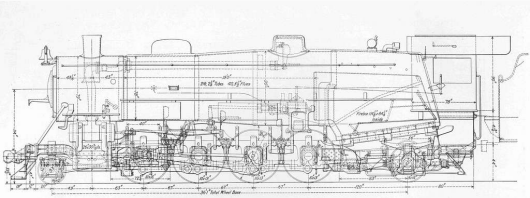
Uniform Substitution for Differential Refinement Logic

Enguerrand Prebet André Platzer

5th July 2024

Karlsruhe Institute of Technology

Cyber-Physical Systems

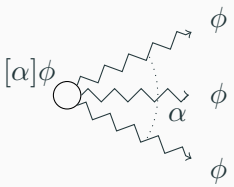


$\Updownarrow$

[train]safe(x) ?

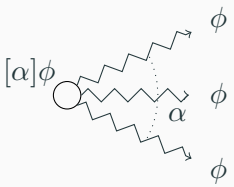
Relating the two systems...	$\rightarrow$ refinement [**?**]
...in a trusted way...	$\rightarrow$ uniform substitution
...and automatically.	$\rightarrow$ decision procedure

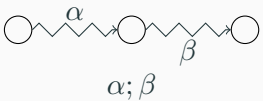
Proving safety for CPS



$$[(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1)^*]x < m$$

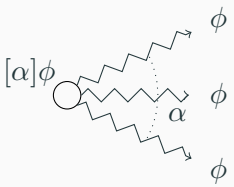
Proving safety for CPS

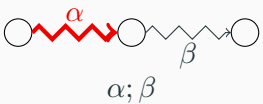




$$[\big(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1\big)^*]x < m$$

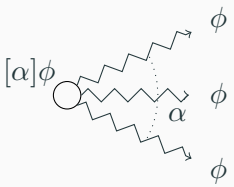
Proving safety for CPS

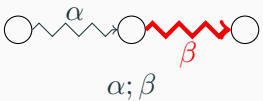




$$[\big(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1\big)^*]x < m$$

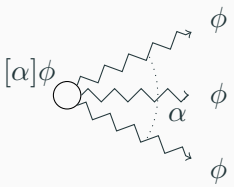
Proving safety for CPS

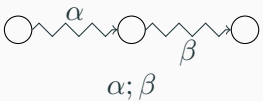




$$[(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1)^*]x < m$$

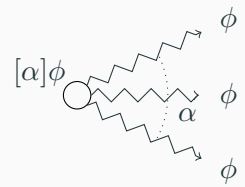
Proving safety for CPS

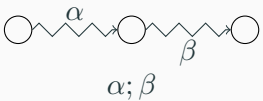


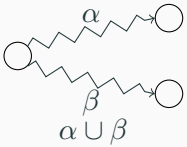


$$[(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1)^*]x < m$$

Proving safety for CPS

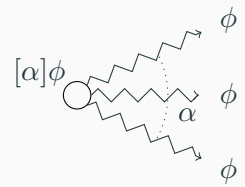


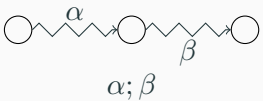


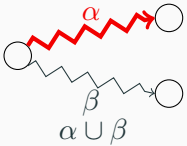


$$[\big(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1\big)^*]x < m$$

Proving safety for CPS

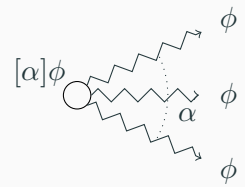


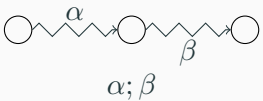


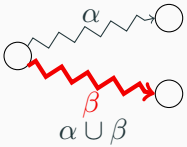


$$[(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1)^*]x < m$$

Proving safety for CPS

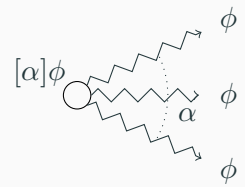


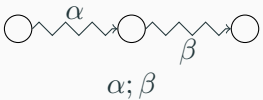


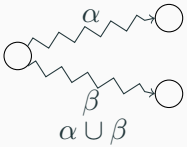


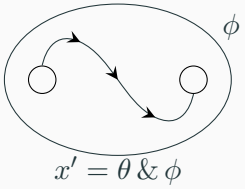
$$[(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1)^*]x < m$$

Proving safety for CPS









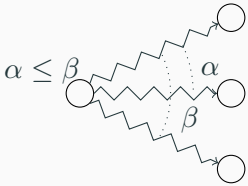
$$[(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1)^*]x < m$$

Differential Refinement Logic (dRL)

FOL $\subseteq$ dL $\subseteq$ dRL

Term:	$\theta, \eta ::= x \mid \theta + \eta \mid \theta \cdot \eta \mid (\theta)'$
Formula:	$\phi, \psi ::= \theta \leq \eta \mid \neg\phi \mid \phi \wedge \psi \mid \forall x\, \phi \mid [\alpha]\phi \mid \alpha \leq \beta$
Program:	$\alpha, \beta ::= ?\psi \mid x := \theta \mid x := * \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid x' = \theta \,\&\, \psi$

Using refinement



Abstraction/rewriting tool:

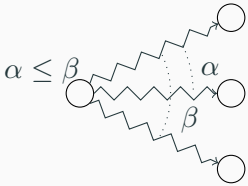
$$\alpha \leq \beta \to ([\beta]\phi \to [\alpha]\phi)$$

$$(x := 3; x := x + 2) = x := 5$$

State dependent:

$$[x := y + 1]\big((?x \leq y; \alpha \cup ?x > y; \beta) = \beta\big)$$

Using refinement



Abstraction/rewriting tool:

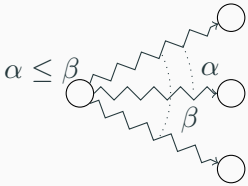
$$\alpha \leq \beta \to ([\beta]\phi \to [\alpha]\phi)$$

$$(x := 3; x := x + 2) = x := 5$$

State dependent:

$$[x := y + 1]\big((?x \leq y; \alpha \cup ?x > y; \beta) = \beta\big)$$

Using refinement



Abstraction/rewriting tool:

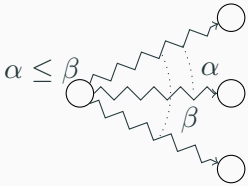
$$\alpha \leq \beta \to ([\alpha]\phi \leftarrow [\beta]\phi)$$

$$(x := 3; x := x + 2) = x := 5$$

State dependent:

$$[x := y + 1]\big((?x \leq y; \alpha \cup ?x > y; \beta) = \beta\big)$$

Using refinement



Abstraction/rewriting tool:

$$\alpha \leq \beta \to ([\alpha]\phi \leftarrow [\beta]\phi)$$

$$(x := 3; x := x + 2) = x := 5$$

State dependent:

$$[x := y + 1]\big((\text{if } x \leq y \text{ then } \alpha \text{ else } \beta) = \beta\big)$$

dRL Calculus and Uniform Substitution

(Some) Axioms of dRL

dL axioms:

([∪]) $[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \land [b]p(\bar{x})$

([;]) $[a;b]p(\bar{x}) \leftrightarrow [a][b]p(\bar{x})$

(G) $\dfrac{p(\bar{x})}{[a]p(\bar{x})}$

dRL-specific axioms:

([≤]) $a \leq b \rightarrow ([a]p(\bar{x}) \leftarrow [b]p(\bar{x}))$

(;) $a;b \leq c;d \leftarrow a \leq c \land [a]b \leq d$

($\leq_t$) $a \leq b \leftarrow (a \leq c \land c \leq b)$

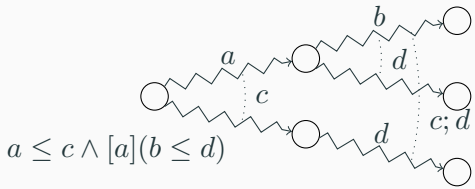
(unloop) $a^* \leq b^* \leftarrow [a^*](a \leq b)$

(ODE) $x' = f(x)\,\&\,p(x) \leq x' = g(x)\,\&\,q(x) \leftrightarrow [x' = f(x)\,\&\,p(x)](x' = g(x) \land q(x))$

KAT axioms,...

Axioms soundness

(;) $a;b \leq c;d \leftarrow a \leq c \wedge [a]b \leq d$



(unloop) $a^* \leq b^* \leftarrow [a^*](a \leq b)$

$[a^*](a \leq b) \rightarrow (\underbrace{a \leq b \wedge [a]a \leq b}_{a;a \leq b;b} \wedge [a;a]a \leq b \wedge \cdots)$

(ODE) $x' = f(x)\,\&\,p(x) \leq x' = g(x)\,\&\,q(x) \leftrightarrow [x' = f(x)\,\&\,p(x)](x' = g(x) \wedge q(x))$

Axioms soundness

(;) $a;b \leq c;d \leftarrow a \leq c \wedge [a]b \leq d$

(unloop) $a^* \leq b^* \leftarrow [a^*](a \leq b)$

$[a^*](a \leq b) \rightarrow (a \leq b \wedge [a]a \leq b \wedge [a;a]a \leq b \wedge \cdots)$

(ODE) $x' = f(x)\,\&\,p(x) \leq x' = g(x)\,\&\,q(x) \leftrightarrow [x' = f(x)\,\&\,p(x)](x' = g(x) \wedge q(x))$

Axioms soundness

(;) $a;b \leq c;d \leftarrow a \leq c \wedge [a]b \leq d$

(unloop) $a^* \leq b^* \leftarrow [a^*](a \leq b)$

$$[a^*](a \leq b) \rightarrow (\underbrace{a \leq b \wedge [a]a \leq b}_{a;a \leq b;b} \wedge [a;a]a \leq b \wedge \cdots)$$

(ODE) $x' = f(x)\,\&\,p(x) \leq x' = g(x)\,\&\,q(x) \leftrightarrow [x' = f(x)\,\&\,p(x)](x' = g(x) \wedge q(x))$

Axioms soundness

(;) $a;b \leq c;d \leftarrow a \leq c \wedge [a]b \leq d$

(unloop) $a^* \leq b^* \leftarrow [a^*](a \leq b)$

$[a^*](a \leq b) \rightarrow (\underbrace{a \leq b \wedge [a]a \leq b}_{a;a \leq b;b} \wedge [a;a]a \leq b \wedge \cdots)$

(ODE) $x' = f(x)\,\&\,p(x) \leq x' = g(x)\,\&\,q(x) \leftrightarrow [x' = f(x)\,\&\,p(x)](x' = g(x) \wedge q(x))$

Uniform Substitution

Axioms are formulas, instantiations are recovered by uniform substitution.

$$\text{(US)}\quad \frac{\phi}{\sigma(\phi)} \text{ if } \sigma(\phi) \text{ defined}$$

Small microkernel: 2k lines of code for KeYmaera X + 300 for dRL (70% of axioms)[1]

Example

For $\sigma = \{p \mapsto \phi, a \mapsto \alpha\}$,

$$\text{US}\frac{\text{V}\dfrac{}{p \to [a]p}}{\phi \to [\alpha]\phi} \qquad \text{if } \sigma(p \to [a]p) \text{ is defined}$$

[1] `https://github.com/LS-Lab/KeYmaeraX-release/tree/dRL`

Uniform Substitution

Axioms are formulas, instantiations are recovered by uniform substitution.

$$\text{(US)}\quad \frac{\phi}{\sigma(\phi)} \text{ if } \sigma(\phi) \text{ defined}$$

Small microkernel: 2k lines of code for KeYmaera X + 300 for dRL (70% of axioms)[1]

Example

For $\sigma = \{p \mapsto \phi, a \mapsto \alpha\}$,

$$\text{US}\frac{\text{V}\dfrac{}{p \to [a]p}}{\phi \to [\alpha]\phi} \qquad \text{if } \mathrm{FV}(\phi) \cap \mathrm{BV}(\alpha) = \emptyset$$

[1] `https://github.com/LS-Lab/KeYmaeraX-release/tree/dRL`

Soundness of Uniform Substitution

"Do not introduce new free variables in a context where they are bound."

Static semantics is subtle.

- $\forall x\, x = 2$? x is bound ✓
- $[x := 1 \cup y := 0]y \geq 0$? equivalent to $[x := 1]y \geq 0 \wedge [y := 0]y \geq 0$. y is both?
- $\mathsf{FV}(\alpha \leq \beta) = \mathsf{FV}(\alpha) \cup \mathsf{FV}(\beta) \cup ((\mathsf{BV}(\alpha) \cup \mathsf{BV}(\beta)) \setminus (\mathsf{MBV}(\alpha) \cap \mathsf{MBV}(\beta)))$

Soundness of Uniform Substitution

"Do not introduce new free variables in a context where they are bound."

Static semantics is subtle.

- $\forall x\, x = 2$? x is bound ✓
- $[x := 1 \cup y := 0]y \geq 0$? equivalent to $[x := 1]y \geq 0 \land [y := 0]y \geq 0$. y is both?
- $\mathsf{FV}(\alpha \leq \beta) = \mathsf{FV}(\alpha) \cup \mathsf{FV}(\beta) \cup ((\mathsf{BV}(\alpha) \cup \mathsf{BV}(\beta)) \setminus (\mathsf{MBV}(\alpha) \cap \mathsf{MBV}(\beta)))$

Soundness of Uniform Substitution

"Do not introduce new free variables in a context where they are bound."

Static semantics is subtle.

- $\forall x\, x = 2$? x is bound ✓
- $[x := 1 \cup y := 0]y \geq 0$? equivalent to $[x := 1]y \geq 0 \land [y := 0]y \geq 0$. y is both?
- $\mathsf{FV}(\alpha \leq \beta) = \mathsf{FV}(\alpha) \cup \mathsf{FV}(\beta) \cup ((\mathsf{BV}(\alpha) \cup \mathsf{BV}(\beta)) \setminus (\mathsf{MBV}(\alpha) \cap \mathsf{MBV}(\beta)))$

Decidability

Subset of interest

Theorem

$(ctrl_1; plant_1)^* \leq (ctrl_2; plant_2)^*$ *is decidable.*

Example of a decidable property

$$\big(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1\big)^*$$

$$\vee\mid$$

$$\big(((?\text{safer}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1\big)^*$$

Subset of interest

Theorem

$(ctrl_1; plant_1)^* \leq (ctrl_2; plant_2)^*$ *is decidable.*

- No symbol
- $ctrl_i$ are loop-free
- $ctrl_2$ is idempotent: $ctrl_2; ctrl_2 = ctrl_2$
- $plant_i$ are "nice" enough (e.g. time is explicit)

Example of a decidable property

$$\big(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1\big)^*$$

$$\vee\text{I}$$

$$\big(((?\text{safer}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1\big)^*$$

Subset of interest

Theorem

$(ctrl_1; plant_1)^* \leq (ctrl_2; plant_2)^*$ *is decidable.*

- No symbol
- $ctrl_i$ are loop-free
- $ctrl_2$ is idempotent: $ctrl_2; ctrl_2 = ctrl_2$
- $plant_i$ are "nice" enough (e.g. time is explicit)

Example of a decidable property

$$\Big(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1\Big)^*$$

$$\vee\mid$$

$$\Big(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 0.5\Big)^*$$

Subset of interest

Theorem

$(ctrl_1; plant_1)^* \leq (ctrl_2; plant_2)^*$ *is decidable.*

- No symbol
- $ctrl_i$ are loop-free
- $ctrl_2$ is idempotent: $ctrl_2; ctrl_2 = ctrl_2$
- $plant_i$ are "nice" enough (e.g. time is explicit)

Example of a decidable property

$$\big(((?\text{safe}(x,m); a := A) \cup a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1\big)^*$$

$$\vee\text{I}$$

$$\big((\text{if safe}(x,m) \text{ then } a := A \text{ else } a := -B); t := 0; x' = v, v' = a, t' = 1 \,\&\, t \leq 1\big)^*$$

Proof outline

Handling discrete programs: $ctrl_1 \leq ctrl_2$

Normalise $ctrl_i$ to $x := *; ?\phi_i(x, x^+)$, then $ctrl_1 \leq ctrl_2$ iff $\phi_1(x, x^+) \to \phi_2(x, x^+)$

Handling continuous programs: $[ctrl_1](\underbrace{\bar{y}' = P(\bar{y}, \bar{u}) \,\&\, \phi}_{plant_1} \leq \underbrace{\bar{y}' = Q(\bar{y}, \bar{u}) \,\&\, \psi}_{plant_2})$

(ODE) $x' = f(x) \,\&\, p(x) \leq x' = g(x) \,\&\, q(x) \leftrightarrow [x' = f(x) \,\&\, p(x)](x' = g(x) \wedge q(x))$

Remaining goal: $[ctrl_1][plant_1](P(\bar{y}, \bar{u}) = Q(\bar{y}, \bar{u}) \wedge \psi)$

Decidable when:

- $\bar{y}' = P(\bar{y}, \bar{u})$ solvable
- or ψ of the form $\bigwedge_i \bigvee_j Q_{ij}(x) = 0$ [?]

Proof outline

Handling discrete programs: $ctrl_1 \leq ctrl_2$

Normalise $ctrl_i$ to $x := *; ?\phi_i(x, x^+)$, then $ctrl_1 \leq ctrl_2$ iff $\phi_1(x, x^+) \to \phi_2(x, x^+)$

Handling continuous programs: $[ctrl_1](\underbrace{\bar{y}' = P(\bar{y}, \bar{u}) \,\&\, \phi}_{plant_1} \leq \underbrace{\bar{y}' = Q(\bar{y}, \bar{u}) \,\&\, \psi}_{plant_2})$

(ODE) $\quad x' = f(x) \,\&\, p(x) \leq x' = g(x) \,\&\, q(x) \leftrightarrow [x' = f(x) \,\&\, p(x)](x' = g(x) \wedge q(x))$

Remaining goal: $[ctrl_1][plant_1](P(\bar{y}, \bar{u}) = Q(\bar{y}, \bar{u}) \wedge \psi)$

Decidable when:

- $\bar{y}' = P(\bar{y}, \bar{u})$ solvable
- or ψ of the form $\bigwedge_i \bigvee_j Q_{ij}(x) = 0$ [?]

Proof outline

Handling discrete programs: $ctrl_1 \leq ctrl_2$

Normalise $ctrl_i$ to $x := *; ?\phi_i(x, x^+)$, then $ctrl_1 \leq ctrl_2$ iff $\phi_1(x, x^+) \to \phi_2(x, x^+)$

Handling continuous programs: $[ctrl_1](\underbrace{\bar{y}' = P(\bar{y}, \bar{u}) \,\&\, \phi}_{plant_1} \leq \underbrace{\bar{y}' = Q(\bar{y}, \bar{u}) \,\&\, \psi}_{plant_2})$

(ODE) $x' = f(x) \,\&\, p(x) \leq x' = g(x) \,\&\, q(x) \leftrightarrow [x' = f(x) \,\&\, p(x)](x' = g(x) \wedge q(x))$

Remaining goal: $[ctrl_1][plant_1](P(\bar{y}, \bar{u}) = Q(\bar{y}, \bar{u}) \wedge \psi)$

Decidable when:

- $\bar{y}' = P(\bar{y}, \bar{u})$ solvable
- or ψ of the form $\bigwedge_i \bigvee_j Q_{ij}(x) = 0$ [?]

Conclusion

- Hybrid system refinement
  $\rightarrow$ enables proof to use abstraction/rewriting for CPS
- Uniform substitution-style calculus
  $\rightarrow$ easily implemented in KeYmaera X
- Decidability result on $(ctrl; plant)^*$ (yet to be implemented)
  $\rightarrow$ a first step toward completeness

References

Why implications?

$(\cup_r)$ $\quad a \leq b \cup c \leftarrow a \leq b \vee a \leq c$

With $?\phi; a := A \cup ?\neg\phi; a := -B$ ✓

But not $x := *; ?\phi \cup x := *; ?\psi$

$(;)$ $\quad a; b \leq c; d \leftarrow a \leq c \wedge [a]b \leq d$

$(a; b); c \leq a; (b; c) \not\rightarrow a; b \leq a \wedge [a; b]c \leq b; c.$