Review of formal proof techniques

Why do we need proofs in CS?
specification \implies SW

How do we know that the SW respects the specification?
specification \implies formal specification
SW \overset{\text{satisfies?}}{\implies} testing
proving + understanding how a complex program works

Deductive proofs:
- 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 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  \( \text{(note: \( n \) has \( n \) as parameter)} \)

by Def.: there exist \( k \) s.t. \( n = 2k \)

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

by integer axiom: \( 2 \cdot k^2 = k \) is an integer

by Def.: \( n^2 = 2 \cdot k \) is even

Other ways of stating if-then 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 \( \text{pent} \): A if B, i.e., if B then A

only if \( \text{pent} \): A only if B, i.e., if A then B

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

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

\( \lfloor x \rfloor \) = least integer \( \geq x \)

\( \text{Floor} \)

\( \text{Ceiling} \)

Proof: let \( x \) be a real number.

\( \text{then } \lfloor x \rfloor = \lceil x \rceil \text{ iff } x \text{ is an integer.} \)
Proof:

"If-then": we assume \( x \) is an integer and prove \( \lfloor x \rfloor = \lfloor x \rfloor \).

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

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

"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)

\( \ldots \) ceiling: \( \lceil x \rceil \geq x \) \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 \( \lfloor x \rfloor = x \).

Since \( \lfloor x \rfloor \) 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 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: \( R \cup (S \cap T) = (R \cup S) \cap (R \cup T) \)

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: \[ \begin{array}{c|c|c|c} H & C & \text{if } \neg C \text{ then } \neg H \\ \hline true & true & true & true \\ true & false & false & false \\ false & true & true & true \\ false & false & true & true \end{array} \]

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"
proving that "H and not C implies falsehood"

Example: \( H = "U \text{ is an infinite set} \)
\( S \text{ is a finite subset of } U \)
\( T \text{ is the complement of } S \text{ in } U \)

\( C = "T \text{ is infinite}" \)

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. \( |S| = n \) \)
\( T \) is finite \( \Rightarrow \) there is an integer \( m \) \( |T| = m \)

From \( H \) we know:
\( S \cup T = U \)
\( S \cap T = \emptyset \)
\( |S \cup T| = |U| = \infty \)

\( \Rightarrow U \) 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**:
  - \( 0 \) is an integer
  - if \( n \) is an integer, then \( n+1 \) is an integer
  - nothing else is an integer

- **Strings**:
  - \( \varepsilon \) is a string
  - if \( x \) is a string and \( \varepsilon \in \Sigma \), then \( xa \) is a string
  - nothing else is a string

- **Binary Trees**:
  - \( \varepsilon \) simple node is a BT
  - if \( N \) is a simple node and \( T_1, T_2 \) are BT
  - then \( \overline{N} \) is a BT
  - \( T_1, T_2 \)
  - nothing else is a BT

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. 100) (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 $m \geq i$ "$S(m)$ implies $S(m+1)$" then we can conclude $S(n)$ for all $n \geq i$.

N.B. The IP cannot be proved.

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

**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.

\[
\sum_{i=1}^{n} i = \sum_{i=0}^{n} i + (n+1) = \frac{n(n+1)}{2} + (n+1) = \frac{n(n+1) + 2(n+1)}{2} = \frac{(n+2)(n+1)}{2}
\]

by IH

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).