Automated Reasoning for Software Engineering

L. Georgieva and A. Ireland

School of Mathematical and Computer Sciences

Heriot-Watt University

Lilia: Office G.50

Email: lilia@macs.hw.ac.uk

Induction. PVS by Examples

- Motivation.
- Axioms for natural numbers.
- Mathematical Induction (weak induction).
- Complete induction (strong induction).
- Application: correctness of loops.

Motivation

Induction is used, e.g.

- to define an infinite domain.
- to prove the properties of an infinite domain.

Examples of infinite domains: the set of natural numbers, set of all predicate logic formulae, languages generated by finite state automata.
These can be defined recursively.

Example: Predicate logic

Recall the definition of formulae in predicate logics:

- If $t_1, \ldots, t_n$ are terms and P is an n-ary predicate symbol, then $P(t_1, \ldots, t_n)$ is an atomic formula.
- If ϕ and ψ are atomic formulae then so are:

$$\neg\phi, \;\; \phi \lor \psi \;\; \phi \land \psi \;\; \phi \rightarrow \psi \;\; \phi \leftrightarrow \psi$$

- If x is a variable and ϕ is a formula then so are $\forall x \phi$ and $\exists x \phi$.

Formula is defined in terms of itself.

Misuse of induction

Consider the function $f(n) = \frac{1}{100.00001n^2 - n^3}$.

$$
\begin{aligned}
f(1) &= 0.01 \\
&\vdots \\
f(4) &= 0.000651 \\
f(5) &= 0.000421 \\
f(6) &= 0.000296
\end{aligned}
$$

Therefore for every $n \geq 1$ we can conclude that $f(n) \leq 0.01$

Wrong! $f(100) = 10$

It is not sufficient to show that ϕ is true for several n to conclude $\forall n \phi$

Peano arithmetic

How do we define the set of natural numbers $\mathbb{N}$ rigorously?
We use 0 and the successor function: $s(0) : \mathbb{N} \longrightarrow \mathbb{N}$.
Then $s(n) = n + 1$ as expected.

1. 0 is a natural number.
2. If n is a natural number, then so is $s(n)$.
3. 0 is not a successor: $\forall x : 0 \neq s(x)$
4. Uniqueness of successors:
$$\forall x \forall y (s(x) = s(y) \longrightarrow x = y)$$
5. The induction postulate:
$$\phi[0/x] \wedge \forall y (\phi[y/x] \longrightarrow \phi[s(y)/x]) \longrightarrow \phi$$

Addition and Multiplication

We can define $+$ using just the axioms:

$$\forall x (x + 0) = x$$

$$\forall (x + s(y)) = s(x + y)$$

How do we prove that $1 + 1 = 2$ in Peano arithmetic:

$$\begin{aligned} 1 + 1 &= \\ &= s(0) + s(0) = \\ &= s(0 + s(0)) = \\ &= s(s(0)) = \\ &= 2 \end{aligned}$$

Definition of multiplication

$$\forall x x * 0 = 0$$

$$\forall x \forall y x * s(y) = x * y + x$$

Exercise: define $<$

Mathematical Induction

Let ϕ be any formula of Peano arithmetic. Then if

- Base step:

$$\vdash \phi[0/n]$$

- Inductive step:

$$\vdash \forall m(\phi[m/n] \longrightarrow \phi[m+1/n])$$

- Then $\vdash \forall n \phi(n)$ by mathematical induction.

Why is this a valid rule of inference?

Look at the sequent calculi rules and try to prove the sequent.

Exercise: Prove by induction that:

$$\forall n(2 * (n+2)) \leq (n+2)^2$$

Changing the Base case

How do we prove that $2n < n!$ for $n \geq$ by induction?

More generally: how do we show that

$$\forall n (n \geq n_0) \rightarrow \phi$$

1. Base step: $\vdash \phi[n_0/n]$
2. Inductive step: Show

$$\vdash \forall m m \geq n_0 \wedge \phi[m/n] \; rightarrow \phi[m+1/n]$$

3. Conclude that $\forall n n \geq n_0 \rightarrow \phi$.

Exercise: prove that $\forall n (n \geq 4) \rightarrow 2^n < n!$

Complete Induction

Theorem: Let ϕ be a formula in Peano arithmetic such that $x \in FV(\phi)$ and y, z do not occur in ϕ. Then:

$$\phi[0/n] \wedge \forall y[\forall z(z \leq y) \longrightarrow \phi[y+1/x] \longrightarrow \forall x \phi$$

is a theorem in Peano arithmetic, i.e. it is true.

Interpretation:

If you can show that

- ϕ is true at 0 and
- By assuming ϕ is true for every natural number upto and including y, you can prove that $\phi[y+1/x]$ is true.

Then conclude that ϕ is true for every natural number.

Complete Induction

Let ϕ be any formula in Peano arithmetic and let x, y, z be the variables as defined in the theorem for complete induction. Then if

1. Base case:

$$\vdash \phi[0/n] \text{ and}$$

2. Inductive step:

$$\vdash \forall y[\forall z(z \leq y \rightarrow \phi[z/x]) \rightarrow \phi[y+1/x]]$$

Then $\vdash \forall n \phi$ by complete induction.

Application: Correctness of Loops

Assertion: Any statement about a program state.

Definition: Let C be a program statement or a sequence of statements, $\{P\}$ be a precondition (assertion on the initial state) of C and Q be a postcondition (assertion on the final state). Then $\{P\}C\{Q\}$ is a Hoare triple. Examples:

1. $\{True\}a := b\{a = b\}$
2. $\{y \neq 0\}x := 1\{x = 1/y\}$

The while rule

Let C be a piece of code such that $\{D \wedge I\}C\{I\}$. Then

$$\{D \wedge I\} \text{ while } D \text{ do } C \; \{\neg D \wedge I\}$$

$\neg D$ is the exit condition and I is the loop invariant.

Application: correctness of loops

Proof of while rule.

Assume that the loop terminated at the nth iteration. Must show that $\neg D \wedge D$ upon termination. However, $\neg D$ must be true upon termination so it remains to show I.
How?: induction.
I is true before entering the loop so I is true for 0 iterations.
Inductive case: assume that I is true for m iterations, where $0 \geq n \geq m$.
Must show that I is true after $m + 1$ iterations.
But D is true before executing C for the $m + 1$st time since the loop does not terminate after m iterations ($m < n$).
Also I is true before the execution by the inductive hypothesis.
$\{D \wedge I\}$ is a precondition for the $m + 1$ st execution C.
Therefore I is a postcondition since $\{D \wedge I\}C\{I\}$. QED (abbreviation for "quod erat demonstrandum" ("that which was to be demonstrated")).

Checklist for proving correctness of loops

- Check that I is true before the loop.
- I is a loop invariant.
- Execution terminates.
- Use $\neg D \wedge I$ to prove the desired property.

Induction: Exercises

Task:

- Prove that for all $n \in \mathbb{N}$ $n^2 + 5 * n + 1$ is odd.
- Prove by hand that $\forall m, n \in \mathbb{N}$ $m \leq n \longrightarrow m^2 < n^2$
- When proving by induction in PVS: use the commands *grind*, *induct*, *induct-and-simplify*

Induction in PVS

Homework: work through the examples of proving properties of natural numbers in PVS as explained in Chapter 2 of the system guide which is available from the exercise directory of the module website.