University of Udine, Italy
Department of Mathematics, Computer Science and Physics

Composable Partial Multiparty Session Types

Claude Stolze, Marino Miculan, and Pietro Di Gianantonio

January 24, 2022

MPST *à la* Carbone et al.

- Global types

$$G := s \to c : \left\{ \begin{array}{l} \text{login}.c \to a : \text{passwd}(\text{Str}).a \to s : \text{auth}(\text{Bool}), \\ \text{cancel}.c \to a : \text{quit} \end{array} \right\}$$

- Local types

$$\begin{array}{ll} S_s & := c \oplus \left\{ \begin{array}{l} \text{login}.a\&\text{auth}(\text{Bool}), \\ \text{cancel} \end{array} \right\} \\ S_a & := c\& \left\{ \begin{array}{l} \text{passwd}(\text{Str}).s \oplus \text{auth}(\text{Bool}), \\ \text{quit} \end{array} \right\} \\ S_c & := s\& \left\{ \begin{array}{l} \text{login}.a \oplus \text{passwd}(\text{Str}), \\ \text{cancel}.a \oplus \text{quit} \end{array} \right\} \end{array}$$

Partial session?

$$
\begin{aligned}
S_s \quad &:= c\oplus\left\{ \begin{array}{l} \text{login}.a\&\text{auth}(\text{Bool}), \\ \text{cancel} \end{array} \right\} \\
S_a \quad &:= c\&\left\{ \begin{array}{l} \text{passwd}(\text{Str}).s\oplus\text{auth}(\text{Bool}), \\ \text{quit} \end{array} \right\}
\end{aligned}
$$

- Participant c is missing
- No global type
- Are these type compatible? Are there deadlocks?
- What is the behaviour of $S_s \mid S_a$?

Summary of the results

Main achievements

- Multiparty session types for any session (partial or not)
- Process typing is *relative* to *points of view* (sets of participants)
- No distinction between local and global types
- Type system

$$\frac{P \vdash \Gamma_1, x : \langle G_1 \mid S_1 \rangle \quad Q \vdash \Gamma_2, x : \langle G_2 \mid S_2 \rangle \quad S_1 \sharp S_2 \quad G_3 \simeq_{S_1 \uplus S_2} G_1 \; {}^{S_1}\!\vee^{S_2} \; G_2}{P \mid_x Q \vdash \Gamma_1, \Gamma_2, x : \langle G_3 \mid S_1 \uplus S_2 \rangle} \; (\mid)$$

- Merging algorithm for session types
- Deadlocks are detected by the merging algorithm
- Prototype implementation of the merging algorithm

Syntax (inspired by Carbone et al.)

$$
\begin{array}{rcll}
P, Q, R ::= & & \overline{x}^{p\tilde{q}} \triangleright \mathrm{in}_i.P & \text{(choice)} \\
& | & x^{pq} \triangleleft (P, Q) & \text{(case)} \\
& | & \overline{x}^{p\tilde{q}}(y).P & \text{(subsession creation and sending)} \\
& | & x^{pq}(y).(P \parallel Q) & \text{(subsession reception and fork)} \\
& | & \mathrm{close}(x) & \text{(process stop)} \\
& | & \mathrm{wait}(x).P & \text{(session closing)} \\
& | & P + Q & \text{(non-deterministic choice)} \\
& | & (P \mid_x Q) & \text{(composition)} \\
& | & (\nu x)P & \text{(restriction)}
\end{array}
$$

Operational semantics (Labelled Transition System)

$$\overline{x}^{p\tilde{q}}(y).R \mid_x \Pi_i^x(x^{q_i p}(y).(P_i \parallel Q_i)) \xrightarrow{x:p\to\tilde{q}:\langle\cdot\rangle} (\nu y)(R \mid_y \Pi_i^y P_i) \mid_x \Pi_i^x Q_i$$

$$\overline{x}^{p\tilde{q}} \triangleright \mathsf{in}_j.R \mid_x \Pi_i^x x^{q_i p} \triangleleft (P_{1,i}, P_{2,i}) \xrightarrow{x:p\to\tilde{q}:\&\mathsf{in}_j} R \mid_x \Pi_i^x P_{j,i}$$

$$(\nu x)(\mathsf{wait}(x).P) \xrightarrow{\tau} P \quad \text{if } x \notin \mathsf{fn}(P)$$

$$P_1 + P_2 \xrightarrow{+} P_j \quad j \in \{1,2\}$$

$$\frac{P \xrightarrow{x:\gamma} Q}{(\nu x)P \xrightarrow{\tau} (\nu x)Q} \qquad \frac{P \xrightarrow{\alpha} Q \quad \forall\gamma, \alpha \neq x:\gamma}{(\nu x)P \xrightarrow{\alpha} (\nu x)Q} \qquad \frac{P \xrightarrow{\alpha} Q}{P \mid_x R \xrightarrow{\alpha} Q \mid_x R}$$

Delegated choice

$$P_p := (\overline{x}^{pq} \triangleright \text{in}_1.\overline{x}^{pr}(y).\text{wait}(y).\text{close}(x)) + (\overline{x}^{pq} \triangleright \text{in}_2.\text{close}(x))$$
$$P_q := x^{qp} \triangleleft (\overline{x}^{qr} \triangleright \text{in}_1.\text{close}(x), \overline{x}^{qr} \triangleright \text{in}_2.\text{close}(x))$$
$$P_r := x^{rq} \triangleleft (x^{rp}(y)(\text{close}(y) \parallel \text{close}(x)), \text{close}(x))$$

- Example of execution

$$\begin{aligned} P_p \mid_x P_q \mid_x P_r & \xrightarrow{+} \overline{x}^{pq} \triangleright \text{in}_2.\text{close}(x) \mid_x P_q \mid_x P_r \\ & \xrightarrow{x:p\to q:\&\text{in}_2} \overline{x}^{qr} \triangleright \text{in}_2.\text{close}(x) \mid_x P_r \\ & \xrightarrow{x:q\to r:\&\text{in}_2} \text{close}(x) \end{aligned}$$

- Partial execution

$$\begin{aligned} P_p \mid_x P_q & \xrightarrow{+} \overline{x}^{pq} \triangleright \text{in}_2.\text{close}(x) \mid_x P_q \\ & \xrightarrow{x:p\to q:\&\text{in}_2} \overline{x}^{qr} \triangleright \text{in}_2.\text{close}(x) \end{aligned}$$

Syntax of session types

$$
\begin{array}{rlr}
G ::= & p \to \tilde{q} : \&\mathrm{in}_i ; G & \text{(choice sending)} \\
| & p \to \tilde{q} : \langle G \rangle ; G & \text{(subsession sending)} \\
| & G \oplus G & \text{(internal choice)} \\
| & G \,\&\, G & \text{(external choice)} \\
| & \mathrm{end} & \text{(session closing)} \\
| & \mathrm{close} & \text{(process stop)} \\
| & 0 & \text{(empty session)} \\
| & \omega & \text{(deadlock)}
\end{array}
$$

Disjunctive Normal Form

- Messages and communications

$$m ::= \&\mathrm{in}_i \mid \langle G \rangle \qquad C ::= p \to \tilde{q} : m \mid \mathsf{end} \mid \mathsf{close} \mid 0 \mid \omega \mid 1$$

- A chain of communications is a type of the form $C_1; C_2; \ldots; C_n$
- A type is in DNF if it has the form

$$\bigoplus_{i \in I} \mathop{\&}_{j \in J_i} G_{i,j}$$

where the $G_{i,j}$ are chains of communications

- Any type can be rewritten into a DNF

Equivalence of session types

Independence

- C_1 and C_2 are independent for a set of participants S (noted $C_1 \; I_S \; C_2$), if S cannot distinguish $C_1; C_2$ from $C_2; C_1$
- *e.g.* $(p \to \tilde{q} : m) \; I_S \; (p' \to \tilde{q}' : m')$ if $(\{p\} \cup \tilde{q}) \cap (\{p'\} \cup \tilde{q}') \cap S = \varnothing$
- Example: $(p \to q : \langle G_1 \rangle) \; I_{\{q,r\}} \; (p \to r : \langle G_2 \rangle)$

Equivalence relation

- G_1 and G_2 are equivalent for a set of participants S (noted $G_1 \simeq_S G_2$), if the participants in S cannot distinguish G_1 and G_2
- Example:

$$p \to q : \langle G_1 \rangle; p \to r : \langle G_2 \rangle; \mathsf{end} \simeq_{\{q,r\}} p \to r : \langle G_2 \rangle; p \to q : \langle G_1 \rangle; \mathsf{end}$$

Environment

- Sessions x are typed with a session type G for a set of participant S (noted $x : \langle G \mid S \rangle$)
- $\Gamma = x_1 : \langle G_1 \mid S_1 \rangle, \ldots, x_n : \langle G_n \mid S_n \rangle$

$$\frac{\Gamma_1 \simeq \Gamma_2 \quad G_1 \simeq_S G_2}{\Gamma_1, x : \langle G_1 \mid S \rangle \simeq \Gamma_2, x : \langle G_2 \mid S \rangle}$$

- Typing judgments have the form $P \vdash \Gamma$

Typing rules

1/2

$$\frac{P \vdash \Gamma, y : \langle G_1 \mid \{p\}\rangle, x : \langle G_2 \mid \{p\}\rangle}{\overline{x}^{p\tilde{q}}(y).P \vdash \Gamma, x : \langle p \to \tilde{q} : \langle G_1 \rangle ; G_2 \mid \{p\}\rangle} \ (\mathit{send})$$

$$\frac{P \vdash \Gamma_1, y : \langle G_1 \mid \{q\}\rangle \quad Q \vdash \Gamma_2, x : \langle G_2 \mid \{q\}\rangle \quad q \in \tilde{q}}{x^{qp}(y).(P \parallel Q) \vdash \Gamma_1, \Gamma_2, x : \langle p \to \tilde{q} : \langle G_1 \rangle ; G_2 \mid \{q\}\rangle} \ (\mathit{recv})$$

$$\frac{P \vdash \Gamma, x : \langle G \mid \{p\}\rangle}{\overline{x}^{p\tilde{q}} \triangleright \mathrm{in}_i.P \vdash \Gamma, x : \langle p \to \tilde{q} : \&\mathrm{in}_i ; G \mid \{p\}\rangle} \ (\mathit{sel}_i)$$

$$\frac{P \vdash \Gamma, x : \langle G_1 \mid \{q\}\rangle \quad Q \vdash \Gamma, x : \langle G_2 \mid \{q\}\rangle \quad q \in \tilde{q}}{x^{qp} \triangleleft (P, Q) \vdash \Gamma, x : \langle (p \to \tilde{q} : \&\mathrm{in}_1 ; G_1) \,\&\, (p \to \tilde{q} : \&\mathrm{in}_2 ; G_2) \mid \{q\}\rangle} \ (\mathit{case})$$

Typing rules

2/2

$$\frac{P \vdash x_1 : \langle G_1 \mid S_1\rangle, \ldots, x_n : \langle G_n \mid S_n\rangle \quad Q \vdash x_1 : \langle G_1' \mid S_1\rangle, \ldots, x_n : \langle G_n' \mid S_n\rangle}{P + Q \vdash x_1 : \langle G_1 \oplus G_1' \mid S_1\rangle, \ldots, x_n : \langle G_n \oplus G_n' \mid S_n\rangle}\ (+)$$

$$\frac{}{\mathrm{close}(x) \vdash x : \langle \mathrm{close} \mid \varnothing\rangle}\ (\mathit{close}) \qquad \frac{P \vdash \Gamma}{\mathrm{wait}(x).P \vdash \Gamma, x : \langle \mathrm{end} \mid \{p\}\rangle}\ (\mathit{wait})$$

$$\frac{P \vdash \Gamma, x : \langle G \mid S_1\rangle \quad S_2 \sharp \mathrm{fn}(G)}{P \vdash \Gamma, x : \langle G \mid S_1 \cup S_2\rangle}\ (\mathit{extra}) \qquad \frac{P \vdash \Gamma \quad \Gamma \simeq \Gamma'}{P \vdash \Gamma'}\ (\simeq)$$

$$\frac{P \vdash \Gamma_1, x : \langle G_1 \mid S_1\rangle \quad Q \vdash \Gamma_2, x : \langle G_2 \mid S_2\rangle \quad S_1 \sharp S_2 \quad G_3 \simeq_{S_1 \uplus S_2} G_1 \ {}^{S_1}\!\vee^{S_2}\ G_2}{P \mid_x Q \vdash \Gamma_1, \Gamma_2, x : \langle G_3 \mid S_1 \uplus S_2\rangle}\ (\mid)$$

$$\frac{P \vdash \Gamma, x : \langle G \mid S\rangle \quad G \downarrow S}{(\nu x)P \vdash \Gamma}\ (\nu)$$

G is finalized for S (noted $G \downarrow S$) if a session $x : \langle G \mid S\rangle$ can safely be restricted.

- We note D_1 for the following derivation:

$$\dfrac{\dfrac{\mathsf{close}(x) \vdash x : \langle \mathsf{close} \mid \varnothing \rangle}{\mathsf{close}(x) \vdash x : \langle \mathsf{close} \mid \{p\} \rangle}\ (extra)}{\dfrac{\mathsf{wait}(y).\mathsf{close}(x) \vdash x : \langle \mathsf{close} \mid \{p\} \rangle, y : \langle \mathsf{end} \mid \{p\} \rangle}{\overline{x}^{pq}(y).\mathsf{wait}(y).\mathsf{close}(x) \vdash x : \langle p \to q : \langle \mathsf{end} \rangle ; \mathsf{close} \mid \{p\} \rangle}}$$

- We note D_2 for the following derivation:

$$\dfrac{\dfrac{\mathsf{close}(z) \vdash z : \langle \mathsf{close} \mid \varnothing \rangle}{\mathsf{close}(z) \vdash z : \langle \mathsf{close} \mid \{q\} \rangle} \quad \dfrac{\mathsf{close}(x) \vdash x : \langle \mathsf{close} \mid \varnothing \rangle}{\mathsf{close}(x) \vdash x : \langle \mathsf{close} \mid \{q\} \rangle}}{x^{qp}(z).(\mathsf{close}(z) \parallel \mathsf{close}(x)) \vdash x : \langle p \to q : \langle \mathsf{close} \rangle ; \mathsf{close} \mid \{q\} \rangle}$$

- Finally:

$$\dfrac{D_1 \quad D_2 \quad p \to q : \langle \mathsf{end} \rangle ; \mathsf{close} \ {}^{\{p\}}\!\vee^{\{q\}}\ p \to q : \langle \mathsf{close} \rangle ; \mathsf{close} \simeq_{\{p,q\}} p \to q : \langle \mathsf{end} \rangle ; \mathsf{close}}{\overline{x}^{pq}(y).\mathsf{wait}(y).\mathsf{close}(x) \mid_x x^{qp}(z).(\mathsf{close}(z) \parallel \mathsf{close}(x)) \vdash x : \langle p \to q : \langle \mathsf{end} \rangle ; \mathsf{close} \mid \{p, q\} \rangle}$$

Mergeable communications from different viewpoints

$$\frac{\{p\} \cup \tilde{q}_1 \cup \tilde{q}_2 \subseteq S_1 \cup S_2 \Rightarrow (G_1 \; {}^{S_1}\!\vee^{S_2} \; G_2) \downarrow S_1 \cup S_2 \qquad p \in S_1 \Rightarrow \tilde{q}_2 \subseteq \tilde{q}_1 \quad p \in S_2 \Rightarrow \tilde{q}_1 \subseteq \tilde{q}_2 \quad S_1 \cap \tilde{q}_2 \subseteq \tilde{q}_1 \quad S_2 \cap \tilde{q}_1 \subseteq \tilde{q}_2}{p \to \tilde{q}_1 : \langle G_1 \rangle \; {}^{S_1}\!\heartsuit^{S_2} \; p \to \tilde{q}_2 : \langle G_2 \rangle}$$

$$\frac{p \in S_1 \Rightarrow \tilde{q}_2 \subseteq \tilde{q}_1 \quad p \in S_2 \Rightarrow \tilde{q}_1 \subseteq \tilde{q}_2 \quad S_1 \cap \tilde{q}_2 \subseteq \tilde{q}_1 \quad S_2 \cap \tilde{q}_1 \subseteq \tilde{q}_2}{p \to \tilde{q}_1 : \&\mathrm{in}_i \; {}^{S_1}\!\heartsuit^{S_2} \; p \to \tilde{q}_2 : \&\mathrm{in}_i} \qquad \frac{}{1 \; {}^{S_1}\!\heartsuit^{S_2} \; 1}$$

$$\frac{C_2 \; {}^{S_2}\!\heartsuit^{S_1} \; C_1}{C_1 \; {}^{S_1}\!\heartsuit^{S_2} \; C_2} \qquad \frac{(\{p\} \cup \tilde{q}) \sharp S_1}{1 \; {}^{S_1}\!\heartsuit^{S_2} \; p \to \tilde{q} : m} \qquad \frac{}{\mathrm{close} \; {}^{S_1}\!\heartsuit^{S_2} \; \mathrm{close}} \qquad \frac{}{\mathrm{close} \; {}^{S_1}\!\heartsuit^{S_2} \; \mathrm{end}}$$

- The most complicated case is $p \to \tilde{q}_1 : \langle G_1 \rangle \; {}^{S_1}\!\heartsuit^{S_2} \; p \to \tilde{q}_2 : \langle G_2 \rangle$
- $G_1 \; {}^{S_1}\!\vee^{S_2} \; G_2$ should be finalized if all the participants have given their viewpoints
- Involved participants should be present *w.r.t.* viewpoints

Merging communications

If $C_1 \; {}^{S_1}\!\heartsuit^{S_2} \; C_2$, then $\mathsf{mcomm}_{S_1,S_2}(C_1, C_2)$ is defined as:

$$
\begin{aligned}
\mathsf{mcomm}_{S_1,S_2}(p \to \tilde{q} : \&\mathrm{in}_i, p \to \tilde{q}' : \&\mathrm{in}_i) &:= p \to (\tilde{q} \cup \tilde{q}') : \&\mathrm{in}_i \\
\mathsf{mcomm}_{S_1,S_2}(p \to \tilde{q} : \langle G_1 \rangle, p \to \tilde{q}' : \langle G_2 \rangle) &:= p \to (\tilde{q} \cup \tilde{q}') : \langle G_1 \; {}^{S_1}\!\vee^{S_2} \; G_2 \rangle \\
\mathsf{mcomm}_{S_1,S_2}(1, C) &:= C \\
\mathsf{mcomm}_{S_1,S_2}(C, 1) &:= C \\
\mathsf{mcomm}_{S_1,S_2}(C, \mathrm{close}) &:= C \\
\mathsf{mcomm}_{S_1,S_2}(\mathrm{close}, C) &:= C
\end{aligned}
$$

Merging algorithm (simplified)

- Types are rewritten in DNF
- Basic rules

$$
\begin{array}{rcl}
(G_1 \oplus G_2) \ {}^{S_1}\!\vee^{S_2}\ G_3 & := & (G_1 \ {}^{S_1}\!\vee^{S_2}\ G_3) \oplus (G_2 \ {}^{S_1}\!\vee^{S_2}\ G_3) \\
G_1 \ {}^{S_1}\!\vee^{S_2}\ (G_2 \oplus G_3) & := & (G_1 \ {}^{S_1}\!\vee^{S_2}\ G_2) \oplus (G_1 \ {}^{S_1}\!\vee^{S_2}\ G_3) \\
(G_1 \,\&\, G_2) \ {}^{S_1}\!\vee^{S_2}\ G_3 & := & (G_1 \ {}^{S_1}\!\vee^{S_2}\ G_3) \,\&\, (G_2 \ {}^{S_1}\!\vee^{S_2}\ G_3) \\
G_1 \ {}^{S_1}\!\vee^{S_2}\ (G_2 \,\&\, G_3) & := & (G_1 \ {}^{S_1}\!\vee^{S_2}\ G_2) \,\&\, (G_1 \ {}^{S_1}\!\vee^{S_2}\ G_3)
\end{array}
$$

- Otherwise find all rewritings $G_1 \simeq_{S_1} C'_i; G'_i$ and $G_2 \simeq_{S_2} C''_i; G''_i$ s.t. $C'_i \ {}^{S_1}\!\heartsuit^{S_2}\ C''_i$, and

$$G_1 \ {}^{S_1}\!\vee^{S_2}\ G_2 := \mathop{\&}_{i \in I} \{\mathsf{mcomm}_{S_1,S_2}(C'_i, C''_i); (G'_i \ {}^{S_1}\!\vee^{S_2}\ G''_i)\}$$

where I enumerates all possible mergings

Reduction of session types

$$G_1 \oplus G_2 \xrightarrow{+}_S G_i \qquad p \to \tilde{q} : \langle G_1 \rangle ; G_2 \xrightarrow{p \to \tilde{q} : \langle \cdot \rangle}_S G_2$$

$$p \to \tilde{q} : \&\mathrm{in}_i ; G \xrightarrow{p \to \tilde{q} : \&\mathrm{in}_i}_S G \qquad \frac{G_1 \xrightarrow{\gamma}_S G' \quad G_1 \simeq_S G_2}{G_2 \xrightarrow{\gamma}_S G'}$$

Theorems: Subject reduction and equivalence

- Subject equivalence: If $P \vdash \Gamma$ and $P \equiv Q$, then $Q \vdash \Gamma$
- Subject reduction: If $P_1 \vdash \Gamma_1$ and $P_1 \xrightarrow{\alpha} P_2$, then for some Γ_2, we have $P_2 \vdash \Gamma_2$ and $\Gamma_1 \xrightarrow{\alpha} \Gamma_2$

Progress

Preemption

- $x : \langle G \mid S \rangle$ preempts P (noted $x : \langle G \mid S \rangle \gg_c P$) when every local participant in S is ready to trigger its communication described by G
- Example: $x : \langle p \to q : \&\mathrm{in}_1; G \mid \{p\} \rangle \gg_c (\overline{x}^{pq} \triangleright \mathrm{in}_1.P)$
- However: $x : \langle p \to q : \&\mathrm{in}_1; G \mid \{p, q\} \rangle \not\gg_c (\overline{x}^{pq} \triangleright \mathrm{in}_1.P)$
- But: $x : \langle p \to q : \&\mathrm{in}_1; G \mid \{p, q\} \rangle \gg_c (\overline{x}^{pq} \triangleright \mathrm{in}_1.P \mid_x x^{qp} \triangleleft (Q, R))$

Theorem: Progress

- If $G \downarrow S$ and $x : \langle G \mid S \rangle \gg_c P$, then $(\nu x)P$ has a redex
- If $P \vdash \Gamma$ then there is a redex in P, or for some $x : \langle G \mid S \rangle \in \Gamma$ we have $x : \langle G \mid S \rangle \gg_c P$

Semantics overview

To interpret types and the operations we introduce:

- A symmetric monoidal category $\textsc{Comm}$ of *communication structures*
- A symmetric monoidal category $\textsc{View}$ of *viewpoints* (sets of participants)
- A monoidal functor $L : \textsc{View} \to \textsc{Comm}$, and the merge of communications corresponds to its coherence map (structural natural transformation)
- An interpretation of communication structures $F : \textsc{Comm} \to \textsc{Set}$ as sets of sets of traces, and the merge of these sets is the image of L's coherence map under this interpretation
- An interpretation of types as sets of sets of traces, and the merge of types is the lifting of the merge of communication structures

Semantics: COMM

Communication structure

A communication structure is given by $A = (E_A, I_A, 1_A)$ where:

- E_A is a set
- $1_A \in E_A$, $I_A \subseteq E_A \times E_A$ is a symmetric relation called the independence relation
- $\forall x \in E_A, x \; I_A \; 1_A$

Category COMM

- an object is a communication structure
- a morphism $f : (E_A, I_A, 1_A) \to (E_B, I_B, 1_B)$ is a partial function from E_A to E_B such that $f(1_A) = f(1_B)$, and, for any $x, y \in E_A$ such that both $f(x)$ and $f(y)$ are defined, we have that $f(x) \; I_B \; f(y)$ iff $x \; I_A \; y$
- composition and identities are standard

Monoidal product

The monoidal product of $(E_A, I_A, 1_A)$ and $(E_B, I_B, 1_B)$ is $(E_C, I_C, 1_C) = (E_A, I_A, 1_A) \otimes (E_B, I_B, 1_B)$ as follows:

- $E_C = E_A \times E_B$, with projections $\pi_A : E_A \times E_B \to E_A$ and $\pi_B : E_A \times E_B \to E_B$;
- for all $x_A, y_A \in E_A$ and $x_B, y_B \in E_B$, $(x_A, x_B)\ I_C\ (y_A, y_B) \Leftrightarrow x_A\ I_A\ y_A$ and $x_B\ I_B\ y_B$;
- $1_C = (1_A, 1_B)$

Semantics: VIEW and functor L

Category VIEW

- objects are sets of participants, in $\mathcal{P}(\mathfrak{P})$
- there exists a unique morphism $f : A \to B$ iff $A \subseteq B$
- the tensor $A \otimes B$ is the union $A \cup B$, and the unit is $\varnothing$

There is a lax symmetric monoidal functor $(L, \mu, \upsilon) :$ VIEW $\to$ COMM:

- a functor $L :$ VIEW $\to$ COMM
- a natural transformation $\mu_{S_1,S_2} : L(S_1) \otimes L(S_2) \to L(S_1 \cup S_2)$
- a morphism $\upsilon : J \to L(\varnothing)$

$$\begin{array}{ccc}
(L(S_1) \otimes L(S_2)) \otimes L(S_3) & \xrightarrow{\alpha} & L(S_1) \otimes (L(S_2) \otimes L(S_3)) \\
\downarrow \mu_{S_1,S_2} \otimes \mathrm{id} & & \downarrow \mathrm{id} \otimes \mu_{S_2,S_3} \\
L(S_1 \cup S_2) \otimes L(S_3) & & L(S_1) \otimes L(S_2 \cup S_3) \\
\downarrow \mu_{S_1 \cup S_2, S_3} & & \downarrow \mu_{S_1, S_2 \cup S_3} \\
L(S_1 \cup S_2 \cup S_3) & \xleftrightarrow{\mathrm{id}} & L(S_1 \cup S_2 \cup S_3)
\end{array}$$

$$\begin{array}{ccc}
L(S_1) \otimes L(S_2) & \xrightarrow{\gamma} & L(S_2) \otimes L(S_1) \\
\downarrow \mu_{S_1,S_2} & & \downarrow \mu_{S_2,S_1} \\
L(S_1 \cup S_2) & \xleftrightarrow{\mathrm{id}} & L(S_2 \cup S_1)
\end{array}$$

$$\begin{array}{ccc}
L(S) \otimes J & \xrightarrow{\mathrm{id} \otimes \upsilon} & L(S) \otimes L(\varnothing) \\
\downarrow \rho & & \downarrow \mu_{A,\varnothing} \\
L(S) & \xleftrightarrow{\mathrm{id}} & L(S)
\end{array}
\qquad
\begin{array}{ccc}
J \otimes L(S) & \xrightarrow{\upsilon \otimes \mathrm{id}} & L(\varnothing) \otimes L(S) \\
\downarrow \lambda & & \downarrow \mu_{\varnothing,A} \\
L(S) & \xleftrightarrow{\mathrm{id}} & L(S)
\end{array}$$

Equivalence relation

$\simeq_A$ is the smallest equivalence relation on E_A^* such that

- $sabt \simeq_A sbat$ if $a \; I_A \; b$
- $s1_At \simeq_A st$.

Schedulable and trace sets

- A schedulable set (over A) is a set $C \subseteq E_A^*$ closed under $\simeq_A$
- A trace set (over A) is a set B of schedulable sets

Functor $F : \textsc{Comm} \to \textsc{Set}$

- $F(A)$= the set of all trace sets over A
- If $f : A_1 \to A_2$, we define $F(f) : F(A_1) \to F(A_2)$ by $F(f)(B) = \{\{\hat{f}(w) \mid w \in C, \hat{f}(w) \text{ is defined}\} \mid C \in B\}$

Interpreting session types as trace sets

Interpretation $[\![\_]\!]_S$: Types → Set

$$[\![0]\!]_S := \varnothing \qquad [\![\omega]\!]_S := \{\varnothing\}$$
$$[\![G_1 \,\&\, G_2]\!]_S := [\![G_1]\!]_S \Cup [\![G_2]\!]_S \qquad [\![G_1 \oplus G_2]\!]_S := [\![G_1]\!]_S \cup [\![G_2]\!]_S$$
$$[\![C; G]\!]_S := C \cdot_{L(S)} [\![G]\!]_S \qquad [\![\mathrm{close}]\!]_S := \{[\epsilon]_{L(S)}\}$$
$$[\![C]\!]_S := \{[C]_{L(S)}\} \quad \text{otherwise}$$

Theorem: Soundness and completeness of interpretation

- $G_1 \simeq_S G_2$ iff $[\![G_1]\!]_S = [\![G_2]\!]_S$
- $[\![G_1 \;{}^{S_1}\!\vee^{S_2}\; G_2]\!]_{S_1 \cup S_2} = F(\mu)([\![G_1]\!]_{S_1} \otimes [\![G_2]\!]_{S_2})$

As a consequence:

$$G_1 \;{}^{S_1}\!\vee^{S_2}\; G_2 \;\simeq_{S_1 \cup S_2}\; G_2 \;{}^{S_2}\!\vee^{S_1}\; G_1$$
$$G_1 \;{}^{S_1}\!\vee^{S_2 \cup S_3}\; (G_2 \;{}^{S_2}\!\vee^{S_3}\; G_3) \;\simeq_{S_1 \cup S_2 \cup S_3}\; (G_1 \;{}^{S_1}\!\vee^{S_2}\; G_2) \;{}^{S_1 \cup S_2}\!\vee^{S_3}\; G_3$$

Conclusion

- Partial sessions for multiparty session types
- Applicable to open systems (i.e. systems with missing parts)
- Early static deadlock detection
- Semantic interpretation showing soundness of construction
- Proof-of-concept implementation of the merging algorithm
  https://github.com/cstolze/partial-session-types-prototype
- Future work:
  - Decidability result: type-checking, type inference
  - Asynchronous session types
  - Recursion: how to adapt the merging algorithm?
  - Subtyping

$$\begin{aligned} G_1 \mathbin{\&} G_2 &\leqslant_S G_i \\ G_i &\leqslant_S G_1 \oplus G_2 \end{aligned}$$