Separation logic preserves the expressive power of classical logic

Short Presentation

Lozes Etienne LIP - ENS Lyon
46,allée d'Italie - 69364 Lyon - FRANCE
elozes@ens-lyon.fr

ABSTRACT

This paper compares separation logic to a classical fragment of it. We prove that they are equally expressive, and that the separative power is obtained using only monotonic assertions.

1. INTRODUCTION

Imperative programming languages manipulating pointers allow one to change the value a variable refers to without explictly mentioning this variable. Such multiple accesses to data make the axiomatic semantics [3] of these programs difficult to handle using classical logic as an assertion language [5]. Separation logic [6] is a proposal for an extension of the assertion language that nicely handles the subtleties of pointer manipulation. It provides two new connectives: a separative conjunction $P * Q$ asserting that P and Q hold in separate parts of the memory, and a separating implication $P \twoheadrightarrow Q$ allowing one to introduce 'spatial hypotheses' about the memory. In [6], the example proof of an in-place reversal of a list turns out to require complex invariants in the standard classical logic, whereas it has a simple formulation in separation logic.

So separation logic offers more concise and meaningful assertions than classical logic. We may raise the question whether it also provides new assertions, that is assertions that cannot be formulated in classical logic. For several examples, classical logic provides a formulation of any given invariant, although usually through costly and poorly scalable methods, as the list reversal example shows. In other words, separation logic should have the same expressive power as classical logic. Our aim in this work is to give a formal account of this intuition, at least for a simple though significant assertion language.

We consider the spatial assertion language presented in [6], but we exclude some features, such as recursion, quantification over values, and expressions with lookup in order to keep the proof simple. We define a *classical* fragment excluding the connectives $*$ and $\twoheadrightarrow$ and prove it to be as expressive as the whole language. The proof relies on the use of an intensional model equivalence along the lines of [2, 4]

In Section 2 we collect all definitions of the assertion languages; Section 3 defines the intensional equivalence and discusses its structural properties; Section 4 gives the essential facts to establish the translation from separation logic in our classical fragment, and Section 5 gives some concluding remarks.

SPACE '04 Venezia, Italy

2. DEFINITIONS

We consider the assertion language presented in [6], but we exclude the universal quantifier, recursion and lookup.

We assume a countable set Var of variables, ranged over with x, y, and a set Loc of locations such that $\mathrm{Loc} \subseteq \mathbb{N}$. Expressions and assertions are defined by the following grammar:

$$\begin{array}{rcl} e & ::= & x + n \mid \mathsf{nil} \\ P & ::= & (e \mapsto e) \mid e = e \mid \mathsf{emp} \mid \bot \mid P \Rightarrow P \\ & & \mid P * P \mid P \twoheadrightarrow P \end{array}$$

We write $\mathsf{v}(P)$ for the set of variables occuring in P. Assertions express properties of memory states, modelled as a pair consisting of a store and a heap, as follows:

$$\begin{array}{rcl} \mathrm{Val} & \stackrel{\mathrm{def}}{=} & \mathrm{Loc} \sqcup \{\mathsf{nil}\} \\ \mathrm{Store} & \stackrel{\mathrm{def}}{=} & \mathrm{Var} \to \mathrm{Val} \\ \mathrm{Heap} & \stackrel{\mathrm{def}}{=} & \mathrm{Loc} \rightharpoonup_{fin} \mathrm{Val} \\ \mathrm{State} & \stackrel{\mathrm{def}}{=} & \mathrm{Stack} \times \mathrm{Heap} \end{array}$$

where $\rightharpoonup_{fin}$ stands for a partial function with finite domain. We range over stores with s, over heaps with h, and over states with σ. We note $\sigma_1 \perp \sigma_2$ for $s_1 = s_2$ and $dom(h_1) \cap dom(h_2) = \emptyset$, and, when this holds, $\sigma_1 * \sigma_2$ is the state defined by keeping the same store and by setting $h_1 * h_2(x) = h_1(x)$ or $h_2(x)$. We note $\sigma \leq \sigma'$ the partial order defined if there is a state σ_1 such that $\sigma * \sigma_1 = \sigma'$, and $\sigma_1 \wedge \sigma_2$ for the meet of σ_1, σ_2.

We note $[\![e]\!]\sigma$ for the evaluation of e in σ, that is $[\![x+n]\!]\sigma = s(x) + n$ and $[\![\mathsf{nil}]\!]\sigma = \mathsf{nil}$. The condition for a state σ to match an assertion P, written $\sigma \models P$, is defined inductively by:

$$
\begin{array}{lll}
\sigma \models \bot & \text{never} & \\
\sigma \models (e \mapsto e') & \text{iff} & \mathit{dom}(h) = \{[\![e]\!]\sigma\} \text{ and} \\
 & & h([\![e]\!]\sigma) = [\![e']\!]\sigma \\
\sigma \models e_1 = e_2 & \text{iff} & [\![e_1]\!]\sigma = [\![e_2]\!]\sigma \\
\sigma \models \mathsf{emp} & \text{iff} & \mathit{dom}(h) = \emptyset \\
\sigma \models P_1 \Rightarrow P_2 & \text{iff} & \sigma \models P_1 \text{ implies } \sigma \models P_2 \\
\sigma \models P_1 * P_2 & \text{iff} & \text{there exist } \sigma_1 \text{ and } \sigma_2 \text{ such that} \\
 & & \sigma = \sigma_1 * \sigma_2;\ \sigma_1 \models P_1 \text{ and } \sigma_2 \models P_2 \\
\sigma \models P_1 -\!\!* P_2 & \text{iff} & \text{for all } \sigma_1 \text{ such that } \sigma \bot \sigma_1, \\
 & & \sigma_1 \models P_1 \text{ implies } \sigma * \sigma_1 \models P_2
\end{array}
$$

We may define as usual the connectives $\wedge, \vee, \top, \neg, \Leftrightarrow$ in the obvious way. We also introduce two *monotonic*[1] assertions:

$$
\begin{array}{rcl}
(e \hookrightarrow e') & \stackrel{\text{def}}{=} & (e \mapsto e') * \top \\
\mathsf{size} \geq n & \stackrel{\text{def}}{=} & \underbrace{\neg\,\mathsf{emp} * \ldots * \neg\,\mathsf{emp}}_{n \text{ times}}
\end{array}
$$

In the remainder, we take these as primitive, which allows us to encode $e \mapsto e'$ and emp assertions by boolean combinations (on the contrary, it is not possible to encode $(e \hookrightarrow e')$ and $\mathsf{size} \geq n$ from $e \mapsto e'$ and emp using boolean combinations – this point is also discussed in conclusion). We call *classical fragment* the set of assertions given by the following grammar:

$$P \ ::= \ P \Rightarrow P \ \mid \ \bot \ \mid (e \hookrightarrow e') \ \mid \ e_1 = e_2 \ \mid \ \mathsf{size} \geq n\,.$$

We will note $w(P)$ for the maximal n such that $\mathsf{size} \geq n$ is a subassertion of P, and $E(P)$ for the set of subexpressions of P. We call *pointing assertion* an assertion of the form $e \hookrightarrow e'$, *equality assertion* one of the form $e_1 = e_2$, and *size assertion* one of the form $\mathsf{size} \geq n$; any such assertion is said to be *atomic*.

Our main result is the following:

THEOREM 2.1. *For all assertion P, there exists a classical assertion P' such that $\models P \Leftrightarrow P'$.*

At the same time, we also prove the following result: the monotonic (indeed atomic) fragment is as separative as the whole language, that is if two states satisfy the same monotonic assertions, then they satisfy the same assertions.

3. INTENSIONAL EQUIVALENCE

Before presenting the definition of the intensional equivalence, some interesting properties of atomic classical assertions are worth mentioning:

- Pointing assertions and equality assertions are *stable*, that is if $\sigma, \sigma' \models P$ and there is τ s.t. $\tau \geq \sigma, \sigma'$, then $\sigma \cap \sigma' \models P$. We hence may define, given a statisfiable theory Φ of pointing or equality assertions and a model σ, the minimal submodel σ_Φ such that $\sigma_\Phi \leq \sigma$ and $\sigma_\Phi \models \Phi$.
- Pointing assertions are *local*, that is $\sigma * \sigma' \models P$ implies $\sigma \models P$ or $\sigma' \models P$.
- Equality assertions are *global*, that is $\sigma \models P$ implies $\sigma' \models P$ when $\sigma' \leq \sigma$ or $\sigma' \geq \sigma$. In other words, equality assertions depend only on the store.

[1]or *intuitionistic*, using the terminology of [6], that is assertions P such that $\sigma \models P$ implies $\sigma' \models P$ for all $\sigma' \geq \sigma$.

Note that the last two points would fail when considering expressions with lookup.

The encoding is based on a notion of equivalence between states that is reminiscent of intensional bisimilarity in the context of process algebras [2]. Let E be a finite set of expressions, and w and integer. We say that two states σ and σ' are intensionally equivalent for E, w, written $\sigma \approx_{E,w} \sigma'$, if for all classical assertion P with $E(P) \subseteq E$ and $w(P) \leq w$ $\sigma \models P \ \Leftrightarrow \ \sigma' \models P$.

Remarks:

- This definition amounts to say that σ and σ' satisfy the same atomic classical assertions P with $E(P) \subseteq E$ and $w(P) \leq w$.
- Let us write $w(\sigma) = \sharp \mathit{dom}(h)$. Given three natural numbers a, b, w, we write $a =_w b$ if either $a = b$ or $a, b \geq w$. Then $\sigma \approx_{E,w} \sigma'$ iff $w(\sigma) =_w w(\sigma')$ and σ, σ' satisfy the same pointing and equality assertions with expressions in E.

We now state some structural properties of $\approx_{E,w}$:

LEMMA 3.1 (COMPOSITION). *For all $\sigma_1, \sigma_2, \sigma_1', \sigma_2'$ such that $\sigma_1 \bot \sigma_2$ and $\sigma_1' \bot \sigma_2'$, if $\sigma_1 \approx_{E,w} \sigma_1'$ and $\sigma_2 \approx_{E,w} \sigma_2'$ then $\sigma_1 * \sigma_2 \approx_{E,w} \sigma_1' * \sigma_2'$.*

PROOF. The theory of equality assertions of $\sigma_1 * \sigma_2$ is the same as for σ_1 and σ_2, and the theory of pointing assertions of $\sigma_1 * \sigma_2$ is the union of the theories for σ_1 and σ_2. As for size assertions, we have $w_1 =_w w_2$, $w_1' =_w w_2'$ implies $w_1 + w_2 =_w w_1' + w_2'$. □

LEMMA 3.2 (ORTHOGONALISATION). *For all $\sigma_1, \sigma_2, \sigma_1'$ such that $\sigma_1 \bot \sigma_2$, and $\sigma_1 \approx_{E,w} \sigma_1'$, there is σ_2' such that $\sigma_2 \approx_{E,w} \sigma_2'$ and $\sigma_1' \bot \sigma_2'$.*

PROOF. Since $\sigma_1 \approx_{E,w} \sigma_1'$, for all variables $x, y \in \mathsf{v}(E)$, it holds that $s(x) = s(y)$ iff $s'(x) = s'(y)$. Let $\psi : s(\mathsf{v}(E)) \rightarrow s'(\mathsf{v}(E))$ be the bijective function such that $\psi \circ s(x) = s'(x)$ for all $x \in \mathsf{v}(E)$. Let σ_2' be defined with the same store s' as σ_1', and the heap $h_2' = h_{def} * h_{garb}$, as follows:

1) $\mathsf{dom}(h_{def}) = \psi(s(\mathsf{v}(E)) \cap \mathsf{dom}(h_2))$, and for all $x \in \mathsf{v}(E) \cap s^{-1}(\mathsf{dom}(h_2))$, $h_2' \circ \psi \circ s(x) = \psi \circ h_2 \circ s(x)$;

2) $\mathsf{dom}(h_{garb})$ and $\mathsf{codom}(h_{garb})$ are taken fresh, with the only condition that $w(h_{garb}) = w(h_2) - w(h_{def})$.

Then $\mathsf{dom}(h_1') \cap \mathsf{dom}(h_2') = \mathsf{dom}(h_1') \cap \mathsf{dom}(h_{def}') = \psi(\mathsf{dom}(h_1) \cap \mathsf{dom}(h_2) \cap s(\mathsf{v}(E))) = \emptyset$, so $\sigma_1' \bot \sigma_2'$. Moreover σ_2, σ_2' satisfy the same equality assertions since this was already the case for σ_1, σ_1' with the same stores. They also satisfy the same pointing assertions by construction, and $w(\sigma * \sigma_2) = w(\sigma_1) + w(\sigma_2) =_w w(\sigma_1') + w(\sigma_2) = w(\sigma_1' * \sigma_2')$. □

LEMMA 3.3 (ARITHMETIC SPLITTING). *For all $n_1, n_2, m_1, m_2, m', w$ such that $n_1 + m_1 + n_2 + m_2 =_{2w} n_1 + n_2 + m'$, there are m_1', m_2' such that $m' = m_1' + m_2'$ and $n_\epsilon + m_\epsilon =_w n_\epsilon + m_\epsilon'$ ($\epsilon = 1, 2$).*

PROOF. If $n_1 + m_1 < w$ and $n_2 + m_2 < w$ we have the standard equality and the lemma is trivial. Otherwise we may assume by symmetry that $n_1 + m_1 \geq w$. We then set $m_2' = \max(0, \min(m_2, w - n_2))$ and $m_1' = m' - m_2'$.

Then $n_2 + m_2' = n_2 + m_2$ or $= w$, and in both cases $n_2 + m_2' =_w n_2 + m_2$. Substracting $n_2 + m_2'$ in $n_1 + m_1 + n_2 + m_2 =_{2w} n_1 + n_2 + m'$, we get $n_1 + m' - m_2' =_w n_1 + m_1 + m_2 - m_2' \geq n_1 + m_1 \geq w$, so $n_1 + m_1' =_w n_1 + m_1$. □

Lemma 3.4 (Splitting). *For all σ, σ' such that $\sigma \approx_{E,2w} \sigma'$, if $\sigma = \sigma_1 * \sigma_2$, then there are σ_1', σ_2' such that $\sigma' = \sigma_1' * \sigma_2'$ and $\sigma_\epsilon \approx_{E,w} \sigma_\epsilon'$ ($\epsilon = 1, 2$).*

Proof. Let Φ_ϵ be the theory of pointing assertions satisfied in σ_ϵ. Then $\sigma'_{\Phi_1} \perp \sigma'_{\Phi_2}$ (by equality assertions), and the pointing assertions satisfied by σ'_{Φ_ϵ} are exactly Φ_ϵ. We then have $\sigma' = \sigma'_{\Phi_1} * \sigma'_{\Phi_2} * \tau'$, $\sigma_\epsilon = \sigma_{\Phi_\epsilon} * \tau_\epsilon$ with τ', τ_1, τ_2 some states having an empty theory of pointing assertions. The size measure condition is then $w_1 + w_2 + w(\tau_1) + w(\tau_2) =_{2w} w_1 + w_2 + w(\tau')$ where $w_\epsilon = w(\sigma_{\Phi_\epsilon}) = w(\sigma'_{\Phi_\epsilon})$. Applying Lemma 3.3, we have a spliting $\tau' = \tau_1' * \tau_2'$ such that $w_\epsilon + w(\tau_\epsilon) =_w w_\epsilon + w(\tau_\epsilon')$. Then $\sigma_\epsilon' = \sigma_{\Phi_\epsilon} * \tau_\epsilon'$ establishes the lemma. □

4. TRANSLATION

Given an assertion P, we define its splitting degree, $\mathsf{spl}(P)$, by induction as follows:

$$\begin{array}{rcl}
\mathsf{spl}(e \mapsto e') & = & 1 \\
\mathsf{spl}(e_1 = e_2) & = & -\infty \\
\mathsf{spl}(\perp) & = & -\infty \\
\mathsf{spl}(\mathsf{emp}) & = & 0 \\
\mathsf{spl}(P_1 \Rightarrow P_2) & = & \max\big(\mathsf{spl}(P_1), \mathsf{spl}(P_2)\big) \\
\mathsf{spl}(P_1 -\!\!* P_2) & = & \max\big(\mathsf{spl}(P_1), \mathsf{spl}(P_2)\big) \\
\mathsf{spl}(P_1 * P_2) & = & \max\big(\mathsf{spl}(P_1), \mathsf{spl}(P_2)\big) + 1
\end{array}$$

Proposition 4.1 (Correction). *If $\sigma \approx_{E,w} \sigma'$ and P is an assertion such that $E(P) \subseteq E$, $2^{\mathsf{spl}(P)} \leq w$, then*

$$\sigma \models P \quad \Leftrightarrow \quad \sigma' \models P\,.$$

Proof. By induction on P:

- if $\sigma \models (e \mapsto e')$, then $\sigma \models (e \hookrightarrow e')$, and so $\sigma' \models (e \hookrightarrow e')$. Moreover $w(\sigma) = 1 =_2 w(\sigma')$, so $w(\sigma') = 1$, that is $\sigma' \models (e \mapsto e')$.
- $\sigma \models e_1 = e_2$ iff $\sigma' \models e_1 = e_2$ by definition of $\approx_{E,w}$.
- $\sigma \models \mathsf{emp}$ iff $\sigma \models \neg(\mathsf{size} \geq 1)$, that is iff $\sigma' \models \neg(\mathsf{size} \geq 1)$ since $w \geq 1$.
- the cases for assertions $\perp$ and $P_1 \Rightarrow P_2$ are straightforward by induction.
- the case of assertion $P_1 * P_2$ follows from Lemma 3.4.
- if $\sigma \models P_1 -\!\!* P_2$, we consider $\sigma_1' \perp \sigma'$ such that $\sigma_1' \models P_1$, and prove that $\sigma_1' * \sigma' \models P_2$. By Lemma 3.2, there is $\sigma_1 \approx_{E,w} \sigma_1'$ such that $\sigma_1 \perp \sigma$. By induction, $\sigma_1 \models P_1$, so $\sigma * \sigma_1 \models P_2$. By Lemma 3.1, $\sigma_1 * \sigma \approx_{E,w} \sigma_1' * \sigma'$, so by induction $\sigma' * \sigma_1' \models P_2$.

□

We write $\Phi_{E,w}$ for the set of atomic assertions P such that $E(P) \subseteq E$ and $w(P) \leq w$. For E finite, $\Phi_{E,w}$ is finite as well. This has two important consequences:

Proposition 4.2 (Precompactness). *For all w and all finite E, $\approx_{E,w}$ has only finitely many classes.*

Proof. A class is represented by a subset $\Phi \subseteq \Phi_{E,w}$ of atomic assertions that are the ones satisfied by any state of the class. So there are less than $2^{\sharp \Phi_{E,w}}$ distinct classes. □

Proposition 4.3 (Characteristic formula). *For all state σ, for all E, w, there is a classical assertion $F_\sigma^{(E,w)}$ such that*

$$\forall \sigma'.\quad \sigma' \models F_\sigma^{(E,w)} \ \textit{iff}\ \sigma \approx_{E,w} \sigma'\,.$$

Proof. Take

$$\bigwedge_{\sigma \models P,\, P \in \Phi_{E,w}} P \quad \wedge \bigwedge_{\sigma \not\models P,\, P \in \Phi_{E,w}} \neg P\,.$$

□

We may now establish Theorem 2.1 noticing that any assertion P is equivalent to the classical assertion:

$$\bigvee_{C \in \mathrm{State}_{/\approx_{E,w}},\, C \models P} F_C^{(E,n)}\,,$$

where finiteness of this disjunction is ensured by Proposition 4.2.

5. CONCLUSION

We defined a classical fragment of the separation logic, excluding both $*$ and $-\!\!*$, and proved it to be as expressive as the full separation logic. Our approach shows also that all the separative power of the logic lies in the monotonic fragment.

An elimination property for a connective equivalent to $-\!\!*$ has been previously established for another spatial logic [4]. For the separation logic, it is even possible to eliminate spatial conjunction, which cannot hold for other spatial logics where multiple copies of the same structure may coexist. The use of equality assertions is essential for that (Lemmas 3.2 and 3.4), since the $*$ connective does express distinctions between pointers. For instance, $x \hookrightarrow - * y \hookrightarrow -$ says that $x \neq y$. Equalities would probably play also an essential role in an encoding involving quantifiers, as a counterexample in [4] tends to show.

When defining our classical fragment, we had to move from the assertions $e \mapsto e'$ and emp to $e \hookrightarrow e'$ and $\mathsf{size} \geq n$ in order to capture the $*$ connective. This would not happen for an assertion language with lookup and quantifiers, where the only necessary atomic assertions are equality assertions.

We do not study effectiveness of the translation, but it could probably be proved. However, our approach seems independent from decidability issues since, if we admit that it extends to logics with quantifiers, we could establish equivalence between a logic that admits a decidable model-checking problem (the classical one) and a logic that does not have it (as established in [1]). This also happens in the setting of [4].

We do not know wether our result remains true for richer assertion languages and wheter our proof is the right strategy to look at this problem. However, we conjecture that the classical logic should always be liable to express any separative assertion.

6. REFERENCES

[1] C. Calcagno, H. Yang, and P. O'Hearn. Computability and Complexity Results for a Spatial Assertion Language for Data Structures. In *Proceedings of FSTTCS '01*, volume 2245 of *LNCS*. Springer Verlag, 2001.

[2] D. Hirschkoff, E. Lozes, and D. Sangiorgi. Separability, Expressiveness and Decidability in the Ambients Logic. In *17th IEEE Symposium on Logic in Computer Science*, pages 423–432. IEEE Computer Society, 2002.

[3] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, pages 12(10):576–580, october 1969.

[4] E. Lozes. Adjunct elimination in the static ambient logic. In *Proceedings of Express '03*, 2003.

[5] J. Reynolds. Intuitionistic reasoning about shared mutable data structure, 2000.

[6] J. Reynolds. Separation logic: a logic for shared mutable data structures. 2002.