ fh \lor gh = fh \text{ or } fh \lor gh = gh \]

A join inverse category with this property will be called \textit{pattern-matching} category. With disjointness \((\oplus)\), rather \(fh = 0\) or \(gh = 0\).

**The idea:** translate it as a notion of \textit{non decomposability}, to have \(fh \lor gh\) not decomposable.

For this, we need \(h\) to be \textit{not decomposable} either.

**Consistency:** for any morphism \(k\), if \(h\) is \textit{non decomposable}, so is \(kh\).

A join inverse category with this property will be called \textit{consistent} category.

Consistency implies pattern-matching.
The origin of consistency

In \((f \lor g)h\), we want \(h\) to choose:

\[
fh \lor gh = fh \text{ or } fh \lor gh = gh
\]

A join inverse category with this property will be called pattern-matching category. With disjointness \((\oplus)\), rather \(fh = 0\) or \(gh = 0\).

**The idea:** translate it as a notion of non decomposability, to have \(fh \lor gh\) not decomposable.

For this, we need \(h\) to be not decomposable either.

**Consistency:** for any morphism \(k\), if \(h\) is non decomposable, so is \(kh\). A join inverse category with this property will be called consistent category.

Consistency implies pattern-matching.

What about the other way around?
The different kinds

<table>
<thead>
<tr>
<th>Definition (Strongly non decomposable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ 0 \quad \longrightarrow \quad m ]</td>
</tr>
<tr>
<td>Thus ( m = u \lor v \Rightarrow u = 0 ) or ( v = 0 )</td>
</tr>
</tbody>
</table>
The different kinds

**Definition (Strongly non decomposable)**

\[ 0 \longrightarrow m \]

Thus \( m = u \lor v \Rightarrow u = 0 \) or \( v = 0 \)

**Definition (Linearly non decomposable)**

\[ 0 \dotsc f \longrightarrow m \]

Thus \( m = u \lor v \Rightarrow u \leq v \) or \( v \leq u \)
The different kinds

**Definition (Strongly non decomposable)**

\[ 0 \rightarrow m \]

Thus \( m = u \lor v \Rightarrow u = 0 \) or \( v = 0 \)

**Definition (Linearly non decomposable)**

\[ 0 \rightarrow \cdots \rightarrow f \rightarrow m \]

Thus \( m = u \lor v \Rightarrow u \leq v \) or \( v \leq u \)

**Definition (Weakly non decomposable)**

\[ 0 \xrightarrow{g} f \xrightarrow{m} m \]

Thus \( m = u \lor v \Rightarrow u = m \) or \( v = m \)
The different kinds

**Definition (Strongly non decomposable)**

\[ 0 \rightarrow m \]

Thus \( m = u \lor v \Rightarrow u = 0 \) or \( v = 0 \)

**Definition (Linearly non decomposable)**

\[ 0 \rightarrow \cdots \rightarrow f \rightarrow m \]

Thus \( m = u \lor v \Rightarrow u \leq v \) or \( v \leq u \)

**Definition (Weakly non decomposable)**

\[ 0 \xrightarrow{g} f \xrightarrow{m} \]

\[ h \]

Thus \( m = u \lor v \Rightarrow u = m \) or \( v = m \)

**strongly \( \Rightarrow \) linearly \( \Rightarrow \) weakly**
**Theorem**

A weakly pattern-matching category is weakly consistent.
The results

Theorem

A weakly pattern-matching category is weakly consistent.

Theorem

A join inverse category is strongly consistent.
The results

**Theorem**

A weakly pattern-matching category is weakly consistent.

**Theorem**

A join inverse category is strongly consistent.

**Theorem**

A join inverse category is linearly consistent.
### The results

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td>A weakly pattern-matching category is weakly consistent.</td>
</tr>
<tr>
<td><strong>Theorem</strong></td>
</tr>
<tr>
<td>A join inverse category is strongly consistent.</td>
</tr>
<tr>
<td><strong>Theorem</strong></td>
</tr>
<tr>
<td>A join inverse category is linearly consistent.</td>
</tr>
</tbody>
</table>

This allows to not rely on a disjointness ($\oplus$) structure. And to use a category like Plid$_S$ as a sound and adequate denotation.
The results

**Theorem**

A weakly pattern-matching category is weakly consistent.

**Theorem**

A join inverse category is strongly consistent.

**Theorem**

A join inverse category is linearly consistent.

This allows to not rely on a disjointness (⊕) structure. And to use a category like PId$_S$ as a sound and adequate denotation.
### The results

<table>
<thead>
<tr>
<th>Theorem</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A weakly pattern-matching category is weakly consistent.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>A join inverse category is strongly consistent.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>A join inverse category is linearly consistent.</strong></td>
<td></td>
</tr>
</tbody>
</table>

This allows to not rely on a disjointness (⊕) structure. And to use a category like PLd₅ as a sound and adequate denotation.
**Recursion**

$\mathcal{C}$ a join inverse category.
For any $A$, $B$ objects of $\mathcal{C}$, $\text{Hom}_\mathcal{C}(A, B)$ is a DCPO.

**Quantum programming language**

Patterns are vectors in a basis of a Hilbert space.
Work in progress.
Thank you