Bug Catching: Automated Program Verification
15414/15614 Spring 2024
Lecture 13: Review

Ruben Martins

February 27, 2024

Semantics

Expressions	e	$::=$	$c \mid x \mid e_1 + e_2 \mid e_1 - e_2 \mid e_1 * e_2 \mid \ldots$
Formulas	P, Q	$::=$	$e_1 = e_2 \mid e_1 \leq e_2 \mid \top \mid \bot \mid P \wedge Q \mid P \vee Q$
			$\mid P \rightarrow Q \mid \neg P, \mid \forall x.\, P \mid \exists x.\, P \mid \ldots$
Programs	α, β	$::=$	$x \leftarrow e \mid \alpha \,;\, \beta \mid \text{if } P\, \alpha\, \beta \mid \text{while } P\, \alpha \mid ?P$

Semantics

$\omega[\![e]\!] = c \in \mathbb{Z}$ (The value of e in state ω is c)

$$\begin{array}{lcl} \omega[\![c]\!] & = & c \\ \omega[\![x]\!] & = & \omega(x) \\ \omega[\![e_1 + e_2]\!] & = & \omega[\![e_1]\!] + \omega[\![e_2]\!] \\ \omega[\![e_1 - e_2]\!] & = & \omega[\![e_1]\!] - \omega[\![e_2]\!] \\ \omega[\![e_1 * e_2]\!] & = & \omega[\![e_1]\!] \times \omega[\![e_2]\!] \\ \ldots & & \end{array}$$

Semantics

$\omega \models P$ (Formula P is true in state ω)

$\omega \models \top$	always
$\omega \models \bot$	never
$\omega \models e_1 = e_2$	iff $\omega[\![e_1]\!] = \omega[\![e_2]\!]$
$\omega \models e_1 \leq e_2$	iff $\omega[\![e_1]\!] \leq \omega[\![e_2]\!]$
$\omega \models P \wedge Q$	iff $\omega \models P$ and $\omega \models Q$
$\omega \models P \vee Q$	iff $\omega \models P$ or $\omega \models Q$
$\omega \models \neg P$	iff $\omega \not\models P$
$\omega \models P \rightarrow Q$	iff whenever $\omega \models P$ then also $\omega \models Q$
$\omega \models \forall x. P$	iff $\omega[x \mapsto a] \models P$ for all $a \in \mathbb{Z}$
$\omega \models \exists x. P$	iff $\omega[x \mapsto a] \models P$ for some $a \in \mathbb{Z}$

Semantics

$\omega[\![\alpha]\!]\nu$ (Program α relates prestate ω to poststate ν)

$\omega[\![x \leftarrow e]\!]\nu$	iff $\nu = \omega[x \mapsto c]$ where $c = \omega[\![e]\!]$
$\omega[\![\alpha\ ;\ \beta]\!]\nu$	iff there is a μ such that $\omega[\![\alpha]\!]\mu$ and $\mu[\![\beta]\!]\nu$
$\omega[\![\text{if } P\, \alpha\, \beta]\!]\nu$	iff $\omega[\![\alpha]\!]\nu$ when $\omega \models P$ and $\omega[\![\beta]\!]\nu$ when $\omega \not\models P$
$\omega_0[\![\text{while } P\, \alpha]\!]\omega_n$	iff there exist $\omega_1, \ldots, \omega_{n-1}$ such that for all $0 \leq i < n$ we have $\omega_i \models P$ and $\omega_i[\![\alpha]\!]\omega_{i+1}$ and $\omega_n \not\models P$
$\omega[\![?P]\!]\nu$	iff $\omega \models P$ and $\omega = \nu$

Exercises

Semantics for a let

Define the semantics of a let

$$\text{let } x\ e\ \alpha$$

which locally binds x to the value of e while executing α. At the end of α, the value of x should revert to what it was before the let.

Exercises

Semantics of a for-loop

Define the semantics of a for-loop

$$\text{for } x \; e_1 \; e_2 \; \alpha$$

which goes through the values for x between the values of e_1 and e_2. It starts at the value of e_1 and counts up or down to the value of e_2, inclusively, executing α each time.

Exercises

Semantics of a repeat-until

Define the semantics for a repeat-until loop as an alternative to a while loop.

$$\omega[\![\text{repeat}\,\alpha\,P]\!]\nu$$

Informally, the $\text{repeat}\,\alpha\,P$ loop executes α and then tests P. If P is true it exits the loop, and if P is false it repeats it.

Dynamic Logic

$\omega \models [\alpha]P$ iff for every ν, $\omega[\![\alpha]\!]\nu$ implies $\nu \models P$

$\omega \models \langle\alpha\rangle P$ iff there exists a ν such that $\omega[\![\alpha]\!]\nu$ and $\nu \models P$

Axioms for Dynamic Logic:

$$
\begin{array}{lcll}
[x \leftarrow e]Q(x) & \leftrightarrow & \forall x'.\, x' = e \rightarrow Q(x') & (x' \text{ not in } e \text{ or } Q(x)) \\
[\alpha\, ;\, \beta]Q & \leftrightarrow & [\alpha][\beta]Q & \\
[?P]Q & \leftrightarrow & (P \rightarrow Q) & \\
[\text{if } P\, \alpha\, \beta]Q & \leftrightarrow & (P \rightarrow [\alpha]Q) \wedge (\neg P \rightarrow [\beta]Q) & \\
[\text{while } P\, \alpha]Q & \leftrightarrow & (P \rightarrow [\alpha][\text{while } P\, \alpha]Q) \wedge (\neg P \rightarrow Q) &
\end{array}
$$

Exercises

Wrong assignment

Show that the following axiom for assignment is wrong:

$$[x \leftarrow e]P \leftrightarrow (x = e \rightarrow P) \qquad \text{(WRONG)}$$

Exercises

Sequential composition

Prove one direction of the sequential composition axiom:

$$[\alpha\ ;\ \beta]Q \leftrightarrow [\alpha][\beta]Q$$

Exercises

Before α P

We can extend dynamic logic with a corresponding operator $(\!|\alpha|\!)P$ read as "*before* α P". Its semantics is defined by

$$\omega \models (\!|\alpha|\!)P \quad \text{iff for all } \mu \text{ such that } \mu[\![\alpha]\!]\omega \text{ we have } \mu \models P$$

Write an axiom for $(\!|\alpha\,;\,\beta|\!)P$ and prove it using semantics.

Nondeterministic Dynamic Logic

Programs $\alpha, \beta \quad ::= \quad x \leftarrow e \mid \alpha \, ; \beta \mid ?P \mid \alpha \cup \beta \mid \alpha^*$

$\omega[\![\alpha \cup \beta]\!]\nu$ iff $\omega[\![\alpha]\!]\nu$ or $\omega[\![\beta]\!]\nu$

$\omega_0[\![\alpha^*]\!]\omega_n$ iff there exist $\omega_1, \ldots, \omega_{n-1}$ s.t. $\omega_i[\![\alpha]\!]\omega_{i+1}$ for all $0 \leq i < n$.

Nondeterministic Dynamic Logic

Programs $\alpha, \beta ::= x \leftarrow e \mid \alpha \,;\, \beta \mid ?P \mid \alpha \cup \beta \mid \alpha^*$

$$\begin{array}{ll} \omega[\![\alpha \cup \beta]\!]\nu & \text{iff } \omega[\![\alpha]\!]\nu \text{ or } \omega[\![\beta]\!]\nu \\ \omega_0[\![\alpha^*]\!]\omega_n & \text{iff there exist } \omega_1, \ldots, \omega_{n-1} \text{ s.t.} \\ & \omega_i[\![\alpha]\!]\omega_{i+1} \text{ for all } 0 \leq i < n. \end{array}$$

$$\begin{array}{lcl} \text{if } P\,\alpha\,\beta & \triangleq & (?P \,;\, \alpha) \cup (?\neg P \,;\, \beta) \\ \text{while } P\,\alpha & \triangleq & (?P \,;\, \alpha)^* \,;\, ?\neg P \end{array}$$

Nondeterministic Dynamic Logic

$$\text{Programs} \quad \alpha, \beta \quad ::= \quad x \leftarrow e \mid \alpha \,;\, \beta \mid ?P \mid \alpha \cup \beta \mid \alpha^*$$

$$\begin{array}{ll} \omega[\![\alpha \cup \beta]\!]\nu & \text{iff } \omega[\![\alpha]\!]\nu \text{ or } \omega[\![\beta]\!]\nu \\ \omega_0[\![\alpha^*]\!]\omega_n & \text{iff there exist } \omega_1, \ldots, \omega_{n-1} \text{ s.t.} \\ & \omega_i[\![\alpha]\!]\omega_{i+1} \text{ for all } 0 \leq i < n. \end{array}$$

$$\begin{array}{lcl} \text{if } P\,\alpha\,\beta & \triangleq & (?P \,;\, \alpha) \cup (?\neg P \,;\, \beta) \\ \text{while } P\,\alpha & \triangleq & (?P \,;\, \alpha)^* \,;\, ?\neg P \end{array}$$

$$\begin{array}{lcl} [\alpha \cup \beta]P & \leftrightarrow & [\alpha]P \wedge [\beta]P \\ [\alpha^*]P & \leftrightarrow & P \wedge [\alpha][\alpha^*]P \end{array}$$

Nondeterministic Dynamic Logic

$$[\alpha^*]Q \leftrightarrow Q \land [\alpha^*](Q \rightarrow [\alpha]Q)$$

We define P is valid

$$\omega \models \Box P \text{ iff } \nu \models P \text{ for any } \nu$$

We then can prove an axiom

$$\Box P \rightarrow [\alpha]P$$

Our axiom for reasoning with invariants then becomes

$$[\alpha^*]Q \leftarrow Q \land \Box(Q \rightarrow [\alpha]Q)$$

Strengthening the loop invariant:

$$[\alpha^*]Q \leftarrow J \land \Box(J \rightarrow [\alpha]J) \land \Box(J \rightarrow Q)$$

Nondeterministic Dynamic Logic

Recall the definition

$$\text{while } P\,\alpha \quad \triangleq \quad (?P\,;\,\alpha)^*\,;\,?\neg P$$

We can plug this in to the axiom we have for repetition and reason, assuming we have settled on a loop invariant J.

$$\begin{aligned}
[\text{while } P\,\alpha]Q \quad &\leftrightarrow [(?P\,;\,\alpha)^*\,;\,?\neg P]Q \\
&\leftrightarrow [(?P\,;\,\alpha)^*][?\neg P]Q \\
&\leftrightarrow [(?P\,;\,\alpha)^*](\neg P \to Q) \\
&\leftarrow J \land \Box(J \to [?P\,;\,\alpha]J) \land \Box(J \to (\neg P \to Q)) \\
&\leftrightarrow J \land \Box(J \to (P \to [\alpha]J)) \land \Box(J \land \neg P \to Q) \\
&\leftrightarrow J \land \Box(J \land P \to [\alpha]J) \land \Box(J \land \neg P \to Q)
\end{aligned}$$

Exercises

Find a program that, when substituted for α, makes the judgment hold.

$$\omega[x \mapsto 0, y \mapsto 42] \models [(?(x \neq y); x \leftarrow x + 1; \alpha)^*; (?(x = y))]\bot$$

$$\omega[x \mapsto 0, y \mapsto 0] \models \neg[\alpha](x \neq y \vee \langle\alpha\rangle(x = y))$$

Exercises

Find a state that, when substituted for ω, makes the judgment hold.

$$\omega \models [(?(x \neq y); x \leftarrow x + 1; y \leftarrow y - 1)^*; (?(x = y))](x \neq y)$$

$$\omega \models \neg[(x \leftarrow x + 1; y \leftarrow 2x)^*](x = y)$$

States as Arrays

```
module NDL

  use int.Int

  type state
  type var

  (* "array" operations and axioms *)
  function read (omega : state) (x : var) : int
  function write (omega : state) (x : var) (v : int) : state

  axiom read_eq : forall x y omega v.
    x = y -> read (write omega x v) y = v
  axiom read_ne : forall x y omega v.
    x <> y -> read (write omega x v) y = read omega y

  (* extensionality *)
  axiom ext : forall omega nu.
    (forall x. read omega x = read nu x) -> omega = nu

end
```

Induction in Why3

```
module SimpleInduction

  use Int

  predicate p int

  axiom base: p 0

  axiom induction_step: forall n:int. 0 <= n -> p n -> p (n
      +1)

  lemma SimpleInduction : forall n:int. 0 <= n -> p n

end
```

Exercises

Induction in Why3

Prove the following using induction in Why3:

$$\sum_{i=0}^{i=n} i = \frac{n(n+1)}{2}$$

Exercises

```
theory SumSquare1

  use int.Int
  use int.EuclideanDivision

  let rec function sum (a : int) (b : int) : int =
  variant { b - a }
  if a > b then 0 else sum a (b-1) + b

  predicate sum_square (n : int) =
    n >= 0 -> sum 0 n = div (n*(n+1)) 2

  clone int.SimpleInduction
    with predicate p = sum_square, lemma base, lemma
        induction_step

  goal G : forall n:int. n >= 0 -> sum_square n

end
```

Diamonds

$$\langle x \leftarrow e \rangle Q(x) \leftrightarrow \forall x'. x' = e \rightarrow Q(x') \quad (x' \text{ not in } e,\ Q(x))$$

Diamonds

$$\langle x \leftarrow e \rangle Q(x) \leftrightarrow \forall x'.x' = e \rightarrow Q(x') \quad (x' \text{ not in } e,\ Q(x))$$

$$\langle \alpha\ ;\ \beta \rangle Q \leftrightarrow \langle \alpha \rangle (\langle \beta \rangle Q)$$

Diamonds

$$\langle x \leftarrow e \rangle Q(x) \leftrightarrow \forall x'. x' = e \rightarrow Q(x') \quad (x' \text{ not in } e,\ Q(x))$$

$$\langle \alpha \,;\, \beta \rangle Q \leftrightarrow \langle \alpha \rangle (\langle \beta \rangle Q)$$

$$\langle \alpha \cup \beta \rangle Q \leftrightarrow \langle \alpha \rangle Q \vee \langle \beta \rangle Q$$

Diamonds

$$\langle x \leftarrow e \rangle Q(x) \leftrightarrow \forall x'. x' = e \rightarrow Q(x') \quad (x' \text{ not in } e,\ Q(x))$$

$$\langle \alpha \,;\, \beta \rangle Q \leftrightarrow \langle \alpha \rangle (\langle \beta \rangle Q)$$

$$\langle \alpha \cup \beta \rangle Q \leftrightarrow \langle \alpha \rangle Q \vee \langle \beta \rangle Q$$

$$\langle ?P \rangle Q \quad \leftrightarrow \quad P \wedge Q$$

Diamonds

$$\langle \alpha^* \rangle Q \quad \leftrightarrow \quad Q \vee \langle \alpha \rangle \langle \alpha^* \rangle Q$$

Diamonds

$$\langle \alpha^* \rangle Q \quad \leftrightarrow \quad Q \vee \langle \alpha \rangle \langle \alpha^* \rangle Q$$

$$\begin{aligned} \langle \alpha^* \rangle Q \quad \leftarrow \quad & (\exists n.\, n \geq 0 \wedge V(n)) \\ & \wedge \Box(\forall n.\, n > 0 \wedge V(n) \rightarrow \langle \alpha \rangle V(n-1)) \\ & \wedge \Box(V(0) \rightarrow Q) \\ & (n \text{ not in } \alpha \text{ or } Q) \end{aligned}$$

Exercises

For each of the following two implications, either prove its validity or find a counterexample.

$$[\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)$$

$$\langle\alpha\rangle(P \rightarrow Q) \rightarrow (\langle\alpha\rangle P \rightarrow \langle\alpha\rangle Q)$$

Weakest Preconditions

The weakest precondition can be specified as follows when translated into our semantic framework:

(i) $\text{wp}(\alpha)Q$ is a precondition for Q (it is *sufficient* for Q):

If $\omega \models \text{wp}(\alpha)Q$ *and* $\omega[\![\alpha]\!]\nu$ *then* $\nu \models Q$

(ii) $\text{wp}(\alpha)Q$ is the weakest precondition for Q (it is *necessary* for Q):

Whenever $\omega[\![\alpha]\!]\nu$ *implies* $\nu \models Q$ *for all* ν, *then* $\omega \models \text{wp}(\alpha)Q$.

These two together are precisely the semantic definition of $[\alpha]Q$, namely

$\omega \models [\alpha]Q$ *iff for all* ν *with* $\omega[\![\alpha]\!]\nu$ *we have* $\nu \models Q$

Weakest Preconditions

$$
\begin{array}{lcll}
\mathrm{wp}(\alpha\,;\,\beta)Q & = & \mathrm{wp}(\alpha)(\mathrm{wp}(\beta)Q) & \\
\mathrm{wp}(\alpha \cup \beta)Q & = & \mathrm{wp}(\alpha)Q \wedge \mathrm{wp}(\beta)Q & \\
\mathrm{wp}(?P)Q & = & P \rightarrow Q & \\
\mathrm{wp}(\alpha^*)Q & = & Q \wedge \mathrm{wp}(\alpha)(\mathrm{wp}(\alpha^*)Q) & \\
\mathrm{wp}(x \leftarrow e)Q(x) & = & \forall x'.\, x' = e \rightarrow Q(x') & (x' \notin e, Q(x)) \\
\mathrm{wp}(x \leftarrow e)Q(x) & = & (e/x)(Q(x)) & \text{(equivalently)}
\end{array}
$$

Exercises

Calculate the weakest precondition in each of the following examples.

$\text{wp}(x \leftarrow x + 1)(x = 3)$

$\text{wp}(x \leftarrow x + 1 \cup x \leftarrow x + 2)(x = 3)$

Strongest Postconditions

The strongest postcondition can be specified as follows:

(i) $\text{sp}(\alpha)P$ is a postcondition
for P (it is *necessarily* true after executing α in any state satisfying P):
For all ν *and* ω, *if* $\omega \models P$ *and* $\omega[\![\alpha]\!]\nu$ *then* $\nu \models \text{sp}(\alpha)P$

(ii) $\text{sp}(\alpha)P$ is sufficient for all postconditions of P (it implies all other postconditions):
Whenever $\nu \models \text{sp}(\alpha)P$ *then there is an* ω *such that* $\omega \models P$ *and* $\omega[\![\alpha]\!]\nu$.

Strongest Postconditions

$$
\begin{array}{lcl}
\mathsf{sp}(\alpha\,;\,\beta)Q & = & \mathsf{sp}(\beta)(\mathsf{sp}(\alpha)Q) \\
\mathsf{sp}(\alpha \cup \beta)Q & = & \mathsf{sp}(\alpha)Q \vee \mathsf{sp}(\beta)Q \\
\mathsf{sp}(?P)Q & = & P \wedge Q \\
\mathsf{sp}(\alpha^*)P & = & P \vee \mathsf{sp}(\alpha^*)(\mathsf{sp}(\alpha)P) \\
\mathsf{sp}(x \leftarrow e(x))(P(x)) & = & \exists x'.x = e(x') \wedge P(x') \quad (x' \notin e(x), P(x))
\end{array}
$$

Assignment

The case for assignment is again somewhat tricky. Let's assume our precondition is $P(x)$ and we assign to x. Now P no longer holds of x!

$$
\begin{aligned}
\mathrm{sp}(x \leftarrow 3)(x = 4) &= x = 3 \\
\mathrm{sp}(x \leftarrow x + 1)(x = 4) &= x = 5 \\
\mathrm{sp}(x \leftarrow x + 1)(0 \leq x \leq 3) &= 1 \leq x \leq 4
\end{aligned}
$$

Assignment

The case for assignment is again somewhat tricky. Let's assume our precondition is $P(x)$ and we assign to x. Now P no longer holds of x!

$$\begin{array}{lcl} \mathrm{sp}(x \leftarrow 3)(x = 4) & = & x = 3 \\ \mathrm{sp}(x \leftarrow x + 1)(x = 4) & = & x = 5 \\ \mathrm{sp}(x \leftarrow x + 1)(0 \leq x \leq 3) & = & 1 \leq x \leq 4 \end{array}$$

Let's revisit the examples:

$$\begin{array}{lclll} \mathrm{sp}(x \leftarrow 3)(x = 4) & = & \exists x'. x = 3 \wedge x' = 4 & \text{iff} & x = 3 \\ \mathrm{sp}(x \leftarrow x + 1)(x = 4) & = & \exists x'. x = x' + 1 \wedge x' = 4 & \text{iff} & x = 5 \\ \mathrm{sp}(x \leftarrow x + 1)(0 \leq x \leq 3) & = & \exists x'. x = x' + 1 \wedge 0 \leq x' \leq 3 & \text{iff} & 1 \leq x \leq 4 \end{array}$$

Exercises

Calculate the strongest postcondition in each of the following examples.

$\mathsf{sp}(x \leftarrow x + 1)(x = 3)$

$\mathsf{sp}(x \leftarrow x + 1 \cup x \leftarrow x + 2)(x = 3)$

Sequent Calculus

$$\frac{}{\Gamma, P \vdash P, \Delta}\ \text{id} \qquad \frac{\Gamma \vdash P, \Delta \quad \Gamma, P \vdash \Delta}{\Gamma \vdash \Delta}\ \text{cut}$$

$$\frac{\Gamma, P \vdash \Delta}{\Gamma \vdash \neg P, \Delta}\ \neg R \qquad \frac{\Gamma \vdash P, \Delta}{\Gamma, \neg P \vdash \Delta}\ \neg L$$

$$\frac{\Gamma \vdash P, \Delta \quad \Gamma \vdash Q, \Delta}{\Gamma \vdash P \wedge Q, \Delta}\ \wedge R \qquad \frac{\Gamma, P, Q \vdash \Delta}{\Gamma, P \wedge Q \vdash \Delta}\ \wedge L$$

$$\frac{\Gamma \vdash P, Q, \Delta}{\Gamma \vdash P \vee Q, \Delta}\ \vee R \qquad \frac{\Gamma, P \vdash \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \vee Q \vdash \Delta}\ \vee L$$

$$\frac{\Gamma, P \vdash Q, \Delta}{\Gamma \vdash P \rightarrow Q, \Delta}\ \rightarrow R \qquad \frac{\Gamma \vdash P, \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \rightarrow Q \vdash \Delta}\ \rightarrow L$$

Sequent Calculus

$$\frac{\Gamma \vdash P, P, \Delta}{\Gamma \vdash P, \Delta}\ \text{contraction}R \qquad \frac{\Gamma, P, P \vdash \Delta}{\Gamma, P \vdash \Delta}\ \text{contraction}L$$

$$\frac{\Gamma \vdash P(a), \Delta}{\Gamma \vdash \forall x.\, P(x), \Delta}\ \forall R^{a} \qquad \frac{\Gamma, P(e) \vdash \Delta}{\Gamma, \forall x.\, P(x) \vdash \Delta}\ \forall L$$

$$\frac{\Gamma \vdash P(e), \Delta}{\Gamma \vdash \exists x.\, P(x), \Delta}\ \exists R \qquad \frac{\Gamma, P(a) \vdash \Delta}{\Gamma, \exists x.\, P(x) \vdash \Delta}\ \exists L^{a}$$

Exercises

Show that $P \rightarrow (Q \rightarrow R) \vdash (P \wedge Q) \rightarrow R$.

Show that $((\exists x.P(x)) \rightarrow Q) \vdash \forall x.(P(x) \rightarrow Q)$.