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 $\stackrel{\text{satisfies?}}{\leftrightarrow}$ testing
- proving = understanding how a complex program works

Deductive proof:

- start from a set $H$ of hypothesis (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?

This is an integer $k$ s.t. $n = 2k$. 
H: \( n \) is even  \( \text{(note: } H \text{ has } n \text{ as parameter)} \)

by Def.: there exist \( k \in \mathbb{Z} \) s.t. \( n = 2k \)

by rules of math: \( n^2 = (2k)^2 = 2^2 \cdot k^2 = 4 \cdot k^2 \)

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

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

If-and-only-if statements:

\( A \) if-and-only-if \( B \)

iff 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 \( \text{"If part"} \)

and the "Only-if part"

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

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

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 $[x] = [x]

We use the definition: if $x$ is an integer $[x] = x$

$[x] = x

$\Rightarrow [x] = [x]

"Only if part": we assume $[x] = [x]$ and prove that $x$ is an integer.

Definition of floor: $[x] \leq x \ (1)$

... ceiling: $\lfloor x \rfloor \geq x \ (2)$

Hypothesis: $[x] = [x] \ (3)$

Substituting $[x]$ in place of $[x]$, we get from (1)

$[x] \leq x$ and with (2) and arithmetic laws,

we get $[x] = x$

Since $[x]$ is an integer, so is $x$

Other forms of proofs:

"Proving equivalences of sets"

To 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:

<table>
<thead>
<tr>
<th>( H )</th>
<th>( C )</th>
<th>if ( H ) then ( C )</th>
<th>if not ( C ) then not ( H )</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

Example: "if \( n \) is even, then \( n^2 \) is ...
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 in 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. \(|S| = n\) or the number of elements of S)

S is finite \(\Rightarrow\) there is an \(n\) s.t. \(|S| = n\)
T is finite \(\Rightarrow\) \(\cdots\)

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

\(\Rightarrow U\) 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: \[
\begin{align*}
0 & \text{ is an integer} \\
\text{if } m & \text{ is an integer, then } n+m \text{ is an integer} \\
\text{nothing else is an integer}
\end{align*}
\]

- strings: \[
\begin{align*}
\varepsilon & \text{ is a string} \\
\text{if } x & \text{ is a string and } a \in \Sigma, \text{ then } xa \text{ is a string} \\
\text{nothing else is a string}
\end{align*}
\]

- binary trees: \[
\begin{align*}
e & \text{ empty node is a BT} \\
\text{if } N & \text{ is an empty node and } T_1, T_2 \text{ are BT} \\
& \text{ then } N(T_1, T_2) \text{ is a BT} \\
& \text{nothing else is a BT}
\end{align*}
\]

Induction on integers:

We want to prove a statement \( S(m) \) about integer \( m \).

We show:

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

2) We assume \( m \geq i \) and show \( \text{if } S(m) \text{ then } S(m+1) \) (inductive step)
We then revert to the Induction Principle.

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

N.B. The IP cannot be proved.

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

Base case: $m = 0$: \( \sum_{i=0}^{0} i = 0 \)

Inductive case: Assume $m \geq 0$

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

(*) is called the Inductive Hypothesis

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

by IH

\[ = \frac{m(m+1)}{2} + \frac{2(m+1)}{2} = \frac{(m+2)(m+1)}{2} \]

Generalization of the basic induction scheme

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

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