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 $\mathbf{wp}_\vee$ from the original model.

This object is defined as such :

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

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

An interesting point is that predicates that couldn't be defined in the original theory using the base langugage, and thus needed to be explicitly defined to be used, can now be defined using this new $\mathbf{wp}_\exists$ object.

$$
\begin{aligned}
\mathrm{proj}(t) \quad &:= \quad \mathbf{wp}_\exists\, t\, \{\mathsf{True}\} \\
t_1 \preceq t_2 \quad &\dashv\vdash \quad \mathbf{wp}_\vee\, [1 : t_1]\, \{\mathbf{wp}_\exists\, [2 : t_2]\, \{v(1) = v(2) \wedge s(1) = s(2)\}\}
\end{aligned}
$$

WP$_\forall$-TRIV

$\vdash \mathbf{wp}_\forall\, t\, \{\mathsf{True}\}$

WP$_\forall$-CONS

$$\frac{\forall v.\; Q(v) \vdash Q'(v)}{\mathbf{wp}_\forall\, t\, \{Q\} \vdash \mathbf{wp}_\forall\, t\, \{Q'\}}$$

WP$_\forall$-ALL

$\forall x.\, \mathbf{wp}_\forall\, t\, \{Q(x)\} \dashv\vdash \mathbf{wp}_\forall\, t\, \{\forall x.\, Q(x)\}$

WP$_\forall$-FRAME

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

WP$_\forall$-IMPL-R

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

WP$_\forall$-SUBST

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

WP$_\forall$-IDX

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

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

WP$_\forall$-VAR

$\vdash \mathbf{wp}_\forall\, [i\colon \mathsf{x}]\, \{\lambda r.\, r(i) = \mathsf{x}(i)\}$

WP$_\forall$-VAL

$\vdash \mathbf{wp}_\forall\, [i\colon v]\, \{\lambda r.\, r(i) = v\}$

WP$_\forall$-SKIP

$\mathbf{wp}_\forall\, (t \cdot [i\colon \mathbf{skip}])\, \{Q\} \dashv\vdash \mathbf{wp}_\forall\, t\, \{Q\}$

WP$_\forall$-PRIM$_\oplus$

$\vdash \mathbf{wp}_\forall\, [i\colon v_1 \oplus v_2]\, \{\lambda r.\, r(i) = (v_1\, [\![\oplus]\!]\, v_2)\}$

WP$_\forall$-SEQ$_I$

$\mathbf{wp}_\forall\, [i\colon t_i \mid i \in I]\, \left\{\mathbf{wp}_\forall\, [i\colon t_i' \mid i \in I]\, \{Q\}\right\} \dashv\vdash \mathbf{wp}_\forall\, [i\colon (t_i\,;\, t_i') \mid i \in I]\, \{Q\}$

WP$_\forall$-ASSIGN$_I$

$$\frac{\forall i \in I.\, (\mathsf{x}_i, i) \notin \mathrm{pvar}(Q)}{\mathbf{wp}_\forall\, [i\colon e_i \mid i \in I]\, \{Q\} \vdash \mathbf{wp}_\forall\, [i\colon \mathsf{x}_i := e_i \mid i \in I]\, \{\lambda r.\, Q(r) \wedge \bigwedge_{i \in I} r(i) = \mathsf{x}_i(i)\}}$$

WP$_\forall$-IF$_I$

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

WP$_\forall$-WHILE$_I$

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

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

WP$_\forall$-NEST

$$\mathbf{wp}_\forall \, t_1 \, \{\lambda v. \, \mathbf{wp}_\forall \, t_2 \, \{\lambda w. \, Q(v \cdot w)\}\} \dashv\vdash \mathbf{wp}_\forall \, (t_1 \cdot t_2) \, \{Q\}$$

WP$_\forall$-CONJ

$$\frac{\mathrm{idx}(Q_1) \cap \mathrm{supp}(t_2) \subseteq \mathrm{supp}(t_1) \qquad \mathrm{idx}(Q_2) \cap \mathrm{supp}(t_1) \subseteq \mathrm{supp}(t_2)}{\mathbf{wp}_\forall \, t_1 \, \{Q_1\} \wedge \mathbf{wp}_\forall \, t_2 \, \{Q_2\} \vdash \mathbf{wp}_\forall \, (t_1 + t_2) \, \{Q_1 \wedge Q_2\}}$$

WP$_\forall$-PROJ

$$\frac{}{\Pi_I. \, \big(\mathrm{proj}(t_2) \Rightarrow \mathrm{proj}(t_1) \wedge \mathbf{wp}_\forall \, (t_1 \cdot t_2) \, \{Q\}\big) \vdash \mathbf{wp}_\forall \, t_2 \, \{\hat{\Pi}_I. \, Q\}} \quad I = \mathrm{supp}(t_1)$$

Fig. 3. Hyper-structure laws from LHC

WP$_\forall$-IDX-PASS

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

WP$_\forall$-IDX-SWAP

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

WP$_\forall$-IDX-MERGE

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

WP$_\forall$-IDX-POST

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

Fig. 4. Reindexing rules from LHC

SOUNDNESS PROOFS

- WP-TRIV:

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

- WP$_\exists$-CONS:
  Suppose $\forall v. \, Q(v) \vdash Q'(v)$.
  $\mathbf{wp}_\exists \, t \, \{Q\}$ 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, one gets $Q'(v)(s')$, thus $\mathbf{wp}_\exists \, t \, \{Q'\}$.

- WP$_\exists$-EXISTS:
  Given that t does not depend on x,

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

- WP$_\exists$-FRAME:
  Suppose $\Gamma \vdash \mathbf{wp}_\exists \, t \, \{Q\}$ and $\mathrm{pvar}(P) \cap \mathrm{mods}(t) = \emptyset$. The first assumption implies a terminating trace for t verifying Q.
  Given that no variable of P is modified by t, for every execution of the term, P will equally hold at both ends of the trace.
  Thus $\Gamma, P \vdash \mathbf{wp}_\exists \, t \, \{\lambda r. P \wedge Q(r)\}$.

WP∃-CONS
$$\frac{\exists v.\ Q(v) \vdash Q'(v)}{\mathbf{wp}_\exists\, t\, \{Q\} \vdash \mathbf{wp}_\exists\, t\, \{Q'\}}$$

WP∃-EXISTS
$$\exists x.\, \mathbf{wp}_\exists\, t\, \{Q(x)\} \dashv\vdash \mathbf{wp}_\exists\, t\, \{\exists x.\, Q(x)\}$$

WP∃-FRAME
$$\frac{\Gamma \vdash \mathbf{wp}_\exists\, t\, \{Q\} \qquad \mathrm{pvar}(P) \cap \mathrm{mods}(t) = \emptyset}{\Gamma, P \vdash \mathbf{wp}_\exists\, t\, \{\lambda r.\, P \wedge Q(r)\}}$$

WP∃-IMPL-R
$$\frac{\mathrm{pvar}(P) \cap \mathrm{mods}(t) = \emptyset}{P \Rightarrow \mathbf{wp}_\exists\, t\, \{Q\} \dashv\vdash \mathbf{wp}_\exists\, t\, \{\lambda r.\, P \Rightarrow Q(r)\}}$$

WP∃-SUBST
$$\frac{\mathsf{x} \notin \mathrm{mods}(t)}{\mathsf{x}(i) = v \wedge \mathbf{wp}_\exists\, \big([i{:}\,t[v/\mathsf{x}]] \cdot t'\big)\, \{Q\} \vdash \mathbf{wp}_\exists\, ([i{:}\,t] \cdot t')\, \{Q\}}$$

WP∃-IDX
$$\frac{\pi\ \text{bijective}}{(\mathbf{wp}_\exists\, t\, \{Q\})(\!|\pi|\!) \vdash \mathbf{wp}_\exists\, t(\!|\pi|\!)\, \{Q(\!|\pi|\!)\}}$$

WP∃-SEQ$_I$
$$\mathbf{wp}_\exists\, [i{:}\,t_i \mid i \in I]\, \big\{\mathbf{wp}_\exists\, [i{:}\,t_i' \mid i \in I]\, \{Q\}\big\} \dashv\vdash \mathbf{wp}_\exists\, [i{:}\,(t_i\,;\, t_i') \mid i \in I]\, \{Q\}$$

WP∃-ASSIGN$_I$
$$\frac{\exists i \in I.\, (\mathsf{x}_i, i) \notin \mathrm{pvar}(Q)}{\mathbf{wp}_\exists\, [i{:}\,e_i \mid i \in I]\, \{Q\} \vdash \mathbf{wp}_\exists\, [i{:}\,\mathsf{x}_i := e_i \mid i \in I]\, \{\lambda r.\, Q(r) \wedge \bigwedge_{i \in I} r(i) = \mathsf{x}_i(i)\}}$$

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

WP∃-WHILE$_I$
$$\frac{\forall \alpha.\, P(\alpha) \vdash \mathbf{wp}_\exists\, [i{:}\,g_i \mid i \in I]\, \big\{\lambda b.\, (b =_I 0 \wedge R) \vee (b \neq_I 0 \wedge \mathbf{wp}_\exists\, [i{:}\,t_i \mid i \in I]\, \{\exists \alpha'.\, P(\alpha') \wedge \alpha' < \alpha\})\big\}}{P(\alpha_0) \vdash \mathbf{wp}_\exists\, [i{:}\,\mathbf{while}\ g_i\ \mathbf{do}\ t_i \mid i \in I]\, \{R\}}$$

WP∃-NEST
$$\mathbf{wp}_\exists\, t_1\, \{\lambda v.\, \mathbf{wp}_\exists\, t_2\, \{\lambda w.\, Q(v \cdot w)\}\} \dashv\vdash \mathbf{wp}_\exists\, (t_1 \cdot t_2)\, \{Q\}$$

WP∃-CONJ
$$\frac{\mathrm{idx}(Q_1) \cap \mathrm{supp}(t_2) \subseteq \mathrm{supp}(t_1) \qquad \mathrm{idx}(Q_2) \cap \mathrm{supp}(t_1) \subseteq \mathrm{supp}(t_2)}{\mathbf{wp}_\exists\, t_1\, \{Q_1\} \wedge \mathbf{wp}_\exists\, t_2\, \{Q_2\} \vdash \mathbf{wp}_\exists\, (t_1 + t_2)\, \{Q_1 \wedge Q_2\}}$$

WP∃-IDX-PASS
$$\frac{i, j \notin \mathrm{supp}(t)}{(\mathbf{wp}_\exists\, t\, \{Q\})(\!|i/j|\!) \vdash \mathbf{wp}_\exists\, t\, \{Q(\!|i/j|\!)\}}$$

WP∃-IDX-SWAP
$$\frac{i \notin \mathrm{idx}(Q)}{(\mathbf{wp}_\exists\, ([j{:}\,t] \cdot t')\, \{Q\})(\!|i/j|\!) \vdash \mathbf{wp}_\exists\, ([i{:}\,t] \cdot t')\, \{Q(\!|i/j|\!)\}}$$

WP∃-IDX-MERGE
$$\big(\mathbf{wp}_\exists\, ([i{:}\,t, j{:}\,t] \cdot t')\, \{Q\}\big)(\!|i/j|\!) \vdash \mathbf{wp}_\exists\, ([i{:}\,t] \cdot t')\, \{Q(\!|i/j|\!)\}$$

WP∃-IDX-POST
$$\frac{\Gamma \vdash \mathbf{wp}_\exists\, t\, \{Q\} \qquad j \notin \mathrm{supp}(t) \cup \mathrm{idx}(\Gamma)}{\Gamma \vdash \mathbf{wp}_\exists\, t\, \{Q(\!|i/j|\!)\}}$$

WP-TRIV

$\vdash \mathbf{wp}_\forall t \, \{\mathbf{wp}_\exists t \, \{\mathsf{True}\}\}$

WP$_\forall$-WP$_\exists$-SWAP

$$\frac{\forall i. \, \mathbf{wp}_\forall \, [1 : t_2(i)] \, \{\mathbf{wp}_\exists \, [2 : t_1(i)] \, \{Q\}\}}{\mathbf{wp}_\exists \, t_2 \, \{\mathbf{wp}_\forall \, t_1 \, \{Q\}\} \vdash \mathbf{wp}_\forall \, t_1 \, \{\mathbf{wp}_\exists \, t_2 \, \{Q\}\}}$$

Fig. 6. Rules combining $\mathbf{wp}_\forall$ and $\mathbf{wp}_\exists$ rules

- WP$_\exists$-IMPL-R:
  Similar to WP$_\exists$-FRAME.
- WP$_\exists$-IDX:
  The proof relies on the fact that the existence of some terminating trace for a certain hyperterm is unchanged by a bijective reindexing of said hyperterm.
- WP$_\exists$-NEST:
  Suppose $\mathbf{wp}_\exists \, (t_1 \cdot t_2) \, \{Q\}$. It says that for all input states, there exists an output state satisfying Q.
  Given that there are no side effects between the terms of an hyperterm, one can "divide" the trace as two separate traces in t_1 and t_2. Thus, forall input state, there exists a state that coincides with the output state given by the assumption for the domain of t_1, and with the input state for the domain of t_2 (it is here of prime importance that t_1 and t_2 are disjoint). Therefore, $\mathbf{wp}_\exists \, t_1 \, \{\mathbf{wp}_\exists \, t_2 \, \{Q\}\}$