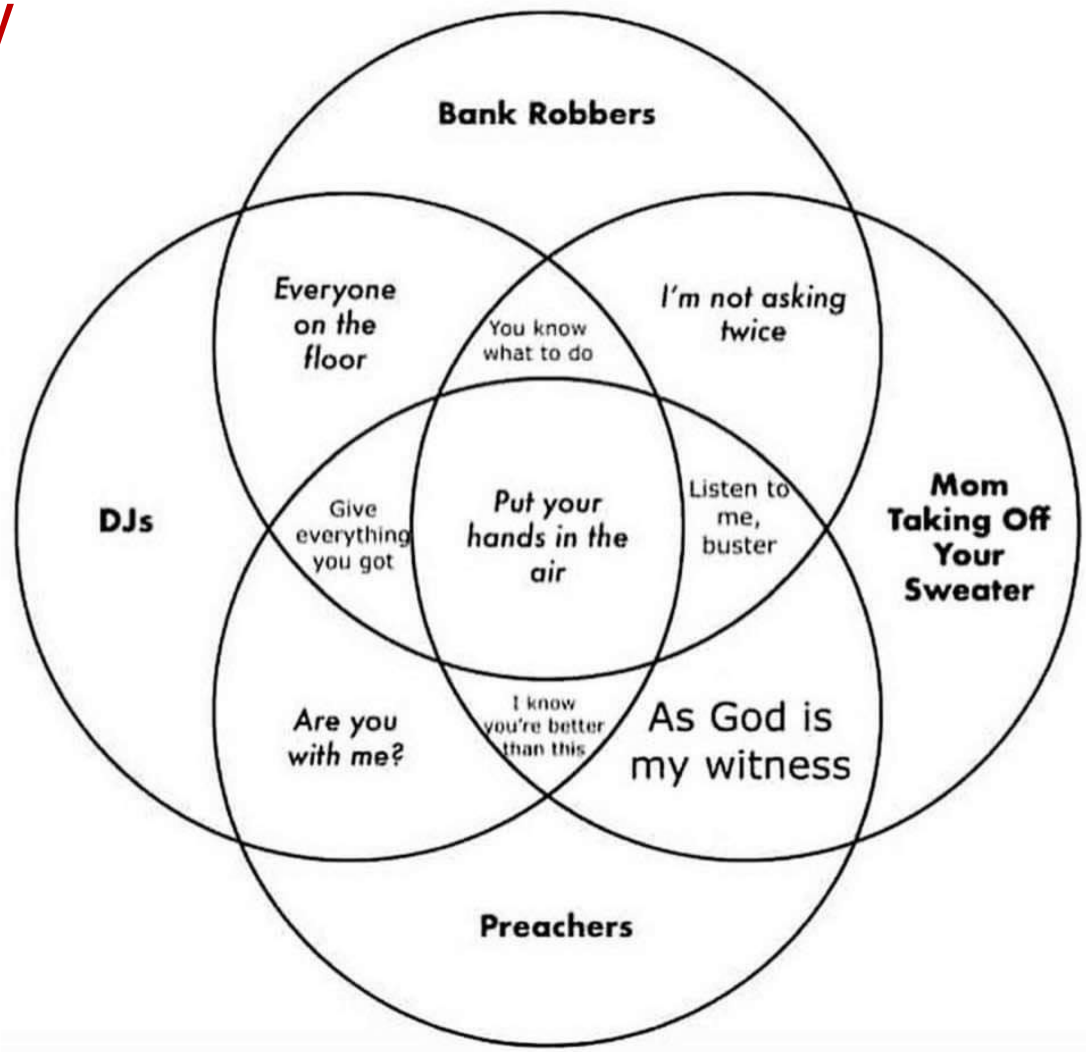
CSE 311: Foundations of Computing

Topic 6: Set Theory



Sets

Sets are collections of objects called **elements.**

Write $a \in B$ to say that a is an element of set B,
and $a \notin B$ to say that it is not.

Some simple examples
$A = \{1\}$
$B = \{1, 3, 2\}$
$C = \{\square, 1\}$
$D = \{\{17\}, 17\}$
$E = \{1, 2, 7, \text{cat}, \text{dog}, \varnothing, \alpha\}$

Some Common Sets

$\mathbb{N}$ is the set of **Natural Numbers;** $\mathbb{N}$ = {0, 1, 2, …}
$\mathbb{Z}$ is the set of **Integers**; $\mathbb{Z}$ = {…, -2, -1, 0, 1, 2, …}
$\mathbb{Q}$ is the set of **Rational Numbers**; e.g. ½, -17, 32/48
$\mathbb{R}$ is the set of **Real Numbers**; e.g. 1, -17, 32/48, π, $\sqrt{2}$
[n] is the set **{1, 2, …, n}** when **n** is a natural number
∅ = **{}** is the **empty set**; the *only* set with no elements

Sets can be elements of other sets

For example

$A = \{\{1\},\{2\},\{1,2\},\emptyset\}$

$B = \{1,2\}$

Then $B \in A$.

Definition: Equality

A and B are *equal* if they have the same elements

$$A = B \ := \ \forall x\, (x \in A \leftrightarrow x \in B)$$

Examples:

- $\{1\} = \{1, 1, 1\}$
- $\varnothing$ is **the** empty set

Definition: Equality

A and B are *equal* if they have the same elements

$$A = B \ := \ \forall x\,(x \in A \leftrightarrow x \in B)$$

A = {1, 2, 3}
B = {3, 4, 5}
C = {3, 4}
D = {4, 3, 3}
E = {3, 4, 3}
F = {4, {3}}

Which sets are equal?

Definition: Subset

A is a *subset* of B if every element of A is also in B

$$A \subseteq B \ := \ \forall x \, (x \in A \rightarrow x \in B)$$

Definition: Subset

A is a *subset* of B if every element of A is also in B

$$A \subseteq B \ := \ \forall x\,(x \in A \rightarrow x \in B)$$

$A = \{1, 2, 3\}$
$B = \{3, 4, 5\}$
$C = \{3, 4\}$

<u>QUESTIONS</u>

$A \subseteq B$?

$C \subseteq B$?

$\varnothing \subseteq A$?

Definition: Subset

A is a *subset* of B if every element of A is also in B

$$A \subseteq B \;:=\; \forall x\,(x \in A \rightarrow x \in B)$$

Note the domain restriction!

We will use a shorthand restriction to a set

$\forall x \in A\,(P(x))$ means $\forall x\,(x \in A \rightarrow P(x))$

Restricting all quantified variables improves *clarity*

Definitions

- A and B are *equal* if they have the same elements

$$A = B \ := \ \forall x \, (x \in A \leftrightarrow x \in B)$$

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

$$A \subseteq B \ := \ \forall x \, (x \in A \rightarrow x \in B)$$

- Notes:

$$(A = B) \equiv (A \subseteq B) \land (B \subseteq A)$$

$A \supseteq B$ means $B \subseteq A$

$A \subset B$ means $A \subseteq B$

Sets & Logic

Proofs About Sets

1. $A \subseteq B$	Given
2. $B \subseteq A$	Given
?. $A = B$	??

Proofs About Sets

1. $A \subseteq B$	Given
2. $B \subseteq A$	Given
3. $\forall x\,(x \in A \rightarrow x \in B)$	Def of Subset: 1
4. $\forall x\,(x \in B \rightarrow x \in A)$	Def of Subset: 2
?. $A = B$	??

Proofs About Sets

1. $A \subseteq B$	Given
2. $B \subseteq A$	Given
3. $\forall x\,(x \in A \rightarrow x \in B)$	Def of Subset: 1
4. $\forall x\,(x \in B \rightarrow x \in A)$	Def of Subset: 2
?. $\forall x\,(x \in A \leftrightarrow x \in B)$	??
?. $A = B$	Def of Same Set

Proofs About Sets

1. $A \subseteq B$ Given
2. $B \subseteq A$ Given
3. $\forall x\,(x \in A \rightarrow x \in B)$ Def of Subset: 1
4. $\forall x\,(x \in B \rightarrow x \in A)$ Def of Subset: 2

   Let y be arbitrary.

   5.?. $y \in A \leftrightarrow y \in B$??

5. $\forall x\,(x \in A \leftrightarrow x \in B)$ Intro $\forall$
6. $A = B$ Def of Same Set: 5

Proofs About Sets

1. $A \subseteq B$ — Given
2. $B \subseteq A$ — Given
3. $\forall x\,(x \in A \rightarrow x \in B)$ — Def of Subset: 1
4. $\forall x\,(x \in B \rightarrow x \in A)$ — Def of Subset: 2

    Let y be arbitrary.

    5.1. $y \in A \rightarrow y \in B$ — Elim ∀: 3

    5.2. $y \in B \rightarrow y \in A$ — Elim ∀: 4

    5.?. $y \in A \leftrightarrow y \in B$ — ??

5. $\forall x\,(x \in A \leftrightarrow x \in B)$ — Intro ∀
6. $A = B$ — Def of Same Set: 5

Proofs About Sets

1.	$A \subseteq B$	Given
2.	$B \subseteq A$	Given
3.	$\forall x\, (x \in A \rightarrow x \in B)$	Def of Subset: 1
4.	$\forall x\, (x \in B \rightarrow x \in A)$	Def of Subset: 2
	Let y be arbitrary.	
	5.1. $y \in A \rightarrow y \in B$	Elim $\forall$: 3
	5.2. $y \in B \rightarrow y \in A$	Elim $\forall$: 4
	5.3. $(y \in A \rightarrow y \in B) \wedge (y \in B \rightarrow y \in A)$	Intro $\wedge$: 5.1, 5.2
	5.4. $y \in A \leftrightarrow y \in B$	Equivalent: 5.3
5.	$\forall x\, (x \in A \leftrightarrow x \in B)$	Intro $\forall$
6.	$A = B$	Def of Same Set: 5

Building Sets from Predicates

Every set S **defines a predicate** $P(x) :=$ "$x \in S$"

We can also define a set from a predicate P**:**

$$S := \{x : P(x)\}$$

S **= the set of all** x **for which** $P(x)$ **is true**

$$S := \{x \in U : P(x)\} = \{x : (x \in U) \wedge P(x)\}$$

Inference Rules on Sets

S := {x : P(x)}

**When a set is defined this way,
we can reason about it using its definition:**

1.	$x \in S$	Given
2.	$P(x)$	Def of S
	…	
8.	$P(y)$	
9.	$y \in S$	Def of S

This will be our **only** inference rule for sets!

Proofs About Sets

$$A := \{x : P(x)\}$$

$$B := \{x : Q(x)\}$$

Suppose we want to prove $A \subseteq B$.

We have a definition of subset:

$$A \subseteq B := \forall x\, (x \in A \rightarrow x \in B)$$

We need to show that is definition holds

Proofs About Sets

$A := \{x : P(x)\}$

$B := \{x : Q(x)\}$

9. $A \subseteq B$??

Proofs About Sets

$A := \{x : P(x)\}$ $B := \{x : Q(x)\}$

8. $\forall x\,(x \in A \to x \in B)$	??
9. $A \subseteq B$	Def of Subset: 8

Proofs About Sets

$A := \{x : P(x)\}$ $B := \{x : Q(x)\}$

Let x be arbitrary

1.1. $x \in A \rightarrow x \in B$??

1. $\forall x\ (x \in A \rightarrow x \in B)$ Intro $\forall$: 1

2. $A \subseteq B$ Def of Subset: 2

Proofs About Sets

$A := \{x : P(x)\}$ $B := \{x : Q(x)\}$

Let x be arbitrary

1.1.1. $x \in A$ Assumption

1.1.?. $x \in B$??

1..1. $x \in A \rightarrow x \in B$ Direct Proof

1. $\forall x\,(x \in A \rightarrow x \in B)$ Intro ∀: 1

2. $A \subseteq B$ Def of Subset: 2

Proofs About Sets

A := {x : P(x)}

B := {x : Q(x)}

Let x be arbitrary

1.1.1. $x \in A$ Assumption

1.1.2. $P(x)$ Def of A

1.1.?. $Q(x)$??

1.1.?. $x \in B$ Def of B

1..1. $x \in A \rightarrow x \in B$ Direct Proof

1. $\forall x\,(x \in A \rightarrow x \in B)$ Intro ∀: 1

2. $A \subseteq B$ Def of Subset: 2

Proofs About Sets

$A := \{x : P(x)\}$ $B := \{x : Q(x)\}$

Prove that $A \subseteq B$**.**

Proof: Let x be an arbitrary object.

Suppose that $x \in A$. By definition of A, this means P(x).

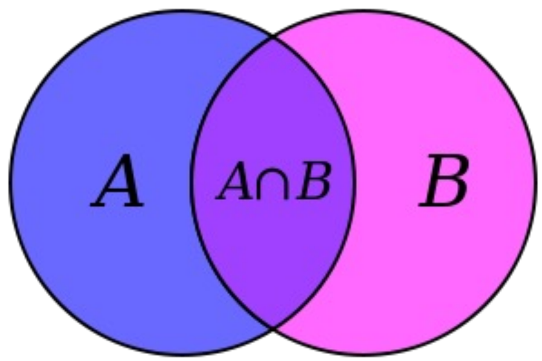
…

Thus, we have Q(x). By definition of B, this means $x \in B$.

Since x was arbitrary, we have shown, by definition, that $A \subseteq B$.

English *template* for a Subset Proof

Operations on Sets



Set Operations

$$A \cup B := \{\, x : (x \in A) \lor (x \in B)\}$$ **Union**

$$A \cap B := \{\, x : (x \in A) \land (x \in B)\}$$ **Intersection**

$$A \setminus B := \{\, x : (x \in A) \land (x \notin B)\}$$ **Set Difference**

A = {1, 2, 3}
B = {3, 5, 6}
C = {3, 4}

QUESTIONS

Using A, B, C and set operations, make…

[6] =

{3} =

{1,2} =

More Set Operations

$$A \oplus B := \{\, x : (x \in A) \oplus (x \in B) \}$$

Symmetric Difference

$$\overline{A} = A^C := \{\, x : x \in U \wedge x \notin A \,\}$$

(with respect to universe U)

Complement

A = {1, 2, 3}
B = {1, 2, 4, 6}
Universe:
U = {1, 2, 3, 4, 5, 6}

$A \oplus B$ = {3, 4, 6}
$\overline{A}$ = {4,5,6}

Note that $A \cup \overline{A} = U$

De Morgan's Laws

$$\overline{A \cup B} = \bar{A} \cap \bar{B}$$

$$\overline{A \cap B} = \bar{A} \cup \bar{B}$$

De Morgan's Laws

Prove that $(A \cup B)^C = A^C \cap B^C$

Formally, prove $\forall x\,(x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C)$

Proof: Let x be an arbitrary object.

Since x was arbitrary, we have shown, by definition, that $(A \cup B)^C = A^C \cap B^C$.

Proof technique:
To show C = D show
$x \in C \rightarrow x \in D$ and
$x \in D \rightarrow x \in C$

De Morgan's Laws

Formally, prove $\forall x\, (x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C)$

1. Let x be arbitrary

    2.1. $x \in (A \cup B)^C$ Assumption

    ...

    2.3. $x \in A^C \cap B^C$

2. $x \in (A \cup B)^C \rightarrow x \in A^C \cap B^C$ Direct Proof

    3.1. $x \in A^C \cap B^C$ Assumption

    ...

    3.3. $x \in (A \cup B)^C$

3. $x \in A^C \cap B^C \rightarrow x \in (A \cup B)^C$ Direct Proof
4. $(x \in (A \cup B)^C \rightarrow x \in A^C \cap B^C) \wedge (x \in A^C \cap B^C \rightarrow x \in (A \cup B)^C)$ Intro $\wedge$: 2, 3
5. $x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C$ Biconditional: 4
6. $\forall x\, (x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C)$ Intro $\forall$: 1-5

De Morgan's Laws

Prove that $(A \cup B)^C = A^C \cap B^C$

Formally, prove $\forall x \, (x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C)$

Proof: Let x be an arbitrary object.

Suppose $x \in (A \cup B)^C$.

...

Thus, we have $x \in A^C \cap B^C$.

De Morgan's Laws

Prove that $(A \cup B)^C = A^C \cap B^C$

Formally, prove $\forall$x $(x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C)$

Proof: Let x be an arbitrary object.

Suppose $x \in (A \cup B)^C$. Then, by the definition of complement, we have $\neg(x \in A \cup B)$.

…

Thus, we have $x \in A^C \cap B^C$.

De Morgan's Laws

Prove that $(A \cup B)^C = A^C \cap B^C$

Formally, prove $\forall x\ (x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C)$

Proof: Let x be an arbitrary object.

Suppose $x \in (A \cup B)^C$. Then, by the definition of complement, we have $\neg(x \in A \cup B)$. The latter says, by the definition of union, that $\neg(x \in A \lor x \in B)$.

…

Thus, we have $x \in A^C \cap B^C$.

De Morgan's Laws

Prove that $(A \cup B)^C = A^C \cap B^C$

Formally, prove $\forall x\ (x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C)$

Proof: Let x be an arbitrary object.

Suppose $x \in (A \cup B)^C$. Then, by the definition of complement, we have $\neg(x \in A \cup B)$. The latter says, by the definition of union, that $\neg(x \in A \lor x \in B)$.

…

Thus, $x \in A^C$ and $x \in B^C$, so we we have $x \in A^C \cap B^C$ by the definition of intersection.

De Morgan's Laws

Prove that $(A \cup B)^C = A^C \cap B^C$

Formally, prove $\forall \mathrm{x}\ (x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C)$

Proof: Let x be an arbitrary object.

Suppose $x \in (A \cup B)^C$. Then, by the definition of complement, we have $\neg(x \in A \cup B)$. The latter says, by the definition of union, that $\neg(x \in A \vee x \in B)$.

…

Thus, $\neg(x \in A)$ and $\neg(x \in B)$, so $x \in A^C$ and $x \in B^C$ by the definition of compliment, and we can see that $x \in A^C \cap B^C$ by the definition of intersection.

De Morgan's Laws

Prove that $(A \cup B)^C = A^C \cap B^C$

Formally, prove $\forall x\ (x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C)$

Proof: Let x be an arbitrary object.

Suppose $x \in (A \cup B)^C$. Then, by the definition of complement, we have $\neg(x \in A \cup B)$. The latter says, by the definition of union, that $\neg(x \in A \vee x \in B)$, or equivalently $\neg(x \in A) \wedge \neg(x \in B)$ by De Morgan's law. Thus, we have $x \in A^C$ and $x \in B^C$ by the definition of compliment, and we can see that $x \in A^C \cap B^C$ by the definition of intersection.

Proof technique:
To show C = D show
$x \in C \rightarrow x \in D$ and
$x \in D \rightarrow x \in C$

De Morgan's Laws

Prove that $(A \cup B)^C = A^C \cap B^C$

Formally, prove $\forall \mathrm{x}\ (x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C)$

Proof: Let x be an arbitrary object.

Suppose $x \in (A \cup B)^C$.... Then, $x \in A^C \cap B^C$.

Suppose $x \in A^C \cap B^C$. Then, by the definition of intersection, we have $x \in A^C$ and $x \in B^C$. That is, we have $\neg(x \in A) \land \neg(x \in B)$, which is equivalent to $\neg(x \in A \lor x \in B)$ by De Morgan's law. The last is equivalent to $\neg(x \in A \cup B)$, by the definition of union, so we have shown $x \in (A \cup B)^C$, by the definition of complement.

Proofs About Set Equality

A lot of *repetitive* work to show $\longrightarrow$ and $\longleftarrow$.

Do we have a way to prove $\leftrightarrow$ directly?

Recall that $A \equiv B$ and $(A \leftrightarrow B) \equiv T$ are the same

We can use an equivalence chain to prove that a biconditional holds.

De Morgan's Laws

Prove that $(A \cup B)^C = A^C \cap B^C$

Formally, prove $\forall x\ (x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C)$

Proof: Let x be an arbitrary object.

The stated biconditional holds since:

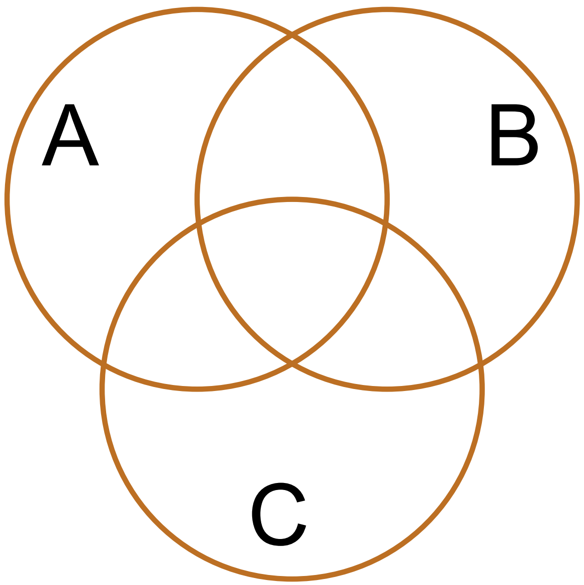
$x \in (A \cup B)^C$	$\equiv \neg(x \in A \cup B)$	Def of Comp
	$\equiv \neg(x \in A \lor x \in B)$	Def of Union
	$\equiv \neg(x \in A) \land \neg(x \in B)$	De Morgan
	$\equiv x \in A^C \land x \in B^C$	Def of Comp
	$\equiv x \in A^C \cap B^C$	Def of Intersection

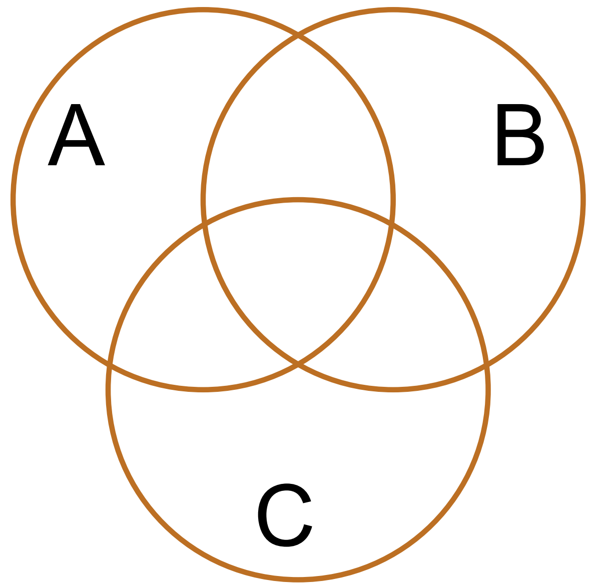
Chains of equivalences are often easier to read like this rather than as English text

Since x was arbitrary, we have shown the sets are equal. ■

Distributive Laws

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$
$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$





It’s Propositional Logic again!

The Meta Theorem Template

Meta-Theorem: Translate any Propositional Logic equivalence into "=" relationship between sets by replacing ∪ with ∨, ∩ with ∧, and $\cdot^C$ with ¬.

"Proof": Let x be an arbitrary object.

The stated bi-condition holds since:

$x \in$ left side ≡ replace set ops with propositional logic

≡ apply Propositional Logic equivalence

≡ replace propositional logic with set ops

≡ $x \in$ right side

Since x was arbitrary, we have shown the sets are equal. ■

Power Set

- Power Set of a set A = set of all subsets of A

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

- e.g., let Days={M,W,F} and consider all the possible sets of days in a week you could ask a question in class

$\mathcal{P}$(Days)=?

$\mathcal{P}(\varnothing)$=?

Power Set

- **Power Set of a set A = set of all subsets of A**

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

- **e.g., let Days={M,W,F} and consider all the possible sets of days in a week you could ask a question in class**

$\mathcal{P}(\text{Days})=\{\{M, W, F\}, \{M, W\}, \{M, F\}, \{W, F\}, \{M\}, \{W\}, \{F\}, \varnothing\}$

$\mathcal{P}(\varnothing)=?$

Power Set

- Power Set of a set A = set of all subsets of A

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

- e.g., let Days={M,W,F} and consider all the possible sets of days in a week you could ask a question in class

$\mathcal{P}$(Days)=$\{\{M, W, F\}, \{M, W\}, \{M, F\}, \{W, F\}, \{M\}, \{W\}, \{F\}, \varnothing\}$

$\mathcal{P}(\varnothing)=\{\varnothing\} \neq \varnothing$

Cartesian Product

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

$\mathbb{R} \times \mathbb{R}$ is the real plane. You've seen ordered pairs before.

These are just for arbitrary sets.

$\mathbb{Z} \times \mathbb{Z}$ is "the set of all pairs of integers"

If A = {1, 2}, B = {a, b, c}, then A × B = {(1,a), (1,b), (1,c),
(2,a), (2,b), (2,c)}.

Cartesian Product

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

$\mathbb{R} \times \mathbb{R}$ is the real plane. You've seen ordered pairs before.

These are just for arbitrary sets.

$\mathbb{Z} \times \mathbb{Z}$ is "the set of all pairs of integers"

If A = {1, 2}, B = {a, b, c}, then A × B = {(1,a), (1,b), (1,c),
(2,a), (2,b), (2,c)}.

What is $A \times \emptyset$?

Cartesian Product

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

$\mathbb{R} \times \mathbb{R}$ is the real plane. You've seen ordered pairs before.

These are just for arbitrary sets.

$\mathbb{Z} \times \mathbb{Z}$ is "the set of all pairs of integers"

If $A = \{1, 2\}$, $B = \{a, b, c\}$, then $A \times B = \{(1,a), (1,b), (1,c), (2,a), (2,b), (2,c)\}$.

$$\boldsymbol{A} \times \emptyset = \{(\boldsymbol{a}, \boldsymbol{b}) : \boldsymbol{a} \in \boldsymbol{A} \land \boldsymbol{b} \in \emptyset\} = \{(\boldsymbol{a}, \boldsymbol{b}) : \boldsymbol{a} \in \boldsymbol{A} \land \mathbf{F}\} = \emptyset$$

More Set Builder Notation

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

- **This can be written more concisely as follows…**

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

- **within set builder variables are implicitly ∃-quantified**

  this is the one <u>exception</u> to the rule that
  unbound variables are implicitly ∀-quantified

More Set Builder Notation

$$S := \{\ x : P(x)\ \}$$

"filter"

- Then $x \in S$ tells us that $\mathrm{P}(x)$ holds

$$T := \{\ f(x) :\ x \in U\ \}$$

"map"

- Then $y \in T$ tells us that $y = f(x)$ for some $x \in U$

More Set Builder Notation

- **Both notations can be used together, e.g.**

$$V := \{\, f(x) : x \in U, P(x)\}$$

- **Then $y \in V$ tells us that $y = f(x)$ for *some* x such that $\mathrm{P}(x)$ holds**

these two notations can be thought of as "filter" and "map"
they are widely used operations in programming as well

Russell's Paradox

$$S := \{x : x \notin x\}$$

Suppose that $S \in S$**...**

Russell's Paradox

$$S := \{x : x \notin x\}$$

Suppose that $S \in S$. Then, by the definition of S, $S \notin S$, but that's a contradiction.

Suppose that $S \notin S$. Then, by the definition of S, $S \in S$, but that's a contradiction too.

This is reminiscent of the truth value of the statement "This statement is false."

Representing Sets Using Bits

- Suppose universe U is $\{1,2,\ldots,n\}$
- Can represent set $B \subseteq U$ as a vector of bits:

  $b_1 b_2 \ldots b_n$ where $b_i = 1$ when $i \in B$

  $b_i = 0$ when $i \notin B$

  – Called the *characteristic vector* of set B

- Given characteristic vectors for A and B
  – What is characteristic vector for $A \cup B$? $A \cap B$?

Bitwise Operations

$$\begin{array}{rl} & 01101101 \\ \vee & \underline{00110111} \\ & 01111111 \end{array}$$

Java: `z=x|y`

$$\begin{array}{rl} & 00101010 \\ \wedge & \underline{00001111} \\ & 00001010 \end{array}$$

Java: `z=x&y`

$$\begin{array}{rl} & 01101101 \\ \oplus & \underline{00110111} \\ & 01011010 \end{array}$$

Java: `z=x^y`

Recursive Definitions of Functions

Recursive definitions of functions

- $0! = 1;\ \ (n+1)! = (n+1)\cdot n!$ **for all** $n \geq\ 0$**.**
- $F(0) = 0;\ \ F(n+1) = F(n) - 1$ **for all** $n \geq\ 0$**.**
- $G(0) = 1;\ \ G(n+1) = 2\cdot G(n)$ **for all** $n \geq\ 0$**.**
- $H(0) = 1;\ \ H(n+1) = 2^{H(n)}$ **for all** $n \geq\ 0$**.**

Prove $n! \leq n^n$ for all $n \geq 1$

1. **Let** P(n) **be** "$n! \leq n^n$". **We will show that** P(n) **is true for all integers** $n \geq 1$ **by induction.**
2. **Base Case** (n=1): $1! = 1 \cdot 0! = 1 \cdot 1 = 1 = 1^1$ **so** P(1) **is true.**
3. **Inductive Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** $k \geq 1$**. I.e., suppose** $k! \leq k^k$.

Prove $n! \le n^n$ for all $n \ge 1$

1. Let P(n) be "$n! \le n^n$". We will show that P(n) is true for all integers $n \ge 1$ by induction.
2. Base Case (n=1): $1!=1 \cdot 0!=1 \cdot 1=1=1^1$ so P(1) is true.
3. Inductive Hypothesis: Suppose that P(k) is true for some arbitrary integer $k \ge 1$. I.e., suppose $k! \le k^k$.
4. Inductive Step:

Goal: Show P(k+1), i.e. show $(k+1)! \le (k+1)^{k+1}$

$(k+1)!$	$= (k+1) \cdot k!$	by definition of !
	$\le (k+1) \cdot k^k$	by the IH
	$\le (k+1) \cdot (k+1)^k$	since $k \ge 0$
	$= (k+1)^{k+1}$	

Therefore P(k+1) is true.

5. Thus P(n) is true for all $n \ge 1$, by induction.

More Recursive Definitions

Suppose that $h: \mathbb{N} \to \mathbb{R}$.

Then we have familiar summation notation:

$$\sum_{i=0}^{0} h(i) = h(0)$$

$$\sum_{i=0}^{n+1} h(i) = h(n+1) + \sum_{i=0}^{n} h(i) \text{ for } n \geq 0$$

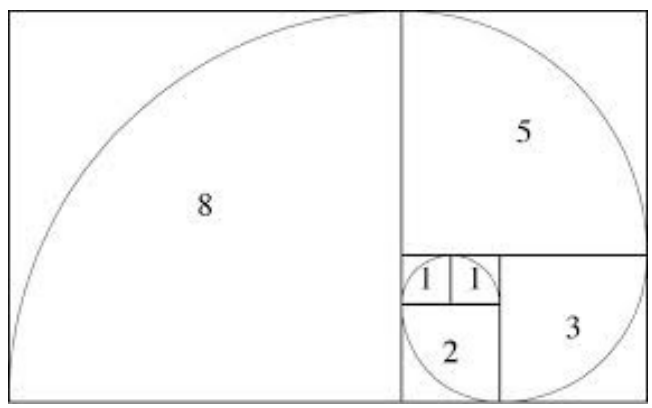
There is also product notation:

$$\prod_{i=0}^{0} h(i) = h(0)$$

$$\prod_{i=0}^{n+1} h(i) = h(n+1) \cdot \prod_{i=0}^{n} h(i) \text{ for } n \geq 0$$

Fibonacci Numbers

$$f_0 = 0$$
$$f_1 = 1$$
$$f_n = f_{n-1} + f_{n-2} \text{ for all } n \geq 2$$



Fibonacci Numbers

$$f_0 = 0$$
$$f_1 = 1$$
$$f_n = f_{n-1} + f_{n-2}$$ **for all** $n \geq 2$

Tamás Görbe
@TamasGorbe

A Mathematician's Way* of Converting Miles to Kilometers

$$\begin{aligned} 3 \text{ mi} &\approx 5 \text{ km} \\ 5 \text{ mi} &\approx 8 \text{ km} \\ 8 \text{ mi} &\approx 13 \text{ km} \end{aligned}$$

f_n mi ≈ f_{n+1} km

Bounding Fibonacci: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be "**$f_n < 2^n$**". We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**

$f_0 = 0 \quad f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be** "$f_n < 2^n$". **We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Case:** $f_0 = 0 < 1 = 2^0$ **so** P(0) **is true.**

$f_0 = 0 \quad f_1 = 1$
$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be "**$f_n < 2^n$**". We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Cases:** $f_0 = 0 < 1 = 2^0$ **so** P(0) **is true and**
   $f_1 = 1 < 2 = 2^1$ **so** P(1) **is true.**

$f_0 = 0 \quad f_1 = 1$
$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be "**$f_n < 2^n$**". We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Cases:** $f_0=0 < 1= 2^0$ **so** P(0) **is true and**
   $f_1 = 1 < 2 = 2^1$ **so** P(1) **is true.**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 1$**, we have** $f_j < 2^j$ **for every integer** j **from** 0 **to** k**.**

$$f_0 = 0 \quad f_1 = 1$$
$$f_n = f_{n-1} + f_{n-2} \text{ for all } n \geq 2$$

Bounding Fibonacci: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be "**$f_n < 2^n$**". We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Cases:** $f_0=0 < 1= 2^0$ **so** P(0) **is true and**
   $f_1 = 1 < 2 = 2^1$ **so** P(1) **is true.**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 1$**, we have** $f_j < 2^j$ **for every integer** j **from** 0 **to** k**.**
4. **Inductive Step: Goal: Show** P(k+1)**; that is,** $f_{k+1} < 2^{k+1}$

$f_0 = 0 \quad f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be "**$f_n < 2^n$**". We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Cases:** $f_0=0 < 1= 2^0$ **so** P(0) **is true and**
   $f_1 = 1 < 2 = 2^1$ **so** P(1) **is true.**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 1$**, we have** $f_j < 2^j$ **for every integer** j **from** 0 **to** k**.**
4. **Inductive Step:** **Goal: Show** P(k+1)**; that is,** $f_{k+1} < 2^{k+1}$

   Since $k+1 \geq 2$, $f_{k+1} = f_k + f_{k-1}$ **by definition**

   $< 2^k + 2^{k-1}$ **by the IH since** $k-1 \geq 0$

   $< 2^k + 2^k$

   $= 2 \cdot 2^k$

   $= 2^{k+1}$

   so P(k+1) **is true.**

$f_0 = 0 \quad f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be** "$f_n < 2^n$". **We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Cases:** $f_0=0 < 1= 2^0$ **so** P(0) **is true and**
   $f_1 = 1 < 2 = 2^1$ **so** P(1) **is true.**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 1$, **we have** $f_j < 2^j$ **for every integer** j **from** 0 **to** k.
4. **Inductive Step:** **Goal: Show** P(k+1); **that is,** $f_{k+1} < 2^{k+1}$

   Since $k+1 \geq 2$, $f_{k+1} = f_k + f_{k-1}$ **by definition**

   $< 2^k + 2^{k-1}$ **by the IH since** $k-1 \geq 0$

   $< 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}$

   so P(k+1) **is true.**
5. **Therefore, by strong induction,** $f_n < 2^n$ **for all integers** $n \geq 0$.

$f_0 = 0 \quad f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Recursive Definitions of Sets

Recursive Definitions of Sets (Data)

Natural numbers

Basis: $0 \in S$

Recursive: If $x \in S$, then $x+1 \in S$

Even numbers

Basis: $0 \in S$

Recursive: If $x \in S$, then $x+2 \in S$

Recursive Definition of Sets

Recursive definition of set S

- **Basis Step:** $0 \in S$
- **Recursive Step: If** $x \in S$**, then** $x + 2 \in S$

The only elements in S are those that follow from the basis step and a finite number of recursive steps

Recursive Definitions of Sets

Natural numbers

Basis: $0 \in S$

Recursive: If $x \in S$, then $x+1 \in S$

Even numbers

Basis: $0 \in S$

Recursive: If $x \in S$, then $x+2 \in S$

Powers of 3:

Basis: $1 \in S$

Recursive: If $x \in S$, then $3x \in S$.

Basis: $(0, 0) \in S$, $(1, 1) \in S$

Recursive: If $(n-1, x) \in S$ and $(n, y) \in S$, then $(n+1, x + y) \in S$. ?

Recursive Definitions of Sets

Natural numbers

Basis: $0 \in S$

Recursive: If $x \in S$, then $x+1 \in S$

Even numbers

Basis: $0 \in S$

Recursive: If $x \in S$, then $x+2 \in S$

Powers of 3:

Basis: $1 \in S$

Recursive: If $x \in S$, then $3x \in S$.

Basis: $(0, 0) \in S$, $(1, 1) \in S$

Recursive: If $(n-1, x) \in S$ and $(n, y) \in S$, then $(n+1, x + y) \in S$.

Fibonacci numbers

Last time: Recursive definitions of functions

- Before, we considered only simple data
  - inputs and outputs were just integers
- Proved facts about those functions with **induction**
  - $n! \leq n^n$
  - $f_n < 2^n$ and $f_n \geq 2^{n/2-1}$
- How do we prove facts about functions that work with more complex (recursively defined) data?
  - we need a more sophisticated form of induction

Structural Induction

How to prove $\forall\, x \in S, P(x)$ is true:

Base Case: Show that $P(u)$ is true for all specific elements u of S mentioned in the *Basis step*

Inductive Hypothesis: Assume that P is true for some arbitrary values of *each* of the existing named elements mentioned in the *Recursive step*

Inductive Step: Prove that $P(w)$ holds for each of the new elements w constructed in the *Recursive step* using the named elements mentioned in the Inductive Hypothesis

Conclude that $\forall\, x \in S, P(x)$

Structural Induction

Basis: $0 \in S$
Recursive: If $x \in S$, then $x+2 \in S$

How to prove $\forall\, x \in S, P(x)$ is true:

Base Case: Show that $P(u)$ is true for all specific elements u of S mentioned in the *Basis step*

Inductive Hypothesis: Assume that P is true for some arbitrary values of *each* of the existing named elements mentioned in the *Recursive step*

Inductive Step: Prove that $P(w)$ holds for each of the new elements w constructed in the *Recursive step* using the named elements mentioned in the Inductive Hypothesis

Conclude that $\forall\, x \in S, P(x)$

Structural Induction vs. Ordinary Induction

Ordinary induction is a special case of structural induction:

Recursive definition of $\mathbb{N}$

Basis: $0 \in \mathbb{N}$

Recursive step: If $k \in \mathbb{N}$ then $k+1 \in \mathbb{N}$

Structural induction follows from ordinary induction:

Define $Q(n)$ to be "for all $x \in S$ that can be constructed in at most n recursive steps, $P(x)$ is true."

Using Structural Induction

- Let S be given by…
  - **Basis:** $6 \in S$; $15 \in S$
  - **Recursive:** if $x, y \in S$ then $x + y \in S$.

Claim: Every element of S is divisible by 3.

Claim: Every element of S is divisible by 3.

1. Let P(x) be "3|x". We prove that P(x) is true for all x ∈ S by structural induction.

Basis: $6 \in S$; $15 \in S$

Recursive: if $x, y \in S$, then $x + y \in S$

Claim: Every element of S is divisible by 3.

1. **Let** P(x) **be** "3|x". **We prove that** P(x) **is true for all** x ∈ S **by structural induction.**
2. **Base Case:** 3|6 **and** 3|15 **so** P(6) **and** P(15) **are true**

Basis: $6 \in S$; $15 \in S$

Recursive: if $x, y \in S$, then $x + y \in S$

Claim: Every element of S is divisible by 3.

1. **Let** P(x) **be** "3|x". **We prove that** P(x) **is true for all** x ∈ S **by structural induction.**
2. **Base Case:** 3|6 **and** 3|15 **so** P(6) **and** P(15) **are true**
3. **Inductive Hypothesis: Suppose that** P(x) **and** P(y) **are true for some arbitrary** x,y ∈ S
4. **Inductive Step:** **Goal: Show** P(x+y)

Basis: $6 \in S$; $15 \in S$

Recursive: if $x, y \in S$, then $x + y \in S$

Claim: Every element of S is divisible by 3.

1. **Let** P(x) **be** "3|x". **We prove that** P(x) **is true for all** x ∈ S **by structural induction.**
2. **Base Case:** 3|6 **and** 3|15 **so** P(6) **and** P(15) **are true**
3. **Inductive Hypothesis: Suppose that** P(x) **and** P(y) **are true for some arbitrary** x,y ∈ S
4. **Inductive Step:** **Goal: Show** P(x+y)

   Since P(x) **is true,** 3|x **and so** x=3m **for some integer** m **and since** P(y) **is true,** 3|y **and so** y=3n **for some integer** n. **Therefore** x+y=3m+3n=3(m+n) **and thus** 3|(x+y).

   Hence P(x+y) **is true.**

Basis: $6 \in S$; $15 \in S$

Recursive: if $x, y \in S$, then $x + y \in S$

Claim: Every element of S is divisible by 3.

1. **Let** P(x) **be** "3|x". **We prove that** P(x) **is true for all** x ∈ S **by structural induction.**
2. **Base Case:** 3|6 **and** 3|15 **so** P(6) **and** P(15) **are true**
3. **Inductive Hypothesis: Suppose that** P(x) **and** P(y) **are true for some arbitrary** x,y ∈ S
4. **Inductive Step:** **Goal: Show** P(x+y)

   Since P(x) **is true,** 3|x **and so** x=3m **for some integer** m **and since** P(y) **is true,** 3|y **and so** y=3n **for some integer** n. **Therefore** x+y=3m+3n=3(m+n) **and thus** 3|(x+y).

   Hence P(x+y) **is true.**
5. **Therefore by induction** 3|x **for all** x ∈ S.

Basis: $6 \in S$; $15 \in S$

Recursive: if $x, y \in S$, then $x + y \in S$

Using Structural Induction

- Let T be given by...
  - **Basis:** $6 \in T$; $15 \in T$
  - **Recursive:** if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$

- Two base cases and two *recursive* cases

Claim: Every element of T is also in S.

Claim: Every element of T is an element of S

1. **Let** P(x) **be** "x ∈ S". **We prove that** P(x) **is true for all** x ∈ T **by structural induction.**

Basis: $6 \in S$; $15 \in S$	**Basis:** $6 \in T$; $15 \in T$
Recursive: if $x, y \in S$, then $x + y \in S$	**Recursive:** if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$

Claim: Every element of T is an element of S

1. **Let** P(x) **be** "x ∈ S". **We prove that** P(x) **is true for all** x ∈ T **by structural induction.**
2. **Base Case:** 6 ∈ S **and** 15 ∈ S **so** P(6) **and** P(15) **are true**

Basis: $6 \in S$; $15 \in S$	**Basis:** $6 \in T$; $15 \in T$
Recursive: if $x, y \in S$, then $x + y \in S$	**Recursive:** if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$

Claim: Every element of T is an element of S

1. Let P(x) be "x ∈ S". We prove that P(x) is true for all x ∈ T by structural induction.
2. Base Case: 6 ∈ S and 15 ∈ S so P(6) and P(15) are true
3. Inductive Hypothesis: Suppose that P(x) is true for some arbitrary x ∈ T

Basis: $6 \in S$; $15 \in S$	**Basis:** $6 \in T$; $15 \in T$
Recursive: if $x, y \in S$, then $x + y \in S$	**Recursive:** if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$

Claim: Every element of T is an element of S

1. **Let** P(x) **be** "x ∈ S". **We prove that** P(x) **is true for all** x ∈ T **by structural induction.**
2. **Base Case:** 6 ∈ S **and** 15 ∈ S **so** P(6) **and** P(15) **are true**
3. **Inductive Hypothesis: Suppose that** P(x) **is true for some arbitrary** x ∈ T
4. **Inductive Step: Goal: Show** P(x+6) **and** P(x+15)

Basis: $6 \in S$; $15 \in S$	**Basis:** $6 \in T$; $15 \in T$
Recursive: if $x, y \in S$, then $x + y \in S$	**Recursive:** if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$

Claim: Every element of T is an element of S

1. Let P(x) be "x ∈ S". We prove that P(x) is true for all x ∈ T by structural induction.
2. Base Case: 6 ∈ S and 15 ∈ S so P(6) and P(15) are true
3. Inductive Hypothesis: Suppose that P(x) is true for some arbitrary x ∈ T
4. Inductive Step: Goal: Show P(x+6) and P(x+15)

   Since P(x) holds, we have x ∈ S. From the recursive step of S, we can see that x + 6 ∈ S, so P(x+6) is true, and we can see that x + 15 ∈ S, so P(x+15) is true.

Basis: $6 \in S$; $15 \in S$	**Basis:** $6 \in T$; $15 \in T$
Recursive: if $x, y \in S$, then $x + y \in S$	**Recursive:** if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$

Claim: Every element of T is an element of S

1. **Let** P(x) **be** "x ∈ S". **We prove that** P(x) **is true for all** x ∈ T **by structural induction.**
2. **Base Case:** 6 ∈ S **and** 15 ∈ S **so** P(6) **and** P(15) **are true**
3. **Inductive Hypothesis: Suppose that** P(x) **is true for some arbitrary** x ∈ T
4. **Inductive Step:** **Goal: Show** P(x+6) **and** P(x+15)

   Since P(x) **holds, we have** x ∈ S. **From the recursive step of** S**,**

   we can see that x + 6 ∈ S, **so** P(x+6) **is true, and**

   we can see that x + 15 ∈ S, **so** P(x+15) **is true.**
5. **Therefore** P(x) **for all** x ∈ T **by induction.**

Basis: $6 \in S$; $15 \in S$	**Basis:** $6 \in T$; $15 \in T$
Recursive: if $x, y \in S$, then $x + y \in S$	**Recursive:** if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$

Lists of Integers

- **Basis:** $\text{nil} \in$ **List**
- **Recursive step:**

  if $L \in$ **List and** $a \in \mathbb{Z}$**,**

  then $a :: L \in$ **List**

Examples:

- nil
- $1 :: \text{nil}$ 1
- $1 :: 2 :: \text{nil}$ $1 \rightarrow 2$
- $1 :: 2 :: 3 :: \text{nil}$ $1 \rightarrow 2 \rightarrow 3$

Functions on Lists

Length:

$\text{len}(\text{nil}) := 0$

$\text{len}(a :: L) := \text{len}(L) + 1$ **for any** L ∈ **List and** a ∈ $\mathbb{Z}$

Concatenation:

$\text{concat}(\text{nil}, R) := R$ **for any** R ∈ **List**

$\text{concat}(a :: L, R) := a :: \text{concat}(L, R)$ **for any** L, R ∈ **List and any** a ∈ $\mathbb{Z}$

Structural Induction

Basis: $\mathrm{nil} \in$ **List**
Recursive step:
  if $\mathrm{L} \in$ **List and** $\mathrm{a} \in \mathbb{Z}$**,**
  then $\mathrm{a} :: \mathrm{L} \in$ **List**

How to prove $\forall\, x \in S, P(x)$ is true:

Base Case: Show that $P(u)$ is true for all specific elements u of S mentioned in the *Basis step*

Inductive Hypothesis: Assume that P is true for some arbitrary values of *each* of the existing named elements mentioned in the *Recursive step*

Inductive Step: Prove that $P(w)$ holds for each of the new elements w constructed in the *Recursive step* using the named elements mentioned in the Inductive Hypothesis

Conclude that $\forall\, x \in S, P(x)$

Claim: $\text{concat}(\text{L}, \text{nil}) = \text{L}$ **for all** $\text{L} \in$ **List**

Claim: $\text{concat}(\text{L}, \text{nil}) = \text{L}$ **for all** $\text{L} \in$ **List**

Let $\text{P}(\text{L})$ **be** "$\text{concat}(\text{L}, \text{nil}) = \text{L}$" .

We will prove $\text{P}(\text{L})$ **for all** $\text{L} \in$ **List by structural induction.**

Claim: $\text{concat}(L, \text{nil}) = L$ for all $L \in$ **List**

Let $P(L)$ **be** "$\text{concat}(L, \text{nil}) = L$" .

We will prove $P(L)$ **for all** $L \in$ **List by structural induction.**

Base Case (nil): **By the definition of concat, we can see that** $\text{concat}(\text{nil}, \text{nil}) = \text{nil}$, **which is** $P(\text{nil})$**.**

Claim: $\text{concat}(L, \text{nil}) = L$ **for all** $L \in$ **List**

Let P(L) **be** "$\text{concat}(L, \text{nil}) = L$" .
We will prove P(L) **for all** L ∈ **List by structural induction.**

Base Case (nil): **By the definition of concat, we can see that** $\text{concat}(\text{nil}, \text{nil}) = \text{nil}$**, which is** P(nil)**.**

Inductive Hypothesis: Assume that P(L) **is true for some arbitrary** L ∈ List**, i.e.,** $\text{concat}(L, \text{nil}) = L$.

Inductive Step: **Goal: Show that** $P(a :: L)$ **is true for any** $a \in \mathbb{Z}$

Claim: $\text{concat}(L, \text{nil}) = L$ for all $L \in$ **List**

Let P(L) **be** "$\text{concat}(L, \text{nil}) = L$".
We will prove P(L) **for all** L ∈ **List by structural induction.**

Base Case (nil): **By the definition of concat, we can see that** $\text{concat}(\text{nil}, \text{nil}) = \text{nil}$, **which is** P(nil).

Inductive Hypothesis: Assume that P(L) **is true for some arbitrary** L ∈ List, **i.e.,** $\text{concat}(L, \text{nil}) = L$.

Inductive Step: **Goal: Show that** P(a :: L) **is true for any** $a \in \mathbb{Z}$.

Let $a \in \mathbb{Z}$ **be arbitrary. We can calculate as follows**

$$\begin{aligned} \text{concat}(a :: L, \text{nil}) &= a :: \text{concat}(L, \text{nil}) && \textbf{def of } \text{concat} \\ &= a :: L && \textbf{IH} \end{aligned}$$

which is P(a :: L).

By induction, we have shown the claim holds for all L ∈ **List**.

Claim: $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** $L, R \in$ **List**

Claim: $\text{len}(\text{concat}(\text{L}, \text{R})) = \text{len}(\text{L}) + \text{len}(\text{R})$ **for all** $\text{L}, \text{R} \in$ **List**

Let P(L) **be** "$\text{len}(\text{concat}(\text{L}, \text{R})) = \text{len}(\text{L}) + \text{len}(\text{R})$ **for all** R ∈ **List**" .
We prove P(L) **for all** L ∈ **List by structural induction.**

Claim: $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** $L, R \in$ **List**

Let $P(L)$ **be** "$\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** $R \in$ **List**" .
We prove $P(L)$ **for all** $L \in$ **List by structural induction.**

Base Case (nil): **Let** $R \in$ **List be arbitrary. Then,**

Length:

$\text{len}(\text{nil}) := 0$
$\text{len}(a :: L) := \text{len}(L) + 1$

Concatenation:

$\text{concat}(\text{nil}, R) := R$
$\text{concat}(a :: L, R) := a :: \text{concat}(L, R)$

Claim: $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** $L, R \in$ **List**

Let P(L) **be** "$\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** R ∈ **List**".
We prove P(L) **for all** L ∈ **List by structural induction.**

Base Case (nil): **Let** R ∈ **List be arbitrary. Then,**

$$\begin{aligned} \text{len}(\text{concat}(\text{nil}, R)) &= \text{len}(R) && \textbf{def of concat} \\ &= 0 + \text{len}(R) \\ &= \text{len}(\text{nil}) + \text{len}(R) && \textbf{def of len} \end{aligned}$$

Since R **was arbitrary,** P(nil) **holds.**

Claim: $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** L, R ∈ **List**

Let P(L) **be** "$\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** R ∈ **List**".
We prove P(L) **for all** L ∈ **List by structural induction.**

Base Case (nil): **Let** R ∈ **List be arbitrary. Then,** $\text{len}(\text{concat}(\text{nil}, R))$ $= \text{len}(R) = 0 + \text{len}(R) = \text{len}(\text{nil}) + \text{len}(R)$, **showing** P(nil).

Inductive Hypothesis: Assume that P(L) **is true for some arbitrary** L ∈ List, **i.e.,** $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** R ∈ **List**.

Claim: $\mathrm{len}(\mathrm{concat}(L, R)) = \mathrm{len}(L) + \mathrm{len}(R)$ **for all** L, R ∈ **List**

Let P(L) **be** "$\mathrm{len}(\mathrm{concat}(L, R)) = \mathrm{len}(L) + \mathrm{len}(R)$ **for all** R ∈ **List**".
We prove P(L) **for all** L ∈ **List by structural induction.**

Base Case (nil): **Let** R ∈ **List be arbitrary. Then,** $\mathrm{len}(\mathrm{concat}(\mathrm{nil}, R)) = \mathrm{len}(R) = 0 + \mathrm{len}(R) = \mathrm{len}(\mathrm{nil}) + \mathrm{len}(R)$, **showing** P(nil).

Inductive Hypothesis: Assume that P(L) **is true for some arbitrary** L ∈ List, **i.e.,** $\mathrm{len}(\mathrm{concat}(L, R)) = \mathrm{len}(L) + \mathrm{len}(R)$ **for all** R ∈ **List**.

Inductive Step: **Goal: Show that** P(a :: L) **is true for any** $a \in \mathbb{Z}$**.**

Claim: $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** $L, R \in$ **List**

Let P(L) **be** "$\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** R ∈ **List**".
We prove P(L) **for all** L ∈ **List by structural induction.**

Base Case (nil): **Let** R ∈ **List be arbitrary. Then,** $\text{len}(\text{concat}(\text{nil}, R))$ $= \text{len}(R) = 0 + \text{len}(R) = \text{len}(\text{nil}) + \text{len}(R)$, **showing** P(nil).

Inductive Hypothesis: Assume that P(L) **is true for some arbitrary** L ∈ List, **i.e.,** $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** R ∈ **List**.

Inductive Step: **Goal: Show that** P(a :: L) **is true for any** $a \in \mathbb{Z}$.

Let $a \in \mathbb{Z}$ **and** R ∈ **List be arbitrary. Then,**

Length:

$\text{len}(\text{nil}) := 0$
$\text{len}(a :: L) := \text{len}(L) + 1$

Concatenation:

$\text{concat}(\text{nil}, R) := R$
$\text{concat}(a :: L, R) := a :: \text{concat}(L, R)$

Claim: $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** $L, R \in$ **List**

Let P(L) **be** "$\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** R ∈ **List**".
We prove P(L) **for all** L ∈ **List by structural induction.**

Base Case (nil)**: Let** R ∈ **List be arbitrary. Then,** $\text{len}(\text{concat}(\text{nil}, R))$ $= \text{len}(R) = 0 + \text{len}(R) = \text{len}(\text{nil}) + \text{len}(R)$**, showing** P(nil)**.**

Inductive Hypothesis: Assume that P(L) **is true for some arbitrary** L ∈ List**, i.e.,** $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** R ∈ **List.**

Inductive Step: **Goal: Show that** P(a :: L) **is true for any** $a \in \mathbb{Z}$**.**

Let $a \in \mathbb{Z}$ **and** R ∈ **List be arbitrary. Then, we can calculate**

$$
\begin{aligned}
\text{len}(\text{concat}(a :: L, R)) &= \text{len}(a :: \text{concat}(L, R)) && \textbf{def of } \text{concat} \\
&= 1 + \text{len}(\text{concat}(L, R)) && \textbf{def of } \text{len} \\
&= 1 + \text{len}(L) + \text{len}(R) && \textbf{IH} \\
&= \text{len}(a :: L) + \text{len}(R) && \textbf{def of } \text{len}
\end{aligned}
$$

Since R **was arbitrary, we have shown** P(a :: L)**.**

By induction, we have shown the claim holds for all L ∈ **List.**

Claim: concat(L, concat(R, S)) = concat(concat(L, R), S) **for all** L,R,S ∈ **List**

Let P(L) **be** "concat(concat(L, R, S)) = concat(concat(L, R), S) **for all** R, S ∈ **List**".
We prove P(L) **for all** L ∈ **List by structural induction.**

Claim: concat(L, concat(R, S)) = concat(concat(L, R), S) **for all** L,R,S ∈ **List**

Let P(L) **be** "concat(concat(L, R), S) = concat(concat(L, R), S) **for all** R, S ∈ **List**".
We prove P(L) **for all** L ∈ **List by structural induction.**

Base Case (nil): **Let** R, S **be arbitrary lists. Then, we can see that**

$$
\begin{aligned}
&\text{concat}(\text{concat}(\text{nil}, R), S) \\
&\quad = \text{concat}(R, S) && \textbf{def of } \text{concat} \\
&\quad = \text{concat}(\text{concat}(\text{nil}, R), S) && \textbf{def of } \text{concat}
\end{aligned}
$$

which is P(nil).

Concatenation:

concat(nil, R) := R
concat(a :: L, R) := a :: concat(L, R)

Claim: $\text{concat}(L, \text{concat}(R, S)) = \text{concat}(\text{concat}(L, R), S)$ **for all** L,R,S ∈ **List**

Let P(L) **be** "$\text{concat}(\text{concat}(L, R), S) = \text{concat}(\text{concat}(L, R), S)$ **for all** R, S ∈ **List**".
We prove P(L) **for all** L ∈ **List by structural induction.**

Base Case (nil): **...**

Inductive Hypothesis: Assume that P(L) **is true for an arbitrary** L ∈ List,
i.e., $\text{concat}(L, \text{concat}(R, S)) = \text{concat}(\text{concat}(L, R), S)$ **for all** R, S.

Inductive Step: **Goal: Show that** P(a :: L) **is true for any** a ∈ ℤ.
Let a ∈ ℤ **and** R, S ∈ **List be arbitrary. Then, we can calculate**

Concatenation:

$\text{concat}(\text{nil}, R) := R$
$\text{concat}(a :: L, R) := a :: \text{concat}(L, R)$

Claim: concat(L, concat(R, S)) = concat(concat(L, R), S) **for all** L,R,S ∈ **List**

Let P(L) **be** "concat(concat(L, R), S) = concat(concat(L, R), S) **for all** R, S ∈ **List**".
We prove P(L) **for all** L ∈ **List by structural induction.**

Base Case (nil): **Let** R, S **be arbitrary lists. Then, we can see that** concat(nil, concat(R, S)) = concat(R, S) = concat(concat(nil, R), S), **by the definition of** concat**. This is** P(nil)**.**

Inductive Hypothesis: Assume that P(L) **is true for an arbitrary** L ∈ List, **i.e.**, concat(L, concat(R, S)) = concat(concat(L, R), S) **for all** R, S.

Inductive Step: **Goal: Show that** P(a :: L) **is true for any** a ∈ ℤ**.**

Let a ∈ ℤ **and** R, S ∈ **List be arbitrary. Then, we can calculate**

concat(a :: L, concat(R, S))
= a :: concat(L, concat(R, S)) **def of** concat
= a :: concat(concat(L, R), S) **IH**
= concat(a :: concat(L, R), S) **def of** concat
= concat(concat(a :: L, R), S) **def of** concat

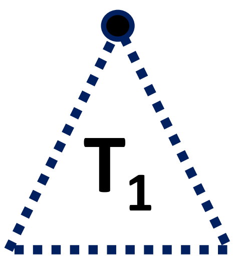
Since R **was arbitrary, we have shown** P(a :: L).
By induction, we have shown the claim holds for all L ∈ **List**.

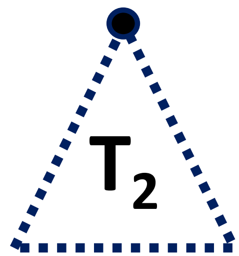
Rooted Binary Trees

- Basis: •   is a rooted binary tree

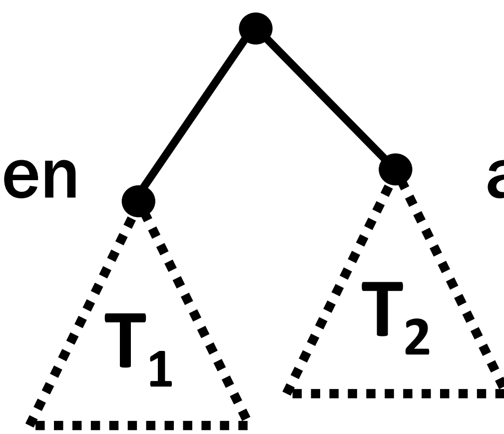
Rooted Binary Trees

- **Basis:**  is a rooted binary tree
- **Recursive step:**

If 

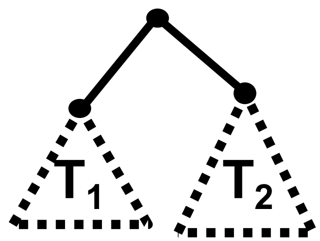
and 

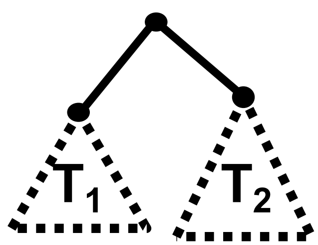
are rooted binary trees,

then 

also is a rooted binary tree.

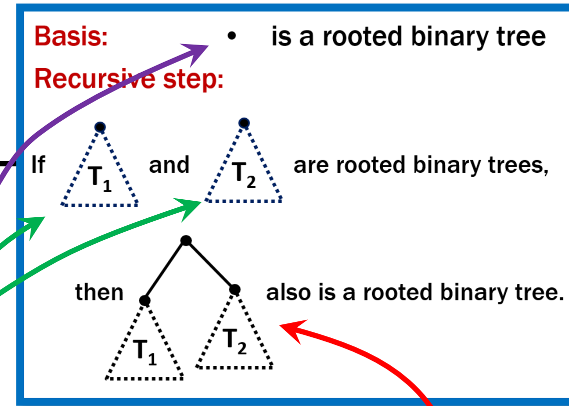
Defining Functions on Rooted Binary Trees

- size(•) ::= 1
- size (

) ::= 1 + size($\mathbf{T_1}$) + size($\mathbf{T_2}$)
- height(•) ::= 0
- height (

) ::= 1 + max{height($\mathbf{T_1}$), height($\mathbf{T_2}$)}

Last time: Structural Induction



How to prove $\forall\, x \in S, P(x)$ is true:

Base Case: Show that $P(u)$ is true for all specific elements u of S mentioned in the ***Basis step***

Inductive Hypothesis: Assume that P is true for some arbitrary values of *each* of the existing named elements mentioned in the ***Recursive step***

Inductive Step: Prove that $P(w)$ holds for each of the new elements w constructed in the ***Recursive step*** using the named elements mentioned in the Inductive Hypothesis

Conclude that $\forall\, x \in S, P(x)$

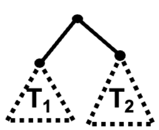
Claim: **For every rooted binary tree T,** size(**T**) $\leq 2^{\text{height}(\mathbf{T}) + 1} - 1$

Claim: **For every rooted binary tree T,** $\text{size}(\mathbf{T}) \le 2^{\text{height}(\mathbf{T})+1} - 1$

---

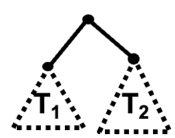
1. Let $P(\mathbf{T})$ **be** "$\text{size}(T) \le 2^{\text{height}(T)+1} - 1$". **We prove** $P(\mathbf{T})$ **for all rooted binary trees T by structural induction.**

size(•) ::= 1

size (

) ::= 1 + size($\mathbf{T_1}$) + size($\mathbf{T_2}$)

height(•) ::= 0

height (

) ::= 1 + max{height($\mathbf{T_1}$), height($\mathbf{T_2}$)}

Claim: **For every rooted binary tree T,** size(**T**) $\leq 2^{\text{height}(\mathbf{T}) + 1} - 1$

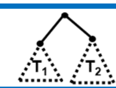
---

1. **Let** P(**T**) **be** "size(T) $\leq 2^{\text{height}(T)+1}-1$". **We prove** P(**T**) **for all rooted binary trees T by structural induction.**
2. **Base Case:** size(•)=1, height(•)=0, **and** $2^{0+1}-1=2^1-1=1$ **so** P(•) **is true.**

Claim: **For every rooted binary tree T,** $\text{size}(\mathbf{T}) \le 2^{\text{height}(\mathbf{T})+1} - 1$

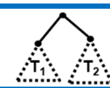
---

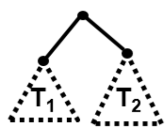
1. **Let** $P(\mathbf{T})$ **be** "$\text{size}(T) \le 2^{\text{height}(T)+1} - 1$". **We prove** $P(\mathbf{T})$ **for all rooted binary trees T by structural induction.**
2. **Base Case:** $\text{size}(\bullet)=1$, $\text{height}(\bullet)=0$, **and** $2^{0+1}-1=2^1-1=1$ **so** $P(\bullet)$ **is true.**
3. **Inductive Hypothesis: Suppose that** $P(\mathbf{T}_1)$ **and** $P(\mathbf{T}_2)$ **are true for some rooted binary trees** $\mathbf{T}_1$ **and** $\mathbf{T}_2$**, i.e.,** $\text{size}(\mathbf{T}_k) \le 2^{\text{height}(\mathbf{T}_k)+1} - 1$ **for** $k=1,2$
4. **Inductive Step:**

Goal: Prove P().

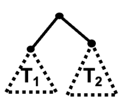
Claim: **For every rooted binary tree T,** size(**T**) $\leq 2^{\text{height}(\mathbf{T})+1} - 1$

1. **Let** P(**T**) **be** "size(T) $\leq 2^{\text{height(T)+1}}$–1". **We prove** P(**T**) **for all rooted binary trees T by structural induction.**
2. **Base Case:** size(•)=1, height(•)=0, **and** 2^{0+1}–1=2^1–1=1 **so** P(•) **is true.**
3. **Inductive Hypothesis: Suppose that** P($\mathbf{T_1}$) **and** P($\mathbf{T_2}$) **are true for some rooted binary trees** $\mathbf{T_1}$ **and** $\mathbf{T_2}$**, i.e.,** size($\mathbf{T_k}$) $\leq 2^{\text{height}(\mathbf{T_k})+1} - 1$ **for** k=1,2
4. **Inductive Step:**

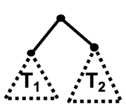
Goal: Prove P().

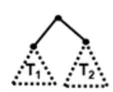
size()

size(•) ::= 1

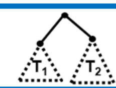
size () ::= 1 + size($\mathbf{T_1}$) + size($\mathbf{T_2}$)

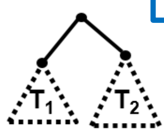
height(•) ::= 0

height () ::= 1 + max{height($\mathbf{T_1}$), height($\mathbf{T_2}$)}

$\leq 2^{\text{height(}}$  $^{)+1} - 1$

Claim: For every rooted binary tree T, size(T) ≤ $2^{\text{height}(\mathbf{T}) + 1} - 1$

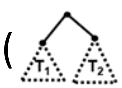
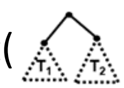
1. Let P(T) be "size(T) ≤ $2^{\text{height}(T)+1} - 1$". We prove P(T) for all rooted binary trees T by structural induction.
2. **Base Case:** size(•)=1, height(•)=0, and $2^{0+1}-1=2^1-1=1$ so P(•) is true.
3. **Inductive Hypothesis:** Suppose that P(T_1) and P(T_2) are true for some rooted binary trees T_1 and T_2, i.e., size(T_k) ≤ $2^{\text{height}(T_k)+1} - 1$ for k=1,2
4. **Inductive Step:** Goal: Prove P().

   By def, size() $=1+\text{size}(\mathbf{T_1})+\text{size}(\mathbf{T_2})$

   $\leq 1+2^{\text{height}(\mathbf{T_1})+1}-1+2^{\text{height}(\mathbf{T_2})+1}-1$ by IH for $\mathbf{T_1}$ and $\mathbf{T_2}$

   $= 2^{\text{height}(\mathbf{T_1})+1}+2^{\text{height}(\mathbf{T_2})+1}-1$

   $\leq 2(2^{\max(\text{height}(\mathbf{T_1}),\text{height}(\mathbf{T_2}))+1})-1$

   = 2(2^height()) – 1 = 2^height()+1 – 1

   which is what we wanted to show.
5. So, the P(T) is true for all rooted binary trees by structural induction.

Strings

- An *alphabet* Σ is any finite set of characters

- The set Σ^* of *strings* over the alphabet Σ
  – example: $\{0,1\}^*$ is the set of *binary strings*
    0, 1, 00, 01, 10, 11, 000, 001, … and “”

- Σ^* is defined recursively by
  – **Basis:** $\varepsilon \in \Sigma^*$ (ε is the empty string, i.e., “”)
  – **Recursive:** if $w \in \Sigma^*$, $a \in \Sigma$, then $wa \in \Sigma^*$

Last time: Structural Induction

Basis: $\varepsilon \in \Sigma^*$
Recursive Steps:
if $w \in \Sigma^*$ and $a \in \Sigma$,
then $wa \in \Sigma^*$

How to prove $\forall\, x \in S, P(x)$ is true:

Base Case: Show that $P(u)$ is true for all specific elements u of S mentioned in the *Basis step*

Inductive Hypothesis: Assume that P is true for some arbitrary values of *each* of the existing named elements mentioned in the *Recursive step*

Inductive Step: Prove that $P(w)$ holds for each of the new elements w constructed in the *Recursive step* using the named elements mentioned in the Inductive Hypothesis

Conclude that $\forall\, x \in S, P(x)$

Functions on Recursively Defined Sets (on Σ^*)

Length:

$\text{len}(\varepsilon) := 0$

$\text{len}(wa) := \text{len}(w) + 1$ for $w \in \Sigma^*$, $a \in \Sigma$

Concatenation:

$x \bullet \varepsilon := x$ for $x \in \Sigma^*$

$x \bullet wa := (x \bullet w)a$ for $x \in \Sigma^*$, $a \in \Sigma$

Reversal:

$\varepsilon^R := \varepsilon$

$(wa)^R := \varepsilon a \bullet w^R$ for $w \in \Sigma^*$, $a \in \Sigma$

Number of c's in a string:

$\#_c(\varepsilon) := 0$

$\#_c(wc) := \#_c(w) + 1$ for $w \in \Sigma^*$

$\#_c(wa) := \#_c(w)$ for $w \in \Sigma^*$, $a \in \Sigma$, $a \neq c$

separate cases for
c vs $a \neq c$

Claim: len(x•y) = len(x) + len(y) for all x,y $\in \Sigma^*$

Let P(y) **be** "len(x•y) = len(x) + len(y) for all x $\in \Sigma^*$" .
We prove P(y) **for all** y $\in \Sigma^*$ **by structural induction.**

Base Case (y = ε): **Let** x $\in \Sigma^*$ **be arbitrary. Then,**

$$\begin{aligned} \text{len}(x \bullet \varepsilon) &= \text{len}(x) && \textbf{def of } \bullet \\ &= \text{len}(x) + 0 \\ &= \text{len}(x) + \text{len}(\varepsilon) && \textbf{def of } \text{len} \end{aligned}$$

Since x **was arbitrary,** P(ε) **holds.**

len(ε)	:= 0
len(wa)	:= len(w) + 1

Claim: len(x•y) = len(x) + len(y) **for all** x,y ∈ Σ^*

Let P(y) **be** "len(x•y) = len(x) + len(y) for all x ∈ Σ^*" .
We prove P(y) **for all** y ∈ Σ^* **by structural induction.**

Base Case (y = ε): **Let** x ∈ Σ^* **be arbitrary. Then,** len(x • ε) = len(x) = len(x) + len(ε) **since** len(ε)=0. **Since** x **was arbitrary,** P(ε) **holds.**

Inductive Hypothesis: Assume that P(w) **is true for some arbitrary** w ∈ Σ^***, i.e.,** len(x•w) = len(x) + len(w) for all x

Claim: $\text{len}(x \bullet y) = \text{len}(x) + \text{len}(y)$ for all $x, y \in \Sigma^*$

Let $P(y)$ **be** "$\text{len}(x \bullet y) = \text{len}(x) + \text{len}(y)$ for all $x \in \Sigma^*$" .
We prove $P(y)$ **for all** $y \in \Sigma^*$ **by structural induction.**

Base Case ($y = \varepsilon$): **Let** $x \in \Sigma^*$ **be arbitrary. Then,** $\text{len}(x \bullet \varepsilon) = \text{len}(x) = \text{len}(x) + \text{len}(\varepsilon)$ **since** $\text{len}(\varepsilon)=0$. **Since** x **was arbitrary,** $P(\varepsilon)$ **holds.**

Inductive Hypothesis: **Assume that** $P(w)$ **is true for some arbitrary** $w \in \Sigma^*$**, i.e.,** $\text{len}(x \bullet w) = \text{len}(x) + \text{len}(w)$ for all x

Inductive Step: **Goal: Show that** $P(wa)$ **is true for every** $a \in \Sigma$

Let $a \in \Sigma$ **and** $x \in \Sigma^*$. **Then,**

$\text{len}(\varepsilon)$	$:= 0$
$\text{len}(wa)$	$:= \text{len}(w) + 1$

Claim: len(x•y) = len(x) + len(y) **for all** x,y ∈ Σ^*

Let P(y) **be** "len(x•y) = len(x) + len(y) for all x ∈
We prove P(y) **for all** y ∈ Σ^* **by structural indu**

Does this look familiar?

Base Case (y = ε): **Let** x ∈ Σ^* **be arbitrary. Then,** len(x • ε) = len(x) = len(x) + len(ε) **since** len(ε)=0. **Since** x **was arbitrary,** P(ε) **holds.**

Inductive Hypothesis: Assume that P(w) **is true for some arbitrary** w ∈ Σ^***, i.e.,** len(x•w) = len(x) + len(w) for all x

Inductive Step: **Goal: Show that** P(wa) **is true for every** a ∈ Σ

Let a ∈ Σ **and** x ∈ Σ^*. **Then** len(x•wa) = len((x•w)a) **def of •**
= len(x•w)+1 **def of len**
= len(x)+len(w)+1 **IH**
= len(x)+len(wa) **def of len**

Therefore, len(x•wa)= len(x)+len(wa) **for all** x ∈ Σ^***, so** P(wa) **is true.**

So, by induction len(x•y) = len(x) + len(y) **for all** x,y ∈ Σ^*

Recall: $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** L, R ∈ **List**

Let P(L) **be** "$\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** R ∈ **List**".
We prove P(L) **for all** L ∈ **List by structural induction.**

Base Case (nil): **Let** $a \in \mathbb{Z}$ **be arbitrary. Then,** $\text{len}(\text{concat}(\text{nil}, R)) = \text{len}(R) = \text{len}(\text{nil}) + \text{len}(R)$. **Since** a **was arbitrary,** $P(\text{nil})$ **holds.**

Inductive Hypothesis: Assume that P(L) **is true for some arbitrary** L ∈ List, **i.e.,** $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** R ∈ **List**.

Inductive Step: **Goal: Show that** $P(a :: L)$ **is true for any** $a \in \mathbb{Z}$**.**

Let $a \in \mathbb{Z}$ **and** R ∈ **List be arbitrary. Then, we can calculate**

$$\begin{aligned}
\text{len}(\text{concat}(a :: L, R)) &= \text{len}(a :: \text{concat}(L, R)) && \textbf{def of } \text{concat} \\
&= 1 + \text{len}(\text{concat}(L, R)) && \textbf{def of } \text{len} \\
&= 1 + \text{len}(L) + \text{len}(R) && \textbf{IH} \\
&= \text{len}(a :: L) + \text{len}(R) && \textbf{def of } \text{len}
\end{aligned}$$

Since R **was arbitrary, we have shown** $P(a :: L)$.

By induction, we have shown the claim holds for all L ∈ **List**.

Lists versus Strings

- **Our strings are basically lists**
  except **that we draw them backward**

[1, 2, 3]	1 :: 2 :: 3 :: nil	1 → 2 → 3
“abc”	εabc	a ⇠ b ⇠ c

  – **would be represented the same way in memory**
  – **but we think of head as the right-most not left-most**

Claim: $\text{len}(x^R) = \text{len}(x)$ for all $x \in \Sigma^*$

Let $P(x)$ **be** "$\text{len}(x^R) = \text{len}(x)$".

We will prove $P(x)$ **for all** $x \in \Sigma^*$ **by structural induction.**

Length:

$\text{len}(\varepsilon) ::= 0$

$\text{len}(wa) ::= \text{len}(w) + 1$ for $w \in \Sigma^*$, $a \in \Sigma$

Reversal:

$\varepsilon^R ::= \varepsilon$

$(wa)^R ::= \varepsilon a \bullet w^R$ for $w \in \Sigma^*$, $a \in \Sigma$

Claim: $\text{len}(x^R) = \text{len}(x)$ **for all** $x \in \Sigma^*$

Let P(x) **be** "$\text{len}(x^R) = \text{len}(x)$".

We will prove P(x) **for all** $x \in \Sigma^*$ **by structural induction.**

Base Case ($x = \varepsilon$): **Then,** $\text{len}(\varepsilon^R) = \text{len}(\varepsilon)$ **by def of string reverse.**

Claim: $\text{len}(x^R) = \text{len}(x)$ for all $x \in \Sigma^*$

Let P(x) **be** "$\text{len}(x^R) = \text{len}(x)$".

We will prove P(x) **for all** $x \in \Sigma^*$ **by structural induction.**

Base Case ($x = \varepsilon$): **Then,** $\text{len}(\varepsilon^R) = \text{len}(\varepsilon)$ **by def of string reverse.**

Inductive Hypothesis: Assume that P(w) **is true for some arbitrary** $w \in \Sigma^*$, **i.e.,** $\text{len}(w^R) = \text{len}(w)$.

Inductive Step: **Goal: Show that** $\text{len}((wa)^R) = \text{len}(wa)$ **for every** a

Length:
$\text{len}(\varepsilon) ::= 0$
$\text{len}(wa) ::= \text{len}(w) + 1$ for $w \in \Sigma^*$, $a \in \Sigma$

Reversal:
$\varepsilon^R ::= \varepsilon$
$(wa)^R ::= \varepsilon a \bullet w^R$ for $w \in \Sigma^*$, $a \in \Sigma$

Claim: $\mathrm{len}(x^R) = \mathrm{len}(x)$ for all $x \in \Sigma^*$

Let $P(x)$ **be** "$\mathrm{len}(x^R) = \mathrm{len}(x)$".

We will prove $P(x)$ **for all** $x \in \Sigma^*$ **by structural induction.**

Base Case ($x = \varepsilon$): **Then,** $\mathrm{len}(\varepsilon^R) = \mathrm{len}(\varepsilon)$ **by def of string reverse.**

Inductive Hypothesis: Assume that $P(w)$ **is true for some arbitrary** $w \in \Sigma^*$, **i.e.,** $\mathrm{len}(w^R) = \mathrm{len}(w)$.

Inductive Step: **Goal: Show that** $\mathrm{len}((wa)^R) = \mathrm{len}(wa)$ **for every** a

Let $a \in \Sigma$. **Then,**

$\mathrm{len}((wa)^R)$	$= \mathrm{len}(\varepsilon a \bullet w^R)$	**def of reverse**
	$= \mathrm{len}(\varepsilon a) + \mathrm{len}(w^R)$	**by previous result**
	$= \mathrm{len}(\varepsilon a) + \mathrm{len}(w)$	**IH**
	$= 1 + \mathrm{len}(w)$	**def of len (twice)**
	$= \mathrm{len}(wa)$	**def of len**

Therefore, $\mathrm{len}((wa)^R) = \mathrm{len}(wa)$, **so** $P(wa)$ **is true for every** $a \in \Sigma$.

So, we have shown $\mathrm{len}(x^R) = \mathrm{len}(x)$ **for all** $x \in \Sigma^*$ **by induction.**

More Theorems

Structural induction is the tool used to prove many more interesting theorems

- General associativity follows from our one rule
  - likewise for generalized De Morgan's laws

- Okay to substitute y for x everywhere in a modular equation when we know that $x \equiv_m y$

- More coming shortly…