# Categorical Computer Science

## Semantics of computational effects

Jean-Guillaume Dumas, Dominique Duval, Laurent Fousse, Jean-Claude Reynaud.

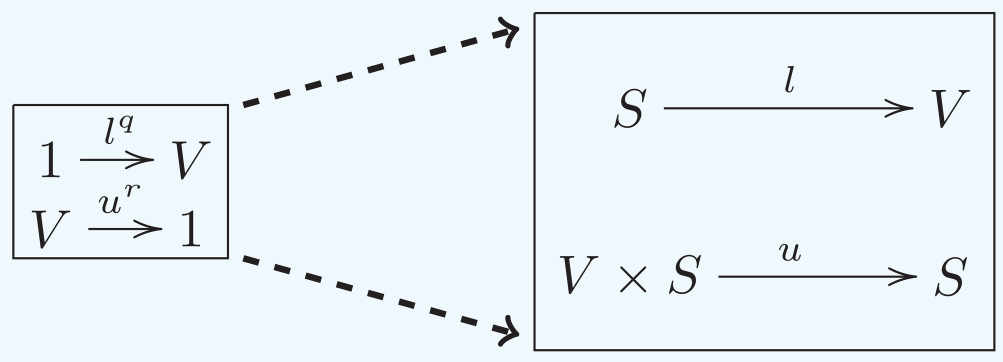
### Zooms for effects

A zooming process looks at the **syntax**, where the effect is partially hidden, and expands it, so that the effect becomes explicit. The **semantics** of the effect can be obtained from either view, thanks to a categorical **adjunction**. We have defined zooms from **morphisms** of diagrammatic logics (see below).

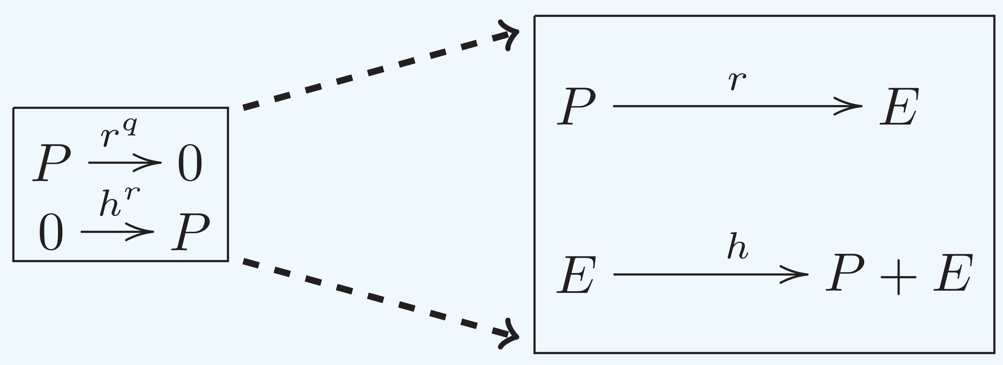

### Two dual effects

We have discovered a **duality** between two fundamental effects.

The **STATE** effect: **comonad** $T(X) = X \times S$ where $V$ = the type of values, $S$ = the type of states, $l$ = **lookup**, $u$ = **update**.




The **EXCEPTION** effect: **monad** $T(X) = X + E$ where $P$ = the type of parameters, $E$ = the type of exceptions, $r$ = **raise**, $h$ = **handle**.

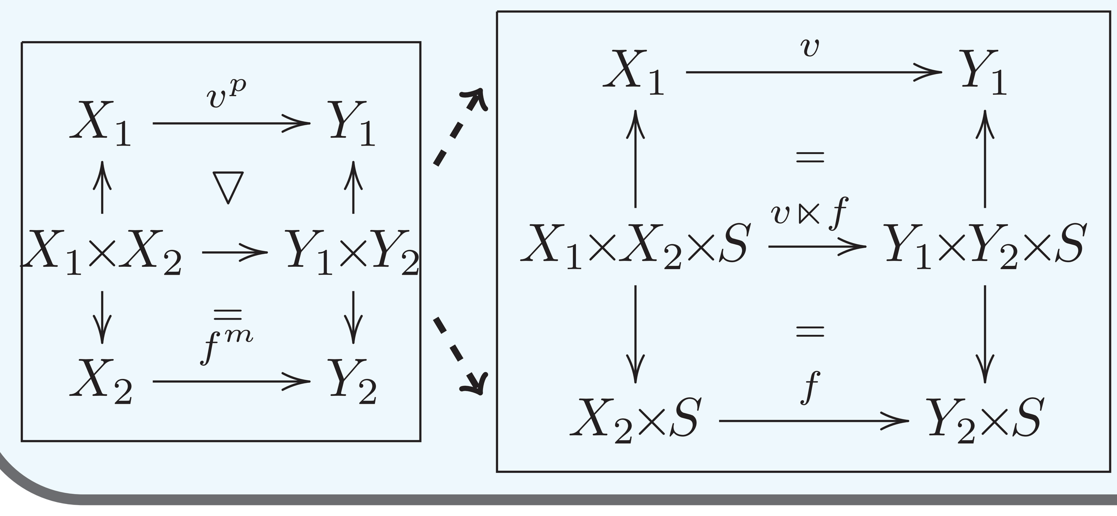



### Sequential product

Without effects, the order of evaluation of the arguments of a multivariate function does not matter. This is seen as a categorical **product**.

$$\begin{tikzcd}
X_1 \arrow[r, "v_1"] & Y_1 \\
X_1 \times X_2 \arrow[u] \arrow[d] \arrow[r, "v_1 \times v_2"] \arrow[ur, phantom, "="] \arrow[dr, phantom, "="] & Y_1 \times Y_2 \arrow[u] \arrow[d] \\
X_2 \arrow[r, "v_2"] & Y_2
\end{tikzcd}$$

When there are effects, the result depends on the order of evaluation of the arguments. We have defined **sequential products** for this purpose. For instance for states:




## Diagrammatic logic

César Domínguez, Dominique Duval.

A diagrammatic logic is defined as an **adjunction**

$$\begin{tikzcd}
\mathbf{S} \arrow[r, bend left, "L"] \arrow[r, phantom, "\perp"] & \mathbf{T} \arrow[l, bend left, "R"]
\end{tikzcd}$$

where (equivalently):

- $R$ is full and faithful
- $L$ is a localizer

- **S** is the category of **specifications**
- **T** is the category of **theories**

Example:

- **S** is the category of **equational specifications**
- **T** is the category of **cartesian categories**

The **models** of a theory $\Theta$ are the functors from $\Theta$

The **proofs** tend to build $L(\Sigma)$ from a specification $\Sigma$

Proofs are morphisms in a **bicategory** of fractions

<table>
<tr><td>An inference rule written $\frac{\mathcal{H}}{\mathcal{C}}$<br>is a categorical fraction $\frac{\mathcal{C}}{\mathcal{H}}$!</td><td>An inference step<br>is a composition of fractions.</td></tr>
<tr><td>$\mathcal{H} \xleftarrow{\rho} \mathcal{C}$    $\mathcal{H} \underset{}{\overset{\tau}{\rightleftarrows}} \mathcal{H}' \xleftarrow{\sigma} \mathcal{C}$</td><td>$\begin{tikzcd} \mathcal{H} \arrow[dr, "\kappa"'] & & \mathcal{C} \arrow[ll, "\rho"'] \arrow[dl, "\kappa\circ\rho"] \\ & \Sigma & \end{tikzcd}$    $\begin{tikzcd} \mathcal{H} \arrow[r, shift left, "\tau"] \arrow[d] & \mathcal{H}' \arrow[l, dashed, shift left] \arrow[d] & \mathcal{C} \arrow[l, "\sigma"'] \arrow[dl] \\ \Sigma_{\mathcal{H}} \arrow[r, shift left] \arrow[dr, dashed, shift left] & \Sigma_{\mathcal{C}} \arrow[l, dashed, shift left] \arrow[d, dashed, shift left] & \\ & \Sigma \arrow[ul, dashed, shift left] \arrow[u, dashed, shift left] & \end{tikzcd}$</td></tr>
</table>

## Graph rewriting

Dominique Duval, Rachid Echahed, Frédéric Prost.

$L, K, R, G, D, H$ are **graphs**, arrows are **graph morphisms**.

<table>
<tr><td>Input:<br>a rule</td><td>and<br>a matching of $L$</td><td>Construction:<br>a (?) and a pushout (PO)</td><td>Output:<br>a matching of $R$</td></tr>
<tr><td>$L \leftarrow K \rightarrow R$</td><td>$\begin{tikzcd} L \arrow[d] \\ G \end{tikzcd}$</td><td>$\begin{tikzcd} L \arrow[d] \arrow[dr, phantom, "(?)"] & K \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "(PO)"] & R \arrow[d] \\ G & D \arrow[l] \arrow[r] & H \end{tikzcd}$</td><td>$\begin{tikzcd} R \arrow[d] \\ H \end{tikzcd}$</td></tr>
</table>

where **(?)** can be:

- a pushout complement
- a terminal pullback complement

We have introduced variants of graph rewriting for cloning and deleting nodes and vertices.