Probabilistic liveness for multiparty session types

Dylan McDermott

University of Oxford

Funded by Project TaRDIS (Horizon Europe)

dylan@dylanm.org

Aim of this work

(Receive-)liveness: if someone starts waiting to receive a message, then such a message will arrive

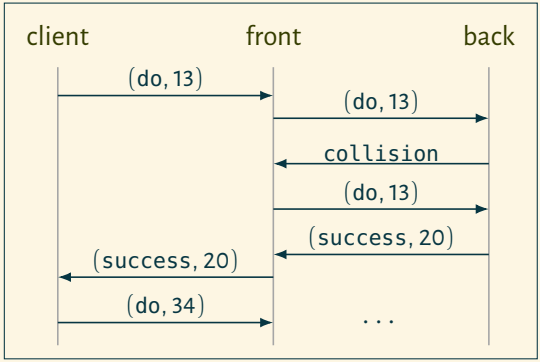
- Problem: many distributed systems are inherently probabilistic, and this definition doesn't work well for them
- Solution: formalize a notion of ***almost-sure liveness***, with tools (MPST framework, model checker) for working with it

Running example: retry on transient error

$$\mathbb{T}_{\text{front}} = \mu Y.\, \text{client}\&\text{do}\langle \textbf{int} \rangle.\, \mu X.\, \text{back} \oplus \text{do}\langle \textbf{int} \rangle.\, \text{back}\& \begin{cases} \texttt{success}\langle \textbf{int} \rangle.\, \text{client} \oplus \texttt{success}\langle \textbf{int} \rangle.\, Y \\ \texttt{collision}.\, X \\ \texttt{error}.\, \text{client} \oplus \texttt{error}.\, Y \end{cases}$$

$$\mathbb{T}_{\text{client}} = \mu Y.\, \text{front} \oplus \text{do}\langle \textbf{int} \rangle.\, \text{front}\& \begin{cases} \texttt{success}\langle \textbf{int} \rangle.\, Y \\ \texttt{error}.\, Y \end{cases}$$

$$\mathbb{T}_{\text{back}} = \mu X.\, \text{front}\&\text{do}\langle \textbf{int} \rangle.\, \text{front} \oplus \begin{cases} \texttt{success}\langle \textbf{int} \rangle.\, X \\ \texttt{collision}.\, X \\ \texttt{error}.\, X \end{cases}$$

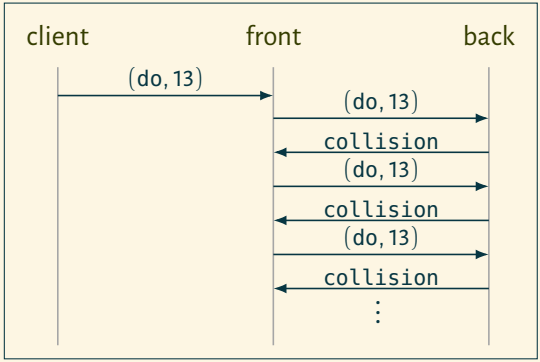


Running example: retry on transient error

$$\mathbb{T}_{\text{front}} = \mu Y.\, \text{client}\&\text{do}\langle\textbf{int}\rangle.\, \mu X.\, \text{back} \oplus \text{do}\langle\textbf{int}\rangle.\, \text{back}\& \begin{cases} \texttt{success}\langle\textbf{int}\rangle.\, \text{client} \oplus \texttt{success}\langle\textbf{int}\rangle.\, Y \\ \texttt{collision}.\, X \\ \texttt{error}.\, \text{client} \oplus \texttt{error}.\, Y \end{cases}$$

$$\mathbb{T}_{\text{client}} = \mu Y.\, \text{front} \oplus \text{do}\langle\textbf{int}\rangle.\, \text{front}\& \begin{cases} \texttt{success}\langle\textbf{int}\rangle.\, Y \\ \texttt{error}.\, Y \end{cases}$$

$$\mathbb{T}_{\text{back}} = \mu X.\, \text{front}\&\text{do}\langle\textbf{int}\rangle.\, \text{front} \oplus \begin{cases} \texttt{success}\langle\textbf{int}\rangle.\, X \\ \texttt{collision}.\, X \\ \texttt{error}.\, X \end{cases}$$



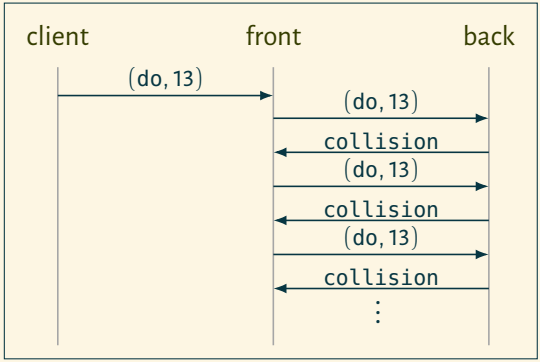
Running example: retry on transient error

$$\mathbb{T}_{\text{front}} = \mu Y.\, \text{client}\&\text{do}\langle\textbf{int}\rangle.\, \mu X.\, \text{back} \oplus \text{do}\langle\textbf{int}\rangle.\, \text{back}\& \begin{cases} \texttt{success}\langle\textbf{int}\rangle.\, \text{client} \oplus \texttt{success}\langle\textbf{int}\rangle.\, Y \\ \texttt{collision}.\, X \\ \texttt{error}.\, \text{client} \oplus \texttt{error}.\, Y \end{cases}$$

$$\mathbb{T}_{\text{client}} = \mu Y.\, \text{front} \oplus \text{do}\langle\textbf{int}\rangle.\, \text{front}\& \begin{cases} \texttt{success}\langle\textbf{int}\rangle.\, Y \\ \texttt{error}.\, Y \end{cases}$$

$$\mathbb{T}_{\text{back}} = \mu X.\, \text{front}\&\text{do}\langle\textbf{int}\rangle.\, \text{front} \oplus \begin{cases} \texttt{success}\langle\textbf{int}\rangle.\, X \\ \texttt{collision}.\, X \\ \texttt{error}.\, X \end{cases}$$

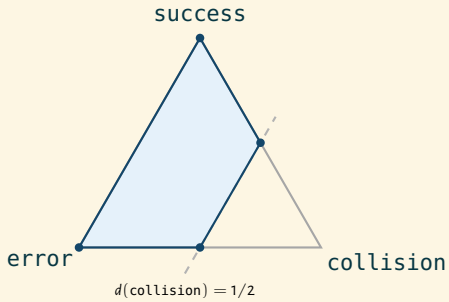
If $\Pr(\texttt{collision}) \leqslant \frac{1}{2}$, this is ***almost-surely*** live



Imprecise probabilities

Instead of having a fixed probability distribution, have a finitely generated convex *set* of distributions

For instance, $\left\{ d \in \mathrm{Dist}\left\{ \begin{matrix} \texttt{success}, \\ \texttt{collision}, \\ \texttt{error} \end{matrix} \right\} \,\middle|\, d(\texttt{collision}) \leqslant 1/2 \right\}$:

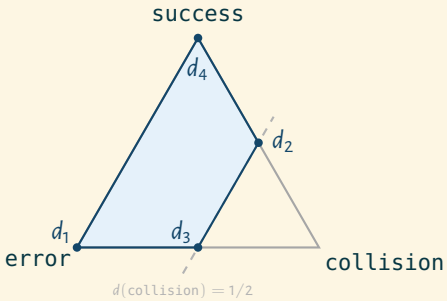


Imprecise probabilities

Instead of having a fixed probability distribution, have a finitely generated convex *set* of distributions

For instance, $\left\{ d \in \mathrm{Dist}\left\{ \begin{matrix} \texttt{success}, \\ \texttt{collision}, \\ \texttt{error} \end{matrix} \right\} \middle| \, d(\texttt{collision}) \leqslant 1/2 \right\}$:

	success	collision	error
d_1	0	0	1
d_2	1/2	1/2	0
d_3	0	1/2	1/2
d_4	1	0	0

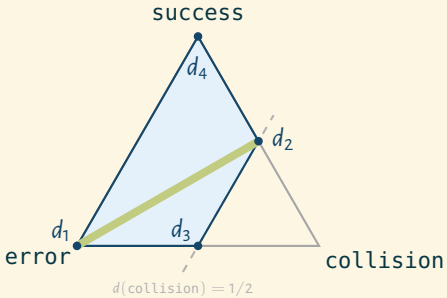


Imprecise probabilities

Combine nondeterministic with probabilistic choice:

$$\textbf{if } x = 0 \textbf{ then } \text{front!}\texttt{error} \\ \textbf{else } (\text{front!}(\texttt{success}, 20) \oplus_{1/2} \text{front!}\texttt{collision})$$

	success	collision	error
d_1	0	0	1
d_2	1/2	1/2	0
d_3	0	1/2	1/2
d_4	1	0	0

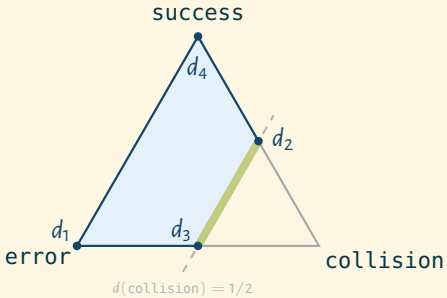


Imprecise probabilities

Combine nondeterministic with probabilistic choice:

$$(\textbf{if } x = 0 \textbf{ then } \text{front}!\texttt{error} \textbf{ else } \text{front}!(\texttt{success}, 20)) \oplus_{1/2} \text{front}!\texttt{collision}$$

	success	collision	error
d_1	0	0	1
d_2	1/2	1/2	0
d_3	0	1/2	1/2
d_4	1	0	0



Multiparty session types with probabilities

Describe protocols followed by message-passing computations:

$$\mathbb{T} ::= X \mid \mu X.\, \mathbb{T} \quad \text{type variables and guarded recursion}$$

$$\mid \mathsf{end} \quad \text{inaction}$$

$$\mid \mathsf{p} \oplus^{\mathcal{C}} \begin{cases} \ell_1\langle \mathsf{b}_1 \rangle.\, \mathbb{T}_1 \\ \vdots \\ \ell_n\langle \mathsf{b}_n \rangle.\, \mathbb{T}_n \end{cases} \quad \text{internal choice, sending a message to } \mathsf{p}$$

$$\mid \mathsf{p}\& \begin{cases} \ell_1\langle \mathsf{b}_1 \rangle.\, \mathbb{T}_1 \\ \vdots \\ \ell_n\langle \mathsf{b}_n \rangle.\, \mathbb{T}_n \end{cases} \quad \text{external choice, receiving a message from } \mathsf{p}$$

where $\mathcal{C}$ is a satisfiable finite conjunction of linear inequalities in variables $\Pr(\ell_i)$

$$r_1 \cdot \Pr(\ell_1) + \cdots + r_n \cdot \Pr(\ell_n) \leqslant r'$$

(these represent finitely generated convex sets of distributions over $\{\ell_1, \ldots, \ell_n\}$)

Multiparty session types with probabilities

$$\mathbb{T}_{\text{front}} = \mu Y.\, \text{client}\&\text{do}\langle\textbf{int}\rangle.\, \mu X.\, \text{back} \oplus \text{do}\langle\textbf{int}\rangle.\, \text{back}\& \begin{cases} \texttt{success}\langle\textbf{int}\rangle.\, \text{client} \oplus \texttt{success}\langle\textbf{int}\rangle.\, Y \\ \texttt{collision}.\, X \\ \texttt{error}.\, \text{client} \oplus \texttt{error}.\, Y \end{cases}$$

$$\mathbb{T}_{\text{client}} = \mu Y.\, \text{front} \oplus \text{do}\langle\textbf{int}\rangle.\, \text{front}\& \begin{cases} \texttt{success}\langle\textbf{int}\rangle.\, Y \\ \texttt{error}.\, Y \end{cases}$$

$$\mathbb{T}_{\text{back}} = \mu X.\, \text{front}\&\text{do}\langle\textbf{int}\rangle.\, \text{front} \oplus^{\mathcal{C}} \begin{cases} \texttt{success}\langle\textbf{int}\rangle.\, X \\ \texttt{collision}.\, X \\ \texttt{error}.\, X \end{cases} \qquad \text{where } \mathcal{C} = \{\Pr(\texttt{collision}) \leqslant 1/2\}$$

(unlabelled = empty conjunction of inequalities = unrestricted choice)

Contexts and reductions

- A *context* Γ assigns a session type $\Gamma(\mathsf{p})$ to each p
- Context *reduction* $\boxed{\Gamma \xrightsquigarrow{\mathsf{p}\to\mathsf{q}:\ell} \Gamma'}$
- For each Γ, p, q, have a set of probability distributions $\mathcal{K}_\Gamma(\mathsf{p},\mathsf{q})$ over labels ℓ

Contexts and reductions

- A *context* Γ assigns a session type $\Gamma(\mathsf{p})$ to each p
- Context *reduction* $\boxed{\Gamma \xrightarrow{\mathsf{p}\to\mathsf{q}:\ell} \Gamma'}$
- For each Γ, p, q, have a set of probability distributions $\mathcal{K}_\Gamma(\mathsf{p},\mathsf{q})$ over labels ℓ

Almost-sure (receive-)liveness of Γ, informally:

For every Γ' reachable from Γ, if $\Gamma'(\mathsf{q})$ is an external choice on p then, with probability 1, a randomly-chosen fair path

$$\varpi = \Gamma' \xrightarrow{\mathsf{p}_1\to\mathsf{q}_1:\ell_1} \Gamma'_1 \xrightarrow{\mathsf{p}_2\to\mathsf{q}_2:\ell_2} \Gamma'_2 \xrightarrow{\mathsf{p}_3\to\mathsf{q}_3:\ell_3} \cdots$$

has $(\mathsf{p}_k, \mathsf{q}_k) = (\mathsf{p}, \mathsf{q})$ for some k

Events and lower probabilities

- An *event* E from Γ is a set of fair paths, all from Γ

  An element of the smallest σ-algebra containing the *cylinder sets*

  $$\mathrm{Cyl}(\pi) = \{\varpi \mid \pi \preccurlyeq \varpi\} \qquad (\pi \text{ a finite path})$$

  Key example: $\boxed{E_{\Gamma,\mathsf{p}\to\mathsf{q}} = \{\varpi \mid \varpi \text{ contains a } \mathsf{p} \to \mathsf{q} \text{ action}\}}$
- A *probability measure* μ assigns a probability to each event E from Γ
- μ is *consistent* if it respects the constraints in the session types
- The *lower probability* of an event E:

$$\underline{\Pr}(E) = \inf\{\mu(E) \mid \mu \text{ is a consistent probability measure}\}$$

Almost-sure liveness and lower probabilities

Almost-sure (receive-)liveness of Γ: for every Γ' reachable from Γ, if $\Gamma'(\mathsf{q})$ is an external choice on q, then

$$\underline{\Pr}(E_{\Gamma',\mathsf{p}\to\mathsf{q}}) = 1$$

where $E_{\Gamma',\mathsf{p}\to\mathsf{q}} = \{\varpi \mid \varpi \text{ contains a } \mathsf{p}\to\mathsf{q} \text{ action}\}$

A logic of lower probabilities

Aim: describe a logic in which

- we express context properties (e.g. deadlock-freedom, *almost-sure liveness*) as formulae Ψ,
- we can decide whether $\Gamma \in [\![\Psi]\!]$.

This gives a bottom-up approach to probabilistic MPST

Less Is More: Multiparty Session Types Revisited

ALCESTE SCALAS, Imperial College London, UK
NOBUKO YOSHIDA, Imperial College London, UK

A logic of lower probabilities

$$\text{Path formula } \phi ::= \Psi \mid \phi \vee \phi' \mid \phi \wedge \phi' \mid \mathbf{X}_{p \to q:\ell}\phi \mid \phi \mathbf{U} \phi' \qquad \text{(neXt, Until)}$$

$$\text{State formula } \Psi ::= \top \mid \Psi \wedge \Psi' \mid \neg\Psi \mid [p \to q : \ell]\Psi \mid Z \mid \nu Z.\, \Psi \mid r \leqslant \underline{\Pr}(\phi)$$

Formula interpretation (for closed formulas):

$[\![\phi]\!]_\Gamma$ is an event from Γ $\qquad$ $[\![\Psi]\!]$ is a set of typing contexts

"Eventually p sends to q" as a path formula:

$$\phi_{p \to q} = \top \mathbf{U} \left(\bigvee_{\ell \in \mathcal{L}} (\mathbf{X}_{(p \to q:\ell)} \top) \right) \qquad [\![\phi_{p \to q}]\!]_\Gamma = E_{\Gamma, p \to q}$$

Computing the lower probability of a path formula

Key fact: lower probabilities of path formulae are the least solution to

$$\underline{\Pr}([\![\phi]\!]_\Gamma) = \sum_\ell d(\ell) \cdot \underline{\Pr}([\![\phi'_\ell]\!]_{\Gamma'_\ell}) \quad \text{for all } p, q \text{ with } \mathcal{K}_\Gamma(p,q) \neq \emptyset$$

where
$$\begin{cases} \Gamma \overset{p \to q:\ell}{\rightsquigarrow} \Gamma'_\ell \\ \phi \overset{p \to q:\ell}{\longmapsto} \phi'_\ell \quad (\text{“formula progression”}) \\ d \in \mathcal{K}_\Gamma(p,q) \text{ is a generator} \end{cases}$$

Computing the lower probability of a path formula

Key fact: lower probabilities of path formulae are the least solution to

$$\underline{\Pr}(\llbracket \phi \rrbracket_\Gamma) = \sum_\ell d(\ell) \cdot \underline{\Pr}(\llbracket \phi'_\ell \rrbracket_{\Gamma'_\ell}) \quad \text{for all } p, q \text{ with } \mathcal{K}_\Gamma(p,q) \neq \emptyset$$

$$\text{where } \begin{cases} \Gamma \overset{p \to q:\ell}{\rightsquigarrow} \Gamma'_\ell \\ \phi \overset{p \to q:\ell}{\longmapsto} \phi'_\ell \quad (\text{“formula progression”}) \\ d \in \mathcal{K}_\Gamma(p,q) \text{ is a generator} \end{cases}$$

For our running example (with $\Gamma \overset{\text{client}\to\text{front:do}}{\rightsquigarrow} \Gamma'$), the lower probability $\alpha = \underline{\Pr}\llbracket \phi_{p\to q} \rrbracket_{\Gamma'}$ minimizes

$$\alpha = d(\texttt{success}) + d(\texttt{collision}) \cdot \alpha + d(\texttt{error})$$

where d is one of the 4 generating distributions of $d(\texttt{collision}) \leqslant 1/2$.

Computing the lower probability of a path formula

Key fact: lower probabilities of path formulae are the least solution to

$$\underline{\Pr}([\![\phi]\!]_\Gamma) = \sum_\ell d(\ell) \cdot \underline{\Pr}([\![\phi'_\ell]\!]_{\Gamma'_\ell}) \quad \text{for all } p, q \text{ with } \mathcal{K}_\Gamma(p,q) \neq \emptyset$$

$$\text{where } \begin{cases} \Gamma \overset{p \to q:\ell}{\rightsquigarrow} \Gamma'_\ell \\ \phi \overset{p \to q:\ell}{\longmapsto} \phi'_\ell \quad (\text{“formula progression”}) \\ d \in \mathcal{K}_\Gamma(p,q) \text{ is a generator} \end{cases}$$

This gives us a way to:

- ***compute*** the lower probability $\Pr([\![\phi]\!]_\Gamma)$,
- hence to ***decide*** whether $\Gamma \in [\![\Psi]\!]$.

Conclusions

- Liveness is sometimes too strong, almost-sure liveness is good enough
- Almost-sure liveness, and other probabilistic context properties, are decidable
- Probabilistic session types naturally involve imprecise probabilities, especially finitely generated convex sets

Work in progress: implementation, schedulers

Assigning session types to processes

$$P ::= \mathbf{0} \mid \mathsf{p}!\ell\langle e\rangle.\, P \mid \sum_{i\in I} \mathsf{p}?\ell_i\langle x_i\rangle.\, P_i \mid \textbf{if } e \textbf{ then } P_1 \textbf{ else } P_2 \mid P_1 \oplus_r P_2$$

$$\frac{\Theta \vdash P_1 : \mathbb{T}_1 \qquad \Theta \vdash P_2 : \mathbb{T}_2 \qquad \mathbb{T}_1 \oplus_r \mathbb{T}_2 <: \mathbb{U}}{\Theta \vdash P_1 \oplus_r P_2 : \mathbb{U}}$$

Formula progression

Analogous to the work of Kabanza et al.[1][2]

$$\frac{\vDash \Phi}{\Phi \overset{\beta}{\longmapsto} \top} \qquad \frac{\nvDash \Phi}{\Phi \overset{\beta}{\longmapsto} \bot}$$

$$\frac{\beta \neq \beta' \qquad \beta \not\asymp \beta'}{\mathbf{X}_{\beta'}\phi_0 \overset{\beta}{\longmapsto} \bot} \qquad \frac{\beta \smile \beta' \qquad \phi_0 \overset{\beta}{\longmapsto} \phi_1}{\mathbf{X}_{\beta'}\phi_0 \overset{\beta}{\longmapsto} \mathbf{X}_{\beta'}\phi_1} \qquad \frac{\beta = \beta'}{\mathbf{X}_{\beta'}\phi_0 \overset{\beta}{\longmapsto} \phi_0}$$

$$\frac{\phi_0 \overset{\beta}{\longmapsto} \phi_1 \qquad \psi_0 \overset{\beta}{\longmapsto} \psi_1}{\phi_0 \mathbf{U} \psi_0 \overset{\beta}{\longmapsto} \psi_1 \vee (\phi_1 \wedge \phi_0 \mathbf{U} \psi_0)}$$

[1] Fahiem Bacchus and Froduald Kabanza, 1996. Using Temporal Logic to Control Search in a Forward Chaining Planner.

[2] Froduald Kabanza and Sylvie Thiebaux, 2005. Search Control in Planning for Temporally Extended Goals.