A Game-Theoretic Approach to Concurrent Separation Logic

Paul-André Melliès[1] **Léo Stefanesco**[2]

November 28, 2016

[1] IRIF/CNRS/Université Paris Diderot

[2] ENS Lyon

Syntax

$$
\begin{aligned}
B &::= \mathbf{true} \mid \mathbf{false} \mid B \wedge B \mid B \vee B \mid E = E' \\
E &::= 0 \mid 1 \mid \cdots \mid x \mid E + E \mid E - E \mid E * E \\
C &::= x \coloneqq E \mid x \coloneqq [y] \mid [x] \coloneqq [y] \\
&\qquad \mid \texttt{while}\ B\ \texttt{do}\ C \mid \texttt{if}\ B\ \texttt{then}\ C\ \texttt{else}\ C \\
&\qquad \mid \texttt{resource}\ r\ \texttt{do}\ C \mid \texttt{with}\ r\ \texttt{when}\ B\ \texttt{do}\ C \\
&\qquad \mid x \coloneqq \texttt{alloc}(E) \mid \texttt{dispose}(E) \\
&\qquad \mid C; C \mid C \parallel C
\end{aligned}
$$

Separation Logic

Predicates

$$P, Q, R, J ::= B \mid P \vee Q \mid P \wedge Q \mid \neg P \mid P \Rightarrow Q \mid \forall X.P \mid \mathrm{Own}_p(x)$$
$$\mid \exists X.P \mid \mathbf{emp} \mid E_1 \overset{p}{\mapsto} E_2 \mid P * Q$$

Semantics

$$\begin{aligned}
\sigma \vDash E_1 = E_2 &\iff [\![E_1]\!](\sigma) = [\![E_2]\!](\sigma) \wedge \mathrm{fv}(E_1 = E_2) \subseteq \mathrm{dom}(\sigma) \\
\sigma \vDash P \wedge Q &\iff \sigma \vDash P \text{ et } \sigma \vDash Q \\
\sigma \vDash P_1 * P_2 &\iff \exists \sigma_1, \sigma_2, \sigma = \sigma_1 \uplus \sigma_2 \wedge \sigma_1 \vDash P_1 \wedge \sigma_2 \vDash P_2 \\
\sigma \vDash \mathrm{Own}_p(x) &\iff \exists v \in \mathbf{Val}, \sigma(x) = (v, p)
\end{aligned}$$

Judgments

$$\Gamma \vdash \{P\}\, C\, \{Q\}$$

Contexts

$$\Gamma ::= () \mid \Gamma, r : J$$

Inference rules

$$\frac{\Gamma \vdash \{P_1\}\, C_1\, \{Q_1\} \quad \Gamma \vdash \{P_2\}\, C_2\, \{Q_2\}}{\Gamma \vdash \{P_1 * P_2\}\, C_1 \parallel C_2\, \{Q_1 * Q_2\}}\ \textsc{Par}$$

$$\frac{\Gamma, r : J \vdash \{P\}\, C\, \{Q\}}{\Gamma \vdash \{P * J\}\, \texttt{resource}\ r\ \texttt{do}\ C\, \{Q * J\}}\ \textsc{Res}$$

$$\frac{\Gamma \vdash \{(P * J) \wedge B\}\, C\, \{Q * J\}}{\Gamma, r : J \vdash \{P\}\, \texttt{with}\ r\ \texttt{when}\ B\ \texttt{do}\ C\, \{Q\}}\ \textsc{When}$$

Example: Producer – Consumer

$POP(n) : \texttt{with } r_n \texttt{ when } n > 0$

$\qquad \texttt{do } n{-}{-};\ f \coloneqq f.1$

$ADD(n) : \texttt{with } r_n \texttt{ when } \mathbf{true}$

$\qquad \texttt{do } n{+}{+};\ b \coloneqq b.1$

$$r_n : \{\mathrm{Own}_{0.5}(f) * \mathrm{Own}_{0.5}(b) * \mathrm{Own}_{\top}(n_S) * \mathrm{listseg}\, n\ f\ b\}$$

$$\{\mathrm{Own}_{\top}(tc) * \mathrm{Own}_{\top}(front) * \mathrm{Own}_{0.5}(f) * \mathrm{Own}_{\top}(n_P) * (front = f \wedge \mathbf{emp})\}$$

$\texttt{while } \mathbf{true} \texttt{ do}$

$\qquad tc \coloneqq front;$

$\qquad POP(n);$

$\qquad front \coloneqq front.1$

$\qquad \mathrm{consume}(tc.0)$

$\qquad \mathrm{dispose}(tc)$

$\|$

$$\{\mathrm{Own}_{\top}(tp) * \mathrm{Own}_{\top}(back) * \mathrm{Own}_{0.5}(b) * \mathrm{Own}_{\top}(n_V) * (back = b \wedge back \mapsto \_, \_)\}$$

$\texttt{while } \mathbf{true} \texttt{ do}$

$\qquad back.0 \coloneqq \mathrm{produce}()$

$\qquad tp \coloneqq \mathrm{cons}()$

$\qquad back.1 \coloneqq tp;$

$\qquad ADD(n);$

$\qquad back \coloneqq tp$

Traces

The semantics of a command C is a transition system $[\![C]\!]$ of traces of the form:

$$\mathfrak{s}_1 \xrightarrow{env} \mathfrak{s}_2 \xrightarrow{m_1} \mathfrak{s}_3 \xrightarrow{env} \mathfrak{s}_4 \xrightarrow{m_2} \mathfrak{s}_5 \xrightarrow{env} \cdots \xrightarrow{env} \mathfrak{s}_{2p} \xrightarrow{m_p} \mathfrak{s}_{2p+1} \xrightarrow{env} \mathfrak{s}_{2p+2}$$

where $\mathfrak{s} = (\sigma, L)$ and m_i are instructions:

- $x \coloneqq E$
- $x \coloneqq [y]$
- $[x] \coloneqq y$
- `nop`
- $P(r)$
- $V(r)$
- $x \coloneqq$ `alloc`(E)
- `dispose`(E)

Transition System

$\mathbf{T} = (T, \left| T \right|)$ with $T \subseteq \mathbf{Traces}$ and $\left| T \right| \subseteq T$, such that T is closed by odd prefixes.

Sequential Composition

$$
\begin{aligned}
[\![C]\!];[\![C']\!] &= [\![C]\!] \cup \{t \cdot t' \mid t \in \left|[\![C]\!]\right|, t' \in [\![C']\!] \text{ and } \partial_1 t = \partial_0 t'\} \\
\left|[\![C]\!];[\![C']\!]\right| &= \{t \cdot t' \mid t \in \left|[\![C]\!]\right|, t' \in |[\![C']\!]| \text{ and } \partial_1 t = \partial_0 t'\}
\end{aligned}
$$

Parallel composition

$[\![C \parallel C']\!]$ is the set of interleavings of the traces $[\![C]\!]$ and of $[\![C']\!]$

Toward the Soundness Theorem

- The graph of separated states
- Separation Games
- Strategies, Winning Strategies
- Soundness Theorem

The Graph of Separated States

Definition (The Graph of Separated States)

Its *nodes* are all the tuples

$$(\sigma_C, \boldsymbol{\sigma}, \sigma_F) \in \mathbb{S} \times (\mathbf{LockName} \to \mathbb{S} + \{C, F\}) \times \mathbb{S}$$

such that

$$\sigma_C * \left\{ \bigcircledast_{r \in \mathrm{dom}(\boldsymbol{\sigma})} \boldsymbol{\sigma}(r) \right\} * \sigma_F$$

is well defined, and it has two kinds of edges:

- the Eve edges: $(\sigma_C, \boldsymbol{\sigma}, \sigma_F) \xrightarrow{r:\sigma_U} (\sigma'_C, \boldsymbol{\sigma} \uplus [r \mapsto \sigma_U], \sigma_F)$,
- the Adam edges $(\sigma_C, \boldsymbol{\sigma}, \sigma_F) \xrightarrow{r:\sigma_L} (\sigma_C * \sigma_L, \boldsymbol{\sigma}', \sigma'_F)$.

The Game Induced by $\Gamma \vdash \{P\}\, C\, \{Q\}$ and $t \in [\![C]\!]$

$$\mathfrak{s}_1 \xrightarrow{\quad env \quad} \mathfrak{s}_2 \xrightarrow{\quad m_1 \quad} \mathfrak{s}_3 \xrightarrow{\quad env \quad} \mathfrak{s}_3 \xrightarrow{\quad m_2 \quad} \mathfrak{s}_5$$

The Game Induced by $\Gamma \vdash \{P\}\, C\, \{Q\}$ and $t \in [\![C]\!]$

$$\begin{tikzcd}
P \arrow[d, dash, dashed] & & & & Q \arrow[d, dash, dashed] \\
\mathfrak{s}_1 \arrow[r, "env"] & \mathfrak{s}_2 \arrow[r, "m_1"] & \mathfrak{s}_3 \arrow[r, "env"] & \mathfrak{s}_3 \arrow[r, "m_2"] & \mathfrak{s}_5
\end{tikzcd}$$

The Game Induced by $\Gamma \vdash \{P\}\, C\, \{Q\}$ and $t \in [\![C]\!]$

$$
\begin{tikzcd}
\{P, \Gamma, \mathbf{true}\} \arrow[r, "r_1 : J_1"] \arrow[d, dash, dashed] & \{\mathbf{true}, \Gamma, \mathbf{true}\} \arrow[r, "r_2 : J_1'"] \arrow[d, dash, dashed] & \{\mathbf{true}, \Gamma, \mathbf{true}\} \arrow[r, "r_3 : J_2"] \arrow[d, dash, dashed] & \{\mathbf{true}, \Gamma, \mathbf{true}\} \arrow[r, "r_4 : J_2'"] \arrow[d, dash, dashed] & \{Q, \Gamma, \mathbf{true}\} \arrow[d, dash, dashed] \\
\mathfrak{s}_1 \arrow[r, "env"] & \mathfrak{s}_2 \arrow[r, "m_1"] & \mathfrak{s}_3 \arrow[r, "env"] & \mathfrak{s}_3 \arrow[r, "m_2"] & \mathfrak{s}_5
\end{tikzcd}
$$

with $J_i = \Gamma(\mathrm{lock}^+(m_i))$ and $J_i' = \Gamma(\mathrm{lock}^-(m_i))$.

$$
\begin{tikzcd}
(\sigma_C^1, \boldsymbol{\sigma}^1, \sigma_F^1) \arrow[r, "r_1 : \sigma_L"] \arrow[d, dash, dashed] & (\sigma_C^1 * \sigma_L, \boldsymbol{\sigma}^2, \sigma_F^2) \arrow[r, "r_2 : \sigma_U"] \arrow[d, dash, dashed] & (\sigma_C^2, \boldsymbol{\sigma}^3, \sigma_F^2) \arrow[r, "r_3 : \sigma'_L"] \arrow[d, dash, dashed] & (\sigma_C^2 * \sigma'_L, \boldsymbol{\sigma}^4, \sigma_F^3) \arrow[r, "r_4 \sigma'_U"] \arrow[d, dash, dashed] & (\sigma_C^3, \boldsymbol{\sigma}^5, \sigma_F^3) \arrow[d, dash, dashed] \\
\{P, \Gamma, \mathbf{true}\} \arrow[r, "r_1 : J_1"] \arrow[d, dash, dashed] & \{\mathbf{true}, \Gamma, \mathbf{true}\} \arrow[r, "r_2 : J'_1"] \arrow[d, dash, dashed] & \{\mathbf{true}, \Gamma, \mathbf{true}\} \arrow[r, "r_3 : J_2"] \arrow[d, dash, dashed] & \{\mathbf{true}, \Gamma, \mathbf{true}\} \arrow[r, "r_4 : J'_2"] \arrow[d, dash, dashed] & \{Q, \Gamma, \mathbf{true}\} \arrow[d, dash, dashed] \\
\mathfrak{s}_1 \arrow[r, "env"] & \mathfrak{s}_2 \arrow[r, "m_1"] & \mathfrak{s}_3 \arrow[r, "env"] & \mathfrak{s}_3 \arrow[r, "m_2"] & \mathfrak{s}_5
\end{tikzcd}
$$

with $J_i = \Gamma(\mathrm{lock}^+(m_i))$ and $J'_i = \Gamma(\mathrm{lock}^-(m_i))$.

Theorem (Soundness)

Suppose that $\Gamma \vdash \{P\}\, C\, \{Q\}$ and let $t \in [\![C]\!]$.

Then there exists a winning strategy for the induced game.

Corollary

Suppose $\varnothing \vdash \{P\}\, C\, \{Q\}$, let $t \in \left|[\![C]\!]\right|$ a trace where the Environment is *silent*.

$$\mathfrak{s}_1 \xrightarrow{env} \mathfrak{s}_1 \xrightarrow{m_1} \mathfrak{s}_2 \xrightarrow{env} \mathfrak{s}_2 \xrightarrow{m_2} \cdots \xrightarrow{m_{k-1}} \mathfrak{s}_k \xrightarrow{env} \mathfrak{s}_k$$

Then $\mathfrak{s}_1 \vDash P * \mathbf{true} \Rightarrow \mathfrak{s}_k \vDash Q * \mathbf{true}$.

Data Races & True Concurrency

- We can add a partial order $\leq$ on instructions in traces (*program-order* + *synchronizes-with*).
- If $m_1 \#_t m_2$ in a trace t, we can define a commuting tile as:

$$
\begin{tikzcd}
t : & & \mathfrak{s}_2 \arrow[r, "env"] & \mathfrak{s}_3 \arrow[r, "m_{i+1}"] & \mathfrak{s}_4 \arrow[dr, "env"] & & \\
\cdots \arrow[r] & \mathfrak{s}_1 \arrow[ur, "m_i"] \arrow[dr, "m_{i+1}"'] & & & & \mathfrak{s}_5 \arrow[r] & \cdots \\
t' : & & \mathfrak{s}'_2 \arrow[r, "env"] & \mathfrak{s}'_3 \arrow[r, "m_i"] & \mathfrak{s}'_4 \arrow[ur, "env"] & &
\end{tikzcd}
$$

- 2 problems:
  - Environnement can add *synchronizes-with* dependencies;
  - we must constraint on the Environnement transitions in t'.

Resource Tracking

Adam's edges become

$$(\sigma_C, \boldsymbol{\sigma}, \sigma_F) \xrightarrow[\rho]{r:\sigma_L} (\sigma_C * \sigma_L, \boldsymbol{\sigma}', \sigma_F')$$

where $\rho \subseteq \mathbf{LockName}$ contains the locks touched by Env:

$$\forall r \in \mathbf{LockName} \setminus \rho,\ \boldsymbol{\sigma}'(r) = \boldsymbol{\sigma}(r).$$

Definition (Concurrent Instructions)

$$\begin{tikzcd}
p: & n_1 \arrow[r, "r_1 : \sigma_{U1}"] \arrow[d, dashed, no head] & n_2 \arrow[r, "r_2 : \sigma_L", "\rho"'] \arrow[d, dashed, no head] & n_3 \arrow[r, "r3 : \sigma_{U2}"] \arrow[d, dashed, no head] & n_4 \arrow[d, dashed, no head] \\
t: & \mathfrak{s}_1 \arrow[r, "m_1"] & \mathfrak{s}_2 \arrow[r, "env"] & \mathfrak{s}_3 \arrow[r, "m_2"] & \mathfrak{s}_4
\end{tikzcd}$$

$$m_1 \#_p m_2 \quad \Leftrightarrow \quad m_1 \#_t m_2 \ \wedge\ (\rho \ \cap\ (\mathrm{lock}(m_1) \cup \mathrm{lock}(m_2)) = \varnothing)$$

Commuting Tile

$$
\begin{tikzcd}
 & & (\sigma_C^3, \boldsymbol{\sigma}^3, \sigma_F^2) \arrow[dr, "r_3 : \sigma_{L2}", "\rho_2"'] & & \\
 & (\sigma_C^2, \boldsymbol{\sigma}^2, \sigma_F^2) \arrow[ur, "r_2 : \sigma_{U1}"] & & (\sigma_C^4, \boldsymbol{\sigma}^4, \sigma_F^3) \arrow[dr, "r_4 : \sigma_{U2}"] & \\
(\sigma_C^1, \boldsymbol{\sigma}^1, \sigma_F^1) \arrow[ur, "r_1 : \sigma_{L1}", "\rho_1"'] \arrow[dr, "\rho_1", "r_3 : \sigma_{L2}"'] & & & & (\sigma_C^5, \boldsymbol{\sigma}^5, \sigma_F^3) \\
 & (\sigma_C'^2, \boldsymbol{\sigma}'^2, \sigma_F^2) \arrow[dr, "r_4 : \sigma_{U2}"'] & & (\sigma_C'^4, \boldsymbol{\sigma}'^4, \sigma_F^3) \arrow[ur, "r_2 : \sigma_{U1}"'] & \\
 & & (\sigma_C'^3, \boldsymbol{\sigma}'^3, \sigma_F^2) \arrow[ur, "\rho_2", "r_1 : \sigma_{L1}"'] & &
\end{tikzcd}
$$

$$\begin{tikzcd}
 & \bullet \arrow[rr] & & \bullet \arrow[dr] & \\
\bullet \arrow[ur] \arrow[rr] \arrow[dr] & & \bullet \arrow[ur] \arrow[dr] & & \bullet \\
 & \bullet \arrow[rr] & & \bullet \arrow[ur] &
\end{tikzcd}$$

$$\Updownarrow$$

$$\begin{tikzcd}
 & \bullet \arrow[rr] \arrow[dr] & & \bullet \arrow[dr] & \\
\bullet \arrow[ur] \arrow[dr] & & \bullet \arrow[rr] & & \bullet \\
 & \bullet \arrow[rr] \arrow[ur] & & \bullet \arrow[ur] &
\end{tikzcd}$$

Merci !