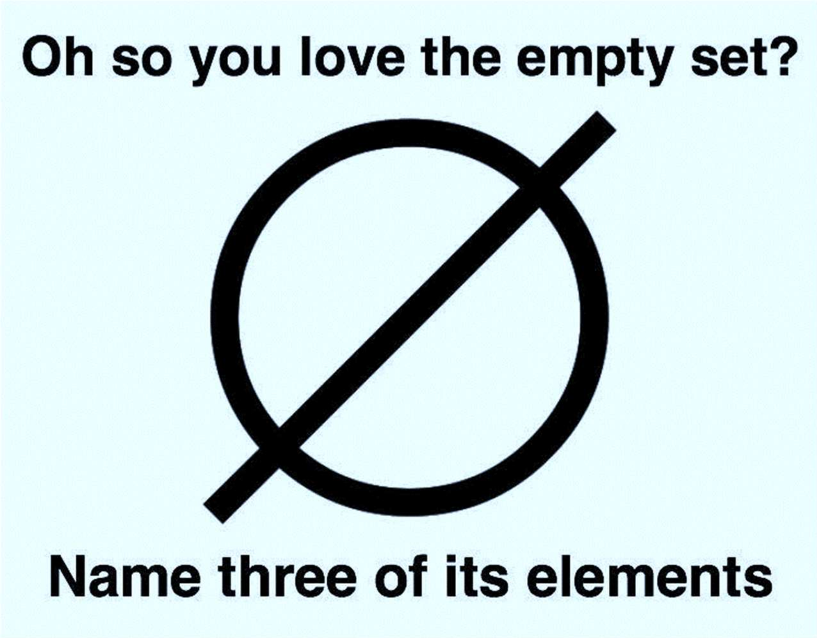


Set Theory Part 2

CSE 311: Foundations of Computing I
Lecture 11

Sets

- Definition:
- A set is an unordered collection of distinct objects, called elements.

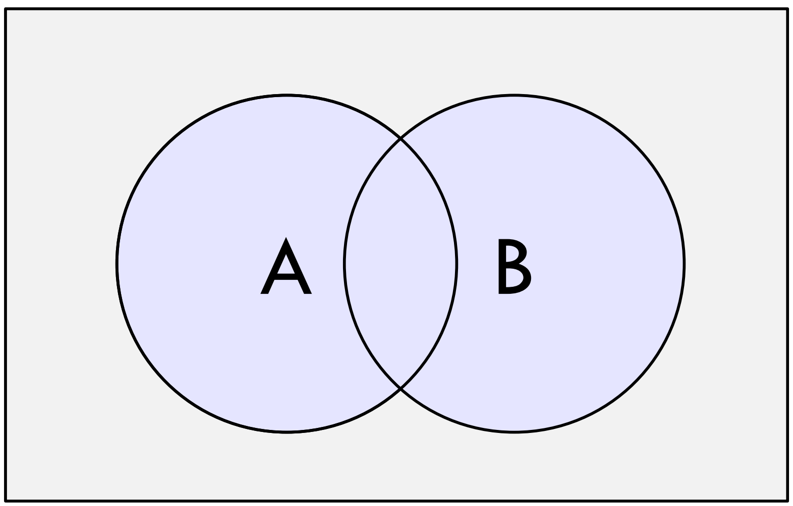
- Set A is a subset of B if every element of A is also in B.

- In predicate logic, $A \subseteq B$ is defined as:
- $\forall x(x \in A \rightarrow x \in B)$

Set Operations

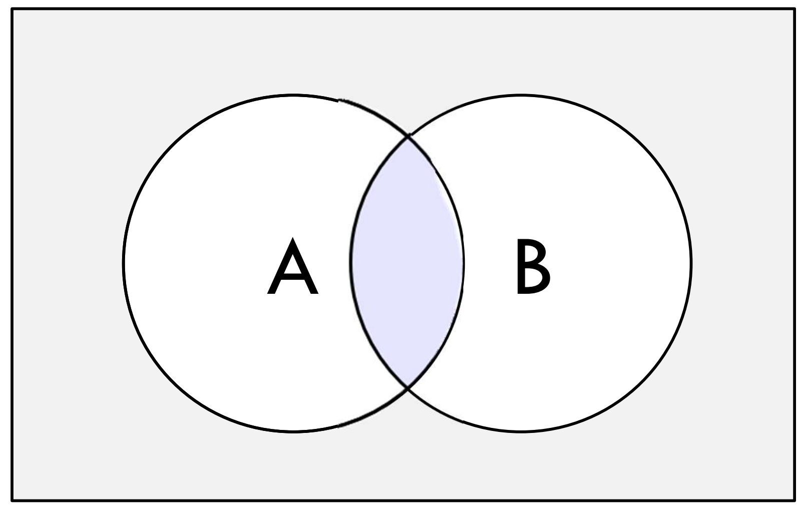
Union: $A \cup B$

$A \cup B = \{x : x \in A \vee x \in B\}$



Intersection: $A \cap B$

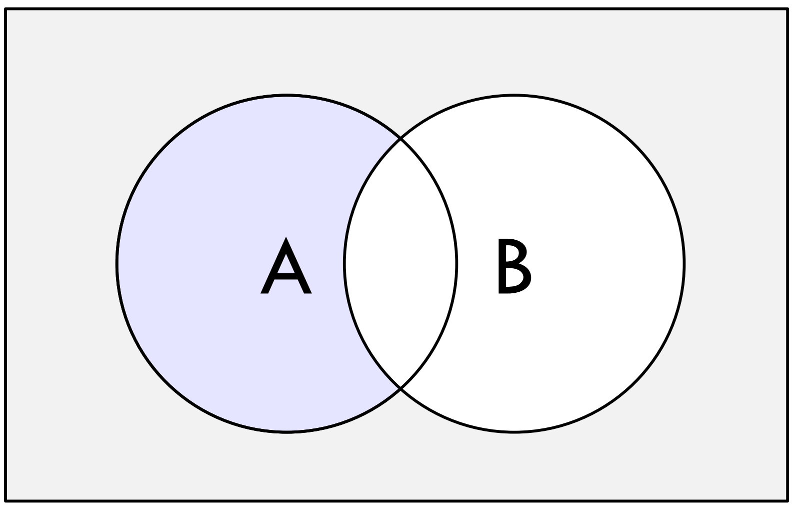
$A \cap B = \{x : x \in A \wedge x \in B\}$



Set Operations

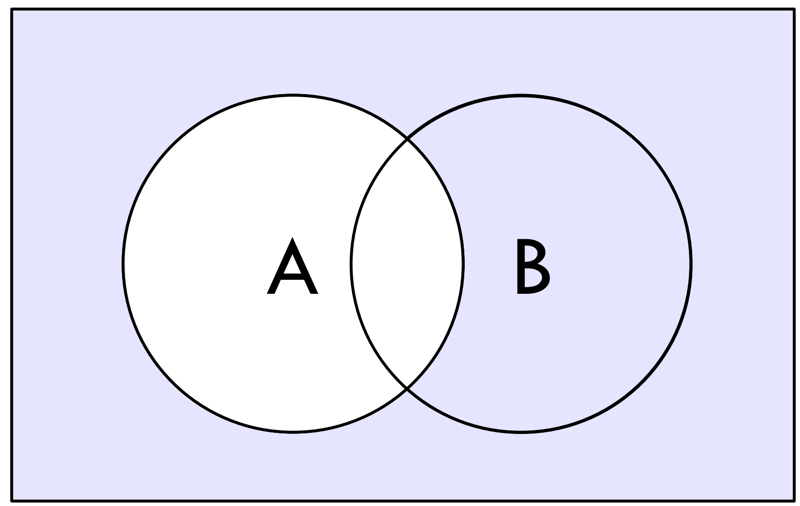
Set Difference: $A \setminus B$

$$A \setminus B = \{x : x \in A \land x \notin B\}$$



Set Complement: $\bar{A} = A^c$
(with respect to the universe $\mathcal{U}$)

$$\bar{A} = \{x \in \mathcal{U}: x \notin A\}$$



Set Operations

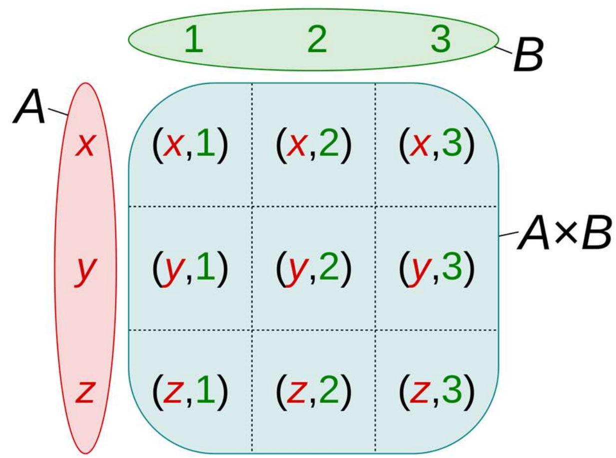
Powerset: $\mathcal{P}(A)$

$$\mathcal{P}(A) = \{B : B \subseteq A\}$$

$$\mathcal{P}(\{1,2,3\}) = \left\{ \begin{array}{ccc} & \{\ \} & \\ \{1\} & \{2\} & \{3\} \\ \{1,2\} & \{1,3\} & \{2,3\} \\ & \{1,2,3\} & \end{array} \right\}$$

Cartesian Product: $A \times B$

$$A \times B = \{(a,b) \colon a \in A, b \in B\}$$



Claim 1

Definitions
$A \subseteq B \equiv \forall x(x \in A \to x \in B)$
$A \cup B = \{x : x \in A \vee x \in B\}$
$A \cap B = \{x : x \in A \wedge x \in B\}$

Claim 1: For all sets A, B, we have $A \cap B \subseteq A \cup B$.

Proof

Let A, B be arbitrary sets. Let $x \in A \cap B$ be arbitrary. Then by definition of intersection, $x \in A$ and $x \in B$. So certainly $x \in A$ or $x \in B$. Thus by definition of union, $x \in A \cup B$. Since x was arbitrary, $A \cap B \subseteq A \cup B$. Since A, B were arbitrary sets, the claim holds for all sets A, B.

Proving Subsets

To prove that $X \subseteq Y$, we let $x \in X$ be arbitrary and prove that $x \in Y$.

Claim 2

Claim 2: For all sets A, B if $A \subseteq B$ then $\mathcal{P}(A) \subseteq \mathcal{P}(B)$.

- Intuition (Example)

$A = \{1,2\} \quad \mathcal{B} = \{1,2,3\}$

$\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, \{1,2\}\}$

$\mathcal{P}(B) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}\}$

Claim 2

Definitions
$A \subseteq B \equiv \forall x(x \in A \rightarrow x \in B)$
$\mathcal{P}(A) = \{B : B \subseteq A\}$

- Claim 2: For all sets A, B if $A \subseteq B$ then $\mathcal{P}(A) \subseteq \mathcal{P}(B)$.
- Proof
- Let A, B be arbitrary sets. Suppose $A \subseteq B$. Let $X \in \mathcal{P}(A)$ be arbitrary. Then by definition of powerset, $X \subseteq A$.

- We know $X \subseteq A$ and $A \subseteq B$. We aim to show that $X \subseteq B$. Let $x \in X$ be arbitrary. Since $x \in X$ and $X \subseteq A$, then $x \in A$. Since $x \in A$ and $A \subseteq B$, then $x \in B$. Since x was arbitrary, $X \subseteq B$.

- Now since we have that $X \subseteq B$, by definition of powerset $X \in \mathcal{P}(B)$. Since X was arbitrary, we have shown that $\mathcal{P}(A) \subseteq \mathcal{P}(B)$

Symbols and Sets

- Note that when writing set proofs, we follow various conventions.

- We DO tend use symbols like $\in$, $\subseteq$, $\cup$, $\cap$, $\times$ etc. (instead of writing out the symbol in English).
- E.g. "Let $x \in A$ be arbitrary"

- We DO NOT tend to use symbols like $\wedge$, $\vee$, $\neg$ (but rather write them out in English).**
- E.g. "Then $x \in A$ and $x \in B$"

- **There are exceptions to this if logical symbols provide clarity when applying equivalence rules (Absorption, DeMorgan's Laws, etc.). The proof of Claim 3 will be an example of that.

Set Equality Proofs

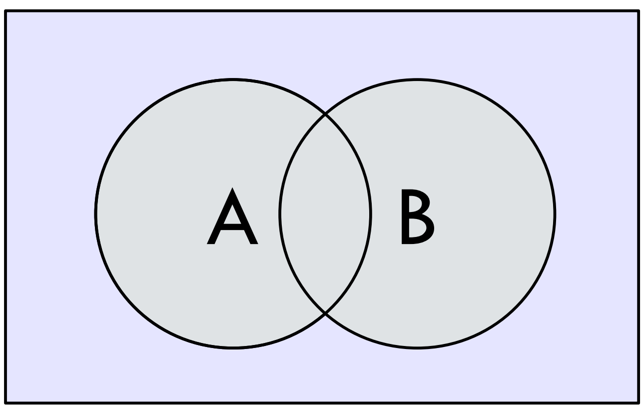
Claim 3 (DeMorgan's Law for Sets)

Definitions
$A = B \equiv A \subseteq B \wedge B \subseteq A$
$A \cup B = \{x : x \in A \vee x \in B\}$
$A \cap B = \{x : x \in A \wedge x \in B\}$
$\bar{A} = \{x \in \mathcal{U}: x \notin A\}$

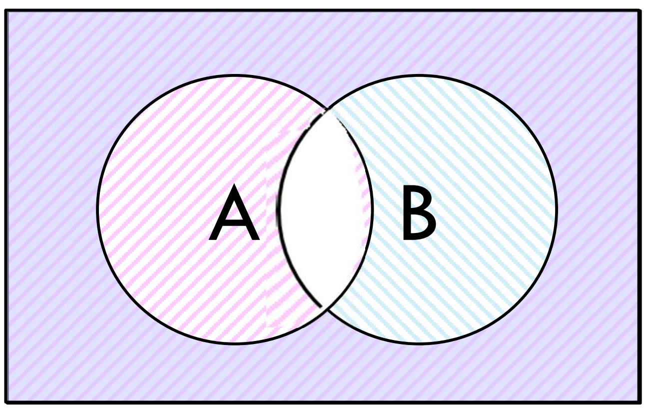
Claim 3: For all sets A, B, $\overline{A \cup B} = \bar{A} \cap \bar{B}$

Intuition (Venn Diagram)

$\overline{A \cup B}$



$\bar{A} \cap \bar{B}$



Claim 3

Definitions

$A = B \equiv A \subseteq B \land B \subseteq A$
$A \cup B = \{x : x \in A \lor x \in B\}$
$A \cap B = \{x : x \in A \land x \in B\}$
$\bar{A} = \{x \in \mathcal{U} : x \notin A\}$

- Claim 3: For all sets A, B, $\overline{A \cup B} = \bar{A} \cap \bar{B}$

- Proof Strategy
- Let A, B be arbitrary sets.
- Prove that $\overline{A \cup B} \subseteq \bar{A} \cap \bar{B}$.
- Prove that $\bar{A} \cap \bar{B} \subseteq \overline{A \cup B}$.

Claim 3

Claim 3: For all sets A, B, $\overline{A \cup B} = \bar{A} \cap \bar{B}$.

Definitions
$A = B \equiv A \subseteq B \wedge B \subseteq A$
$A \cup B = \{x : x \in A \vee x \in B\}$
$A \cap B = \{x : x \in A \wedge x \in B\}$
$\bar{A} = \{x \in \mathcal{U} : x \notin A\}$

Proof (Method 1)

Let A, B be arbitrary sets.

$\Rightarrow$ First we show that $\overline{A \cup B} \subseteq \bar{A} \cap \bar{B}$. Let $x \in \overline{A \cup B}$ be arbitrary. By definition of complement, we have that $\neg(x \in A \cup B)$. Then by definition of union, $\neg(x \in A \vee x \in B)$. So by DeMorgan's Law, $x \notin A \wedge x \notin B$. Then by definition of complement, $x \in \bar{A}$ and $x \in \bar{B}$. By definition of intersection, $x \in \bar{A} \cap \bar{B}$. Since x was arbitrary, $\overline{A \cup B} \subseteq \bar{A} \cap \bar{B}$.

$\Leftarrow$ Now we show that $\bar{A} \cap \bar{B} \subseteq \overline{A \cup B}$. Let $x \in \bar{A} \cap \bar{B}$ be arbitrary. By definition of intersection, we have that $x \in \bar{A}$ and $x \in \bar{B}$. By definition of complement, we have $\neg(x \in A) \wedge \neg(x \in B)$. Apply DeMorgan's Law, we have $\neg(x \in A \vee x \in B)$. Then by definition of union, $\neg(x \in A \cup B)$. Then by definition of complement, $x \in \overline{A \cup B}$. Since x was arbitrary, $\bar{A} \cap \bar{B} \subseteq \overline{A \cup B}$.

Thus we have shown that $\overline{A \cup B} = \bar{A} \cap \bar{B}$. Since A, B were arbitrary, the claim holds.

Claim 4

Definitions
$A \times B = \{(a, b): a \in A, b \in B\}$
$A \cup B = \{x : x \in A \vee x \in B\}$

- Claim 4: For all sets A, B, C, $A \times (B \cap C) = (A \times B) \cap (A \times C)$
- Intuition (Diagram)

$A \times (B \cap C)$

$(A \times B) \cap (A \times C)$

First Direction (Other way is an exercise)

Let A, B, C, be arbitrary sets.

Assume that $x \in (A \times (B \cap C)$ is arbitrary. We therefore know by definition of cartesian product that there exists an $a \in A$ and a $b \in B \cap C$ such that $x = (a, b)$.

Therefore, by definition of intersection, $b \in B$ and $b \in C$. Therefore, we have that $a \in A$ and $b \in B$, and $a \in a$ and $b \in C$. Since $x = (a, b)$, we have by definition of cartesian product that $x \in A \times B$ and $x \in A \times C$. Thus, by definition of intersection, $x \in (A \times B) \cap (A \times C)$.

Since x was arbitrary, we have that $(A \times (B \cap C) \subseteq (A \times B) \cap (A \times C)$.

Prove for all integers n, $n^2 \equiv_4 0$ or $n^2 \equiv_4 1$

Let n be an arbitrary integer. We will argue by cases:

Case 1: n is even

If n is even, that means that n=2k for some integer k. squaring both sides, we get $n^2 = 4k^2$. By definition of divides, we have that $4 \mid n^2 - 0$ which means that $n^2 \equiv_4 0$

Case 2: n is odd

If n is odd, this means that n = 2k + 1 for some integer k. Squaring both sides, we get $n^2 = 4k^2 + 4k + 1$, simplifying, we get

$n^2 - 1 = 4(k^2 + k)$. By definition of divides and congruence, we get that $n^2 \equiv_4 1$.

Therefore, $n^2 \equiv_4 0$ or $n^2 \equiv_4 1$

Example Proof By Contradiction

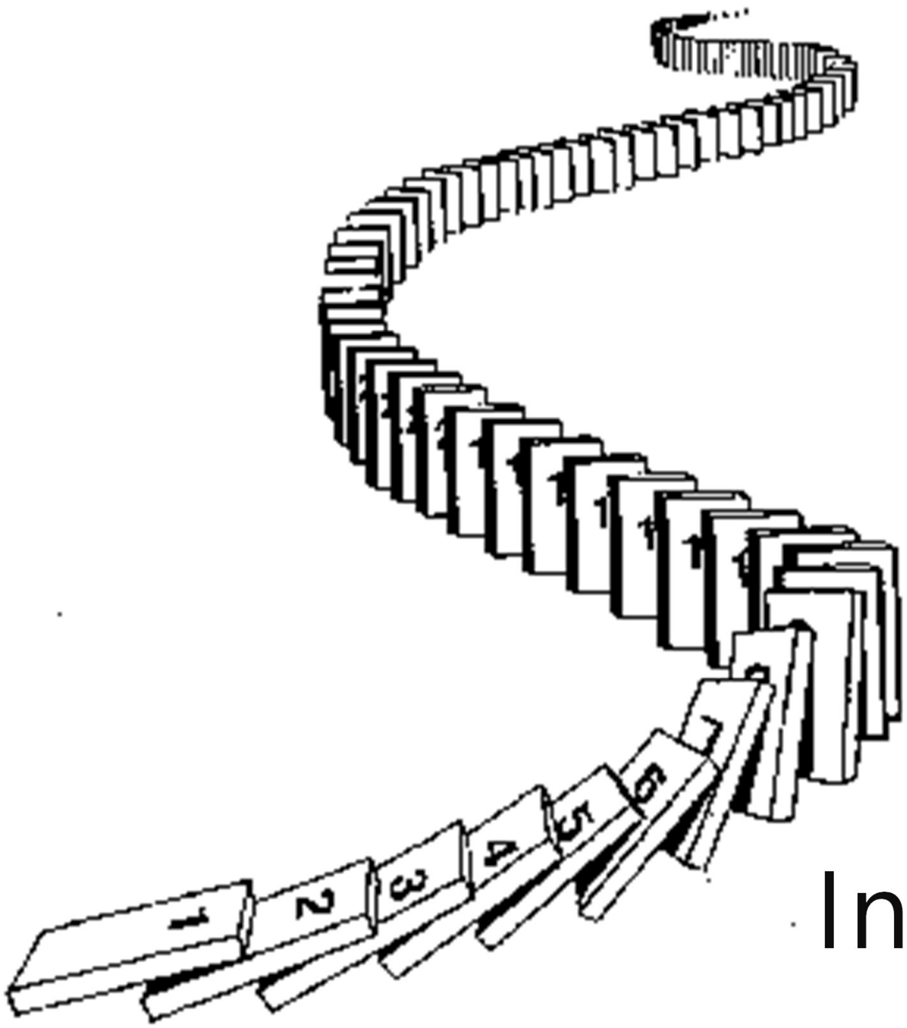
Prove: “No integer is both even and odd.”

Suppose that x is an integer that is both even and odd. Then, $x=2a$ for some integer a, and $x=2b+1$ for some integer b. This means $2a=x=2b+1$ and hence $2a-2b=1$ and so $a-b=½$. But a-b is an integer while ½ is not, so they cannot be equal. This is a contradiction.

Proof Strategies so Far

- Direct Proof
- Proof by Contrapositive
- Proof of Biconditional
- Proof by Cases
- Existence Proof

There are claims we cannot prove using these strategies!



Induction

CSE 311 Winter 2024
Lecture 12

How do we know recursion works?

```
//Assume i is a nonnegative integer
//returns 2^i.
public int CalculatesTwoToTheI(int i){
     if(i == 0)
          return 1;
     else
          return 2*CaclulatesTwoToTheI(i-1);
}
```

Why does `CalculatesTwoToTheI(4)` calculate `2^4?`
Convince the people around you!

How do we know recursion works?

Something like this:

Well, as long as `CalculatesTwoToTheI(3) = 8`, we get 16…

Which happens as long as `CalculatesTwoToTheI(2) = 4`

Which happens as long as `CalculatesTwoToTheI(1) = 2`

Which happens as long as `CalculatesTwoToTheI(0) = 1`

And it is! Because that’s what the base case says.

How do we know recursion works?

There's really only two cases.

The Base Case is Correct

`CalculatesTwoToTheI(0) = 1 (which it should!)`

And that means `CalculatesTwoToTheI(1) = 2`, (like it should)

And that means `CalculatesTwoToTheI(2) = 4`, (like it should)

And that means `CalculatesTwoToTheI(3) = 8,` (like it should)

And that means `CalculatesTwoToTheI(4) = 16,` (like it should)

IF the recursive call we make is correct
THEN our value is correct.

How do we know recursion works?

The code has two big cases,

So our proof had two big cases

“The base case of the code produces the correct output”

“IF the calls we rely on produce the correct output THEN the current call produces the right output”

A bit more formally…

“The base case of the code produces the correct output”

“IF the calls we rely on produce the correct output THEN the current call produces the right output”

Let $P(i)$ be “`CalculatesTwoToTheI(i)` returns 2^i.”

How do we know $P(4)$?

$P(0)$ is true.

And $P(0) \rightarrow P(1)$, so $P(1)$.

And $P(1) \rightarrow P(2)$, so $P(2)$.

And $P(2) \rightarrow P(3)$, so $P(3)$.

And $P(3) \rightarrow P(4)$, so $P(4)$.

A bit more formally…

This works alright for $P(4)$.

What about $P(1000)$? $P(1000000000)$?

At this point, we'd need to show that implication $P(k) \to P(k+1)$ for A BUNCH of values of k.

But the code is the same each time.

And so was the argument!

We should instead show $\forall k[P(k) \to P(k+1)]$.

Induction

Your new favorite proof technique!

How do we show $\forall n, P(n)$?

Show $P(0)$

Show $\forall k (P(k) \rightarrow P(k+1))$

Induction

```
//Assume i is a nonnegative integer
public int CalculatesTwoToTheI(int i){
       if(i == 0)
              return 1;
       else
              return 2*CaclulatesTwoToTheI(i-1);
}
```

Let $P(i)$ be "`CalculatesTwoToTheI(i)` returns 2^i."

Note that if the input i is 0, then the if-statement evaluates to true, and $1 = 2\char`^0$ is returned, so $P(0)$ is true.

Suppose $P(k)$ holds for an arbitrary $k \geq 0$.

Consider the code run on $k + 1$. Since $k \geq 0$, $k + 1 \geq 1$ and we are in the else branch. By inductive hypothesis, `CalculatesTwoToTheI(`k`)` returns 2^k, so the code run on $k + 1$ returns $2 \cdot 2^k = 2^{k+1}$.

So $P(k + 1)$ holds.

Therefore $P(n)$ holds for all $n \geq 0$ by the principle of induction.

Making Induction Proofs Pretty

Let $P(i)$ be the predicate "`CalculatesTwoToTheI(i)` returns 2^i." We prove $P(n)$ holds holds for all natural numbers n by induction on n.

Base Case ($i = 0$) Note that if the input i is 0, then the if-statement evaluates to true, and $1 = 2^\wedge 0$ is returned, so $P(0)$ is true.

Inductive Hypothesis: Suppose $P(k)$ holds for an arbitrary $k \geq 0$.

Inductive Step: Since $k \geq 0, k + 1 \geq 1$, so the code goes to the recursive case. We will return $2 \cdot$ `CalculatesTwoToTheI(k)`. By Inductive Hypothesis,

`CalculatesTwoToTheI(k)` = 2^k. Thus we return $2 \cdot 2^k = 2^{k+1}$.

So $P(k + 1)$ holds.

Therefore $P(n)$ holds for all $n \geq 0$ by the principle of induction.

Making Induction Proofs Pretty

All of our induction proofs will come in 5 easy(?) steps!

1. Define $P(n)$. State that your proof is by induction on n.
2. Show $P(0)$ i.e. show the base case
3. Suppose $P(k)$ for an arbitrary k.
4. Show $P(k+1)$ (i.e. get $P(k) \rightarrow P(k+1)$)
5. Conclude by saying $P(n)$ is true for all n by induction.

Some Other Notes

Always state where you use the inductive hypothesis when you're using it in the inductive step.

It's usually the key step, and the reader really needs to focus on it.

Be careful about what values you're assuming the Inductive Hypothesis for – the smallest possible value of k should assume the base case but nothing more.

The Principle of Induction (formally)

$$\text{Principle of Induction}\quad \frac{P(0);\forall k(P(k) \rightarrow P(k+1))}{\therefore \quad \forall n(P(n))}$$

Informally: if you knock over one domino, and every domino knocks over the next one, then all your dominoes fell over.

More induction!

More Induction

Induction doesn't only work for code!

Show that $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$.

More Induction

Induction doesn't only work for code!

Show that $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$.

Let $P(n) =$"$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$."

We show $P(n)$ holds for all natural numbers n by induction on n.

Base Case ( )

Inductive Hypothesis:

Inductive Step:

$P(n)$ holds for all $n \geq 0$ by the principle of induction.

More Induction

Induction doesn't only work for code!

Show that $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$.

Let $P(n) =$"$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$."

We show $P(n)$ holds for all natural numbers n by induction on n.

Base Case $(n = 0)$ $\sum_{i=0}^{0} 2^i = 1 = 2 - 1 = 2^{0+1} - 1$.

Inductive Hypothesis: Suppose $P(k)$ holds for an arbitrary $k \geq 0$.

Inductive Step: We show $P(k+1)$. Consider the summation$\sum_{i=0}^{k+1} 2^i = 2^{\mathrm{k}+1} + \sum_{i=0}^{k} 2^i = 2^{k+1} + 2^{k+1} - 1$, where the last step is by IH.

Simplifying, we get: $\sum_{i=0}^{k+1} 2^i = 2^{k+1} + 2^{k+1} - 1 = 2 \cdot 2^{k+1} - 1 = 2^{(k+1)+1} - 1$.

$P(n)$ holds for all $n \geq 0$ by the principle of induction.