Stala

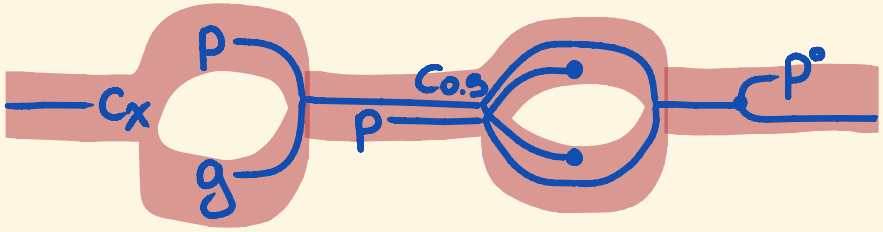
23 December 2025

PROGRAM LOGICS VIA DISTRIBUTIVE MONOIDAL CATEGORIES

Filippo Bonchi, U. of Pisa
Elena Di Lavore, U. of Oxford
Mario Román, U. of Oxford
Sam Staton, U. of Oxford

THE PIRAÑHA PROBLEM

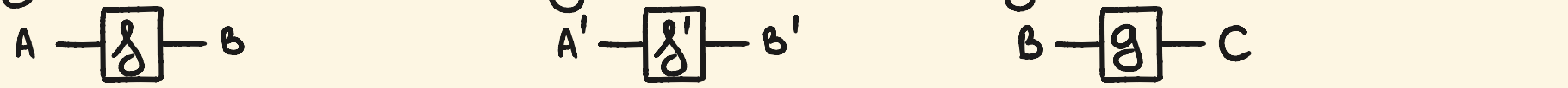
```
if coin(x)
| then F1 ← piranha
| else F1 ← goldFish
end
F2 ← piranha
if coin(0.5)
| then s ← F1
| else s ← F2
end
observe(s = piranha)
```



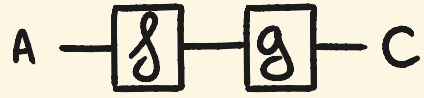
⇝ can we bound the probability of observing piranha ?

MORPHISMS AS PROGRAMS

$f : A \to B$ $\quad f' : A' \to B'$ $\quad g : B \to C$

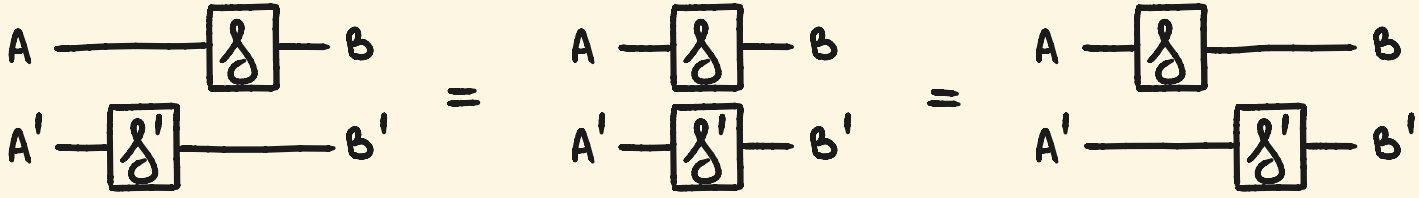


- Sequential composition. $f ; g : A \to C$



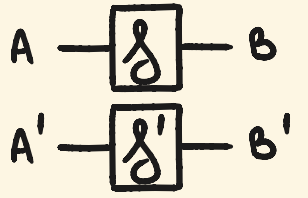
- Identity. $id_A : A \to A$

- Parallel composition. $f \otimes f' : A \otimes A' \to B \otimes B'$



- Symmetries. $\sigma_{A,B} : A \otimes B \to B \otimes A$

TWO KINDS OF PARALLEL COMPOSITION



execute f and f' ⇝ data flow

execute f or f' ⇝ control flow

ex for $T : \mathrm{Set} \to \mathrm{Set}$ x-monoidal monad,

$(\mathrm{Kl}T, \otimes, \{*\})$ ⇝ 'and'

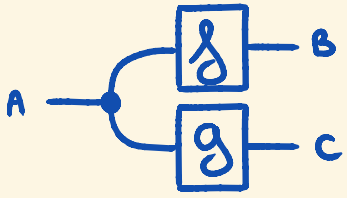
$(\mathrm{Kl}T, +, \emptyset)$ ⇝ 'or'

We need both kinds of parallel compositions

⇒ distributive monoidal categories $(\mathcal{C}, \otimes, 1, +, 0)$

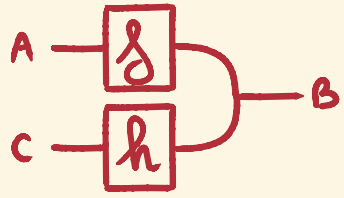
TWO MONOIDAL STRUCTURES

data flow



$A \longrightarrow B \otimes C$

control flow



$A + C \longrightarrow B$

[Arbib, Manes (1980)] [Walters (1992)]
[Jacobs (2010, 2016)]

OUTLINE

- [The guarded command language]
- Imperative categories
- Program logics

THE GUARDED COMMAND LANGUAGE (GCL)

- Signature

  $Var, \Sigma_E, \Sigma_G, \Sigma_P, \Sigma_C, \Sigma_D$

- Expressions

  $E ::= x \in Var \mid e(E, \ldots, E),\ e \in \Sigma_E$

- Guards

  $G ::= b(E, \ldots, E),\ b \in \Sigma_G \mid \bot \mid \top \mid G \wedge G \mid \neg G$

- Predicates

  $P ::= p(E, \ldots, E),\ p \in \Sigma_P \mid \top \mid \bot \mid P \wedge P \mid P +_G P \mid \check{G}$

- Commands

  $C ::= c \in \Sigma_C \mid skip \mid abort \mid x \leftarrow E \mid (x_1, \ldots, x_m) \xleftarrow{\$} \sigma,\ \sigma \in \Sigma_D \mid C;C$
  $\mid assert\ P \mid if\ B\ then\ C\ else\ C \mid while\ B\ do\ C$

INTERPRETING THE GUARDED COMMAND LANGUAGE

- Choose a type X for variables and fix $N := \#\mathrm{Var}$.
- Commands are endomorphisms of $X^N = \underbrace{X \otimes \cdots \otimes X}_{N \text{ times}}$
  ⇝ state-passing translation
- Expressions are deterministic and total $X^N \longrightarrow X$
- Guards are total $X^N \longrightarrow 1+1$
- Predicates are $X^N \longrightarrow 1$

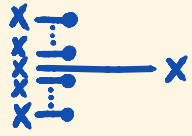
→ we also constructed an internal language (in the paper)
  ↳ succint with few primitives
  ↳ programs with arbitrary types

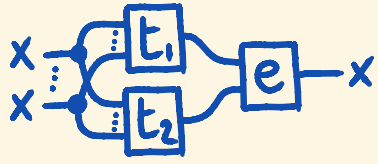
INTERPRETING THE GCL : EXPRESSIONS

$E ::= x_i \in Var \mid e(E, \dots, E),\ e \in \Sigma_E(n)$

Choose $\bar{e} : X^n \longrightarrow X$ for each $e \in \Sigma_E(n)$.

Define

$[\![x_i]\!] := \pi_i =$ 

$[\![e(t_1, \dots, t_m)]\!] := X^N \to m \,;\, ([\![t_1]\!] \otimes \cdots \otimes [\![t_m]\!]) \,;\, \bar{e} \overset{m=2}{=}$ 

⇒ copy-discard structure for variable handling
↳ compatible cocommutative comonoid on every object

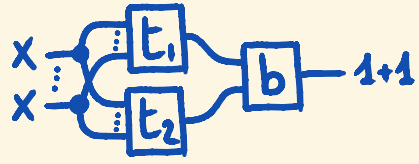
INTERPRETING THE GCL : GUARDS

$G ::= b(E,\dots,E),\ b \in \Sigma_G(n) \mid L \mid R \mid G \wedge G \mid \neg G$

Choose $\mathfrak{b} : X^n \longrightarrow 1+1$ total for each $b \in \Sigma_G(n)$

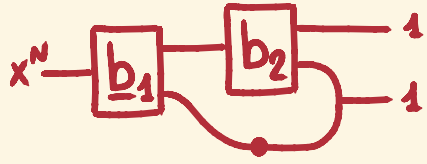
Define

$[\![b(t_1,\dots,t_m)]\!] := X^N \to X^m ; ([\![t_1]\!] \otimes \dots \otimes [\![t_m]\!]) ; \mathfrak{b}$ $\overset{m=2}{=}$



$[\![L]\!] := X^N \to ; \vdash 1,1 = X^N \to \vdash 1,1$

$[\![b_1 \wedge b_2]\!] :=$



$[\![R]\!] := X^N \to ; \dashv 1,1 = X^N \dashv 1,1$

$[\![\neg b]\!] :=$

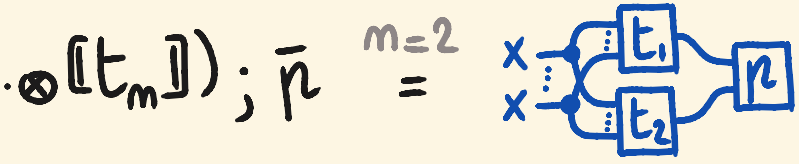
$\Rightarrow$ coproducts for choices

INTERPRETING THE GCL : PREDICATES

$P ::= p(E,\dots,E),\ p \in \Sigma_P(n) \mid \top \mid \bot \mid P \wedge P \mid P +_G P \mid \check{G}$

Choose $\bar{p} : X^n \longrightarrow 1$ for each $p \in \Sigma_P(n)$

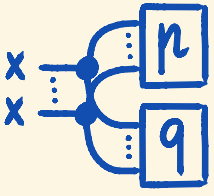
Define

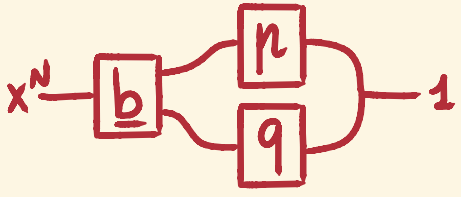
$[\![p(t_1,\dots,t_m)]\!] :=$ 

$[\![\top]\!] :=$ 

$[\![\bot]\!] :=$ 

$[\![\check{b}]\!] :=$ 

$[\![p \wedge q]\!] :=$ 

$[\![p +_b q]\!] :=$ 

INTERPRETING THE GCL : COMMANDS (1)

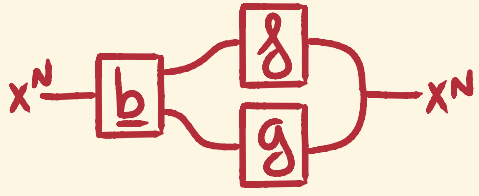
$C ::= c \in \Sigma_C \mid \text{skip} \mid \text{abort} \mid x \leftarrow E \mid (x_1, \ldots, x_m) \overset{\$}{\leftarrow} \sigma, \sigma \in \Sigma_D \mid C;C$
$\mid \text{assert } P \mid \text{if } B \text{ then } C \text{ else } C \mid \text{while } B \text{ do } C$

Choose $\bar{c} : X^N \to X^N$ for each $c \in \Sigma_C$
and $\bar{\sigma} : 1 \to X^m$ for each $\sigma \in \Sigma_D$

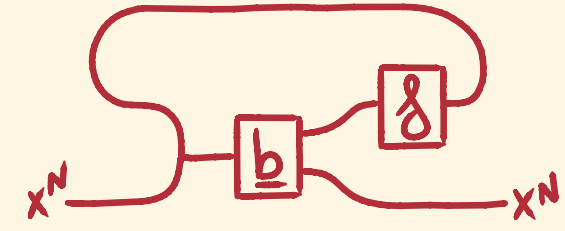
Define

$[\![\text{abort}]\!] := X^N \dashv\vdash X^N$

$[\![\text{if } b \text{ then } f \text{ else } g]\!]$
$:= [\![\underline{b}]\!] ; [[\![f]\!], [\![g]\!]]$



$[\![\text{while } b \text{ do } f]\!]$
$:= tr_{X^N}(\supset X^N ; [\![\underline{b}]\!] ; ([\![f]\!] + id_{X^N}))$



$\Rightarrow$ Traces for while loops

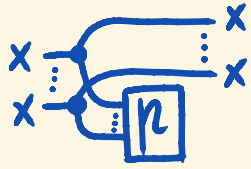
INTERPRETING THE GCL : COMMANDS (2)

$C ::= c \in \Sigma_C \mid \mathsf{skip} \mid \mathsf{abort} \mid x \leftarrow E \mid (x_1, \ldots, x_m) \xleftarrow{\$} \sigma,\ \sigma \in \Sigma_D \mid C;C$

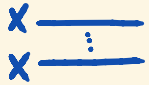
$\mid \mathsf{assert}\ P \mid \mathsf{if}\ B\ \mathsf{then}\ C\ \mathsf{else}\ C \mid \mathsf{while}\ B\ \mathsf{do}\ C$

Define

$[\![\mathsf{assert}\ p]\!] := X^N \to C ; (id_{X^N} \otimes [\![p]\!])$ =



$[\![\mathsf{skip}]\!] := id_{X^N}$ =

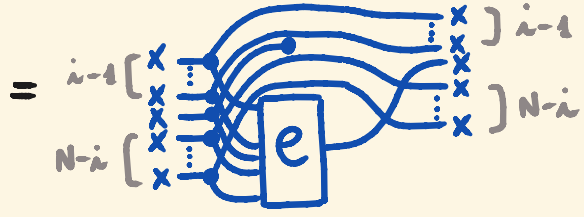


$[\![g;g]\!] := [\![g]\!]; [\![g]\!]$ =

$[\![x_i \leftarrow e]\!]$

$:= X^N \to C ; (\pi_{-i} \otimes [\![e]\!]) ; \chi_{X^{N-i}}^{X^{i-1}}$

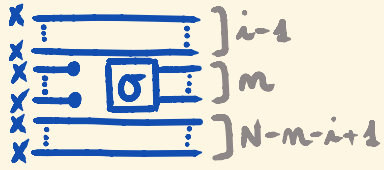
=



$[\![(x_i, \ldots, x_{i+m}) \xleftarrow{\$} \sigma]\!]$

$:= id_{X^{i-1}} \otimes (X^m \to \bullet ; \sigma - X^m) \otimes id_{X^{N-m-i+1}}$

=



OUTLINE

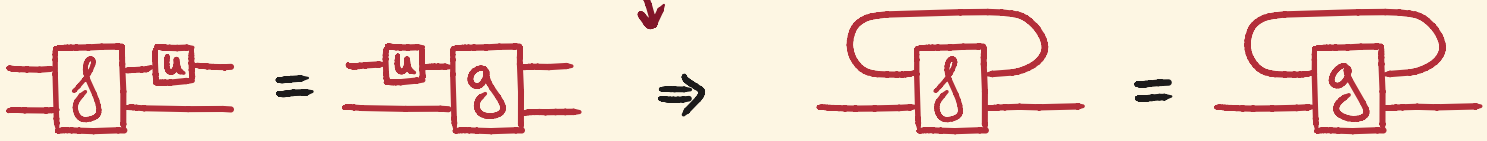
- The guarded command language
- [Imperative categories]
- Program logics

IMPERATIVE CATEGORIES

Category $\mathcal{C}$ with

- a copy-discard structure $(\otimes, 1)$
- finite coproducts $(+, 0)$
- uniform coproduct-trace $tr_S : \mathcal{C}(S+A, S+B) \to \mathcal{C}(A, B)$

$\mapsto$ distributive cd (injections det & tot)



$(\mathcal{C}, \otimes, I)$ symmetric monoidal category with coproducts and

$$\delta^L_{A,B,C} : (A \otimes B) + (A \otimes C) \xrightarrow{\simeq} A \otimes (B + C) \qquad \jmath^L_A : 0 \xrightarrow{\simeq} 0 \otimes A$$

$$\delta^R_{A,B,C} : (A \otimes C) + (B \otimes C) \xrightarrow{\simeq} (A + B) \otimes C \qquad \jmath^R_A : 0 \xrightarrow{\simeq} A \otimes 0$$

UNIFORMITY GIVES DIVERGENCE

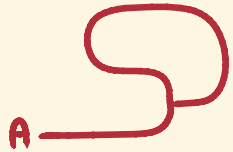
PROPOSITION

$\mathcal{C}$ category with coproducts and uniform trace

$\Rightarrow$ *0 is terminal*

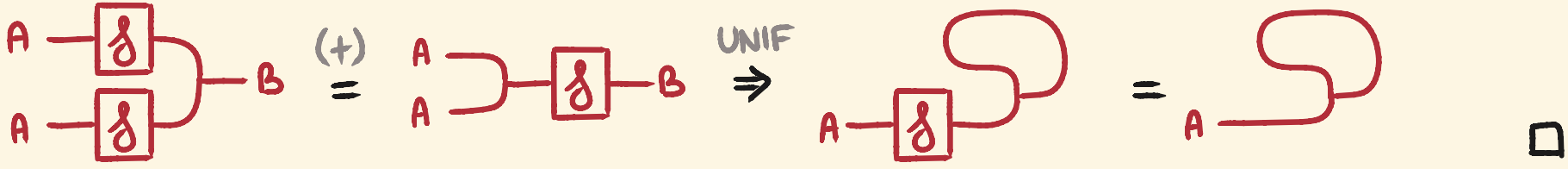
PROOF

Consider the morphisms $A\dashv : A \longrightarrow 0$ defined

as $A\dashv := tr_A[id_A, id_A] =$ 

$\rightsquigarrow$ while TRUE do skip

By uniformity, we show that they are natural.



□

EXAMPLES

Kleisli categories of partially additive monads

- $\mathbf{Par}$, sets and partial functions
- $\mathbf{Rel}$, sets and relations
- $\mathbf{Stoch}_{\leq}$, sets and countably supported partial stochastic kernels
- $\mathbf{BorelStoch}_{\leq}$, standard Borel spaces and measurable partial stochastic kernels

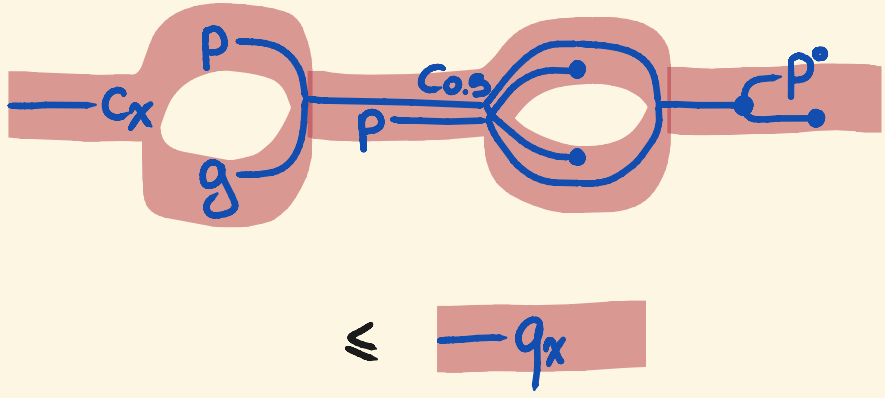
[Jacobs (2010), Jlasuo (2006)]

OUTLINE

- The guarded command language
- Imperative categories
- [Program logics]

THE PIRAÑHA PROBLEM

```
if coin(x)
| then F₁ ← piranha
| else F₁ ← goldFish
end
F₂ ← piranha
if coin(0.5)
| then s ← F₁
| else s ← F₂
end
observe(s = piranha)
```

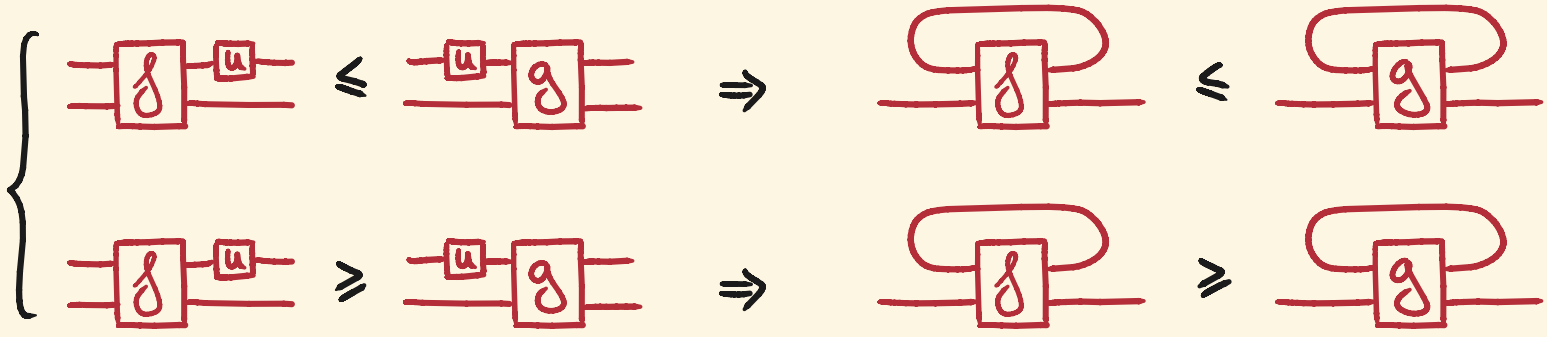


⇝ can we bound the probability of observing piranha ?
↳ we need an order on morphisms

POSETAL IMPERATIVE CATEGORY

A imperative category with

- a poset distributive enrichment
- posetally-uniform coproduct-trace



EXAMPLES

All our examples are poset-enriched.

- $\mathbf{Par}$, $f \leq g$ if $\forall a \in \mathrm{dom}\, f \quad f(a) = g(a)$
- $\mathbf{Rel}$, $f \leq g$ if $f \subseteq g$ as relations
- $\mathbf{Stoch}_{\leq}$, $f \leq g$ if $\forall a \in A \; \forall b \in B \quad f(b|a) \leq g(b|a)$
- $\mathbf{BorelStoch}_{\leq}$, $f \leq g$ if $\forall a \in A \; \forall E \in \sigma_B \quad f(E|a) \leq g(E|a)$

PROGRAM TRIPLES

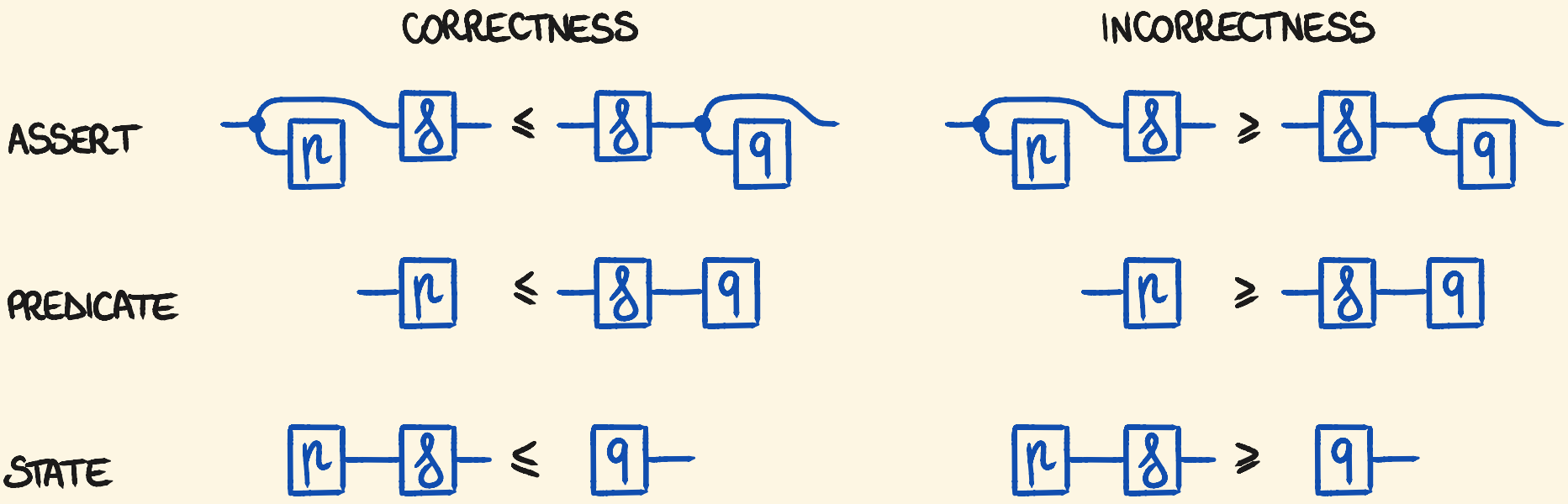
Statements of the form $\{p\}\ S\ \{q\}$.

program

precondition

postcondition

When is a triple valid?



HOARE LOGIC (ASSERT-CORRECTNESS) [Hoare (1969)]

$$\frac{}{\{p\}\ \mathrm{skip}\ \{p\}} \qquad \frac{\{p\}\ f\ \{q\} \quad \{q\}\ g\ \{r\}}{\{p\}\ f;g\ \{r\}} \qquad \frac{p \leqslant p' \quad \{p'\}\ f\ \{q'\} \quad q' \leqslant q}{\{p\}\ f\ \{q\}} \qquad \frac{\{p_1\}\ f\ \{q_1\} \quad \{p_2\}\ f\ \{q_2\}}{\{p_1 \wedge p_2\}\ f\ \{q_1 \wedge q_2\}}$$

$$\frac{}{\{p\}\ \mathrm{abort}\ \{q\}} \qquad \frac{}{\{p(x_1,\dots,x_{i-1},e,x_{i+1},\dots,x_N)\}\ x_i \leftarrow e\ \{p\}} \qquad \frac{}{\{q\}\ \mathrm{assert}\ p\ \{p \wedge q\}} \qquad \frac{q \wedge r \leqslant \bot}{\{r +_b q\}\ \mathrm{assert}\ r\ \{p \wedge \overline{b}\}}$$

$$\frac{}{\{\bot\}\ f\ \{q\}} \qquad \frac{}{\{p\}\ f\ \{\top\}} \qquad \frac{\{p\}\ f\ \{q\} \quad \{p\}\ g\ \{q\}}{\{p\}\ \mathrm{if}\ b\ \mathrm{then}\ f\ \mathrm{else}\ g\ \{q\}} \qquad \frac{\{p \wedge b\}\ f\ \{q\} \quad \{p \wedge \neg b\}\ g\ \{q\} \quad b\ \mathrm{det}}{\{p\}\ \mathrm{if}\ b\ \mathrm{then}\ f\ \mathrm{else}\ g\ \{q\}}$$

$$\frac{\{p\}\ \mathrm{if}\ b\ \mathrm{then}\ (f;\mathrm{while}\ b\ \mathrm{do}\ f)\ \mathrm{else}\ \mathrm{skip}\ \{q\}}{\{p\}\ \mathrm{while}\ b\ \mathrm{do}\ f\ \{q\}} \qquad \frac{\{p\}\ f\ \{p\}}{\{p\}\ \mathrm{while}\ b\ \mathrm{do}\ f\ \{p\}} \qquad \frac{\{p \wedge b\}\ f\ \{p\} \quad b\ \mathrm{det}}{\{p\}\ \mathrm{while}\ b\ \mathrm{do}\ f\ \{p \wedge \neg b\}}$$

THEOREM

The rules of assert-correctness program logics are valid in any posetal imperative category where, for all morphisms f and all predicates p, $\mathrm{abort} \leqslant f$ and $p \leqslant \top$.

INCORRECTNESS LOGIC (STATE-INCORRECTNESS)

[de Vries, Koutavas (2011), O'Hearn (2019)]

$$\frac{}{\{s\}\ \mathsf{skip}\ \{s\}} \qquad \frac{\{s\}\ f\ \{t\} \quad \{t\}\ g\ \{u\}}{\{s\}\ f;g\ \{u\}} \qquad \frac{\{s\}\ f\ \{\bot\}}{\{s\}\ f;g\ \{\bot\}} \qquad \frac{s \geqslant s' \quad \{s'\}\ f\ \{t'\} \quad t' \geqslant t}{\{s\}\ f\ \{t\}} \qquad \frac{\{s_1\}\ f\ \{t_1\} \quad \{s_2\}\ f\ \{t_2\} \quad b\ \text{const}}{\{s_1 +_b s_2\}\ f\ \{t_1 +_b t_2\}}$$

$$\frac{}{\{s\}\ x_i \leftarrow e\ \{s[x_i \backslash e]\}} \qquad \frac{}{\{s\}\ (x_{i_1}, \ldots, x_{i_m}) \xleftarrow{\$} s_0\ \{\coprod_{x_i} s \cdot s_0\}} \qquad \frac{}{\{s\}\ \mathsf{abort}\ \{\bot\}} \qquad \frac{}{\{s\}\ f\ \{\bot\}} \qquad \frac{}{\{s\}\ \mathsf{assert}\ p\ \{s \downarrow p\}}$$

$$\frac{\{s \downarrow b\}\ f\ \{t\}}{\{s\}\ \mathsf{if}\ b\ \mathsf{then}\ f\ \mathsf{else}\ g\ \{t\}} \qquad \frac{\{s \downarrow \neg b\}\ g\ \{t\}}{\{s\}\ \mathsf{if}\ b\ \mathsf{then}\ f\ \mathsf{else}\ g\ \{t\}} \qquad \frac{\{s \downarrow b\}\ f; \mathsf{while}\ b\ \mathsf{do}\ f\ \{t\}}{\{s\}\ \mathsf{while}\ b\ \mathsf{do}\ f\ \{t\}} \qquad \frac{}{\{s\}\ \mathsf{while}\ b\ \mathsf{do}\ f\ \{s \downarrow \neg b\}}$$

THEOREM

The rules of state-incorrectness program logics are valid in any positive imperative category where, for all morphisms f, $\mathsf{abort} \leqslant f$.

OUTCOME LOGIC (PREDICATE-CORRECTNESS)

[Zilberstein, Dreyer, Silva (2023)]

$$\frac{}{\{p\}\ \mathsf{skip}\ \{p\}} \qquad \frac{\{p\}\,f\,\{q\} \quad \{q\}\,g\,\{r\}}{\{p\}\,f;g\,\{r\}} \qquad \frac{p \leqslant p' \quad \{p'\}\,f\,\{q'\} \quad q' \leqslant q}{\{p\}\,f\,\{q\}} \qquad \frac{\{p_1\}\,f\,\{q_1\} \quad \{p_2\}\,f\,\{q_2\} \quad b\ \text{const}}{\{p_1 +_b p_2\}\,f\,\{q_1 +_b q_2\}} \qquad \frac{\neg b \wedge q \leqslant \bot \quad b\ \text{det}}{\{p +_b q\}\ \mathsf{assert}\ b\ \{p\}}$$

$$\frac{}{\{p(x_1,\ldots,x_{i-1},e,x_{i+1},\ldots,x_N)\}\ x_i \leftarrow e\ \{p\}} \qquad \frac{}{\{p(x_i,\ldots,x_{i+m} \setminus s)\}\ x_i,\ldots,x_{i+m} \xleftarrow{\$} s\ \{p\}}$$

$$\frac{\{p\}\ \mathsf{if}\ b\ \mathsf{then}\ (f;\ \mathsf{while}\ b\ \mathsf{do}\ f)\ \mathsf{else}\ \mathsf{skip}\ \{q\}}{\{p\}\ \mathsf{while}\ b\ \mathsf{do}\ f\ \{q\}}$$

$$\frac{\{p \wedge b\}\,f\,\{q\} \quad \{p \wedge \neg b\}\,g\,\{q\} \quad b\ \text{det}}{\{p\}\ \mathsf{if}\ b\ \mathsf{then}\ f\ \mathsf{else}\ g\ \{q\}} \qquad \frac{\{p\}\,f\,\{q\} \quad \{p\}\,g\,\{q\}}{\{p\}\ \mathsf{if}\ b\ \mathsf{then}\ f\ \mathsf{else}\ g\ \{q\}} \qquad \frac{}{\{\bot\}\,f\,\{q\}} \qquad \frac{}{\{p \wedge q\}\ \mathsf{assert}\ p\ \{q\}}$$

THEOREM

The rules of predicate-correctness program logics are valid in any posetal imperative category where, for all morphisms f, $\mathsf{abort} \leqslant f$.

RELATIONAL PROGRAM TRIPLES

Statements of the form $\{p\}\ f \sim g\ \{q\}$.

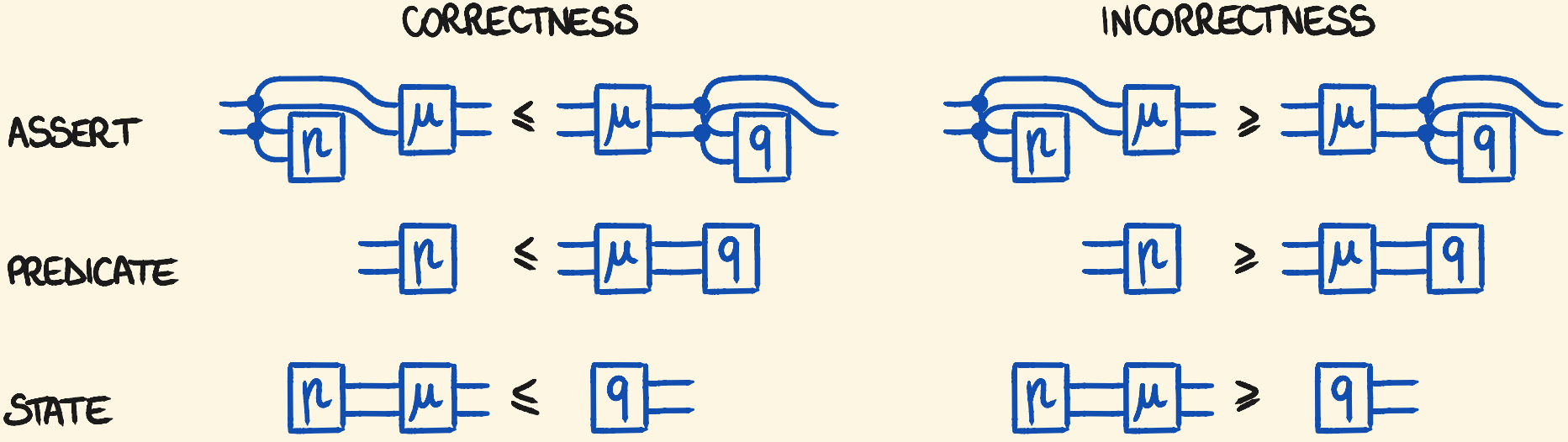
programs

precondition

postcondition

When is a triple valid?

There is a coupling $\mu \triangleright f \& g$ such that



[Benton (2004)] [Avanzini, Barthe, Davoli, Lyrégoire (2025)]

COUPLINGS

A strict coupling μ of $f : A \to A$ and $g : B \to B$ is $\mu : A \otimes B \to A \otimes B$ st

$\rightsquigarrow$ this forces f and g to have the same termination behaviour

A coupling μ of $f : A \to A$ and $g : B \to B$ is $\mu : A \otimes B \to A \otimes B + A + B$ st

RELATIONAL HOARE LOGIC (ASSERT-CORRECTNESS)

$$\frac{}{\{p\}\ \mathsf{skip} \sim \mathsf{skip}\ \{p\}} \qquad \frac{\{p\}\ c_1 \sim d_1\ \{q\} \quad \{q\}\ c_2 \sim d_2\ \{r\}}{\{p\}\ c_1 c_2 \sim d_1 d_2\ \{r\}} \qquad \frac{p \leqslant p' \quad \{p'\}\ c \sim d\ \{q'\} \quad q' \leqslant q}{\{p\}\ c \sim d\ \{q\}} \qquad \frac{\{p\}\ c \sim d\ \{q\}}{\{\sigma p\}\ c \sim d\ \{\sigma q\}}$$

$$\frac{}{\{p[(u_1,u_2)\backslash(e_1,e_2)]\}\ u_1 \leftarrow e_1 \sim u_2 \leftarrow e_2\ \{p\}} \qquad \frac{\{p\}\ c_1 \sim c_2\ \{q\} \quad \{p\}\ c_1 \sim d_2\ \{q\} \quad \{p\}\ d_1 \sim c_2\ \{q\} \quad \{p\}\ d_1 \sim d_2\ \{q\}}{\{p\}\ \mathsf{if}\ b_1\ \mathsf{then}\ c_1\ \mathsf{else}\ d_1 \sim \mathsf{if}\ b_2\ \mathsf{then}\ c_2\ \mathsf{else}\ d_2\ \{q\}}$$

$$\frac{\{(\check{b}_1 \otimes \check{b}_2) \wedge p\}\ c_1 \sim c_2\ \{q\} \quad \{(\neg\check{b}_1 \otimes \neg\check{b}_2) \wedge p\}\ d_1 \sim d_2\ \{q\} \quad b_1, b_2\ \mathsf{det}}{\{(b_1 = b_2) \wedge p\}\ \mathsf{if}\ b_1\ \mathsf{then}\ c_1\ \mathsf{else}\ d_1 \sim \mathsf{if}\ b_2\ \mathsf{then}\ c_2\ \mathsf{else}\ d_2\ \{q\}}$$

$$\frac{\{(\check{b}_1 \otimes \check{b}_2) \wedge p\}\ c_1 \sim c_2\ \{(b_1 = b_2) \wedge p\} \quad b_1, b_2\ \mathsf{det}}{\{(b_1 = b_2) \wedge p\}\ \mathsf{while}\ b_1\ \mathsf{do}\ c_1 \sim \mathsf{while}\ b_2\ \mathsf{do}\ c_2\ \{(\neg\check{b}_1 \otimes \neg\check{b}_2) \wedge p\}} \qquad \frac{\{p\}\ c_1 \sim c_2\ \{p\} \quad \{p\}\ c_1 \sim \mathsf{skip}\ \{p\} \quad \{p\}\ \mathsf{skip} \sim c_2\ \{p\}}{\{p\}\ \mathsf{while}\ b_1\ \mathsf{do}\ c_1 \sim \mathsf{while}\ b_2\ \mathsf{do}\ c_2\ \{p\}}$$

+ one-sided rules

THEOREM

The rules of relational assert-correctness program logics are valid in any posetal imperative category where, for all morphisms f, $\mathsf{abort} \leqslant f$.

SUMMARY

- Imperative categories are a good semantic universe for imperative programs (with effects).
- Poretal imperative categories derive program logics, existing and new
- Sound and complete syntax (in the paper)

FUTURE WORK

- Separation logic & premonoidality
- Predicate fibrations for richer predicates
- Relative completeness