Review of formal proof techniques

Why do we need proofs in CS

· specification $\Rightarrow$ SW

How do we know that the SW respects the specification?

specification $\Rightarrow$ formal specification

SW ← satisfies? < testing
⋮
proving = understanding how a complex program works

Deductive proof:

- start from a set H of hypotheses (i.e., given statements)
- show that if H is true, then a conclusion C is also true
- this is done through a sequence of steps:
  - for every step a new fact follows from H and/or previously proved facts by some accepted logical principle
  - the final fact of the sequence is C

Note: the hypothesis H may be either true or false

What we have proved when we go from H to C is:
"if H then C"

Note 2: H and C may depend on parameters that affect their truth-value

Example: "If n is even, then n^2 is even"

What does it mean that n is even?

There is an integer k s.t. $n = 2k$.

H : n is even (note: H has n as parameter) 0.2

by Def.: there exist k s.t. $n = 2k$

by rules of mult.: $n^2 = (2 \cdot k)^2 = 2^2 \cdot k^2 = 2 \cdot (2 \cdot k^2)$

by integer arithm.: $2 \cdot k^2 = h$ is an integer

by Def.: $n^2 = 2 \cdot h$ is even

Other ways of stating if-then statements:

if H then C

H implies C

H only if C   C if H

whenever H holds, also C holds

<u>If-and-only-if statements</u>

A if-and-only-if B

if part: A if B   i.e. if B then A

only-if part: A only-if B   i.e., if A then B

To prove "A iff B", we must prove both the "If part" and the "Only-if part"

<u>Example</u>: $\lfloor x \rfloor$ = greatest integer $\leq x$
floor x

$\lceil x \rceil$ = least integer $\geq x$
ceiling x

Prove: Let x be a real number.

Then $\lfloor x \rfloor = \lceil x \rceil$ iff x is an integer.

Proof:

"If-part": we assume x is an integer and prove $\lfloor x \rfloor = \lceil x \rceil$

We use the definition: if x is an integer $\lfloor x \rfloor = x$
$\lceil x \rceil = x$

$\Rightarrow \lfloor x \rfloor = \lceil x \rceil$

"Only-if part": we assume $\lfloor x \rfloor = \lceil x \rceil$ and prove that x is an integer

Def of floor: $\lfloor x \rfloor \leq x$ (1)
" ceiling: $\lceil x \rceil \geq x$ (2)

Hypothesis: $\lfloor x \rfloor = \lceil x \rceil$ (3)

Substituting $\lceil x \rceil$ in place of $\lfloor x \rfloor$, we get from (1) $\lceil x \rceil \leq x$ and with (2) and arithmetic laws, we get $\lceil x \rceil = x$

Since $\lceil x \rceil$ is an integer, so is x

Other forms of proofs:

- Proving equivalences of sets

e.g. show that the language accepted by A_1 is the same as " A_2

To show $E = F$ we have to show
E, F: expressions representing sets

1) $E \subseteq F$, i.e. if $x \in E$ then $x \in F$
2) $F \subseteq E$, i.e. if $x \in F$ then $x \in E$

Example: $R \cup (S \cap T) = (R \cup S) \cap (R \cup T)$

$\underset{E}{|}$ $\underset{F}{|}$

1) If $x \in R \cup (S \cap T)$ then $x \in (R \cup S) \cap (R \cup T)$

See HMU Figure 1.5

2) If $x \in (R \cup S) \cap (R \cup T)$ then $x \in R \cup (S \cap T)$

See HMU Figure 1.6

- Contrapositive:

To prove: "if H then C"

we can prove its contrapositive: "if not C, then not H"

We can easily see that a statement and its contrapositive are logically equivalent (i.e., either both true, or both false)

4 cases:

H	C	if H then C	if not C then not H
true	true	true	true
true	false	false	false
false	true	true	true
false	false	true	true

Example: "if n is even, then n^2 is even

contrapositive: "if n^2 is not even, then n is not even"

Don't confuse contrapositive, with converse.

Note: to prove an iff statement, we prove a statement and its converse

- Proof by contradiction:

To prove "if H then C"
prove that "H and not C implies falsehood"

Example: H = "U is an infinite set
S is a finite subset of U
T is the complement of S wrt U"

C = "T is infinite"

Proof by contradiction of "if H then C"
Assume H and not C, i.e. H and T is finite.
(A set S is finite iff there is an integer n s.t. $\|S\| = n$
number of elements of S)

S is finite $\Rightarrow$ there is n s.t. $\|S\| = n$
T is finite $\Rightarrow$ — " — m $\|T\| = m$

From H we know: $S \cup T = U$
$S \cap T = \emptyset$ $\}$ $\|S \cup T\| = \|U\| = n + m$

$\Rightarrow$ U is finite, which is a contradiction

- Proof by counterexample:

- to prove something is not a theorem is often easier than to prove something is a theorem
It is sufficient to provide a counterexample
e.g. All odd numbers > 1 are prime
9 is not, which is a counterexample

Proof by induction:

Basic proof technique when dealing with recursively defined objects

- integers: $\left\{\begin{array}{l} 0 \text{ is an integer} \\ \text{if } n \text{ is an integer, then } n+1 \text{ is an integer} \\ \text{nothing else is an integer} \end{array}\right.$

- strings $\left\{\begin{array}{l} \varepsilon \text{ is a string} \\ \text{if } x \text{ is a string and } a \in \Sigma \text{, then } x \cdot a \text{ is a string} \\ \text{nothing else is a string} \end{array}\right.$

- binary trees $\left\{\begin{array}{l} \text{a single node is a BT} \\ \text{if } N \text{ is a single node and } T_1, T_2 \text{ are BT} \\ \quad \text{then } \begin{array}{c} N \\ / \quad \backslash \\ T_1 \quad T_2 \end{array} \text{ is a BT} \\ \text{nothing else is a BT} \end{array}\right.$

Induction on integers:

We want to prove a statement $S(n)$ about integer n

We show:

1) We show $S(i)$, for some specific integer i (e.g. $i=0$)
(base step)

2) We assume $n \geq i$ and show "if $S(n)$ then $S(n+1)$"
(inductive step)

We then resort to the <u>Induction Principle</u>

If we prove $S(i)$ and we prove that
for all $n \geq i$ "$S(n)$ implies $S(n+1)$"
then we can conclude $S(n)$ for all $n \geq i$

N.B. The IP cannot be proved

Example: For all $n \geq 0$ $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ $(*)$

base case: $n = 0$: $\sum_{i=1}^{0} i = 0$

inductive case. assume $n \geq 0$

we must prove that $(*)$ implies $\sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2}$

$(*)$ is called the inductive hypothesis

$$\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) \underset{\text{by IH}}{=} \frac{n \cdot (n+1)}{2} + (n+1) =$$

$$= \frac{n \cdot (n+1)}{2} + \frac{2 \cdot (n+1)}{2} = \frac{(n+2)(n+1)}{2}$$

Generalization of the basic induction scheme

1) We can use several base cases, i.e. we prove $S(i), S(i+1), \ldots, S(j)$ for some $j > i$

2) In proving $S(n)$, we use all of $S(i), S(i+1), \ldots, S(n)$ (strong induction)