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 $\xRightarrow{\text{satisfies?}}$ 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, the 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: $H$ 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. $n = 2k$. 

5/10/2003
H: \(n\) is even (note: \(H\) has \(n\) as parameter)

by Def.: there exist \(k\) s.t. \(n = 2k\)
by rules of mult.: \(n^2 = (2k)^2 = 2^2 \cdot k^2 = 2 \cdot (2k^2)\)
by integer even: \(2k^2\) is an integer
by Def.: \(n^2 = 2 \cdot k\) is even

Other ways of stating the same statements:

if \(H\) then \(C\)
\(H\) implies \(C\)
\(H\) only if \(C\)
\(C\) if \(H\)
wherever \(H\) holds, also \(C\) holds

If and only if statements:

A if and only if B

If part: A if B
only if part: A only if B

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

Example: \(\lfloor x \rfloor = \text{greatest integer } \leq x\)

\(\lceil x \rceil = \text{least integer } \geq x\)

Proof: 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 \( \lfloor x \rfloor = \lfloor x \rfloor \).

We use the definition: \( \lfloor x \rfloor = x \) if \( x \) is an integer. 

\[ \lfloor x \rfloor = x \]

\[ \Rightarrow \lfloor x \rfloor = \lfloor x \rfloor \]

"Only - if part": we assume \( \lfloor x \rfloor = \lfloor x \rfloor \) and prove that \( x \) is an integer.

Def of floor: \( \lfloor x \rfloor \leq x \) \hspace{1cm} (1)

"ceil": \( \lceil x \rceil \geq x \) \hspace{1cm} (2)

Hypothesis: \( \lfloor x \rfloor = \lceil x \rceil \) \hspace{1cm} (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:

\[ \lfloor x \rfloor = \lceil x \rceil \]

Since \( \lceil x \rceil \) is an integer, so is \( x \).

Other forms of proofs:

- To show \( E = F \) we have to show

  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: \[ R \cup (S \cap T) = (R \cup S) \cap (R \cup T) \]

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

See HMU Figure 1.5

2) \( x \in (R \cup S) \cap (R \cup T) \) 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:

\[
\begin{array}{ccc}
H & C & \text{if } H \text{ then } C & \text{if not } C \text{ then not } H \\
\hline
\text{true} & \text{true} & \text{true} & \text{true} \\
\text{true} & \text{false} & \text{false} & \text{false} \\
\text{false} & \text{true} & \text{true} & \text{true} \\
\text{false} & \text{false} & \text{true} & \text{true} \\
\end{array}
\]

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 \( S \) and \( U \)

\( C = \) "\( T \) 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 \( m \) s.t. \( |S| = m \)

\( S \) is finite \( \Rightarrow \) there is an \( m \) s.t. \( |S| = m \)

\( T \) is finite \( \Rightarrow \) \( |T| = m \)

From \( H \) we know:

\( |S \cup T| = |U| \quad S \cap T = \emptyset \)

\( |S \cup T| = |U| = m + m \)

\( \Rightarrow U \) 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

  \( S \) is not, which is a counterexample
Proof by induction:

Basic proof technique when dealing with recursively defined objects

- integers: \( \begin{cases} 0 \text{ is an integer} \\ \text{if } n \text{ is an integer, then } n+1 \text{ is an integer} \\ \text{nothing else is an integer} \end{cases} \)

- strings: \( \begin{cases} \epsilon \text{ is a string} \\ \text{if } x \text{ is a string and } \epsilon \in \Sigma, \text{ then } x \cdot \epsilon \text{ is a string} \\ \text{nothing else is a string} \end{cases} \)

- binary trees: \( \begin{cases} \text{every node is a BT} \\ \text{if } N \text{ is a node and } T_1, T_2 \text{ are BT} \\ \text{then } N \text{ is a BT} \\ \text{nothing else is a BT} \end{cases} \)

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., 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 Induction Principle.

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 $n \geq 0$,\[\sum_{i=0}^{n} i = \frac{n(n+1)}{2} \tag{\#} \]

Base case: $n=0$: \[\sum_{i=0}^{0} i = 0\]

Inductive case: Assume $n \geq 0$,

we must prove that $(\#)$ implies \[\sum_{i=0}^{n+1} i = \frac{(n+1)(n+2)}{2}\]

$(\#)$ is called the inductive hypothesis.

$\sum_{i=0}^{n+1} i = \sum_{i=0}^{n} i + (n+1) = \frac{n(n+1)}{2} + (n+1) =$\[\begin{align*} \text{by IH} & = \frac{n(n+1)}{2} + \frac{2(n+1)}{2} = \frac{(n+2)(n+1)}{2} \end{align*}\]

Generalization of the basic induction scheme:

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

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