Extending LHC to Hyperliveness Properties

Report for internship – MPI-SWS Saarbrücken

ANTOINE GUILMIN-CRÉPON, ENS Paris-Saclay, France

1 STATE OF THE ART

Hoare logic was one of the first attempt at creating a formal framework for proving statements on programs. However, it proved itself limited for proving certain properties. In particular, dealing with non-determinism can become tricky, since basic Hoare logic can only deal with a given execution at a time. More generally, hyperproperties are an issue for basic Hoare logic, as they require to reason on multiple executions of the same program, or the execution of different programs.

Thus, multiple frameworks have been designed over the years to deal with those hyperproperties, mainly by extending Hoare logic to directly work on so-called hyperterms, i.e finite sets of terms, instead of regular terms.

LHC (for Logic for Hypertriple Composition) is one of those frameworks,

2 PRESENTATION OF LHC

This section is inspired by the paper presenting LHC. The logic defines, upon an arbitrary toy language, hyperterms, hyperstores, hyperreturn values and hyperproperties.

3 DESCRIPTION OF THE EXTENSION

The main point of the extension is to add an object to the language complementary to the $\mathbf{wp}_\forall$ from the original model.

This object is defined as such :

$\mathbf{wp}_\exists\, \boldsymbol{t}\, \{Q\} ::= \lambda s. \exists s' v.\, \langle t, s\rangle \Downarrow \langle v, s'\rangle \wedge Q(v)$

This object behaves in the same way as $\mathbf{wp}_\forall$ for most of the rules previously described. Those rules are written down for completeness

SOUNDNESS PROOFS

- WP-TRIV:

$$\begin{aligned} \vdash \mathbf{wp}_\forall\, \boldsymbol{t}\, \{\mathbf{wp}_\exists\, \boldsymbol{t}\, \{\mathsf{True}\}\} \quad &:= \quad \forall s\ v_1\ s_1'.\, \exists v_2\ s_2'.\, \langle t, s\rangle \Downarrow \langle v_1, s_1'\rangle \Rightarrow \langle t, s\rangle \Downarrow \langle v_2, s_2'\rangle \\ &\Leftarrow \quad \forall s\ v\ s'.\, \langle t, s\rangle \Downarrow \langle v, s'\rangle \Rightarrow \langle t, s\rangle \Downarrow \langle v, s'\rangle \end{aligned}$$

- WP$_\exists$-CONS:
  Suppose $\forall v.\, Q(v) \vdash Q'(v)$.
  If we have $\mathbf{wp}_\exists\, \boldsymbol{t}\, \{Q\}$, it means there exists an return value v and a output store s' s.t $\langle t, s\rangle \Downarrow \langle v, s'\rangle$ and $Q(v)(s')$. By the first assumption, we have $Q'(v)(s')$, thus $\mathbf{wp}_\exists\, \boldsymbol{t}\, \{Q'\}$.
- WP$_\exists$-EXISTS:

$$\begin{aligned} &\exists x.\, \mathbf{wp}_\exists\, \boldsymbol{t}\, \{Q(x)\} \dashv\vdash \mathbf{wp}_\exists\, \boldsymbol{t}\, \{\exists x.\, Q(x)\} \\ := \quad &\forall s.\, (\exists x\ v\ s'.\, \langle t, s\rangle \Downarrow \langle v, s'\rangle \wedge Q(x)(v)(s') \Leftrightarrow \exists v\ s'.\, \langle t, s\rangle \Downarrow \langle v, s'\rangle \wedge (\exists x.\, Q(x)(v)(s'))) \end{aligned}$$

Author's address: Antoine Guilmin-Crépon, antoine.guilmin@ens-paris-saclay.fr, ENS Paris-Saclay, Gif-Sur-Yvette, France.

$$\textsc{wp}_\forall\textsc{-triv}\quad \vdash \mathbf{wp}_\forall\, \boldsymbol{t}\, \{\mathrm{True}\}$$

$$\textsc{wp}_\forall\textsc{-cons}\quad \frac{\forall \boldsymbol{v}.\ Q(\boldsymbol{v}) \vdash Q'(\boldsymbol{v})}{\mathbf{wp}_\forall\, \boldsymbol{t}\, \{Q\} \vdash \mathbf{wp}_\forall\, \boldsymbol{t}\, \{Q'\}}$$

$$\textsc{wp}_\forall\textsc{-all}\quad \forall x.\, \mathbf{wp}_\forall\, \boldsymbol{t}\, \{Q(x)\} \dashv\vdash \mathbf{wp}_\forall\, \boldsymbol{t}\, \{\forall x.\, Q(x)\}$$

$$\textsc{wp}_\forall\textsc{-frame}\quad \frac{\Gamma \vdash \mathbf{wp}_\forall\, \boldsymbol{t}\, \{Q\} \qquad \mathrm{pvar}(P) \cap \mathrm{mods}(\boldsymbol{t}) = \emptyset}{\Gamma, P \vdash \mathbf{wp}_\forall\, \boldsymbol{t}\, \{\lambda \boldsymbol{r}.\, P \wedge Q(\boldsymbol{r})\}}$$

$$\textsc{wp}_\forall\textsc{-impl-r}\quad \frac{\mathrm{pvar}(P) \cap \mathrm{mods}(\boldsymbol{t}) = \emptyset}{P \Rightarrow \mathbf{wp}_\forall\, \boldsymbol{t}\, \{Q\} \dashv\vdash \mathbf{wp}_\forall\, \boldsymbol{t}\, \{\lambda \boldsymbol{r}.\, P \Rightarrow Q(\boldsymbol{r})\}}$$

$$\textsc{wp}_\forall\textsc{-subst}\quad \frac{\mathsf{x} \notin \mathrm{mods}(t)}{\mathsf{x}(i) = v \wedge \mathbf{wp}_\forall\, \left(\left[i{:}\, t[v/\mathsf{x}]\right] \cdot \boldsymbol{t'}\right) \{Q\} \vdash \mathbf{wp}_\forall\, \left([i{:}\, t] \cdot \boldsymbol{t'}\right) \{Q\}}$$

$$\textsc{wp}_\forall\textsc{-idx}\quad \frac{\pi \text{ bijective}}{(\mathbf{wp}_\forall\, \boldsymbol{t}\, \{Q\})(\!|\pi|\!) \vdash \mathbf{wp}_\forall\, \boldsymbol{t}(\!|\pi|\!)\, \{Q(\!|\pi|\!)\}}$$

Fig. 1. Base rules for $\mathbf{wp}_\forall$ from LHC

$$\textsc{wp}_\forall\textsc{-var}\quad \vdash \mathbf{wp}_\forall\, [i{:}\, \mathsf{x}]\, \{\lambda \boldsymbol{r}.\, \boldsymbol{r}(i) = \mathsf{x}(i)\}$$

$$\textsc{wp}_\forall\textsc{-val}\quad \vdash \mathbf{wp}_\forall\, [i{:}\, v]\, \{\lambda \boldsymbol{r}.\, \boldsymbol{r}(i) = v\}$$

$$\textsc{wp}_\forall\textsc{-skip}\quad \mathbf{wp}_\forall\, (\boldsymbol{t} \cdot [i{:}\, \mathbf{skip}])\, \{Q\} \dashv\vdash \mathbf{wp}_\forall\, \boldsymbol{t}\, \{Q\}$$

$$\textsc{wp}_\forall\textsc{-prim}_\oplus\quad \vdash \mathbf{wp}_\forall\, [i{:}\, v_1 \oplus v_2]\, \{\lambda \boldsymbol{r}.\, \boldsymbol{r}(i) = (v_1\ [\![\oplus]\!]\ v_2)\}$$

$$\textsc{wp}_\forall\textsc{-seq}_I\quad \mathbf{wp}_\forall\, [i{:}\, t_i \mid i \in I]\, \left\{\mathbf{wp}_\forall\, [i{:}\, t_i' \mid i \in I]\, \{Q\}\right\} \dashv\vdash \mathbf{wp}_\forall\, [i{:}\, (t_i;\ t_i') \mid i \in I]\, \{Q\}$$

$$\textsc{wp}_\forall\textsc{-assign}_I\quad \frac{\forall i \in I.\, (\mathsf{x}_i, i) \notin \mathrm{pvar}(Q)}{\mathbf{wp}_\forall\, [i{:}\, e_i \mid i \in I]\, \{Q\} \vdash \mathbf{wp}_\forall\, [i{:}\, \mathsf{x}_i \coloneqq e_i \mid i \in I]\, \{\lambda \boldsymbol{r}.\, Q(\boldsymbol{r}) \wedge \textstyle\bigwedge_{i \in I} \boldsymbol{r}(i) = \mathsf{x}_i(i)\}}$$

$$\textsc{wp}_\forall\textsc{-if}_I\quad \mathbf{wp}_\forall\, [i{:}\, g_i \mid i \in I]\, \left\{\lambda \boldsymbol{b}.\, \mathbf{wp}_\forall \begin{pmatrix}[i{:}\, t_i \mid i \in I, \boldsymbol{b}(i) \neq 0]\cdot \\ [i{:}\, t_i' \mid i \in I, \boldsymbol{b}(i) = 0]\end{pmatrix} \{Q\}\right\} \dashv\vdash \mathbf{wp}_\forall\, [i{:}\, \mathbf{if}\ g_i\ \mathbf{then}\ t_i\ \mathbf{else}\ t_i' \mid i \in I]\, \{Q\}$$

$$\textsc{wp}_\forall\textsc{-while}_I\quad \frac{P \vdash \mathbf{wp}_\forall\, [i{:}\, g_i \mid i \in I]\, \left\{\lambda \boldsymbol{b}.\, (\boldsymbol{b} =_I 0 \wedge R) \vee (\boldsymbol{b} \neq_I 0 \wedge \mathbf{wp}_\forall\, [i{:}\, t_i \mid i \in I]\, \{P\})\right\}}{P \vdash \mathbf{wp}_\forall\, [i{:}\, \mathbf{while}\ g_i\ \mathbf{do}\ t_i \mid i \in I]\, \{R\}}$$

Fig. 2. Lockstep rules for $\mathbf{wp}_\forall$ from LHC

WP$_\forall$-NEST

$$\mathbf{wp}_\forall\, \boldsymbol{t}_1\, \{\lambda \boldsymbol{v}.\, \mathbf{wp}_\forall\, \boldsymbol{t}_2\, \{\lambda \boldsymbol{w}.\, Q(\boldsymbol{v} \cdot \boldsymbol{w})\}\} \dashv\vdash \mathbf{wp}_\forall\, (\boldsymbol{t}_1 \cdot \boldsymbol{t}_2)\, \{Q\}$$

WP$_\forall$-CONJ

$$\frac{\mathrm{idx}(Q_1) \cap \mathrm{supp}(\boldsymbol{t}_2) \subseteq \mathrm{supp}(\boldsymbol{t}_1) \qquad \mathrm{idx}(Q_2) \cap \mathrm{supp}(\boldsymbol{t}_1) \subseteq \mathrm{supp}(\boldsymbol{t}_2)}{\mathbf{wp}_\forall\, \boldsymbol{t}_1\, \{Q_1\} \wedge \mathbf{wp}_\forall\, \boldsymbol{t}_2\, \{Q_2\} \vdash \mathbf{wp}_\forall\, (\boldsymbol{t}_1 + \boldsymbol{t}_2)\, \{Q_1 \wedge Q_2\}}$$

WP$_\forall$-PROJ

$$\frac{}{\Pi_I.\, \big(\mathrm{proj}(\boldsymbol{t}_2) \Rightarrow \mathrm{proj}(\boldsymbol{t}_1) \wedge \mathbf{wp}_\forall\, (\boldsymbol{t}_1 \cdot \boldsymbol{t}_2)\, \{Q\}\big) \vdash \mathbf{wp}_\forall\, \boldsymbol{t}_2\, \{\hat{\Pi}_I.\, Q\}}\ I = \mathrm{supp}(\boldsymbol{t}_1)$$

Fig. 3. Hyper-structure laws from LHC

WP$_\forall$-IDX-PASS

$$\frac{i, j \notin \mathrm{supp}(\boldsymbol{t})}{(\mathbf{wp}_\forall\, \boldsymbol{t}\, \{Q\})(\!|i/j|\!) \vdash \mathbf{wp}_\forall\, \boldsymbol{t}\, \{Q(\!|i/j|\!)\}}$$

WP$_\forall$-IDX-SWAP

$$\frac{i \notin \mathrm{idx}(Q)}{\big(\mathbf{wp}_\forall\, ([j\colon t] \cdot \boldsymbol{t}')\, \{Q\}\big)(\!|i/j|\!) \vdash \mathbf{wp}_\forall\, ([i\colon t] \cdot \boldsymbol{t}')\, \{Q(\!|i/j|\!)\}}$$

WP$_\forall$-IDX-MERGE

$$\big(\mathbf{wp}_\forall\, ([i\colon t, j\colon t] \cdot \boldsymbol{t}')\, \{Q\}\big)(\!|i/j|\!) \vdash \mathbf{wp}_\forall\, ([i\colon t] \cdot \boldsymbol{t}')\, \{Q(\!|i/j|\!)\}$$

WP$_\forall$-IDX-POST

$$\frac{\Gamma \vdash \mathbf{wp}_\forall\, \boldsymbol{t}\, \{Q\} \qquad j \notin \mathrm{supp}(\boldsymbol{t}) \cup \mathrm{idx}(\Gamma)}{\Gamma \vdash \mathbf{wp}_\forall\, \boldsymbol{t}\, \{Q(\!|i/j|\!)\}}$$

Fig. 4. Reindexing rules from LHC

Fig. 5. New $\mathbf{wp}_\exists$ rules