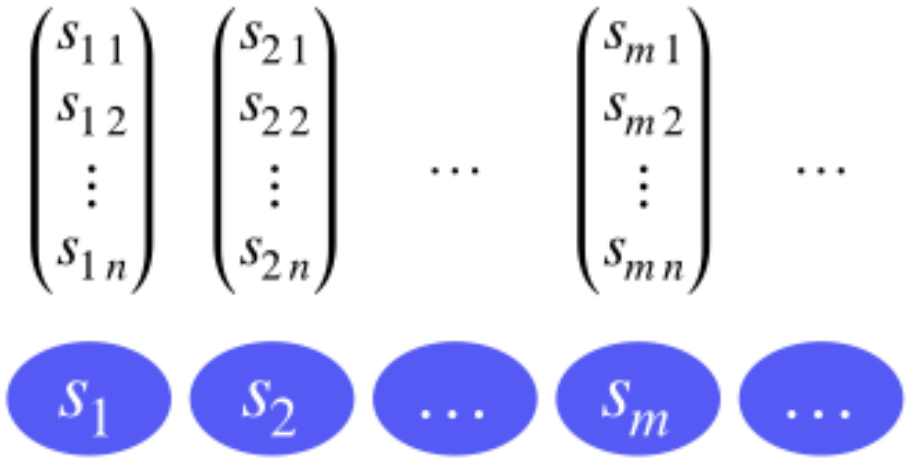
Differential Hoare Logic

Jonathan Julián Huerta y Munive

University of Sheffield, UK

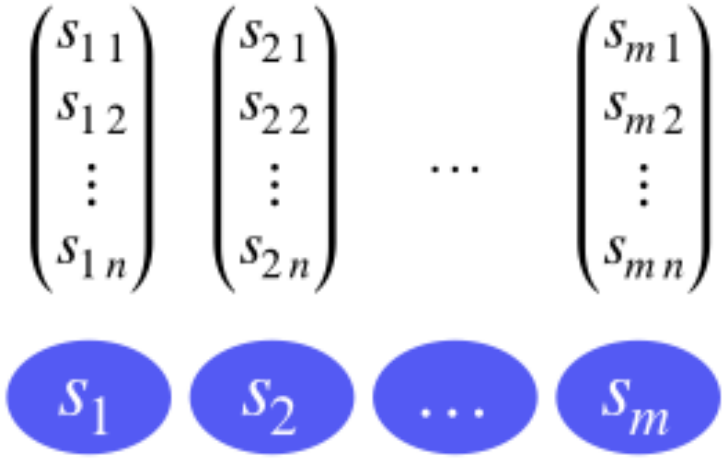
Programs



Modelled as

- Sequences of states ($V \to \mathbb{R}$).
- Monoid actions $\varphi : T \to S \to S$

Programs



Modelled as

- Sequences of states ($V \to \mathbb{R}$).
- Monoid actions $\varphi : T \to S \to S$
- Relations (input/output) $R \subseteq S \times S$
- Functions $F : S \to \mathcal{P}\, S$

Hoare triple

$$\{P\}\alpha\{Q\}$$

If a system starts satisfying a precondition P, then after the execution of program α (assuming α halts), the system satisfies postcondition Q.

Partial correctness specification

Syntax of Hoare Logic

Programs

$$\alpha, \beta ::= a \mid \alpha; \beta \mid \text{if } \varphi \text{ then } \alpha \text{ else } \beta \mid \text{while } \varphi \text{ do } \alpha$$

Formulas

$$\varphi, \psi ::= p \mid \neg\varphi \mid \varphi \to \psi$$

Hoare Triples

$$\{\varphi\}\alpha\{\psi\}$$

Propositional

Syntax of Hoare Logic

Programs

$$\alpha, \beta ::= \mathit{skip} \mid x := t \mid \alpha; \beta \mid \text{if } \varphi \text{ then } \alpha \text{ else } \beta \mid \text{while } \varphi \text{ do } \alpha$$

Formulas

$$t ::= x \mid f\ t_1 \ldots t_n$$
$$\varphi, \psi ::= P\ t_1 \ldots t_n \mid \neg\varphi \mid \varphi \to \psi \mid \forall x.\ \varphi$$

Hoare Triples

$$\{\varphi\}\alpha\{\psi\}$$

Predicate

Rules of Hoare Logic

$$\frac{}{\{\varphi\}\mathit{skip}\{\varphi\}} \qquad \frac{}{\{\varphi[t/x]\}x := t\{\varphi\}} \qquad \frac{\{\varphi\}\alpha\{\gamma\} \quad \{\gamma\}\beta\{\psi\}}{\{\varphi\}\alpha;\beta\{\psi\}}$$

$$\frac{\{\rho \wedge \varphi\}\alpha\{\psi\} \quad \{\neg\rho \wedge \varphi\}\beta\{\psi\}}{\{\varphi\}\text{if } \rho \text{ then } \alpha \text{ else } \beta\{\psi\}} \qquad \frac{\{\varphi \wedge \rho\}\alpha\{\psi\}}{\{\varphi\}\text{while } \rho \text{ do } \alpha\{\neg\rho \wedge \psi\}}$$

$$\frac{\varphi_1 \to \varphi_2 \quad \{\varphi_2\}\alpha\{\psi_2\} \quad \psi_2 \to \psi_1}{\{\varphi_1\}\alpha\{\psi_1\}}$$

over the underlying logic

Semantics

Programs are nondeterministic functions $S \to \mathcal{P}\, S$:

- $f \circ_K g : s \mapsto \bigcup\{g\, \sigma \mid \sigma \in f\, s\}$
- $f_K^0 : s \mapsto \{s\}$ and $f_K^{n+1} = f \circ_K f_K^n$
- $f^* : s \mapsto \bigcup_{n \in \mathbb{N}} f_K^n\, s$

Semantics given by:

- $[\![skip]\!] : s \mapsto \{s\}$
- $[\![x := t]\!] : s \mapsto \{s[x \mapsto [\![t]\!]_s]\}$
- $[\![\alpha; \beta]\!] = [\![\alpha]\!] \circ_K [\![\beta]\!]$
- $[\![\text{if } \rho \text{ then } \alpha \text{ else } \beta]\!] : s \mapsto \text{if } [\![\rho]\!]_s = 1 \text{ then } [\![\alpha]\!]\, s \text{ else } [\![\beta]\!]\, s$
- $[\![\text{while } \rho \text{ do } \alpha]\!] = [\![\rho; \alpha]\!]^* \circ_K [\![\neg\rho]\!]$

Identify $\{s \mid [\![\varphi]\!]_s = 1\}$ $\qquad s \mapsto \{\sigma \mid \sigma = s \wedge [\![\varphi]\!]_s = 1\}$

$$[\![\{\varphi\}\alpha\{\psi\}]\!] \equiv (\bigwedge s\ s'.\ [\![\varphi]\!]_s = 1 \Rightarrow s' \in [\![\alpha]\!]\, s \Rightarrow [\![\psi]\!]_{s'} = 1)$$

Simple verification

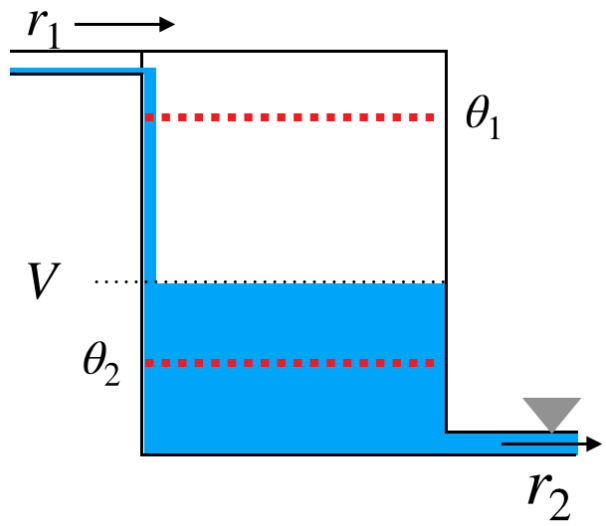
Integer Division[1]

$\{x \geq 0\}$
$q := 0;$
$r := x;$
while $y \leq r$ do
    $q := q + 1$
    $r := r - y$
$\{x = qy + r \wedge r \geq 0 \wedge r < y\}$

by `hoare auto`

Done with proof assistant Isabelle

Hybrid System Verification

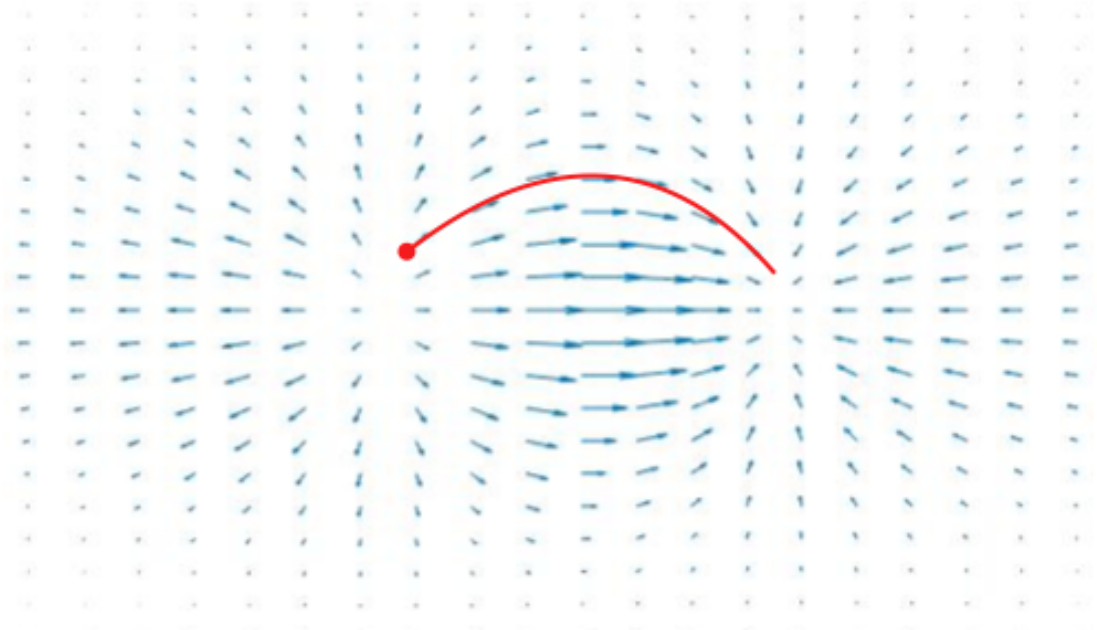


$$\begin{aligned} \text{dynamics} &= h' = f\ h \\ \text{safe} &= \theta_2 \leq h \leq \theta_1 \\ \text{tank} &= (\text{program}; \text{dynamics})^* \\ \{\text{safe}\}\ &\text{tank}\ \{\text{safe}\} \end{aligned}$$

How to put ODEs in Hoare logic?

vector field

$$X'\ \tau = f\ (X\ \tau)$$



where

$$X : T \to S \qquad f : S \to S \qquad X\,0 = s$$

Orbit map $s \mapsto \{X\,\tau \mid \tau \in T\}$

Differential Hoare Logic

Syntax

$\alpha, \beta ::= \mathit{skip} \mid x := t \mid x' = t \;\&\; \gamma \mid \alpha; \beta \mid \text{if } \varphi \text{ then } \alpha \text{ else } \beta \mid \text{while } \varphi \text{ do } \alpha$

Semantics $S \to \mathcal{P}\, S$

- $\text{orbit}_\gamma^X\, s = \{X\, \tau \mid \tau \in T \text{ and } (\bigwedge r \in [0, \tau].\ [\![\psi]\!]_{X\, r} = 1)\}$
- $[\![x' = t \;\&\; \gamma]\!]\, s = \bigcup \{\text{orbit}_\gamma^X\, s \mid X \in \text{Sols}\, t\, T\, s\}$ (orbital)
- $\text{Sols}\, t\, T\, s = \{X : T \to S \mid \bigwedge \tau \in T.\ X'\, \tau = [\![t]\!]_{X\, \tau} \text{ and } X\, 0 = s\}$

Hoare rule

$$\frac{}{\{\forall t_1 \geq 0.\ (\forall t_2.\ 0 \leq t_2 \leq t_1 \to \gamma[\kappa/x]) \to \psi[\kappa/x]\} x' = t \;\&\; \gamma \{\psi\}}$$

provided $T \to S \to S : \tau \mapsto s \mapsto [\![\kappa]\!]_{X\, \tau}$ is the flow
for the Lipschitz continuous $s \mapsto [\![t]\!]_s$

Verification Example

abbreviation $G\ h_x\ k \equiv \mathbf{U}(t \leq (h_x - h_0)/k)$

abbreviation $I\ h_l\ h_h \equiv \mathbf{U}(h_l \leq h \wedge h \leq h_h \wedge (\pi = 0 \vee \pi = 1))$

abbreviation $f\ k \equiv [\pi \mapsto_s 0,\ h \mapsto_s k,\ h_0 \mapsto_s 0,\ t \mapsto_s 1]$

abbreviation $\varphi\ k\ \tau \equiv [h \mapsto_s k * \tau + h,\ t \mapsto_s \tau + t,\ h_0 \mapsto_s h_0,\ \pi \mapsto_s \pi]$

abbreviation $ctrl\ h_l\ h_h \equiv (t ::= 0);(h_0 ::= h);$
$(IF\ (\pi = 0 \wedge h_0 \leq h_l + 1)\ THEN\ (\pi ::= 1)\ ELSE$
$(IF\ (\pi = 1 \wedge h_0 \geq h_h - 1)\ THEN\ (\pi ::= 0)\ ELSE\ skip))$

abbreviation $dyn\ c_i\ c_o\ h_l\ h_h\ \tau \equiv (IF\ (\pi = 1)\ THEN$
$(x' = f\ (c_i - c_o)\ \&\ G\ h_h\ (c_i - c_o)\ on\ \{0..\tau\}\ UNIV\ @\ 0)$
$ELSE\ (x' = f\ (-c_o)\ \&\ G\ h_l\ (-c_o)\ on\ \{0..\tau\}\ UNIV\ @\ 0))$

abbreviation $tank\ c_i\ c_o\ h_l\ h_h\ \tau \equiv LOOP\ (ctrl\ h_l\ h_h\ ;\ dyn\ c_i\ c_o\ h_l\ h_h\ \tau)$

lemma *tank-flow*:
assumes $0 \leq \tau$ **and** $0 < c_o$ **and** $c_o < c_i$
shows $\{I\ h_l\ h_h\}\ tank\ c_i\ c_o\ h_l\ h_h\ \tau\ \{I\ h_l\ h_h\}$

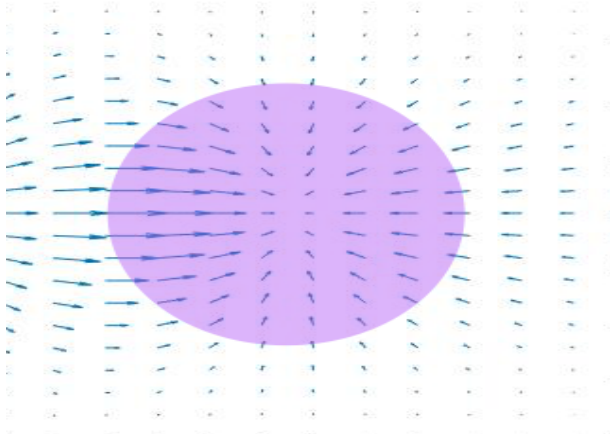
Work Flow

1. write specification statement for hybrid program
   $\{\text{safe}\}\,(\text{program}; \text{dynamics})^*\,\{\text{safe}\}$
2. supply flow for system of ODEs
3. certify unique solution property (via Picard-Lindelöf)
4. generate verification conditions with $\mathrm{d}\mathcal{H}$
5. solve them (. . . if you can)

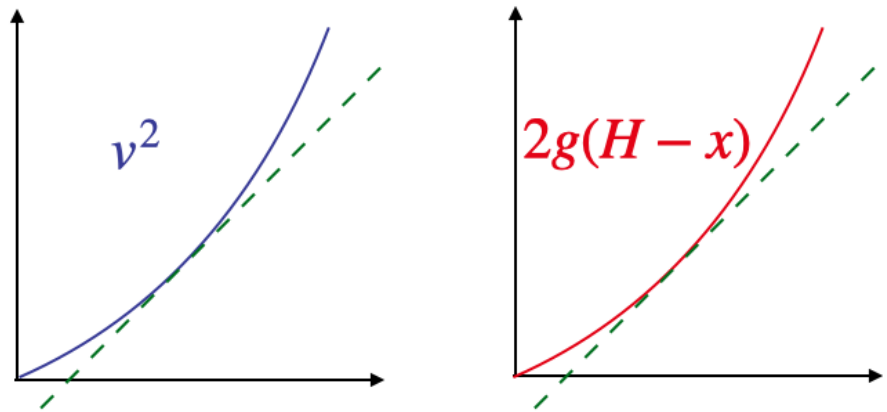
Invariants in d$\mathcal{H}$

Precedents

Invariant sets:



Mean value theorem:



Definition

The formula ι is a d$\mathcal{H}$-invariant for program α iff $\{\iota\}\alpha\{\iota\}$ holds.

$$x' = v \wedge v' = -g \Rightarrow (v^2)' = (2g(H-x))'$$
$$\Leftrightarrow 2vv' = 2gH' - 2x'g$$
$$\Leftrightarrow -2vg = -2vg$$

$\therefore v^2 = 2g(H-x)$ is an invariant

Workflow with invariants

Invariants for vector fields

$$[\![\{\iota\}x' = t \;\&\; \gamma\{\iota\}]\!] \Leftrightarrow \left(\bigcup\{[\![x' = t \;\&\; \gamma]\!]\, s \mid [\![\iota]\!]_s = 1\} \subseteq \{s \mid [\![\iota]\!]_s = 1\}\right)$$

Lemma (Invariant rules[2])

For continuous $s \mapsto [\![t]\!]_s$, differentiable $\mu, \nu : S \to \mathbb{R}$, and $T \subseteq \mathbb{R}$.

1. *If $(\mu \circ X)' = (\nu \circ X)'$ for all $X \in \mathsf{Sols}\, t\, T\, s$, then $[\![\{\mu = \nu\}\, x' = t \;\&\; \gamma\, \{\mu = \nu\}]\!]$,*
2. *if $(\mu \circ X)'\, \tau \leq (\nu \circ X)'\, \tau$ when $\tau > 0$, and $(\mu \circ X)'\, \tau \geq (\nu \circ X)'\, \tau$ when $\tau < 0$, for all $X \in \mathsf{Sols}\, f\, T\, s$, then $[\![\{\mu < \nu\}\, x' = t \;\&\; \gamma\, \{\mu < \nu\}]\!]$.*

Rule

$$\frac{\varphi \to \iota \quad \{\iota\}x' = t \;\&\; \gamma\{\iota\} \quad \gamma \wedge \iota \to \psi}{\{\varphi\}x' = t \;\&\; \gamma\{\psi\}}$$

Verification Example

abbreviation $dI\ h_l\ h_h\ k \equiv$
$\mathbf{U}(h = k \cdot t + h_0 \wedge 0 \leq t \wedge h_l \leq h_0 \wedge h_0 \leq h_h \wedge (\pi = 0 \vee \pi = 1))$

lemma *tank-diff-inv*:
$0 \leq \tau \Longrightarrow$ *diff-invariant* $(dI\ h_l\ h_h\ k)\ (f\ k)\ \{0..\tau\}$ *UNIV 0 Guard*
⟨*proof*⟩

lemma *tank-inv*:
assumes $0 \leq \tau$ **and** $0 < c_o$ **and** $c_o < c_i$
shows $\{I\ h_l\ h_h\}$ *tank-dinv* $c_i\ c_o\ h_l\ h_h\ \tau\ \{I\ h_l\ h_h\}$
apply(*hyb-hoare* $\mathbf{U}(I\ h_l\ h_h \wedge t = 0 \wedge h_0 = h)$)
prefer *4* **prefer** *7* **using** *tank-diff-inv assms* **apply** *force*+
using *assms tank-inv-arith1 tank-inv-arith2* **by** *rel-auto*′

`https://github.com/yonoteam/HybridKAT`

Conclusions

- First standalone Hoare logic $\mathrm{d}\mathcal{H}$ for hybrid programs
- Details:
  - Minimal logic for verification of hybrid systems
  - Shallow embedding in Isabelle/HOL using KAT
  - Formalised with relational and state transformer semantics
- Future work:
  - Integration with a computer algebra system
  - Extended case study
  - Extend to total correctness

References



V. B. F. Gomes and G. Struth.
Program construction and verification components based on Kleene algebra.
Archive of Formal Proofs, 2016.



J. J. Huerta y Munive and G. Struth.
Predicate transformer semantics for hybrid systems: Verification components for Isabelle/HOL.
arXiv:1909.05618 [cs.LO], 2019.