Protocol Choice and Iteration for the Free Cornering

Chad Nester[a], Niels Voorneveld[b]

[a] *University of Tartu*
[b] *Cybernetica AS*

Morphisms of monoidal categories often admit interpretation as *processes*, which produce and consume resources according to their type [1]. The *free cornering* of a monoidal category [3, 2] augments it with *corner cells*, resulting in a double category. Processes become *interacting processes*, with the interaction governed by a simple system of protocol types. Specifically, a cell $\alpha : (U\,{}^{A}_{B}\,W)$ of the resulting double category admits interpretation as a process that consumes an instance of A and produces an instance of B while participating in the interaction protocols U and W along its left and right boundary, respectively.

Alas, the system of protocol types that arises in this way is too simple to form the basis of a serious programming language: it lacks both protocol choice and the ability to iterate protocols. In recent work [4] we extend the free cornering to support protocol choice and iteration. This requires more of the base category, which must now be a distributive monoidal category. Here the coproducts are understood to support a sort of case statement (see e.g., [5]). The extended construction again results in a double category of interacting processes.

We sketch the construction here. Given a distributive monoidal category $\mathbb{A}$ the *free cornering with choice and iteration* of $\mathbb{A}$ contains:

- For each $f : A \to B$ of $\mathbb{A}$ a cell $\ulcorner f \urcorner : (I\,{}^{A}_{B}\,I)$ subject to equations:

  $$\ulcorner fg \urcorner = \frac{\ulcorner f \urcorner}{\ulcorner g \urcorner} \qquad \ulcorner 1_A \urcorner = 1_A \qquad \ulcorner f \otimes g \urcorner = \ulcorner f \urcorner \mid \ulcorner g \urcorner$$

  I denotes the empty interaction protocol, and the cells $\ulcorner f \urcorner$ serve to include the processes of the base category into the new setting.

- For each object A of $\mathbb{A}$, *corner cells* $\mathsf{get}_{\mathsf{L}}^{A} : (A^{\circ}\,{}^{I}_{A}\,I)$, $\mathsf{put}_{\mathsf{R}}^{A} : (I\,{}^{A}_{I}\,A^{\circ})$, $\mathsf{get}_{\mathsf{R}}^{A} : (I\,{}^{I}_{A}\,A^{\bullet})$, and $\mathsf{put}_{\mathsf{L}}^{A} : (A^{\bullet}\,{}^{A}_{I}\,I)$, subject to equations:

  $$\frac{\mathsf{get}_{\mathsf{L}}^{A}}{\mathsf{put}_{\mathsf{R}}^{A}} = id_{A^{\circ}} \qquad \mathsf{put}_{\mathsf{R}}^{A} \mid \mathsf{get}_{\mathsf{L}}^{A} = 1_A \qquad \frac{\mathsf{get}_{\mathsf{R}}^{A}}{\mathsf{put}_{\mathsf{L}}^{A}} = id_{A^{\bullet}} \qquad \mathsf{get}_{\mathsf{R}}^{A} \mid \mathsf{put}_{\mathsf{L}}^{A} = 1_A$$

  A° ($A^{\bullet}$) is the interaction protocols in which the left (right) participant sends the right (left) participant an instance of A. The corner cells allow processes to send and receive things along their left/right boundaries.

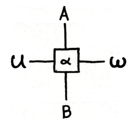
- For each pair U, W of protocols, horizontal projections $\pi_0 : (U{\times}W\,{}^{I}_{I}\,U)$ and $\pi_1 : (U{\times}W\,{}^{I}_{I}\,W)$ and horizontal injections $\perp\!\!\!\perp_0 : (U\,{}^{I}_{I}\,U{+}W)$ and $\perp\!\!\!\perp_1 : (W\,{}^{I}_{I}\,U{+}W)$. Moreover, for each pair of cells $\alpha \in (V\,{}^{A}_{B}\,U)$ and $\beta : (V\,{}^{A}_{B}\,W)$ a unique cell $\alpha \times \beta : (V\,{}^{A}_{B}\,U{\times}W)$ such that $(\alpha \times \beta) \mid \pi_0 = \alpha$ and $(\alpha \times \beta) \mid \pi_1 = \beta$. Dually, for each pair of cells $\alpha : (U\,{}^{A}_{B}\,V)$ and $\beta : (W\,{}^{A}_{B}\,V)$ a unique cell $\alpha + \beta : (U{+}W\,{}^{A}_{B}\,V)$ such that $\perp\!\!\!\perp_0(\alpha + \beta) = \alpha$ and $\perp\!\!\!\perp_1(\alpha + \beta) = \beta$.

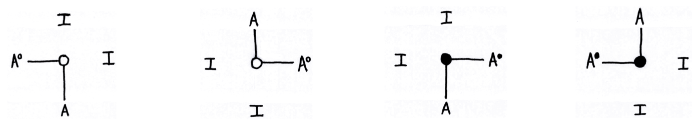
$U + W$ ($U \times W$) is the protocol in which the left (right) participant chooses which of U and W will be carried out. The injections (projections) allow the left (right) participant to make this choice, and the cells $\alpha + \beta$ ($\alpha \times \beta$) specify the response of the right (left) participant in each case.

- For any cells $\alpha : (V\,{}^{A}_{A}\,U)$, $f : (W\,{}^{A}_{B}\,K)$, $g : (W\,{}^{I}_{I}\,V{\otimes}W)$ a unique cell $\alpha^{\times}_{f,g} : (W\,{}^{A}_{B}\,U^{\times}{\otimes}K)$ such that $\alpha^{\times}_{f,g} \mid \frac{\pi_0}{id_K} = f$ and $\alpha^{\times} \mid \frac{\pi_1}{id_K} = g \mid \frac{\alpha}{\alpha^{\times}_{f,g}}$. Dually, for any cells $\alpha : (U\,{}^{A}_{A}\,V)$, $f : (K\,{}^{A}_{B}\,W)$, and $g : (V{\otimes}W\,{}^{I}_{I}\,W)$ a unique cell $\alpha^{+}_{f,g} : (U^{+}{\otimes}K\,{}^{A}_{B}\,W)$ such that $\frac{ip_0}{id_K} \mid \alpha^{+}_{f,g} = f$ and $\frac{\amalg_1}{id_K} \mid \alpha^{+}_{f,g} = \frac{\alpha}{\alpha^{+}_{f,g}} \mid g$.

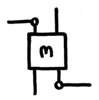
  U^{+} ($U^{\times}$) is the protocol in which the left (right) participant chooses to either end the protocol, or to perform U once before being offered the choice again. We ask that $U^{+} = I + (U \otimes U^{+})$ ($U^{\times} = I \times (U \otimes U^{\times})$) so that this choice can be made using the injections (projections) that are already present.

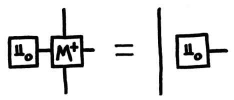
It is helpful to depict cells of the free cornering as string diagrams, with $\alpha : (U\,{}^{A}_{B}\,W)$ depicted as below left, and the corner cells depicted as below right:

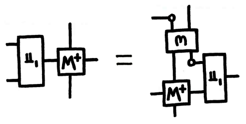




For example, a *mealy machine* in a monoidal category $\mathbb{A}$ is a morphism $m : A \otimes S \to S \otimes B$. Taking $\mathbb{A}$ to be finite sets recovers the classical notion. Mealy machines are usually understood to operate on a sequence of inputs drawn from A, producing a sequence of outputs drawn from B. The state S of the machine is fed forward to future iterations. Let $M : (A^{\circ}\,{}^{S}_{S}\,B^{\circ})$ be the cell below left. Then the cell $M^{+} : ((A^{\circ})^{+}\,{}^{S}_{S}\,(B^{\circ})^{+})$ exhibits the behaviour of the process that the Mealy machine m is intended to define, as below right.







that is, if the there is no more input then the machine produces no more output, and if there is further input then the machine produces output according to m and updates its internal state.

The resulting double category has a number of encouraging properties. For example, It admits the structure of a monoidal double category. The cells supporting protocol choice give products/coproducts in its category of horizontal cells, and the two iteration operations define a monad/comonad on the horizontal cells. Iteration admits a sensible coinductive reasoning principle.

A promising direction for future work is to apply the extended free cornering construction to a suitable base category of computable functions, with the result being a basic model of *interactive* computation. A related question concerns the operational semantics of the extended free cornering, and how one might best make develop a programming language using these ideas.

References

[1] B. Coecke, T. Fritz, and R.W. Spekkens. A Mathematical Theory of Resources. *Information and Computation*, 250:59–86, 2016.

[2] C. Nester. The Structure of Concurrent Process Histories. In *International Conference on Coordination Languages and Models*, pages 209–224. Springer, 2021.

[3] C. Nester. Concurrent Process Histories and Resource Transducers. *Logical Methods in Computer Science*, Volume 19, Issue 1, January 2023.

[4] C. Nester and N. Voorneveld. Protocol choice and iteration for the free cornering. *Journal of Logical and Algebraic Methods in Programming*, 137:100942, 2024.

[5] Robert F. C. Walters. An imperative language based on distributive categories. *Mathematical Structures in Computer Science*, 2(3):249–256, 1992.