Polar Interleavings for Deadlock-Free Message-Passing

Extended Abstract

Matt Earnshaw, Chad Nester, and Mario Román

Tallinn University of Technology

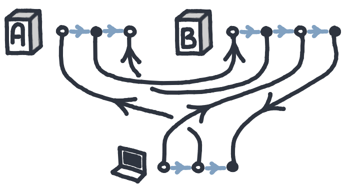
Abstract

We introduce *message theories*, a notion of asynchronous multi-party message passing. We show that the operations of a message theory correspond to *polar interleavings*, a certain kind of directed acyclic graph.

Message Protocols. In modern computer systems it is common to describe interaction in terms of message passing protcols, in which the different parties send and receive a series of messages amongst themselves to achieve the task at hand.

For example, suppose we wish to connect to a server B, but must do so via an authentication server A. From our perspective, *(i)* we send a connection request to the server B, *(ii)* we send an authentication request to the server A and *(iii)* wait for the server B to acknowledge our request. The server B is not allowed to receive messages from unauthenticated machines; it must wait for the authentication server A to confirm our identity before handling our request. That is, the server B *(i)* waits to receive authentication details from A, *(ii)* sends a message to A acknowledging this, *(iii)* receives our connection request and finally *(iv)* answers us. For its part, the authentication server A *(i)* receives our authentication request, *(ii)* sends our authentication details to B and *(iii)* receives acknowledgement from B.

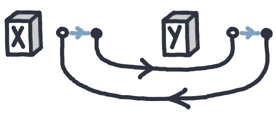
We can depict this chain of events graphically as in Figure 1 on the left: the two different polarities distinguish the sending (•) and receiving (∘) of messages. Alternatively, we might use a more familiar session-like syntax as in Figure 1 on the right, in which the two prefix operators distinguish the sending (?) and receiving (!) of messages.




```
system(?request,?login,!send) {
  serverA(?login,!ok,?ack),
  serverB(?ok,!ack,?request,!send)
}
```

Figure 1: Communication Diagram.

Nothing ensures that a multi-party protocol like this one is possible to carry out. For example, if two parties are each waiting for the other to send a message before the protocol can proceed, they are in *deadlock*, as in Figure 2. Here, server X waits for a message from server Y before doing anything else, while server Y waits for a message from server X. We would like to avoid this situation in our protocols.




```
deadlock() {
  serverX(?y,?x),
  serverY(?x,!y)
}
```

Figure 2: Deadlocked Communication Diagram.

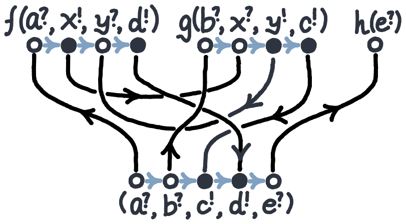
Polar Interleaving. How can we avoid protocols that result in a deadlock? Prohibiting swaps is suitable but it is too strong: our first example contained swapping of wires but it did not result in deadlock. A better answer is that the directed graph corresponding to our protocol must be *acyclic* [CD10], in which case no deadlock can occur. Therefore, our first step towards a logic of deadlock-free protocols will be to encode communication graphs into a combinatorial structure – *polar interleavings* – which are acyclic. Next, we introduce the logic of *message theories* and prove that derivations in this logic correspond bijectively to polar interleavings, thereby establishing that deadlock-free sessions are precisely those described by message theories.

Definition 1. A *polar list* X is a list labelled over the two element set $p\colon X \to \{\circ, \bullet\}$; we call $X^\circ = p^{-1}(\circ)$ and $X^\bullet = p^{-1}(\bullet)$ to their $(\circ)$ and $(\bullet)$-polarized sublists, respectively. A *polar interleaving* from a multiset of polar lists $X_1, \dots, X_n$ to a single polar list Y, is a bijection

$$f\colon X_1^\bullet + \dots + X_n^\bullet + Y^\circ \to Y^\bullet + X_1^\circ + \dots + X_n^\circ$$

such that the directed graph containing all polar lists (as linear posets) and an edge $x \to f(x)$ for each element in $X_1^\bullet + \dots + X_n^\bullet + Y^\circ$, is acyclic.

For instance, a polar shuffle of shape $\mathtt{pShuf}(\circ\bullet\circ\bullet, \circ\circ\bullet\bullet, \circ; \circ\circ\bullet\bullet\circ)$, is given by the following acyclic graph in Figure 3. In black, we depict the edges that come from the graph of a function. In blue, the edges that come from the linear finite posets.




```
main(a?,b?,c!,d!,e?) {
  f(a?,x!,y?,d!),
  g(b,x?,y!,c!),
  h(e?)
}
```

Figure 3: Polar shuffle and its encoding.

Encoding Polar Shuffles. Polar shuffles are ultimately graphs, and they can be encoded as such. We propose a notation suggestive of multiparty session calculi. The *session encoding* of polar shuffles assigns a variable name to each polar list (say, $f, g, h, \dots$) and to each edge of the graph outside a polar list (say, $a, b, c, x, y, z, \dots$). The encoding of a polar shuffle starts by declaring the list of edges incident to the output polar list, together with their polarization. Then, enclosed in braces, we write the polar lists and the edges that incide on them. For instance, the encoding of the polar shuffle of Figure 3 is on the right side.

The implication of this encoding is that, if we label the vertices of a polar shuffle with types, there exists at most one polar shuffle with any distinctly typed sources and targets.

Lemma 2. *Polar shuffles are coherent — there exists at most a single polar shuffle between distinctly typed polar lists: we say that a polar list is distinctly typed if each variable (each type) appears in the premises and the conclusion exactly twice, each time with a different variance.*

Theorem 3. *Polar interleavings form the free polarized physical monoidal multicategory over an element; typed polar shuffles form the free polarized physical monoidal multicategory over a set of types.*

Remark 4. Parsing polar shuffles requires checking whether the graph is acyclic. Checking acyclicity can be done in linear time in the sum of vertices and edges; this means that checking if a polar shuffle is valid is linear in its total length, $\mathcal{O}(\#X_1 + \dots + \#X_n + \#Y)$.

Message Theories, a Logic of Deadlock-Free Sessions

Message passing requires the interplay of at least two mathematical structures: the ability to *interleave* events in time and the ability to connect a *sender and a receiver.* Let us propose a minimally axiomatized algebra of interleaving and sending/receiving: interleaving will correspond to a normal duoidal algebra, and sending/receiving will correspond to polarization.

Definition 5. A *message theory* $\mathbb{M}$ consists of a set of types, $\mathbb{M}_{obj}$ with extra structure: a *send/receive session type* is a polarized list of types; for each session type, we have a collection of *sessions* with that type,

$$\mathbb{M}(X_1^{\bullet_1}, \ldots, X_n^{\bullet_n}), \text{ for each } X_1, \ldots, X_n \in \mathbb{M}_{obj}, \text{ and each polarization } \bullet_i \in \{\circ, \bullet\}.$$

A message theory must contain operations for *(i)* binary shuffling, *(ii)* linking a sent message to immediately receive it, *(iii)* spawning a channel that receives a message and sends it immediately, and *(iv)* and nullary shuffling. Message theories may be better understood in the notation of a logic, as in Figure 4. Types form a free polarized monoid; each term describes a possible communication protocol.

$$\frac{\Gamma \qquad \Delta}{[\Gamma, \Delta]_\sigma}\ (\textsc{shf}_\sigma) \qquad \frac{\Gamma, X^\bullet, X^\circ, \Delta}{\Gamma, \Delta}\ (\textsc{lnk}) \qquad \frac{\Gamma, \Delta}{\Gamma, X^\circ, X^\bullet, \Delta}\ (\textsc{spw}) \qquad \frac{}{\varepsilon}\ (\textsc{nop})$$

Figure 4: Type-theoretic presentation of a message theory.

A message theory must satisfy the axioms of a physical monoidal multicategory [AM10, SS22, Rom23] (for convenience, we include them in Appendix A); they will be exactly an algebra for polar shuffles as a consequence of Theorem 3. Under these axioms, derivations in a message theory are precisely polar shuffles.

Theorem 6. *Message theories over a set of types are equivalently the algebras of the physical monoidal multicategory of polar shuffles over that same set of types.*

Related Work

Honda pioneered *binary session types* in the 90s [Hon93], and further work with Yoshida and Carbone extended them to the multi-party case [HYC08]. Session types [Hon93, HYC08] are the mainstay type formalism for communication protocols, and they have been extensively applied to the π-calculus [SW01].

Lambek [Lam69] first introduced multicategories as the underlying structure that unified Gentzen's sequents and multilinear maps. Hermida's work [Her00] explains that tensors are universal because they represent a relevant multimap structure, a *multicategory.* For instance, the monoidal category of vector spaces with their tensor product represents functions linear in each variable: a linear function $A \otimes B \to C$ is the same as a multilinear function $A, B \to C$.

Nester [Nes21] notices the importance of polarization for message passing via cornerings: single-object proarrow equipments. Further work with Voorneveld [NV23] added choice and iteration, and further work with Boisseau and Román [BNR22] compared cornerings with optics. We now extend this last piece of work: we notice for the first time that the normal monoidal multicategory structure of lenses, studied by Earnshaw, Hefford and Román [EHR23], can be used to characterize message passing universally. Message passing may not only use the structure of double categories but also of monoidal multicategories.

Results for the present work are part of the thesis of the last-named author [Rom23].

Acknowledgements This research was supported by the ESF funded Estonian IT Academy research measure (project 2014-2020.4.05.19-0001) and Estonian Research Council grant PRG-1210.

References

[AM10] Marcelo Aguiar and Swapneel A. Mahajan. *Monoidal functors, species and Hopf algebras*, volume 29. American Mathematical Society Providence, RI, 2010.

[BNR22] Guillaume Boisseau, Chad Nester, and Mario Román. Cornering optics. In *ACT 2022*, volume abs/2205.00842, 2022.

[CD10] Marco Carbone and Søren Debois. A graphical approach to progress for structured communication in web services. In Simon Bliudze, Roberto Bruni, Davide Grohmann, and Alexandra Silva, editors, *Proceedings Third Interaction and Concurrency Experience: Guaranteed Interaction, ICE 2010, Amsterdam, The Netherlands, 10th of June 2010*, volume 38 of *EPTCS*, pages 13–27, 2010.

[EHR23] Matt Earnshaw, James Hefford, and Mario Román. The produoidal algebra of process decomposition. *arXiv preprint arXiv:2301.11867*, 2023.

[Her00] Claudio Hermida. Representable multicategories. *Advances in Mathematics*, 151(2):164–225, 2000.

[Hon93] Kohei Honda. Types for dyadic interaction. In Eike Best, editor, *CONCUR '93, 4th International Conference on Concurrency Theory, Hildesheim, Germany, August 23-26, 1993, Proceedings*, volume 715 of *Lecture Notes in Computer Science*, pages 509–523. Springer, 1993.

[HYC08] Kohei Honda, Nobuko Yoshida, and Marco Carbone. Multiparty asynchronous session types. In George C. Necula and Philip Wadler, editors, *Proceedings of the 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2008, San Francisco, California, USA, January 7-12, 2008*, pages 273–284. ACM, 2008.

[Lam69] Joachim Lambek. Deductive systems and categories II: standard constructions and closed categories. *Category Theory, Homology Theory and their Applications I*, 1969.

[Nes21] Chad Nester. The structure of concurrent process histories. In Ferruccio Damiani and Ornela Dardha, editors, *Coordination Models and Languages - 23rd IFIP WG 6.1 International Conference, COORDINATION 2021, Held as Part of the 16th International Federated Conference on Distributed Computing Techniques, DisCoTec 2021, Valletta, Malta, June 14-18, 2021, Proceedings*, volume 12717 of *Lecture Notes in Computer Science*, pages 209–224. Springer, 2021.

[NV23] Chad Nester and Niels Voorneveld. Protocol choice and iteration for the free cornering. *arXiv preprint arXiv:2305.16899*, 2023.

[Rom23] Mario Román. Monoidal Context Theory. PhD Thesis. *Tallinn University of Technology, available at https://mroman42.github.io/notes/papers/monoidal-context-theory.pdf*, 2023.

[SS22] Brandon T. Shapiro and David I. Spivak. Duoidal structures for compositional dependence. *arXiv preprint arXiv:2210.01962*, 2022.

[SW01] Davide Sangiorgi and David Walker. *The Pi-Calculus - a theory of mobile processes.* Cambridge University Press, 2001.

A Appendix

Axioms of a Message Theory. A message theory must satisfy the following axioms: *(i)* shuffles compose as a symmetric multicategory, where we write $(\sigma \mathbin{⨟}_1 \tau)$ to be the same by associativity as $(\tau' \mathbin{⨟}_2 \sigma')$, we write $(*)$ for the trivial shuffle, and we write $\tilde{\sigma}$ for the symmetric counterpart of σ; *(ii)* linking is natural with respect to the shuffles; *(iii)* spawning is natural with respect to the shuffles; *(iv)* linking is dual to spawning; and *(v)* independent linkings and spawnings commute.

(1a) $\textsc{shf}_\tau(\textsc{shf}_\sigma(m_1 m_2), m_3) = \textsc{shf}_{\sigma'}(m_1, \textsc{shf}_{\tau'}(m_2, m_3))$;
(1b) $\textsc{shf}_*(m, \textsc{nop}) = m$;
(1c) $\textsc{shf}_\sigma(m_1, m_2) = \textsc{shf}_{\tilde{\sigma}}(m_2, m_1)$;
(2a) $\textsc{shf}_{\sigma,\tau}(\textsc{lnk}_x^{\Gamma_1,\Gamma_2}(m_1), m_2) = \textsc{lnk}_x^{\Gamma_1,\Delta_1;\Gamma_2,\Delta_2}(\textsc{shf}_{\sigma,x,\tau}(m_1, m_2))$;
(2b) $\textsc{shf}_{\sigma,\tau}(m_1, \textsc{lnk}_x^{\Delta_1,\Delta_2}(m_2)) = \textsc{lnk}_x^{\Gamma_1,\Delta_1;\Gamma_2,\Delta_2}(\textsc{shf}_{\sigma,x,\tau}(m_1, m_2))$;
(3a) $\textsc{shf}_{\sigma,\tau}(\textsc{spw}_x^{\Gamma_1,\Gamma_2}(m_1), m_2) = \textsc{spw}_x^{\Gamma_1,\Delta_1;\Gamma_2,\Delta_2}(\textsc{shf}_{\sigma,\tau}(m_1, m_2))$;
(3b) $\textsc{shf}_{\sigma,\tau}(m_1, \textsc{spw}_x^{\Delta_1,\Delta_2}(m_2)) = \textsc{spw}_x^{\Gamma_1,\Delta_1;\Gamma_2,\Delta_2}(\textsc{shf}_{\sigma,\tau}(m_1, m_2))$;
(4a) $\textsc{lnk}_x^{\Gamma,X^\circ;\Delta}(\textsc{spw}_x^{\Gamma;X^\circ,\Delta}(m)) = m$;
(4b) $\textsc{lnk}_x^{\Gamma;X^\bullet,\Delta}(\textsc{spw}_x^{\Gamma,X^\bullet;\Delta}(m)) = m$;
(5a) $\textsc{lnk}_x^{\Gamma_1;\Gamma_2 Y \Gamma_3}(\textsc{spw}_y^{\Gamma_1 X \Gamma_2;\Gamma_3}(m)) = \textsc{spw}_y^{\Gamma_1\Gamma_2;\Gamma_3}(\textsc{lnk}_x^{\Gamma_1;\Gamma_2\Gamma_3}(m))$;
(5b) $\textsc{lnk}_y^{\Gamma_1 X \Gamma_2;\Gamma_3}(\textsc{spw}_x^{\Gamma_1;\Gamma_2 Y \Gamma_3}(m)) = \textsc{spw}_x^{\Gamma_1;\Gamma_2\Gamma_3}(\textsc{lnk}_y^{\Gamma_1 X \Gamma_2;\Gamma_3}(m))$;
(5c) $\textsc{spw}_x^{\Gamma_1;\Gamma_2 Y \Gamma_3}(\textsc{spw}_y^{\Gamma_1\Gamma_2;\Gamma_3}(m)) = \textsc{spw}_y^{\Gamma_1 X \Gamma_2;\Gamma_3}(\textsc{spw}_x^{\Gamma_1;\Gamma_2\Gamma_3}(m))$;
(5d) $\textsc{lnk}_x^{\Gamma_1;\Gamma_2\Gamma_3}(\textsc{lnk}_y^{\Gamma_1 X \Gamma_2;\Gamma_3}(m)) = \textsc{lnk}_y^{\Gamma_1\Gamma_2;\Gamma_3}(\textsc{lnk}_x^{\Gamma_1;\Gamma_2 Y \Gamma_3}(m))$.