Towards a Logic of Multi-Party Sessions

Marco Carbone
IT University of Copenhagen
Copenhagen
Denmark
maca@itu.dk

Fabrizio Montesi
University of Southern Denmark
Odense
Denmark
famontesi@gmail.com

Carsten Schürmann
IT University of Copenhagen
Copenhagen
Denmark
carsten@itu.dk

Nobuko Yoshida
Imperial College
London
United Kingdom
n.yoshida@imperial.ac.uk

A two-party computation corresponds to the interaction of two processes and a multi-party computation corresponds therefore to the interaction of multiple (three or more) processes. The process algebra community has developed a thorough understanding of multi-party computation through a mechanism called *global type* that determines a sequential order of the individual send and receive actions among the participating parties.

Viewed from the vantage point of logic, we observe that two-party computations are also well understood through a Curry-Howard correspondence that was first pointed out by Caires and Pfenning in the setting of intuitionistic logic [CP10] and then by Wadler in the setting of classical logic [Wad14]. In the binary case, two processes form a two-party computation if their respective types are dual to each other. Inspired by this, we show how to generalize the notion of duality to coherence, which allows us to capture the essence of multi-party computations and establish a Curry-Howard correspondence between multi-party sessions and an extension of linear logic with coherence proofs. This paper is based on prior work in [CMSY15].

In the reminder of the paper, we illustrate the main contributions of this work using the classical 2-buyer protocol [HYC08] as example. Two buyers B1 and B2 attempt to buy a book together from seller S. B1 sends the title of the book that he intends to purchase. The seller S replies to both, B1 and B2, with a quote. B1 then sends a message to B2 about how much money he is willing to contribute to the purchase, leaving B2 with the decision either to contribute the remaining funds and to complete the purchase or to not buy the book at all. Formally, we write:

$$\begin{array}{ll} 1. & \mathsf{B1} \to \mathsf{S} : \langle \mathbf{str} \rangle;\ \mathsf{S} \to \mathsf{B1} : \langle \mathbf{int} \rangle;\ \mathsf{S} \to \mathsf{B2} : \langle \mathbf{int} \rangle;\ \mathsf{B1} \to \mathsf{B2} : \langle \mathbf{int} \rangle; \\ 2. & \mathsf{B2} \to \mathsf{S} : \& \,(\, \mathsf{B2} \to \mathsf{S} : \langle \mathbf{addr} \rangle; \mathsf{end},\ \mathsf{end}) \end{array} \tag{1}$$

Implicitly, for the purpose of this example, we assume that all communication proceeds through a single channel. Its type depends on the role of each party. When one sends, another receives. The following are the types of the shared channels for each role expressed in Wadler's CP. To be consistent with [Wad14] and in a slight deviation from [CDCYP15], we use $\otimes$ to type outputs and $⅋$ to type inputs.

$$\begin{array}{rl} \mathsf{B1}: & \mathbf{str} \otimes \mathbf{int} ⅋ \mathbf{int} \otimes \mathsf{end} \\ \mathsf{B2}: & \mathbf{int} ⅋ \mathbf{int} ⅋ \big((\mathbf{addr} \otimes \mathsf{end}) \oplus \mathsf{end} \big) \\ \mathsf{S}: & \mathbf{str} ⅋ \mathbf{int} \otimes \mathbf{int} \otimes \big((\mathbf{addr} ⅋ \mathsf{end}) \,\&\, \mathsf{end} \big) \end{array} \tag{2}$$

Above, each formula in classical linear logic (CLL) states how x is used by each process. For instance, B1 outputs ($\otimes$) a string, receives ($⅋$) an integer, sends another integer and eventually terminates (end).

The motivating observation of this work is that CLL is not general enough to express the composition of three or more processes sharing one channel, since the cut-rule can only compose two processes P and Q on one single shared channel x with a compatible types and A and $A^{\perp}$ and not on three.

$$\frac{P \vdash \Delta, x{:}A \qquad Q \vdash \Delta', x{:}A^{\perp}}{(\nu x{:}A)\,(P \mid Q) \vdash \Delta, \Delta'}\ \mathsf{Cut}$$

As a solution to this challenge, we propose to annotate the connectives with roles as the partner of the communication. For details, consult [CDCYP15].

$$\begin{array}{rl} \mathsf{B1}: & \mathbf{str} \otimes^{\mathsf{S}} \mathbf{int} \parr^{\mathsf{S}} \mathbf{int} \otimes^{\mathsf{B2}} \mathsf{end} \\ \mathsf{B2}: & \mathbf{int} \parr^{\mathsf{S}} \mathbf{int} \parr^{\mathsf{B1}} \big((\mathbf{addr} \otimes^{\mathsf{S}} \mathsf{end}) \oplus^{\mathsf{S}} \mathsf{end}\big) \\ \mathsf{S}: & \mathbf{str} \parr^{\mathsf{B1}} \mathbf{int} \otimes^{\mathsf{B1}} \mathbf{int} \otimes^{\mathsf{B2}} \big((\mathbf{addr} \parr^{\mathsf{B2}} \mathsf{end}) \mathbin{\&}^{\mathsf{B2}} \mathsf{end}\big) \end{array} \tag{3}$$

Annotations identify the dual role of each action, e.g., the usage for $\mathsf{B1}$ now reads: send a string to S ($\otimes^{\mathsf{S}}$); receive an integer from S ($\parr^{\mathsf{S}}$); send an integer to $\mathsf{B2}$ ($\otimes^{\mathsf{B2}}$); and, terminate (end). This trick allows us to generalize the standard notion of de Morgan duality that is defined between two processes of type A and $A^{\perp}$, to coherence between a set of processes $\{A_i\}_i$. Coherence is expressed using the judgment $G \vDash p_1 : A_1, \ldots, p_n : A_n$. Here, G is the global type, each p_i denotes a role of type A_i. Coherence is defined by the following rules.

$$\frac{G \vDash p{:}A,\ q{:}C \qquad G' \vDash \Theta,\ p{:}B,\ q{:}D}{p \to q : \langle G \rangle ; G' \vDash \Theta, p{:}A \otimes^{q} B,\ q{:}C \parr^{p} D} \parr\otimes \qquad \frac{}{\mathsf{end} \vDash p{:}1, q_1{:}\perp, \ldots, q_n{:}\perp} 1\perp$$

$$\frac{G_1 \vDash \Theta, p{:}A, q{:}C \qquad G_2 \vDash \Theta, p{:}B, q{:}D}{p \to q :\ \&(G_1, G_2) \vDash \Theta, p{:}A \oplus^{q} B, q{:}C \mathbin{\&}^{p} D} \oplus\& \qquad \frac{G \vDash p{:}A, q{:}B}{?p \to !q : \langle G \rangle \vDash p{:}?A, q{:}!B} !?$$

Building on coherence, we generalize the cut-rule Cut to a multi-cut rule MCut that defines the type of a multi-party computation by combining multiple processes P_i.

$$\frac{P_1 \vdash \Gamma_1, x^{p_1}{:}A_1 \quad \ldots \quad P_n \vdash \Gamma_n, x^{p_n}{:}A_n}{(\nu x{:}G)\,(P_1 \mid \ldots \mid P_n) \vdash \Gamma_1, \ldots, \Gamma_n} \ \mathsf{MCut}, \text{ where } G \vDash p_1{:}A_1^{\perp}, \ldots, p_n{:}A_n^{\perp}$$

In conclusion, this work presents to our knowledge the first formulation of a logic of multi-party sessions. The logic is a conservative extension over Wadler's CP, it is expressive as this example shows, and it is sound as cut-elimination and multi-cut elimination hold.

References

[CDCYP15] Mario Coppo, Mariangiola Dezani-Ciancaglini, Nobuko Yoshida, and Luca Padovani. Global progress for dynamically interleaved multiparty sessions. *MSCS*, 760:1–65, 2015.

[CMSY15] Marco Carbone, Fabrizio Montesi, Carsten Schürmann, and Nobuko Yoshida. Multiparty session types as coherence proofs. In *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015*, pages 412–426, 2015.

[CP10] Luís Caires and Frank Pfenning. Session types as intuitionistic linear propositions. In *CONCUR*, pages 222–236, 2010.

[HYC08] Kohei Honda, Nobuko Yoshida, and Marco Carbone. Multiparty asynchronous session types. In *Proc. of POPL*, volume 43(1), pages 273–284. ACM, 2008.

[Wad14] Philip Wadler. Propositions as sessions. *Journal of Functional Programming*, 24(2-3):384–418, 2014.