Induction and Loops

SFWR ENG 2FA3

Ryszard Janicki

Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada

Introduction

The set $\mathbb{N}$ of natural numbers $\{0, 1, 2, \cdots\}$ is infinite

- How to prove properties of such an infinite set?
- It requires a technique that is of fundamental importance in mathematics and computer science: mathematical induction
- We investigate
  1. Mathematical induction
  2. Induction over sets other than $\mathbb{N}$
- We show how properties of an inductively defined function can be proved using induction
- We show how a program loop can be analysed using induction

Induction over the natural numbers

Claim

$$P(n): \quad +(i \mid 1 \le i \le n : 2i - 1) = n^2$$

- *$P(n)$ is a boolean expression*
- *We can view it as a boolean function $P(n : \mathbb{N})$ of its free variable n*

Example

$1 + 3 = (2 \cdot 1 - 1) + (2 \cdot 2 - 1) = 2^2$

Example

$1 + 3 + 5 = (2 \cdot 1 - 1) + (2 \cdot 2 - 1) + (2 \cdot 3 - 1) = 3^2$

Induction over the natural numbers

We can prove $\forall(n \mid 0 < n : P(n))$ as follows:

- First prove $P(0)$
- Then prove that for all $n \geq 0$, if $P(0), \cdots, P(n-1)$ hold, then so does $P(n)$

$$\forall(n : \mathbb{N} \mid 0 < n : P(0) \wedge P(1) \wedge \cdots \wedge P(n-1) \implies P(n))$$

- We do not really have to prove $P(n)$ in this way (suffices to know that in principle we can do so)
- The proofs of $P(0)$ and $\forall(n : \mathbb{N} \mid 0 < n : P(0) \wedge P(1) \wedge \cdots \wedge P(n-1) \implies P(n))$ are all we need to conclude that $P(n)$ holds for all natural numbers

Mathematical induction is formalised as a single axiom in the predicate calculus as follows, where $P : \mathbb{N} \longrightarrow \mathbb{B}$

Axiom (Mathematical Induction over $\mathbb{N}$)

$$\forall(n : \mathbb{N} \mid : \forall(i \mid 0 \le i < n : P(i)) \implies P(n))$$
$$\implies \forall(n : \mathbb{N} \mid : P(n))$$

Theorem (Mathematical Induction over $\mathbb{N}$)

$$\forall(n : \mathbb{N} \mid : \forall(i \mid 0 \le i < n : P(i)) \implies P(n))$$
$$\iff \forall(n : \mathbb{N} \mid : P(n))$$

Induction over the natural numbers

Theorem (Mathematical Induction over $\mathbb{N}$)

$$P(0) \ \wedge \ (\, \forall(n:\mathbb{N} \ |: \ \forall(i \ | \ 0 \leq i \leq n \ : \ P(i)\,) \implies P(n+1)\,)\,)$$
$$\implies \ \forall(n:\mathbb{N} \ |: \ P(n)\,)$$

- Conjunct $P(0)$ is called the base case of the mathematical induction
- $\forall(n:\mathbb{N} \ |: \ \forall(i \ | \ 0 \leq i \leq n \ : \ P(i)\,) \implies P(n+1)\,)$ is called the inductive case of the mathematical induction

Induction over the natural numbers

Definition (Weak Mathematical Induction over $\mathbb{N}$)

$$P(0) \;\wedge\; \forall(n : \mathbb{N} \;|\!:\; P(n) \implies P(n+1)\,)$$
$$\implies \forall(n : \mathbb{N} \;|\!:\; P(n)\,)$$

- Conjunct $P(0)$ is called the base case of the mathematical induction
- $\forall(n : \mathbb{N} \;|\!:\; P(n) \implies P(n+1)\,)$ is called the inductive case of the weak mathematical induction

Induction over the natural numbers

Theorem (Weak Mathematical Induction over $\mathbb{N}$)

$$P(0) \ \wedge \ \ \forall(n:\mathbb{N} \ |: \ P(n) \implies P(n+1)\,)$$
$$\iff \ \ \forall(n:\mathbb{N} \ |: \ P(n)\,)$$

- Proving $\forall(n:\mathbb{N} \ |: \ P(n) \implies P(n+1)\,)$ is often technically easier than proving
  $\forall(n:\mathbb{N} \ |: \ \forall(i \ | \ 0 \leq i \leq n \ : \ P(i)\,) \implies P(n+1)\,)$
- However sometimes we cannot prove $P(n) \implies P(n+1)$, while we can prove $\forall(i \mid 0 \leq i \leq n : P(i)) \implies P(n+1)$.

Induction over the natural numbers

- When proving $\forall(n:\mathbb{N} \mid: P(n))$ by induction, we often prove the base case and inductive case separately and then assert, in English, that $P(n)$ holds for all natural numbers n
- The proof of the inductive case is typically done by proving $\forall(i \mid 0 \leq i \leq n : P(i)) \implies P(n+1)$ for arbitrary $n \geq 0$
- Further, $\forall(i \mid 0 \leq i \leq n : P(i)) \implies P(n+1)$ is usually proved by assuming $\forall(i \mid 0 \leq i \leq n : P(i))$ and then proving $P(n+1)$

Example

We prove $P(n)$ for all natural numbers.

1. $P(n)$: $+(i \mid 1 \leq i \leq n : 2i-1) = n^2$. (weak induction suffices)
2. $P(n)$: If $n \geq 1$, then n is a product of primes. Assume that 1 is a prime number (weak induction does not work).

Examples 1

- $P(n): \ +(i \mid 1 \leq i \leq n \ : \ 2i-1\,) = n^2$, or, in more popular notation, $\sum_{i=1}^{n}(2i-1) = n^2$, for all natural numbers.

  Proof. Weak induction.

  Base: $P(0)$ means $n = 0$

  $$
  \begin{aligned}
  & +(i \mid 1 \leq i \leq 0 \ : \ 2i-1\,) = 0^2 \\
  = \ & \langle \text{ Range is empty!} \rangle \\
  & 0 \\
  = \ & \langle \text{ Arithmetic} \rangle \\
  & 0 = 0^2
  \end{aligned}
  $$

  Base with $n = 0$ is a little bit artificial. It is OK due to the concept of *quantifier*, but usually one starts with $n = 1$.

- $P(n)$: $+(i \mid 1 \leq i \leq n : 2i-1) = n^2$, or, in more popular notation, $\sum_{i=1}^{n}(2i-1) = n^2$, for all natural numbers.
  Proof. Weak induction.
  Inductive case: Assume $P(n)$ is *true*, i.e.
  $+(i \mid 1 \leq i \leq n : 2i-1) = n^2$ (or, $\sum_{i=1}^{n}(2i-1) = n^2$).
  Consider $P(n+1)$:

$$
\begin{aligned}
& +(i \mid 1 \leq i \leq n+1 : 2i-1) \\
= & \langle \text{ Split } \rangle \\
& +(i \mid 1 \leq i \leq n : 2i-1) + 2(n+1) - 1 \\
= & \langle \text{ Inductive hypothesis } \rangle \\
& n^2 + 2(n+1) - 1 \\
= & \langle \text{ Arithmetic} \rangle \\
& n^2 + 2n + 2 - 1 = n^2 + 2n + 1 = (n+1)^2
\end{aligned}
$$

- Hence $P(n)$ holds for all n.

Examples 2

- A natural number n is *prime* iff for any k, m, if $n = k \cdot m$ then either $k = n, m = 1$ or $k = 1, m = n$.
- $P(n)$: If $n \geq 1$, then n is a product of primes.
  Assume that 1 is a prime number (weak induction does not work here and we have to start with $P(1)$!).
- **Base:** $1 = 1 \cdot 1$.
- **Inductive case:** Assume it is *true* for *all* i where $1 \leq i \leq n$. Consider $n + 1$. Note that there always are such k, m that $n + 1 = k \cdot m$, for example $k = 1, m = n + 1$. If $k = 1, m = n + 1$ is the only decomposition then $n + 1$ is a prime so we are done. If $n + 1 = k \cdot m$ where $k \neq 1, m \neq n + 1$, so $1 \leq k, m \leq n$. Then by our *induction hypothesis*, both k and m are products of primes, so $n + 1 = k \cdot m$ is a product of primes as well.

Induction over the natural numbers

- Induction can be performed over any subset $n_0, n_0+1, n_0+2, \cdots$, of the integers

- The only difference in such a proof is the starting point and thus the base case

Theorem (Mathematical Induction over $\{n_0, n_0+1, \cdots\}$)

$$P(n_0) \wedge (\forall(n:\mathbb{N} \mid n_0 \le n : \forall(i \mid n_0 \le i \le n : P(i)) \implies P(n+1)))$$
$$\implies \forall(n:\mathbb{N} \mid n_0 \le n : P(n))$$

Theorem (Weak Mathematical Induction over $\{n_0, n_0+1, \cdots\}$)

$$P(n_0) \wedge \forall(n:\mathbb{N} \mid n_0 \le n : P(n) \implies P(n+1))$$
$$\implies \forall(n:\mathbb{N} \mid n_0 \le n : P(n))$$

Example (Example of a proof by induction)

1. Prove $2n + 1 < 2^n$, for $n \geq 3$
2. Consider a currency consisting of 2-cent and 5-cent coins. Show that any amount above 3 cents can be represented using these coins.
3. Prove $P(n)$: $\exists(h, k \mid 0 \leq h \wedge 0 \leq k : 2h + 5k = n)$

Note that (3) is the formalisation of (2).

Examples 3: less formal style

- Prove $2n + 1 < 2^n$, for $n \geq 3$ (weak induction is enough)
- **Base:** $2 \cdot 3 + 1 = 7 < 8 = 2^3$
- **Inductive case:** Inductive hypothesis here is $2n + 1 < 2^n$.
  Hence:
  $2 \cdot (n + 1) + 1 = 2 \cdot n + 2 + 1 = 2 \cdot n + 1 + 2 < 2^n + 2 < 2 \times 2^n = 2^{n+1}$.

- Consider a currency consisting of 2-cent and 5-cent coins. Show that any amount above 3 cents can be represented using these coins (weak induction does not work).
  Formally: prove
  $P(n): \quad \exists(h, k \mid 0 \leq h \wedge 0 \leq k : 2h + 5k = n)$
- **Base:** Smallest $n > 3$ is 4 and $4 = 2 + 2$.
- **Inductive case:** We have two cases: either the bag contains a 5-cent coin or it does not:
  Case (a) The bag contains a 5-cent coin. Replacing the 5-cent coin by three 2-cent coins yields a bag of coins whose sum is one greater (as $5 + 1 = 3 \cdot 2$), so $P(n+1)$ holds.
  Case (b) The bag contains only 2-cent coins. It has at least two 2-cent coins, since $4 \leq n$. Replacing two 2-cent coins by a 5-cent coin yields a bag whose sum is one greater, so $P(n+1)$ holds.

Inductive definition

Suppose, we want to define b^n for $b : \mathbb{Z}$ and $n : \mathbb{N}$

- $b^n = \cdot(i \mid 1 \leq i \leq n : b)$
- An alternative style:

$$\begin{cases} b^0 & = & 1 \\ b^{n+1} & = & b \cdot b^n \text{ (for } n \geq 0) \end{cases}$$

- Or,

$$\begin{cases} b^0 & = & 1 \\ b^n & = & b \cdot b^{n-1} \text{ (for } n \geq 1) \end{cases}$$

Example

Prove by mathematical induction that for all natural numbers m and n, $b^{m+n} = b^m \cdot b^n$.
Formally: prove $P(n) : (\forall m : \mathbb{N} \mid: b^{m+n} = b^m \cdot b^n)$

- **Base:** For arbitrary m, we have $b^{m+0} = b^m \cdot 1 = b^m \cdot b^0$.
- **Induction case:** For arbitrary m, we prove $b^{m+(n+1)} = b^m \cdot b^{n+1}$ using *inductive hypothesis* $(\forall m \mid: b^{m+n} = b^m \cdot b^n)$.

$$
\begin{aligned}
& b^{m+(n+1)} \\
= & \ \langle \text{ Arithmetic } \rangle \\
& b^{(m+1)+n} \\
= & \ \langle \text{ Inductive hypothesis } P(n) \text{ with } m := m+1 \ \rangle \\
& b^{m+1} \cdot b^n \\
= & \ \langle \text{ Definition } \rangle \\
& b \cdot b^m \cdot b^n \\
= & \ \langle \text{ Associativity and symmetry of } \cdot \ \rangle \\
& b^m \cdot (b \cdot b^n) \\
= & \ \langle \text{ Definition } \rangle \\
& b^m \cdot b^{n+1}
\end{aligned}
$$

Inductive definition

Problem

A model for the number of lobsters caught per year is based on the assumption that the number of lobsters caught in a year is the average of the number caught in the two previous years. At the beginning of the application of this model, $100,000$ *lobsters were caught in year 1 and* $300,000$ *were caught in year 2.*
Define inductively L_n*, where* L_n *is the number of lobsters caught in year n, under the assumption of this model and its initial conditions.*

- Solution:
- Base case: $L_1 = 100,000$ and $L_2 = 300,000$
- Inductive part: $L_n = \frac{L_{n-2}+L_{n-1}}{2}$
- We can use *recursion* to program calculation of L_n (in any programming language).

Inductive definition

Problem

- *A path is 2 metres wide and n metres long. It is to be paved using paving stones of size $1m \times 2m$. In how many ways can the paving be accomplished? Justify your answer.*

- Solution:
- Base case: $p_1 = 1$ and $p_2 = 2$
- We need base case of at least two elements here!
- Inductive part: $p_n = p_{n-1} + p_{n-2}$
- We can use *recursion* to program calculation of L_n (in any programming language).

Pitfalls of Induction

- *"Theorem"*: For each natural number n and real $a \neq 0$, we have $a^n = 1$.
- *"Proof"*: Base: $a^0 = 1$ by the definition of a^0.
  Inductive step: Assume that $a^k = 1$ for all natural k with $k \leq n$. Then note that

  $$a^{n+1} = \frac{a^n \cdot a^n}{a^{n-1}} = \frac{1 \cdot 1}{1} = 1.$$

  So we are done.
- Where is the error!?

Where is the error!?

- Consider the equation used in the "proof".

$$a^{n+1} = \frac{a^n \cdot a^n}{a^{n-1}} = \frac{1 \cdot 1}{1} = 1.$$

- To be properly used this equation must be defined and valid for all natural k with $k \leq n$! And this includes the case $k = 0$!
- For $k = 0$ we have:

$$a^{0+1} = \frac{a^0 \cdot a^0}{a^{0-1}} = \frac{1 \cdot 1}{?} = ?,$$

  since the value a^{-1} is undefined!, so a^1 is not defined and not equal 1. This means we cannot use this equation in any proof that assumes $0 \leq k \leq n$.
- These kind of errors happen more often than you might think, especially when induction is used for complex structures as trees, automata, steps of algorithms, etc.

Peano Arithmetic Again

Definition. The set of natural numbers $\mathbb{N}$, expressed in terms of 0 and a function S (for *successor*), $S : \mathbb{N} \to \mathbb{N}$, is defined as follows.

(a) 0 is a member of $\mathbb{N}$: $0 \in \mathbb{N}$.

(b) If n is in $\mathbb{N}$, then so is $S(n)$: $n \in \mathbb{N} \Rightarrow S(n) \in \mathbb{N}$.
Traditional notation $S(n) = n + 1$.

(c) The element 0 is not the successor of any natural number: $(\forall n : \mathbb{N} \mid: S(n) \neq 0)$.

(d) S is *one-to-one*, i.e. $(\forall n, m : \mathbb{N} \mid: S(n) = S(m) \Rightarrow n = m)$.

(e) If a subset N of $\mathbb{N}$ (i) contains 0 and (ii) contains the successors of all its elements, then $N = \mathbb{N}$:

$$N \subseteq \mathbb{N} \wedge 0 \in N \wedge (\forall n \mid n \in N : S(n) \in N) \Rightarrow N = \mathbb{N}.$$

- Each part of the definition is necessary to define $\mathbb{N}$ unambiguously.
- Part (e) is actually a form of *weak induction*, but expressed in terms of sets instead of predicates.

Induction and well-founded sets

- We now generalise the notion of mathematical induction to deal with sets other than $\mathbb{N}$ and other relations

- For example, we can use mathematical induction to prove properties of binary trees with the relation "tree t' is a subtree of tree t".

- Let $\prec$ be a boolean function of two arguments of type U

- We want to determine the cases in which $\langle U, \prec \rangle$ admits induction (induction over $\langle U, \prec \rangle$ is sound)

- Not every pair $\langle U, \prec \rangle$ admits induction

We write the principle of mathematical induction over $\langle U, \prec \rangle$ as follows

Axiom (Mathematical induction over $\langle U, \prec \rangle$)

$$\forall (x \;|\!:\; P(x)\,) \iff \forall (x \;|\!:\; \forall (y \mid y \prec x : P(y)\,) \implies P(x)\,)$$

- In the case $\langle U, \prec \rangle = \langle \mathbb{N}, < \rangle$ the above formulation reduces to the induction over $\mathbb{N}$
- We want to show that mathematical induction has two characterizations

Induction and well-founded sets

Definition (Minimal element)

Element y is a minimal element of S if $y \in S$ and $\forall(x \mid x \prec y : x \notin S)$

Example

1. For $\langle \mathbb{N}, < \rangle$, the minimal element of any nonempty subset of $\mathbb{N}$ is its smallest element, in the usual sense.
2. Consider $\langle \mathbb{N}, \mathrm{pdiv} \rangle$, where i pdiv j means "i is a divisor of j and $i < j$"
   - Then the subset $S = \{5, 15, 3, 20\}$ has two minimal elements, 5 and 3
3. Consider $\langle \mathbb{P}, \mathrm{pdiv} \rangle$, where $\mathbb{P}$ is the set of prime numbers
   - All elements of $\langle \mathbb{P}, \mathrm{pdiv} \rangle$ are minimal

We use this notion of minimal element to define well foundedness

Definition (Well foundedness)

$\langle U, \prec \rangle$ is well founded if every nonempty subset of U has a minimal element, i.e.,

$$\forall(S \mid S \subseteq U : S \neq \emptyset \iff \exists(x \mid: x \in S \wedge \forall(y \mid y \prec x : y \notin S)))$$

Example

- $\langle \mathbb{N}, < \rangle$ is well founded
- $\langle \mathbb{Z}, < \rangle$ is not well founded
- Let U be the set of all boolean expressions, and let $x \prec y$ mean "x is a proper subexpression of y ", i.e. x is a subexpression of y but x and y are (syntactically) different. Note that a constant or variable contains no proper subexpression. Since any boolean expression contains at least one constant or variable,$(U, \prec)$ is well founded.
- $(\mathbb{Z}, \prec)$ where, for all $b, c \in \mathbb{Z}$:
$$b \prec c \equiv 0 \leq b < v \vee c < 0 \leq b$$
is well founded. For example: $0 \prec 2 \prec 15 \prec -3$.

We now prove a remarkable fact: well foundedness of $\langle U, \prec \rangle$ and mathematical induction over $\langle U, \prec \rangle$ are equivalent

Theorem (Well-Foundness and Induction)

$\langle U, \prec \rangle$ is well founded iff it admits induction.

Proof.

The proof rests on the fact that for any subset S of U we can construct the expression $P(z) \iff z \notin S$, and for any boolean expression $P(z)$ we can construct the set $S = \{z | \neg P(z)\}$ ∎

$$S \neq \emptyset \iff \exists(x \mid: x \in S \wedge \forall(y \mid y \prec x : y \notin S))$$

$$\iff \langle (\neg p \iff q) \iff (p \iff \neg q) \;\; \& \;\; (X \iff Y) \iff (\neg X \iff \neg Y) \;\; \& \;\; \text{Double negation} \rangle$$

$$S = \emptyset \iff \neg(\exists(x \mid: x \in S \wedge \forall(y \mid y \prec x : y \notin S)))$$

$$\iff \langle \text{De Morgan} \;\; \& \;\; \text{Generalised De Morgan} \rangle$$

$$S = \emptyset \iff \forall(x \mid: x \notin S \vee \neg(\forall(y \mid y \prec x : y \notin S)))$$

$$\iff \langle P(z) \iff z \notin S \text{ –replace occurrences of } S \rangle$$

$$\forall(x \mid: P(x)) \iff \forall(x \mid: P(x) \vee \neg(\forall(y \mid y \prec x : P(y))))$$

$$\iff \langle \text{Law of implication} \rangle$$

$$\forall(x \mid: P(x)) \iff \forall(x \mid: \forall(y \mid y \prec x : P(y)) \implies P(x))$$

There is another characterization of well foundedness, in terms of the decreasing finite chain property

- Consider again $\langle U, \prec \rangle$, and define predicate DCF(x):
  DCF(x) : "every decreasing chain beginning with x is finite"

- We formalize the property of finite chain as follows:

Axiom (Finite chain property)

$$\forall(x \ |: \ \forall(y \ | \ y \prec x \ : \ DCF(y)) \implies DCF(x))$$

Definition

$\langle U, \prec \rangle$ is noetherian iff $\forall(x : U \mid: \mathrm{DCF}(x))$

Theorem

$\langle U, \prec \rangle$ is well founded iff $\langle U, \prec \rangle$ is noetherian

Theorem

If $\langle U, \prec \rangle$ admits induction, then $\prec$ if irreflexive, that is,
$\forall(x \mid x \in U : x \not\prec x)$

Theorem

If $\langle U, \prec \rangle$ admits induction, then
$\forall(x, y \mid x, y \in U : x \prec y \implies y \not\prec x)$

The correctness of loops

- We introduce a theorem concerning the while loop

$$\text{while } B \text{ do } S$$

- The proof of the theorem will show how correctness of a loop is inextricably intertwined with induction
- Following the textbook we write often a while loop using the syntax

$$\text{do } B \longrightarrow S \text{ od}$$

  where boolean expression B is called the guard and statement S is called the repetend.
- Hence: while B do $S \equiv$ do $B \longrightarrow S$ od.

The correctness of loops

Example

$\{Q: \; 0 \leq n\}$

$i, p \; := \; 0, 0;$

$\{P: \; 0 \leq i \leq n \; \wedge \; p = i \cdot x\}$

$\text{do } i \neq n \longrightarrow i, p \; := \; i + 1, p + x \text{ od}$

$\{R: \; p = n \cdot x\}$

- This loop execution requires exactly n iterations
- There is a loop invariant P (i.e., $0 \leq i \leq n \; \wedge \; p = i \cdot x$)

Theorem (Fundamental invariance theorem)

Suppose

- $\{P \ \wedge \ B\}\ S\ \{P\}$ *holds, and*
- $\{P\}$ do $B \longrightarrow S$ od $\{\text{true}\}$ *(i.e., execution of the loop begun in a state in which P is true terminates)*

Then $\{P\}$ do $B \longrightarrow S$ od $\{P \ \wedge \ \neg B\}$ *holds.*

Proof.

By the second hypothesis, the loop terminates, say in $n \geq 0$ iterations. It remains to show that $P \wedge \neg B$ holds upon termination. B is *false* upon termination because the loop can terminate only when B becomes *false*. We prove that P is *true* upon termination of the n iterations by proving (by induction) that it is true after i iterations, $0 \leq i \leq n$.
P is *true* before execution of the loop, so P is true after 0 iterations. Hence the base case holds. For the inductive case, assume P is *true* after i ($i < n$) iterations. Iteration $i + 1$ is executed with P and B *true* and consists of executing S . By the first hypothesis of the theorem, P holds after iteration $i + 1$. Hence the inductive case holds. ∎

Example

Prove the following Hoare triple

$$\{P: \ 0 \leq i \leq n \ \wedge \ p = i \cdot x\}$$
$$\text{do } B: \ \ i \neq n \longrightarrow i, p \ := \ i+1, p+x \text{ od}$$
$$\{P \ \wedge \ i = n\}$$

Main Proof Steps:

- We prove the first hypothesis of the theorem

$$\{P \ \wedge \ B\} \ i, p \ := \ i+1, p+x \ \{P\}$$

- We prove the second hypothesis of the theorem (Execution of the loop terminates)

Then we conclude that the above Hoare triple holds

- **Prove the following Hoare triple**

  $\{P: \ 0 \leq i \leq n \ \wedge \ p = i \cdot x\}$
  do $B: \ i \neq n \longrightarrow i, p \ := \ i+1, p+x$ od
  $\{P \ \wedge \ i = n\}$

- We prove the first hypothesis of the theorem, $\{P \wedge B\} i, p := i+1, p+x \{P\}$. To do this, we calculate the precondition $P[i, p := i+1, p+x]$ and show that it is implied by $P \wedge B$.

  $P[i, p := i+1, p+x]$
  $= \langle$Definition of P; textual substitution$\rangle$
  $0 \leq i+1 \leq n \wedge p + x = (i+1) \cdot x$
  $= \langle$Arithmetic$\rangle$
  $-1 \leq i < n \wedge p = i \cdot x$
  $= \langle$Arithmetic$\rangle$
  $i \neq n \wedge 0 \leq i \leq n \wedge p = i \cdot x$
  $= \langle$Definition of B and P $\rangle$
  $B \wedge P$

- Next, we prove the second hypothesis of the theorem. Since initially $i \leq$ and each iteration increases i by 1 , after a finite number of iterations $i = n$ and the loop guard is *false*. Hence, by Fundamental Invariance Theorem (page 34), we conclude that our Hoare tripe holds.

The correctness of loops

$\{P\}$
do $B \longrightarrow S$ od
$\{R\}$

Checklist for proving loop correct

(a) P is true before execution of the loop
(b) P is a loop invariant: $\{P \ \wedge \ B\} \ S \ \{P\}$
(c) Execution of the loop terminates
(d) R holds upon termination: $P \ \wedge \ \neg B \implies R$

The correctness of loops

Example

Use the checklist to prove that the annotation in this program is correct.

$\{0 \leq n\}$
$i, p \ := \ 0, 0;$
$\{\text{invariant } P : 0 \leq i \leq n \ \wedge \ p = i \cdot x\}$
$\text{do } i \neq n \longrightarrow i, p \ := \ i + 1, p + x \text{ od}$
$\{R : \ \ p = n \cdot x\}$

- Proving point (a) requires proving $0 \leq n \Rightarrow P[i, p := 0, 0]$
- We have already proved (b) and (c)!
- Proving point (d) requires proving $\neg B \wedge P \Rightarrow R$.

The correctness of loops

Problem

Use the checklist to prove that the annotation in this program is correct.

$\{Q: \;\; b \geq 0 \;\wedge\; c > 0\}$

$q, r \;:=\; 0, b;$

$\{invariant\ P : b = q \cdot c + r \;\wedge\; 0 \leq r\}$

do $r \geq c \longrightarrow q, r \;:=\; q + 1, r - c$ od

$\{R: \;\; b = q \cdot c + r \;\wedge\; 0 \leq r < c\}$

$\{Q: \; b \geq 0 \; \wedge \; c > 0\}$

$q, r \; := \; 0, b;$

$\{\text{invariant } P : b = q \cdot c + r \; \wedge \; 0 \leq r\}$

do $r \geq c \longrightarrow q, r \; := \; q + 1, r - c$ od

$\{R: \; b = q \cdot c + r \; \wedge \; 0 \leq r < c\}$

(a) We need to prove $Q \Rightarrow P[q, r := 0, b]$.

$$
\begin{aligned}
& P[q, r := 0, b] \\
= \; & \langle \text{ Definition of } P\text{; textual substitution} \rangle \\
& b = 0 \cdot c + b \wedge 0 \leq b \\
\Leftarrow \; & \langle \text{Arithmetic; definition of } Q \; \rangle \\
& Q
\end{aligned}
$$

(b) $\{P \wedge B\} S \{P\}$, hence we have to prove $P \wedge B \Rightarrow P[q, r := q + 1, r - c]$.

$$
\begin{aligned}
& P[q, r := q + 1, r - c] \\
= \; & \langle \text{Definition of } P \text{ and textual substitution} \rangle \\
& b = (q + 1) \cdot c + (r - c) \wedge 0 \leq r - c \\
= \; & \langle \text{Arithmetic} \rangle \\
& b = q \cdot c + r \wedge r \geq c \\
\Leftarrow \; & \langle \text{Definition of } P \text{ and } B \rangle \\
& P \wedge B
\end{aligned}
$$

(c) Note that each iteration decreases r by c ($c > 0$), so that after a finite number of iterations $r < c$ is achieved.

(d) $P \wedge \neg B \Rightarrow$ is obvious. So we are done.

Consider the following program

$$
\begin{array}{l}
\{0 \leq i = I\} \\
q, r \ := \ 0, b; \\
\{\text{invariant } P : 0 \leq i\} \\
\text{do } 0 \neq i \longrightarrow \quad \text{if} \quad \text{true} \longrightarrow i \ := \ i - 1 \\
\qquad\qquad\qquad\quad [] \quad i \neq 1 \longrightarrow i \ := \ i - 2 \\
\qquad\qquad\qquad\quad \text{fi} \\
\{R : \ \ i = 0\}
\end{array}
$$

- It is readily seen that invariant P is initially true, that the repetend maintains P, and that $P \ \wedge \ \neg(0 \neq i) \ \Longrightarrow \ R$
- We can argue that loop terminates after at most I iterations
- More generally, we can prove the following theorem

Termination of loops

Theorem

To prove that

$\{invariant\colon\ P\}$

$\{bound\ function\colon\ T\}$

do $B \longrightarrow S$ od

terminates, it suffices to find a bound function T, i.e., an integer expression T that is an upper bound on the number of iterations still to be performed. Thus, bound function T satisfies:

(a) *T decreases at each iteration: that is, for v a fresh variable,* $\{P \ \wedge \ B\}\ v \ := \ T; S\ \{T < v\}$

(b) *As long as there is another iteration to perform,* $T > 0 : P \ \wedge \ B \implies T > O.$

Proof.

We prove the theorem by induction on the initial value of T ■

{invariant: P}
{bound function: T}
do $B \longrightarrow S$ od
terminates if

(a) T decreases at each iteration: that is, for v a fresh variable, $\{P \ \wedge \ B\}\ v \ := \ T; S\ \{T < v\}$

(b) As long as there is another iteration to perform, $T > 0 : P \ \wedge \ B \ \Longrightarrow \ T > O$.

- *Base*: $T \leq 0$. Since P is initially *true*, from $P \wedge B \Rightarrow T > 0$ (which is equivalent to $P \wedge T \leq 0 \Rightarrow \neg B$), we conclude $\neg B$, so the loop terminates after 0 iterations.
- *Inductive case*: $T > 0$. We assume as inductive hypothesis that the theorem holds for all initial values of $T \leq k$ for some arbitrary integer $k \geq 0$; we prove the theorem for $T = k + 1$. If B is initially *false*, then the loop terminates immediately and the theorem holds. If B is initially *true*, then execution of one iteration decreases T so that $T \leq k$ (while maintaining P); by the inductive hypothesis, further execution of the loop terminates in at most k iterations.

Termination of Loops

- Consider the following program:
  $\{\text{invariant } P : 0 \leq i \leq n \ \wedge \ p = i \cdot x\}$
  $\{\text{bound function } T : n - 1\}$
  do $i \neq n \longrightarrow i, p \ := \ i + 1, p + x$ od
- We will show that this program terminates.
- *Proof*: Each iteration increases i by 1 and thus decreases $n - i$. Second, we prove $P \wedge B \Rightarrow T > 0$, by transforming $P \wedge B$ to $T > 0$.

$$
\begin{aligned}
& \quad 0 \leq i \leq n \wedge p = i \cdot x \wedge i \neq n \\
\Rightarrow & \ \langle \text{Weakening} \rangle \\
& \quad i < n \\
= & \ \langle \text{Arithmetic} \rangle \\
& \quad 0 < n - i \\
= & \ \langle \text{Definition of } T \rangle \\
& \quad 0 < T
\end{aligned}
$$

Remarks:

- This method of proof does not work with all loops
- Termination proofs might use other well-founded sets

Termination proof using other well-founded sets

- Let $choose(x)$ store an arbitrary natural number in variable x in a nondeterministic fashion.
- Consider the following loop where i is an integer:
  $\{Q : true\}$
  $\{\text{invariant } P : true\}$
  $\text{do } i \neq 0 \rightarrow \text{if } i < 0 \text{ then } choose(i) \text{ else } i := i - 1 \text{ od}$
  $\{R : i = 0\}$
- It is easy to see that this loop terminates.
- However, our previous method of proof of termination cannot be used to prove termination, because there is no a priori upper bound on the number of iterations.
- If initially $i < 0$, then the number of iterations is determined by the value chosen for i during the first iteration, and the value chosen for i is unbounded.

Termination proof using other well-founded sets

- Consider a loop do $B \rightarrow S$ od with an invariant P. Assume $(U, \prec)$ is a well founded set (i.e. its admits induction).
- Consider an expression $T : U$. Suppose that each iteration of the loop changes T to a smaller value:
$$\{P \wedge B\} v := T; S\{v \prec T\}$$
- Suppose further that $P \wedge S \Rightarrow (\exists c : U \mid: u \prec T)$.
- Since every decreasing chain is finite, in a finite number of iterations, T will become a minimal element of U, in which case B is *false* and the loop terminates.

- Consider the program
  $\{Q : true\}$
  $\{\text{invariant } P : true\}$
  do $i \neq 0 \rightarrow$ if $i < 0$ then $choose(i)$ else $i := i - 1$ od
  $\{R : i = 0\}$
- To prove it termination, we can use $(\mathbb{N}, \prec)$ where: for all $b, c \in \mathbb{Z}$:
  $$b \prec c \equiv 0 \leq b < v \vee c < 0 \leq b.$$
  For example: $0 \prec 2 \prec 15 \prec -3$, so any negative i is a “boundary".

Example (Factorial)

Consider the following program

Pr: $\quad i := 1;\ factorial := 1;$

$\quad\quad$ while $i < n$ do

$\quad\quad$ begin $i := i + 1;\ factorial := factorial * i$ end

$\quad\quad$ od.

Let $Q = (factorial = i! \wedge i \leq n)$.

We can show (using rules for assignment) that

$\{(factorial = i! \wedge i \leq n) \wedge i < n\}$

$i := i + 1;\ factorial := factorial * i$

$\{factorial = i! \wedge i \leq n\}$,

so Q is the loop invariant.

Since $\neg(x < n) \wedge Q \iff factorial = n!$, we have

$$\{\text{true}\}\ Pr\ \{factorial = n!\}$$

Example (Factorial-continued)

- Let solve:
  $\{\ ?\ \}$
  $i := i + 1;\ factorial := factorial * i$
  $\{factorial = i! \wedge i \leq n\}$.
- From the definition of sequential composition of two assignment statements we have:
  $\{(factorial = i! \wedge i \leq n)[factorial := factorial * i][i := i + 1]\}$
  $i := i + 1;\ factorial := factorial * i$
  $\{factorial = i! \wedge i \leq n\}$.
- Hence:
  $(factorial = i! \wedge i \leq n)[factorial := factorial * i][i := i + 1]$
  $\iff (factorial * i = i! \wedge i \leq n)[i := i + 1] \iff$
  $factorial * (i + 1) = (i + 1)! \wedge i + 1 \leq n \iff$
  $factorial * (i + 1) = i! * (i + 1) \wedge i < n \iff$
  $factorial = i! \wedge i < n \iff (factorial = i! \wedge i \leq n) \wedge i < n$.
- Which means $\{\ ?\ \} = \{(factorial = i! \wedge i \leq n) \wedge i < n\}$.

Theorem (Fundamental invariance theorem)

Suppose

- $\{P \land B\}\ S\ \{P\}$ *holds, and*
- $\{P\}$ do $B \longrightarrow S$ od $\{\text{true}\}$ *(i.e., execution of the loop begun in a state in which P is true terminates)*

Then $\{P\}$ do $B \longrightarrow S$ od $\{P \land \neg B\}$ *holds.*

Checklist for proving loop correct

(a) P is true before execution of the loop

(b) P is a loop invariant: $\{P \land B\}\ S\ \{P\}$

(c) Execution of the loop terminates

(d) R holds upon termination: $P \land \neg B \implies R$

- Finding P is usually not easy! It requires good understanding of loop behaviour and good intutitions of Hare logic.

Summary of Loops: Termination

Theorem

To prove that

$\{invariant: \ P\}$
$\{bound\ function: \ T\}$
do $B \longrightarrow S$ od

terminates, it suffices to find a bound function T, *i.e.*, *an integer expression* T *that is an upper bound on the number of iterations still to be performed. Thus, bound function* T *satisfies:*

(a) T *decreases at each iteration: that is, for* v *a fresh variable,* $\{P \ \wedge \ B\}\ v \ := \ T; S\ \{T < v\}$

(b) *As long as there is another iteration to perform,* $T > 0 : P \ \wedge \ B \ \Longrightarrow \ T > O.$

- Finding a bound function T is usually easier than an invariant P.
- However, this method of proof does not work with all loops. In some cases we need to use well founded sets, with not obvious predicate (relation) $\prec$ (see page 48 of this Lecture Notes).

- The most popular error is based on the following reasoning:
- *"Theorem"*: All cats have the same colour.
- *"Proof"'*: Let n denote the number of cats is set.
  Base: $n = 1$, the set is $\{c_1\}$, one cat has 'the same colour' (we do not consider which colour).
  Inductive step: Assume that for each $n \geq 1$, if C is a set of cats, all cats in C have the same colour. Consider a set $\{c_1, c_2, \ldots, c_n, c_{n+1}\}$ of cats. The set $\{c_1, c_2, \ldots, c_n\}$ contains n cats so all have the same colour. The set $\{c_2, \ldots, c_n, c_{n+1}\}$ also contains n cats so all have the same colour. Hence c_{n+1} has the same colour as c_2, i.e. it has the same colour as any c_i, $1 \leq i \leq n$. So we are done!
- Where is the error!?

Where is the error!?

- The idea of the proof is based on the assumption that $\{c_1, c_2, \ldots, c_n\} \cap \{c_2, \ldots, c_n, c_{n+1}\} \neq \emptyset$.
- This assumption is assumed to hold for all $n \geq 1$.
- But it does not hold for $n = 1$, as $\{c_1\} \cap \{c_2\} = \emptyset$ if $c_1 \neq c_2$!
- This kind of error is much more difficult to detect for more complex mathematical structures.
- One may argue that the statement "All cats have the same colour" does not make sense for a set with one cat only, so the base should consists of two cats, and then it is obviously false.
- However, where to start with is not always obvious for more complex cases.
- When using induction in real problems, always check several 'initial' cases.

A different kind of error

- *"Theorem"*: For all n, we have $n + 1 < n$
- *"Proof"*: Assume it is true for n, so we have $n + 1 < n$. Then
  $(n + 1) + 1 < (n) + 1 = (n + 1)$
  so
  $(n + 1) + 1 < (n + 1)$
  so we are done.
- Where is the error!?

Where is the error!?

- Actually there might be two different but connected errors.
- If n is integer, or real number, we cannot use induction since integers and real numbers are not well founded! Induction requires some beginning.
- If n is a natural number, it has to be true for $n = 0$, which is not as $0 + 1 < 0$ is *false*.
- False proofs of $P(0)$ (or $P(x_0)$) occur seldom, but it might happen for complex mathematical structures.