Verification

Lecture 28

Martin Zimmermann

UNIVERSITÄT
DES
SAARLANDES

Plan for today

- Deductive verification
  - First-order logic
  - First-order theories

Review: Annotations

```
@pre 0 ≤ l ∧ u < |a|
@post rv ↔ ∃i. l ≤ i ≤ u ∧ a[i] = e
bool LinearSearch(int[] a, int l, int u, int e) {
  for @ L : l ≤ i ∧ (∀j. l ≤ j < i → a[j] ≠ e)
    (int i := l; i ≤ u; i := i + 1) {
    if (a[i] = e) return true;
  }
  return false;
}
```

Review: Basic paths

@pre $0 \le l \wedge u < |a|$
@post $rv \leftrightarrow \exists i.\ l \le i \le u \wedge a[i] = e$
bool LinearSearch(int[] a, int l, int u, int e) {
  for @ $L : l \le i \wedge (\forall j.\ l \le j < i \rightarrow a[j] \ne e)$
    (int $i := l$; $i \le u$; $i := i + 1$) {
    if ($a[i] = e$) return true;
  }
  return false;
}

(1)	@pre $0 \le l \wedge u < \lvert a \rvert$ $i := l$ @ $L : l \le i \wedge (\forall j.\ l \le j < i \rightarrow a[j] \ne e)$	**(3)**	@ $L : l \le i \wedge (\forall j.\ l \le j < i \rightarrow a[j] \ne e)$ assume $i \le u$; assume $a[i] \ne e$; $i := i + 1$ @ $L : l \le i \wedge (\forall j.\ l \le j < i \rightarrow a[j] \ne e)$
(2)	@ $L : l \le i \wedge (\forall j.\ l \le j < i \rightarrow a[j] \ne e)$ assume $i \le u$; assume $a[i] = e$; $rv :=$ true @post $rv \leftrightarrow \exists i.\ l \le i \le u \wedge a[i] = e$	**(4)**	@ $L : l \le i \wedge (\forall j.\ l \le j < i \rightarrow a[j] \ne e)$ assume $i > u$; $rv :=$ false @post $rv \leftrightarrow \exists i.\ l \le i \le u \wedge a[i] = e$

Review: Verification conditions

$$
\boxed{
\begin{array}{ll}
\textbf{(2)} & @L : F : l \leq i \land (\forall j.\ \ l \leq j < i \to a[j] \neq e) \\
 & S_1 : \texttt{assume}\ i \leq u \\
 & S_2 : \texttt{assume}\ a[i] = e \\
 & S_3 : rv := \texttt{true} \\
 & @\texttt{post}\ \ G : rv \leftrightarrow \exists i.\ \ l \leq i \leq u \land a[i] = e
\end{array}
}
$$

The VC of basic path **(2)** is

$$F \to wp(G, S_1; S_2; S_3).$$

We compute

$$
\begin{array}{ll}
 & wp(G, S_1; S_2; S_3) \\
\Leftrightarrow & wp(wp(rv \leftrightarrow \exists i.\, l \leq i \leq u \land a[i] = e, rv :=\ \texttt{true}), S_1; S_2) \\
\Leftrightarrow & wp(\exists i.\, l \leq i \leq u \land a[i] = e, S_1; S_2) \\
\Leftrightarrow & wp(wp(\exists i.\, l \leq i \leq u \land a[i] = e,\ \texttt{assume}\ a[i] = e), S_1) \\
\Leftrightarrow & wp(a[i] = e \to \exists i.\, l \leq i \leq u \land a[i] = e,\ \texttt{assume}\ i \leq u) \\
\Leftrightarrow & i \leq u \to (a[i] = e \to \exists i.\, l \leq i \leq u \land a[i] = e)
\end{array}
$$

Review: Theorem (Verification Conditions)

If for every basic path

$$
\begin{array}{|l|}
\hline
\quad @\, L_1 : F \\
\\
\quad S_1 \\
\quad \vdots \\
\quad S_n \\
\\
\quad @\, L_j : G \\
\hline
\end{array}
$$

of program P, the verification condition

$$\{F\}\ S_1; \ldots; S_n\ \{G\}$$

is valid, then the annotatons are P-inductive, and therefore P-invariant.

If there is a P-invariant annotation, then P is partially correct.

First-order Logic

Propositional Logic (PL)

PL Syntax

Atom	truth symbols $\top$(“true”) and $\bot$(“false”)
	propositional variables $P, Q, R, P_1, Q_1, R_1, \cdots$
Literal	atom α or its negation $\neg\alpha$
Formula	literal or application of a
	logical connective to formulae F, F_1, F_2

$\neg F$	“not”	(negation)
$F_1 \wedge F_2$	“and”	(conjunction)
$F_1 \vee F_2$	“or”	(disjunction)
$F_1 \rightarrow F_2$	“implies”	(implication)
$F_1 \leftrightarrow F_2$	“if and only if”	(iff)

PL Semantics

Formula F + Interpretation I = Truth value (true, false)

Interpretation

$$I : \{P \mapsto \text{true}, Q \mapsto \text{false}, \cdots\}$$

Evaluation of F under I:

F	$\neg F$
0	1
1	0

where 0 corresponds to value false
1 true

F_1	F_2	$F_1 \wedge F_2$	$F_1 \vee F_2$	$F_1 \rightarrow F_2$	$F_1 \leftrightarrow F_2$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

Satisfiability and Validity

F is satisfiable iff there exists an interpretation I such that $I \models F$.
F is valid iff for all interpretations I, $I \models F$.

$$\boxed{F \text{ is valid iff } \neg F \text{ is unsatisfiable}}$$

Satisifability and validity are decidable (truth tables, BDDs, DPLL, . . .)

Example $\quad F : P \wedge Q \rightarrow P \vee \neg Q$

P Q	$P \wedge Q$	$\neg Q$	$P \vee \neg Q$	F
0 0	0	1	1	1
0 1	0	0	0	1
1 0	0	1	1	1
1 1	1	0	1	1

Thus F is valid.

First-Order Logic (FOL)

Also called Predicate Logic or Predicate Calculus

FOL Syntax

variables	$x, y, z, \cdots$
constants	$a, b, c, \cdots$
functions	$f, g, h, \cdots$
terms	variables, constants or n-ary function applied to n terms as arguments $a, x, f(a), g(x, b), f(g(x, g(b)))$
predicates	$p, q, r, \cdots$
atom	$\top$, $\bot$, or an n-ary predicate applied to n terms
literal	atom or its negation $p(f(x), g(x, f(x))), \quad \neg p(f(x), g(x, f(x)))$

Note: 0-ary functions: constant
0-ary predicates: $P, Q, R, \ldots$

Quantifiers

existential quantifier $\exists x.F[x]$
"there exists an x such that $F[x]$"

universal quantifier $\forall x.F[x]$
"for all x, $F[x]$"

FOL formula literal, application of logical connectives
($\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$) to formulae,
or application of a quantifier to a formula

Example: FOL formula

$$\forall x.\ \underbrace{p(f(x),x) \ \to\ (\exists y.\ \underbrace{p(f(g(x,y)),g(x,y))}_{G}) \ \wedge\ q(x,f(x))}_{F}$$

The scope of $\forall x$ is F.
The scope of $\exists y$ is G.
The formula reads:
"for all x,
if $p(f(x),x)$
then there exists a y such that
$p(f(g(x,y)),g(x,y))$ and $q(x,f(x))$"

FOL Semantics

An interpretation $I : (D_I, \alpha_I)$ consists of:

- Domain D_I
  non-empty set of values or objects
  cardinality $|D_I|$ finite (eg, 52 cards),
  countably infinite (eg, integers), or
  uncountably infinite (eg, reals)
- Assignment α_I
  - each variable x assigned value $x_I \in D_I$
  - each n-ary function f assigned

    $$f_I : D_I^n \to D_I$$

    In particular, each constant a (0-ary function) assigned value $a_I \in D_I$
  - each n-ary predicate p assigned

    $$p_I : D_I^n \to \{\text{true}, \text{false}\}$$

    In particular, each propositional variable P (0-ary predicate) assigned truth value (true, false)

Example:

$$F:\ p(f(x,y),z)\ \rightarrow\ p(y,g(z,x))$$

Interpretation $I : (D_I, \alpha_I)$

$D_I = \mathbb{Z} = \{\cdots, -2, -1, 0, 1, 2, \cdots\}$ integers

$\alpha_I : \{f \mapsto +, g \mapsto -, p \mapsto >\}$

Therefore, we can write

$$F_I : x + y > z\ \rightarrow\ y > z - x$$

(This is the way we'll write it in the future!)

Also

$\alpha_I : \{x \mapsto 13, y \mapsto 42, z \mapsto 1\}$

Thus

$$F_I : 13 + 42 > 1\ \rightarrow\ 42 > 1 - 13$$

Compute the truth value of F under I

1.	I	$\models$	$x + y > z$	since $13 + 42 > 1$
2.	I	$\models$	$y > z - x$	since $42 > 1 - 13$
3.	I	$\models$	F	by 1, 2, and $\rightarrow$

F is true under I

Semantics: Quantifiers

x variable.
x-variant of interpretation I is an interpretation $J : (D_J, \alpha_J)$ such that

- $D_I = D_J$
- $\alpha_I[y] = \alpha_J[y]$ for all symbols y, except possibly x

That is, I and J agree on everything except possibly the value of x

Denote $J : I \triangleleft \{x \mapsto \mathrm{v}\}$ the x-variant of I in which $\alpha_J[x] = \mathrm{v}$ for some $\mathrm{v} \in D_I$. Then

- $I \models \forall x.\, F$ iff for all $\mathrm{v} \in D_I$, $I \triangleleft \{x \mapsto \mathrm{v}\} \models F$
- $I \models \exists x.\, F$ iff there exists $\mathrm{v} \in D_I$ s.t. $I \triangleleft \{x \mapsto \mathrm{v}\} \models F$

Example

For $\mathbb{Q}$, the set of rational numbers, consider

$$F : \forall x.\ \exists y.\ 2 \times y = x$$

Compute the value of F_I (F under I):

Let

$J_1 : I \lhd \{x \mapsto \mathrm{v}\}$	$J_2 : J_1 \lhd \{y \mapsto \frac{\mathrm{v}}{2}\}$
x-variant of I	y-variant of J_1

for $\mathrm{v} \in \mathbb{Q}$.

Then

1.	J_2	$\models$	$2 \times y = x$	since $2 \times \frac{\mathrm{v}}{2} = v$
2.	J_1	$\models$	$\exists y.\ 2 \times y = x$	
3.	I	$\models$	$\forall x.\ \exists y.\ 2 \times y = x$	since $\mathrm{v} \in \mathbb{Q}$ is arbitrary

Satisfiability and Validity

F is satisfiable iff there exists I s.t. $I \models F$
F is valid iff for all I, $I \models F$

F is valid iff $\neg F$ is unsatisfiable

- FOL is undecidable (Turing & Church)
  There does not exist an algorithm for deciding if a FOL formula F is valid, i.e. always halt and says "yes" if F is valid or say "no" if F is invalid.
- FOL is semi-decidable
  There is a procedure that always halts and says "yes" if F is valid, but may not halt if F is invalid.

Semantic Argument Method

Proof rules for propositional logic

$$\frac{I \models \neg F}{I \not\models F} \qquad \frac{I \not\models \neg F}{I \models F}$$

$$\frac{I \models F \wedge G}{\begin{array}{l} I \models F \\ I \models G \end{array}} \leftarrow \text{and} \qquad \frac{I \not\models F \wedge G}{I \not\models F \quad | \quad I \not\models G} \;{}^{\nwarrow}\text{or}$$

$$\frac{I \models F \vee G}{I \models F \quad | \quad I \models G} \qquad \frac{I \not\models F \vee G}{\begin{array}{l} I \not\models F \\ I \not\models G \end{array}}$$

$$\frac{I \models F \rightarrow G}{I \not\models F \quad | \quad I \models G} \qquad \frac{I \not\models F \rightarrow G}{\begin{array}{l} I \models F \\ I \not\models G \end{array}}$$

$$\frac{I \models F \leftrightarrow G}{I \models F \wedge G \quad | \quad I \not\models F \vee G} \qquad \frac{I \not\models F \leftrightarrow G}{I \models F \wedge \neg G \quad | \quad I \models \neg F \wedge G}$$

$$\frac{\begin{array}{l} I \models F \\ I \not\models F \end{array}}{I \models \bot}$$

Semantic Argument Method

Proof rules for quantifiers

$$\frac{I \models \forall x.\, F}{I \lhd \{x \mapsto \mathsf{v}\} \models F} \qquad \frac{I \not\models \exists x.\, F}{I \lhd \{x \mapsto \mathsf{v}\} \not\models F}$$

$$\frac{I \models \exists x. F}{I \lhd \{x \mapsto \mathsf{v}\} \models F} \text{ for a } \textit{fresh } \mathsf{v} \in D_I \qquad \frac{I \not\models \forall x. F}{I \lhd \{x \mapsto \mathsf{v}\} \not\models F} \text{ for a } \textit{fresh } \mathsf{v} \in D_I$$

$$\frac{\begin{array}{l} J : I \lhd \{\cdots \mapsto \cdots\} \models p(s_1, \ldots, s_n) \\ K : I \lhd \{\cdots \mapsto \cdots\} \not\models p(t_1, \ldots, t_n) \end{array}}{I \models \bot} \text{ for all } i \in \{1, \ldots, n\}, \alpha_J[s_i] = \alpha_K[t_i]$$

First-order Theories

First-Order Theories

First-order theory T defined by

- Signature Σ - set of constant, function, and predicate symbols
- Set of axioms A_T - set of closed (no free variables) Σ-formulae

Σ-formula constructed of constants, functions, and predicate symbols from Σ, and variables, logical connectives, and quantifiers

The symbols of Σ are just symbols without prior meaning — the axioms of T provide their meaning

A Σ-formula F is valid in theory T (T-valid, also $T \models F$),
if every interpretation I that satisfies the axioms of T,
i.e. $I \models A$ for every $A \in A_T$ (T-interpretation)
also satisfies F,
i.e. $I \models F$

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation (i.e. satisfies all the axioms of T) that satisfies F

Two formulae F_1 and F_2 are equivalent in T (T-equivalent), if
$T \models F_1 \leftrightarrow F_2$,

i.e. if for every T-interpretation I, $I \models F_1$ iff $I \models F_2$

A fragment of theory T is a syntactically-restricted subset of formulae of the theory.

Example: quantifier-free segment of theory T is the set of quantifier-free formulae in T.

A theory T is decidable if $T \models F$ (T-validity) is decidable for every Σ-formula F,

i.e., there is an algorithm that always terminate with "yes", if F is T-valid, and "no", if F is T-invalid.

A fragment of T is decidable if $T \models F$ is decidable for every Σ-formula F in the fragment.

Theory of Equality T_E

Signature

$$\Sigma_= : \{=, a, b, c, \cdots, f, g, h, \cdots, p, q, r, \cdots\}$$

consists of

- $=$, a binary predicate, interpreted by axioms.
- all constant, function, and predicate symbols.

Axioms of T_E

1. $\forall x.\, x = x$ (reflexivity)
2. $\forall x, y.\, x = y \;\rightarrow\; y = x$ (symmetry)
3. $\forall x, y, z.\, x = y \;\wedge\; y = z \;\rightarrow\; x = z$ (transitivity)
4. for each positive integer n and n-ary function symbol f,
   $\forall x_1, \ldots, x_n, y_1, \ldots, y_n.\, \bigwedge_i x_i = y_i \;\rightarrow\; f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$
   (congruence)
5. for each positive integer n and n-ary predicate symbol p,
   $\forall x_1, \ldots, x_n, y_1, \ldots, y_n.\, \bigwedge_i x_i = y_i \;\rightarrow\; (p(x_1, \ldots, x_n) \leftrightarrow p(y_1, \ldots, y_n))$
   (equivalence)

Congruence and Equivalence are axiom schemata. For example,
Congruence for binary function f_2 for $n = 2$:

$$\forall x_1, x_2, y_1, y_2.\, x_1 = y_1 \;\wedge\; x_2 = y_2 \;\rightarrow\; f_2(x_1, x_2) = f_2(y_1, y_2)$$