Multirole Logic and Multiparty Channels

Hongwei Xi and Hanwen Wu

Boston University, Boston, MA, USA
{hwxi,hwwu}@cs.bu.edu

Abstract

We identify multirole logic as a new form of logic in which conjunction/disjunction is interpreted as an ultrafilter on the power set of some underlying set (of roles), and the notion of negation is generalized to endomorphisms on this underlying set. In this talk, we present linear multirole logic (LMRL) as a natural generalization of classical linear logic (CLL). Among various meta-properties established for LMRL, we obtain one named multiparty cut-elimination stating that every cut involving one or more sequents (as a generalization of a binary cut involving exactly two sequents) can be eliminated, thus extending the celebrated result of cut-elimination by Gentzen. An immediate application of LMRL can be found in a formulation of session types for channels that support multiparty communication in distributed programming. Guided by LMRL, we give an interesting interpretation to linear multiplicative conjunction/disjunction as session type constructors that encompasses certain seemingly contradictory ones found in the literature.

Contents

1 Introduction

Session types [7, 8, 10] offer a type-theoretic approach to ensuring that communication protocols be correctly followed between concurrently running programs. Connections of session types to linear logic have long been investigated (e.g., [3, 1, 2]). In particular, certain forms of Curry-Howard correspondence are presented between (dyadic) session types and propositions of linear logic in some recent results (e.g., [11, 12, 4]), where matching actions send/receive are reflected through the duality in linear logical connectives $\otimes/\parr$ (or $\otimes/\supset$) and the communications involved in such actions are reflected through cut-reduction in linear logic for the case of $\otimes/\parr$ (or $\otimes/\supset$).

Multiparty session types [9] are introduced as a type discipline for communications involving more than two parties. The main challenge in seeking a direct correspondence between multiparty session types and (some form of) linear logic is to identify a suitable notion for multiple parties that *naturally* generalizes the standard notion of duality in logic.

We formulate linear multirole logic (LMRL) [15] as a natural generalization of classical linear logic (CLL). In LMRL, conjunction/disjunction is interpreted as an ultrafilter on the power set of some underlying set (of roles), and the notion of negation is generalized to endomorphisms on this underlying set. We also formulate a multi-threaded lambda-calculus (MTLC) where threads communicate on multiparty channels of linear types that are directly rooted in LMRL.

2 LMRL: Linear Multirole Logic

Multirole logic is parameterized over a chosen underlying set of roles, which may be infinite, and we use $\overline{\emptyset}$ to refer to this set. We use R to range over role sets, which are just subsets of $\overline{\emptyset}$. Given any role set R, we use $\overline{R}$ for the complement of R in $\overline{\emptyset}$. Also, we use $R_1 \uplus R_2$ for the disjoint union of R_1 and R_2 (where R_1 and R_2 are assumed to be disjoint).

A filter $\mathcal{F}$ on $\overline{\emptyset}$ is a subset of the power set of $\overline{\emptyset}$ such that

$$(1)\ \overline{\emptyset} \in \mathcal{F};\ \ (2)\ R_1 \in \mathcal{F} \text{ and } R_1 \subseteq R_2 \text{ implies } R_2 \in \mathcal{F};\ \ (3)\ R_1 \in \mathcal{F} \text{ and } R_2 \in \mathcal{F} \text{ implies } R_1 \cap R_2 \in \mathcal{F}.$$

A filter $\mathcal{F}$ on $\overline{\emptyset}$ is an ultrafilter if either $R \in \mathcal{F}$ or $\overline{R} \in \mathcal{F}$ holds for every subset R of $\overline{\emptyset}$. We use $\mathcal{U}$ to range over ultrafilters on $\overline{\emptyset}$. When there is no risk of confusion, we may simply use r for the principal ultrafilter at r, which is defined as $\{R \subseteq \overline{\emptyset} \mid r \in R\}$. If $\overline{\emptyset}$ is finite, then it can be readily proven that each $\mathcal{U}$ on $\overline{\emptyset}$ is a principal filter at some element r.

Given an endomorphism f on $\overline{\emptyset}$, that is, a mapping from $\overline{\emptyset}$ to itself, we use $\neg_f$ for a unary connective. Given an ultrafilter $\mathcal{U}$ on $\overline{\emptyset}$, we use $\&_{\mathcal{U}}$ and $\otimes_{\mathcal{U}}$ for two binary connectives. Note that we may equally choose the name $\oplus_{\mathcal{U}}$ for $\&_{\mathcal{U}}$ (and $\parr_{\mathcal{U}}$ for $\otimes_{\mathcal{U}}$) as the meaning of the named connective solely comes from $\mathcal{U}$. The formulas in LMRL are defined as follows:

$$\text{formulas} \quad A \quad ::= \quad a \mid \neg_f(A) \mid A_1 \&_{\mathcal{U}} A_2 \mid A_1 \otimes_{\mathcal{U}} A_2$$

where a ranges over pre-defined primitive formulas. We may use B and C to range over formulas as well. We may also write $f(A)$ for $\neg_f(A)$.

Given a formula A and a set R of roles, we write $[A]_R$ for an i-formula, which is some sort of interpretation of A based on R. For instance, the interpretation of $\otimes_r$ based on R is conjunction-like if $r \in R$ holds, and it is disjunction-like otherwise. A crucial point, which we learned when studying multiparty session types [13, 14], is that interpretations should be based on sets of roles rather than just individual roles. In other words, one side is allowed to play multiple roles simultaneously. A sequent Γ in multirole logic is a multiset of i-formulas, and such a sequent is inherently many-sided as each R appearing in Γ represents precisely one side. For instance, the inference rules for $\neg_f$ and $\otimes_{\mathcal{U}}$ in LMRL are given as follows:

$$\frac{\vdash \Gamma, [A]_{f^{-1}(R)}}{\vdash \Gamma, [\neg_f(A)]_R}\ (\neg) \qquad \frac{R \notin \mathcal{U} \quad \vdash \Gamma, [A]_R, [B]_R}{\vdash \Gamma, [A \otimes_{\mathcal{U}} B)]_R}\ (\otimes\textbf{-neg}) \qquad \frac{R \in \mathcal{U} \quad \vdash \Gamma_1, [A]_R \quad \vdash \Gamma_2, [B]_R}{\vdash \Gamma_1, \Gamma_2, [A \otimes_{\mathcal{U}} B]_R}\ (\otimes\textbf{-pos})$$

where $f^{-1}(R)$ refers to the pre-image of R under f.

The (binary) cut-elimination of Gentzen can be generalized to the following multiparty cut-elimination involving n sequents for $n \geq 1$:

Assume that $R_1, \ldots, R_n$ are subsets of $\overline{\emptyset}$ for some $n \geq 1$. If $\overline{R}_1 \uplus \cdots \uplus \overline{R}_n = \overline{\emptyset}$ holds, then the following inference rule (mp-cut) is admissible in LMRL:

$$\frac{\overline{R}_1 \uplus \cdots \uplus \overline{R}_n = \overline{\emptyset} \quad \vdash \Gamma_1, [A]_{R_1} \quad \cdots \quad \vdash \Gamma_n, [A]_{R_n}}{\vdash \Gamma_1, \ldots, \Gamma_n}$$

3 Linearly Typed Multiparty Channels

We use A, B and C in the rest of this section both for formulas in LMRL and for session types. We use CH for channels supporting communication between multiple parties (processes) involved in a session.

We assume that each channel CH consists of two or more endpoints and each process holding one endpoint can communicate with the processes holding the other endpoints (of the same channel). We use CH_R to denote one endpoint of CH, where R is a role set (that is, a subset of $\overline{\emptyset}$). At any given time, if CH consists of n endpoints $\mathrm{CH}_{R_1}, \ldots, \mathrm{CH}_{R_n}$, then $R_1 \uplus \cdots \uplus R_n = \overline{\emptyset}$. Note that a process can only hold an endpoint of a channel; it cannot hold a channel per se.

We say that a channel CH is of some session type A if each endpoint CH_R can be assigned the type $\mathbf{chan}(R, A)$ (where **chan** is some linear type constructor). We may write $\mathbf{chan}(A)$ to mean $\mathbf{chan}(R, A)$ for some role set R.

Primitive formulas As an example, let $\mathbf{msg}(r_0, r_1)$ be a primitive formula for any given pair of distinct roles r_0 and r_1. Then the specified action by $\mathbf{msg}(r_0, r_1)$ on an endpoint CH_R can be described as follows:

- Assume $r_0 \in R$ and $r_1 \in R$. There is no action.
- Assume $r_0 \in R$ and $r_1 \notin R$. A message is sent to the endpoint $\mathrm{CH}_{R'}$ for the only R' containing r_1.
- Assume $r_0 \notin R$ and $r_1 \in R$. A message is received (from the endpoint $\mathrm{CH}_{R'}$ for the only R' containing r_0).
- Assume $r_0 \notin R$ and $r_1 \notin R$. There is no action.

In other words, $\mathbf{msg}(r_0, r_1)$ specifies a form of point-to-point messaging from the endpoint CH_R to the endpoint $\mathrm{CH}_{R'}$ for R and R' containing r_0 and r_1, respectively.

Multiplicative conjunction/disjunction Given a role r, we have $\otimes_{\mathcal{U}}$ in LMRL for the principal ultrafilter $\mathcal{U}$ at r. Let us write $\otimes_r$ for this $\otimes_{\mathcal{U}}$. Also, we may write $\mathbf{chan}(R, A \otimes B)$ to mean $\mathbf{chan}(R, A \otimes_r B)$ for some $r \in R$ and $\mathbf{chan}(R, A \parr B)$ to mean $\mathbf{chan}(R, A \otimes_r B)$ for some $r \notin R$. For each CH of session type $A \otimes_r B$, there is exactly one endpoint of type $\mathbf{chan}(A \otimes B)$ and each of the other endpoints is of type $\mathbf{chan}(A \parr B)$. *Intutively, a process holding an endpoint of type* $\mathbf{chan}(R, A \otimes B)$ *turns it into two endpoints of types* $\mathbf{chan}(R, A)$ *and* $\mathbf{chan}(R, B)$ *for being used sequentially (that is, only one of these two endpoints can be used at any given moment) while any process holding an endpoint of type* $\mathbf{chan}(R, A \parr B)$ *turns it into two endpoints of types* $\mathbf{chan}(R, A)$ *and* $\mathbf{chan}(R, B)$ *for being used concurrently. In other words, a process holding an endpoint of type* $\mathbf{chan}(R, A \otimes B)$ *can choose to interleave the interactions specified by A and B in any order while any process holding an endpoint of type* $\mathbf{chan}(R, A \parr B)$ *must be able to handle any chosen order of interleaving of interactions specified by A and B.*

In the literature, $A \otimes B$ ($A \parr B$) is interpreted as *output A and then behave as B* (*input A and then behave as B*) or vice versa [12, 5, 6]. The interpretation we give for $\otimes$ and $\parr$ as session type constructors can actually encompass as a special case the seemingly contradictory interpretations of $\otimes$ and $\parr$ in the literature. In this special case, a process holding an endpoint of type $\mathbf{chan}(R, A \otimes B)$ turns it into two endpoints of types $\mathbf{chan}(R, A)$ and $\mathbf{chan}(R, B)$ and then decides to finish all of the interactions specified by A before initiating the interactions specified by B. Let us use $@_r$ for this special variant of $\otimes_r$.

Unlike $\otimes_r$, there is not need to create new endpoints when an endpoint of type $\mathbf{chan}(A @_r B)$ is split as the endpoint itself can be first used as an endpoint of type $\mathbf{chan}(A)$ and then as another endpoint of type $\mathbf{chan}(B)$. Therefore, there is no difference between $@_r$ and $@_{r'}$ even if r and r' are distinct, implying that we can use $@$ for any $@_r$.

4 Related Work

The most closely related work is probably an extension of CLL by Carbone et al [5, 6] that admits a cut involving multiple sequents. While we may share the very same motivation for seeking a form of cut-rule that can involving more than two sequents, the multirole we take is fundamentally different, offering an alternative solution to the quest for a multi-sequent cut.

References

[1] S. Abramsky. Computational Interpretations of Linear Logic. *Theor. Comput. Sci.*, 1993.

[2] S. Abramsky. Proofs as Processes. *Theor. Comput. Sci.*, 1994.

[3] G. Bellin and P. J. Scott. On the pi-Calculus and Linear Logic. *Theor. Comput. Sci.*, 1994.

[4] L. Caires and F. Pfenning. Session Types as Intuitionistic Linear Propositions. *CONCUR*, 2010.

[5] M. Carbone, S. Lindley, F. Montesi, C. Schürmann, and P. Wadler. Coherence Generalises Duality - A Logical Explanation of Multiparty Session Types. *CONCUR*, 2016.

[6] M. Carbone, F. Montesi, C. Schürmann, and N. Yoshida. Multiparty session types as coherence proofs. *Acta Inf.*, 2017.

[7] K. Honda. Types for Dyadic Interaction. *CONCUR*, 1993.

[8] K. Honda, V. T. Vasconcelos, and M. Kubo. Language Primitives and Type Discipline for Structured Communication-Based Programming. *ESOP*, 1998.

[9] K. Honda, N. Yoshida, and M. Carbone. Multiparty asynchronous session types. *POPL*, 2008.

[10] K. Takeuchi, K. Honda, and M. Kubo. An Interaction-based Language and its Typing System. *PARLE*, 1994.

[11] P. Wadler. Propositions as sessions. *ICFP*, 2012.

[12] P. Wadler. Propositions as sessions. *J. Funct. Program.*, 2014.

[13] H. Xi and H. Wu. Linearly Typed Dyadic Group Sessions for Building Multiparty Sessions. *CoRR*, 2016.

[14] H. Xi and H. Wu. Propositions in Linear Multirole Logic as Multiparty Session Types. *CoRR*, 2016.

[15] H. Xi and H. Wu. Multirole Logic (Extended Abstract). *CoRR*, 2017.