Half baked talk: Invariant logic

Quentin Carbonneaux

November 6, 2015

Motivation

Global invariants often show up:

1. resource safety ($\text{mem} \geq 0$)
2. low-level code analysis (machine not crashed)
3. domain specific (in a car: $\neg(\text{break} \wedge \text{accel})$)

Previous work define *Quantitative Logics*.
I make them instances of *Invariant Logic*.

Programs and Logic

Programs: Syntax

$$p := b \mid \mathsf{skip} \mid \mathsf{break} \mid p; p \mid p + p \mid \mathsf{loop}\ p$$

Programs: Semantics

Program states σ, σ' are elements of S.

The semantics of a base action b is $[\![b]\!] \subseteq \mathrm{S} \times \mathrm{S}$.

The invariants and assertions $A_1, A_2, I \subseteq \mathrm{S}$.

Programs: Semantics

$k := \mathsf{k0} \mid \mathsf{ks}\ p\ k \mid \mathsf{kl}\ p\ k$

1. $\sigma[\![b]\!]\sigma' \implies (\sigma, b, k) \mapsto (\sigma', \mathsf{skip}, k)$
2. $(\sigma, p_1; p_2, k) \mapsto (\sigma, p_1, \mathsf{ks}\ p_2\ k)$
3. $(\sigma, \mathsf{skip}, \mathsf{ks}\ p\ k) \mapsto (\sigma, p, k)$
4. $(\sigma, \mathsf{break}, \mathsf{ks}\ p\ k) \mapsto (\sigma, \mathsf{break}, k)$
5. $(\sigma, p_1 + p_2, k) \mapsto (\sigma, p_1, k)$
6. $(\sigma, p_1 + p_2, k) \mapsto (\sigma, p_2, k)$
7. $(\sigma, \mathsf{loop}\ p, k) \mapsto (\sigma, p, \mathsf{kl}\ p\ k)$
8. $(\sigma, \mathsf{skip}, \mathsf{kl}\ p\ k) \mapsto (\sigma, \mathsf{loop}\ p, k)$
9. $(\sigma, \mathsf{break}, \mathsf{kl}\ p\ k) \mapsto (\sigma, \mathsf{skip}, k)$

Sample actions and invariants

- $[\![x := N]\!] := \{(H, H[x \mapsto N]) \mid \forall H\}$
- $[\![\text{when } e]\!] := \{(H, H) \mid \forall H,\ [\![e]\!]_H \neq 0\}$
- $[\![\text{tick } N]\!] := \{(\{H, c\}, \{H, c - N\}) \mid \forall Hc\}$
- $[\![\lightning]\!] := \{(H, \bot) \mid \forall H\}$

We can now encode:

- $\text{if}(e)\ p_1 \text{ else } p_2 \ \equiv\ (\text{when } e; p_1) + (\text{when } \neg e; p_2)$
- $\text{assert } e \ \equiv\ (\text{when } \neg e; \lightning) + \text{when } e$

Relevant invariants:

$I_t := \{(H, c) \mid \forall Hc,\ c \geq 0\}$ and $I_s := \{H \mid \forall H\}$

Invariant logic (in blue)

$$\frac{}{\vdash_I \{A\}\mathsf{skip}\{A, \bot\}} \qquad \frac{}{\vdash_I \{A\}\mathsf{break}\{\bot, A\}}$$

$$\frac{}{\vdash_I \{\forall \sigma', \sigma[\![b]\!]\sigma' \implies \sigma' \in A \cap I\}b\{A, \bot\}}$$

$$\frac{\vdash_I \{A_1\}p_1\{A_2, B\} \quad \vdash_I \{A_2\}p_2\{A_3, B\}}{\vdash_I \{A_1\}p_1; p_2\{A_3, B\}} \qquad \frac{\vdash_I \{A_1\}p_1\{A_2, B\} \quad \vdash_I \{A_1\}p_2\{A_2, B\}}{\vdash_I \{A_1\}p_1 + p_2\{A_2, B\}}$$

$$\frac{\vdash_I \{A\}p\{A, B\}}{\vdash_I \{A\}\mathsf{loop}\ p\{B, \bot\}} \qquad \frac{\vdash_I \{A_1\}p\{A_2, B\} \quad A_1' \subseteq A_1 \quad A_2 \subseteq A_2' \quad B \subseteq B'}{\vdash_I \{A_1'\}p\{A_2', B'\}}$$

Parenthesis

Why is the full specification power of Hoare logic required when simply proving I?

Parenthesis

Why is the full specification power of Hoare logic required when simply proving I?

$$[\text{compute } 6! \text{ in } x...];\ \textsf{assert}\ (x \equiv 730)$$

We need the functional specification of the code in the ellipsis!

Meaning of Invariant logic triples

We say $A \subseteq I$ is safe for p when

$$\forall(\sigma \in A)\, \sigma',\ (\sigma, p, \mathsf{k0}) \mapsto^* (\sigma', \_, \_) \implies \sigma' \in I$$

If the program p is started in a state in A then, all reachable states are in I.

We shoot for:
$\forall A \subseteq I, \vdash_I \{A\} p \{\_, \_\} \iff A$ safe for p.

Soundness

Step 1: Safety indexing

A configuration (σ, p, k) is *safe* for n steps if:
$\forall m \leq n,\ (\sigma, p, k) \mapsto^m (\sigma', \_, \_) \implies \sigma' \in I.$

We write $\text{safe}_n(\sigma, p, k)$.

Step 1: Safety indexing

A configuration (σ, p, k) is *safe* for n steps if:
$\forall m \leq n,\ (\sigma, p, k) \mapsto^m (\sigma', \_, \_) \implies \sigma' \in I.$

We write $\text{safe}_n(\sigma, p, k)$.

Safety verifies two essential properties:

1. Weaken:
   $\text{safe}_{n+1}\ c \implies \text{safe}_n\ c$
2. Step:
   $(\forall c',\ c \mapsto c' \implies \text{safe}_n\ c') \implies \text{safe}_{n+1}\ c$

Step 2: Compositionality

"A safe for p"

- mentions `k0`
- does not mention the post-condition

It is *non-compositional*, i.e. unsuitable for proofs.

Step 2: Compositionality

"A safe for p"

- mentions `k0`
- does not mention the post-condition

It is *non-compositional*, i.e. unsuitable for proofs.

Solution: Introduce a continuation k and make the post-condition a condition on k.

Step 2: Compositionality

$$\begin{aligned}\models_I \{A_1\}p\{A_2, B\} := \forall k\, n,\\ \forall \sigma' \in B,\ \mathrm{safe}_n(\sigma', \textsf{break}, k) \ \wedge\\ \forall \sigma' \in A_2,\ \mathrm{safe}_n(\sigma', \textsf{skip}, k) \implies\\ \forall \sigma \in A_1,\ \mathrm{safe}_n(\sigma, p, k).\end{aligned}$$

Step 2: Compositionality

$$
\begin{aligned}
\models_I \{A_1\}p\{A_2, B\} := \forall k\, n,& \\
\forall \sigma' \in B,\ \mathrm{safe}_n(\sigma', \mathsf{break}, k) &\ \wedge \\
\forall \sigma' \in A_2,\ \mathrm{safe}_n(\sigma', \mathsf{skip}, k) &\implies \\
&\forall \sigma \in A_1,\ \mathrm{safe}_n(\sigma, p, k).
\end{aligned}
$$

Note:

- We always use the same index, this allows proof by induction for loop.

Step 2: Compositionality

$$\models_I \{A_1\}p\{A_2, B\} := \forall k\, n,\\ \forall \sigma' \in B,\ \text{safe}_n(\sigma', \textsf{break}, k) \ \wedge\\ \forall \sigma' \in A_2,\ \text{safe}_n(\sigma', \textsf{skip}, k) \implies\\ \forall \sigma \in A_1,\ \text{safe}_n(\sigma, p, k).$$

Note:

- We always use the same index, this allows proof by induction for $\textsf{loop}$.
- $\textsf{k0}$ is safe: $\forall n\, \sigma,\ \text{safe}_n(\sigma, \textsf{break}/\textsf{skip}, \textsf{k0})$

Step 2: Compositionality

$$\models_I \{A_1\}p\{A_2, B\} := \forall k\, n,$$
$$\forall \sigma' \in B,\ \text{safe}_n(\sigma', \textsf{break}, k) \ \wedge$$
$$\forall \sigma' \in A_2,\ \text{safe}_n(\sigma', \textsf{skip}, k) \implies$$
$$\forall \sigma \in A_1,\ \text{safe}_n(\sigma, p, k).$$

Note:

- We always use the same index, this allows proof by induction for loop.
- k0 is safe: $\forall n\, \sigma,\ \text{safe}_n(\sigma, \textsf{break}/\textsf{skip}, \textsf{k0})$
  Thus $\models_I \{A\}p\{\_\} \implies A$ safe for p.

Completeness

Recipe for completeness

Idea: prove $\vdash_I \{X\}p\{\_,\_\}$ for a well chosen X,
then weaken to $\vdash_I \{A\}p\{\_,\_\}$
using $\forall \sigma,\ \sigma \in A \implies \sigma \in X$ (*).

Recipe for completeness

Idea: prove $\vdash_I \{X\}p\{\_,\_\}$ for a well chosen X, then weaken to $\vdash_I \{A\}p\{\_,\_\}$ using $\forall\sigma,\ \sigma \in A \implies \sigma \in X$ (*).

1. Look at your assumption: A safe for p, i.e. $\forall\sigma,\ \sigma \in A \implies \forall n,\ \mathrm{safe}_n(\sigma, p, \mathsf{k0})$

Recipe for completeness

Idea: prove $\vdash_I \{X\}p\{\_, \_\}$ for a well chosen X,
then weaken to $\vdash_I \{A\}p\{\_, \_\}$
using $\forall \sigma,\ \sigma \in A \implies \sigma \in X$ (*).

1. Look at your assumption: A safe for p, i.e.
   $\forall \sigma,\ \sigma \in A \implies \forall n,\ \mathrm{safe}_n(\sigma, p, \mathsf{k0})$
2. Match it with (*):
   X is $\{\sigma \mid \forall n,\ \mathrm{safe}_n(\sigma, p, \mathsf{k0})\}$.

Recipe for completeness

Idea: prove $\vdash_I \{X\}p\{\_, \_\}$ for a well chosen X,
then weaken to $\vdash_I \{A\}p\{\_, \_\}$
using $\forall \sigma,\ \sigma \in A \implies \sigma \in X$ (*).

1. Look at your assumption: A safe for p, i.e.
   $\forall \sigma,\ \sigma \in A \implies \forall n,\ \mathrm{safe}_n(\sigma, p, \mathsf{k0})$
2. Match it with (*):
   X is $\{\sigma \mid \forall n,\ \mathrm{safe}_n(\sigma, p, \mathsf{k0})\}$.
3. Make it compositional!

Most general triple

$$X \text{ is } \{\sigma \mid \forall n,\ \text{safe}_n(\sigma, p, \mathsf{k0})\}$$

Once again $\mathsf{k0}$ is mentioned in X and should not.

Most general triple

$$X \text{ is } \{\sigma \mid \forall n,\ \text{safe}_n(\sigma, p, \mathsf{k0})\}$$

Once again $\mathsf{k0}$ is mentioned in X and should not.
To solve that:

1. Abstract $\mathsf{k0}$ (and p) from X:
   $X(p, k) := \{\sigma \mid \forall n,\ \text{safe}(\sigma, p, k)\}$.

Most general triple

$$X \text{ is } \{\sigma \mid \forall n,\ \text{safe}_n(\sigma, p, \mathsf{k0})\}$$

Once again $\mathsf{k0}$ is mentioned in X and should not.
To solve that:

1. Abstract $\mathsf{k0}$ (and p) from X:
   $X(p, k) := \{\sigma \mid \forall n,\ \text{safe}(\sigma, p, k)\}$.
2. Define the *most general triple* $\text{mgt}(p)$:
   $\forall k,\ \vdash_I \{X(p, k)\}p\{X(\mathsf{skip}, k), X(\mathsf{break}, k)\}$.
   (It is a family of triples.)

Most general triple

$$X \text{ is } \{\sigma \mid \forall n,\ \text{safe}_n(\sigma, p, \mathsf{k0})\}$$

Once again $\mathsf{k0}$ is mentioned in X and should not.
To solve that:

1. Abstract $\mathsf{k0}$ (and p) from X:
   $X(p,k) := \{\sigma \mid \forall n,\ \text{safe}(\sigma, p, k)\}$.
2. Define the *most general triple* $\text{mgt}(p)$:
   $\forall k,\ \vdash_I \{X(p,k)\}p\{X(\mathsf{skip},k), X(\mathsf{break},k)\}$.
   (It is a family of triples.)

Remark: To my knowledge, this is original work.

Completeness

$$X(p,k) := \{\sigma \mid \forall n,\ \mathrm{safe}(\sigma, p, k)\}$$

$$\mathrm{mgt}(p) := \forall k, \vdash_I \{X(p,k)\} p \{\_, \_\}$$

Completeness

$$X(p,k) := \{\sigma \mid \forall n,\ \text{safe}(\sigma, p, k)\}$$
$$\text{mgt}(p) := \forall k,\ \vdash_I \{X(p,k)\}p\{\_,\_\}$$

- Show $\forall p, \text{mgt}(p)$ by induction.

Completeness

$$X(p,k) := \{\sigma \mid \forall n,\ \text{safe}(\sigma,p,k)\}$$
$$\text{mgt}(p) := \forall k, \vdash_I \{X(p,k)\}p\{\_,\_\}$$

- Show $\forall p, \text{mgt}(p)$ by induction.
- Remark "A safe for p" is $A \subseteq X(p,\mathsf{k0})$.

Completeness

$$X(p,k) := \{\sigma \mid \forall n,\ \mathrm{safe}(\sigma,p,k)\}$$
$$\mathrm{mgt}(p) := \forall k,\ \vdash_I \{X(p,k)\}p\{\_,\_\}$$

- Show $\forall p, \mathrm{mgt}(p)$ by induction.
- Remark "A safe for p" is $A \subseteq X(p, \mathsf{k0})$.
- Then, the completeness simply falls by weakening:

$$\frac{A \subseteq X(p,\mathsf{k0}) \qquad \vdash_I \{X(p,\mathsf{k0})\}p\{\_,\_\}}{\vdash_I \{A\}p\{\top,\top\}}$$

Concluding Remarks

Relation with Hoare logic

$$\frac{}{\vdash_I \implies \vdash_H} \qquad \frac{}{\vdash_H \not\Longrightarrow \vdash_I}$$

$$\frac{}{\vdash_I \{A_1\}p\{A_2\} \implies \vdash_H \{A_1 \cap I\}p\{A_2 \cap I\}} \; (1)$$

$$\frac{}{\vdash_H \{A_1 \cap I\}p\{A_2 \cap I\} \not\Longrightarrow \vdash_I \{A_1\}p\{A_2\}} \; (2)$$

Invariant logic says more about the process than the outcome.

Coq proof

Available on demand as one Coq file of 403 lines.

Follows very closely the explanations above, probably good for education.

Also exists with function calls. (Non-trivial extension, it requires auxiliary state, handled with a meta-level quantification.)

Questions?