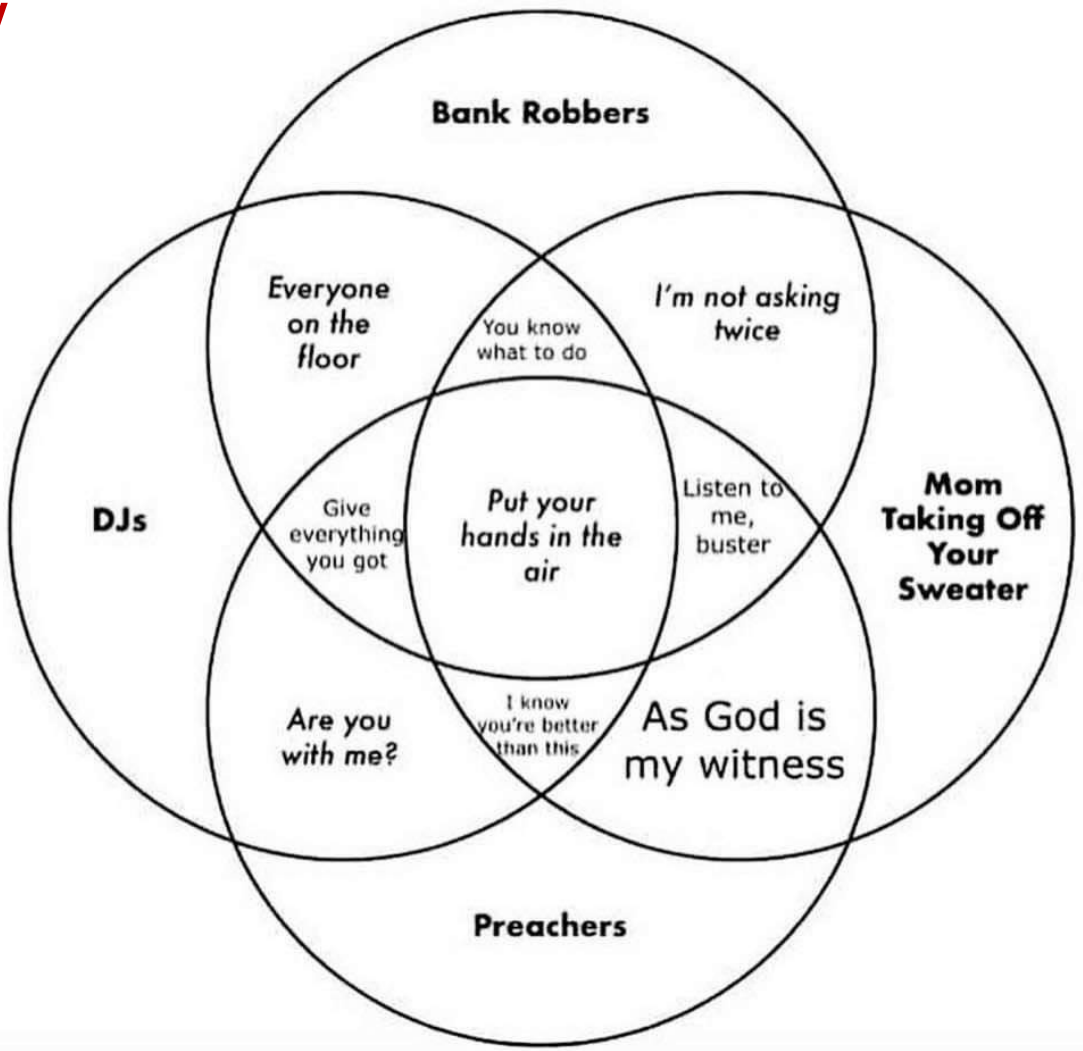
CSE 311: Foundations of Computing

Topic 4: 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 = {□, 1}
D = {{17}, 17}
E = {1, 2, 7, cat, dog, ∅, α}

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
$\varnothing$ = **{}** is the **empty set**; the *only* set with no elements

Sets can be elements of other sets

For example

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

$B = \{1,2\}$

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}
- ∅ 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

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

A = {1, 2}
B = {1, 2, 3}

$A \subseteq B$ is **true**
$B \subseteq A$ is **false**

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}

QUESTIONS

$A \subseteq B$?
$C \subseteq B$?
$\varnothing \subseteq A$?

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:

$$(\boldsymbol{A} = \boldsymbol{B}) \equiv (\boldsymbol{A} \subseteq \boldsymbol{B}) \wedge (\boldsymbol{B} \subseteq \boldsymbol{A})$$

$A \supseteq B$ 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 ∀: 3

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

   5.3. $(y \in A \rightarrow y \in B) \wedge (y \in B \rightarrow y \in A)$ — Intro ∧: 5.1, 5.2

   5.4. $y \in A \leftrightarrow y \in B$ — Biconditional: 5.3

5. $\forall x\,(x \in A \leftrightarrow x \in B)$ — Intro ∀
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. $\forall x\,(x \in A \rightarrow x \in B)$ **??**
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 $\forall$: 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
1.1.2. $P(x)$	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 $\forall$: 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).

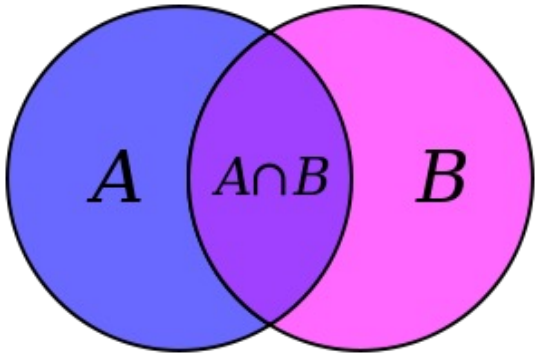
…

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

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

English *template* for a Subset Proof

Operations on Sets



Set Operations

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

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

$$A \setminus B := \{ x : (x \in A) \wedge (x \notin B) \}$$ **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 $\forall x\, (x \in (A \cup B)^C \leftrightarrow x \in A^C \cap B^C)$

Equivalently, prove $(A \cup B)^C \subseteq A^C \cap B^C$ **and** $A^C \cap B^C \subseteq (A \cup B)^C$

Recall: 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 of subset, that $A \subseteq B$.

De Morgan's Laws

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

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

Proof: Let x be an arbitrary object.

Suppose that $x \in (A \cup B)^C$. By the definition of …

By the definition of …, this means $x \in A^C \cap B^C$.

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

De Morgan's Laws

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

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

Proof: Let x be an arbitrary object.

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

By the definition of …, this means $x \in A^C \cap B^C$.

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

De Morgan's Laws

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

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

Proof: Let x be an arbitrary object.

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

…

By the definition of …, this means $x \in A^C \cap B^C$.

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

De Morgan's Laws

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

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

Proof: Let x be an arbitrary object.

Suppose that $x \in (A \cup B)^C$. 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, $x \in A^C$ and $x \in B^C$. By the definition of intersection, this means $x \in A^C \cap B^C$.

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

De Morgan's Laws

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

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

Proof: Let x be an arbitrary object.

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

…

So $\neg(x \in A)$ and $\neg(x \in B)$. Thus, $x \in A^C$ and $x \in B^C$ by the definition of complement. By the definition of intersection, this means $x \in A^C \cap B^C$.

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

De Morgan's Laws

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

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

Proof: Let x be an arbitrary object.

Suppose that $x \in (A \cup B)^C$. 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, $x \in A^C$ and $x \in B^C$ by the definition of complement. By the definition of intersection, this means $x \in A^C \cap B^C$.

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

De Morgan's Laws

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

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

Proof: Let x be an arbitrary object.

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 $\neg(x \in A) \wedge \neg(x \in B)$, which is equivalent to $\neg(x \in A \vee x \in B)$ 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.

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

Proofs About Set Equality

A lot of *repetitive* work to show $\rightarrow$ and $\leftarrow$.

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.

Suppose $x \in A^C \cap B^C$.
Then, by the definition of intersection, we have $x \in A^C$ and $x \in B^C$.
We then have $\neg(x \in A) \wedge \neg(x \in B)$ by the definition of complement.
which is equivalent to $\neg(x \in A \vee x \in B)$ 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 $\rightarrow$ and $\leftarrow$.

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.

Suppose $x \in A^C \cap B^C$.
Then, by the definition of intersection, we have $x \in A^C$ and $x \in B^C$.
We then have $\neg(x \in A) \wedge \neg(x \in B)$ by the definition of complement.
which is equivalent to $\neg(x \in A \vee x \in B)$ 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 $\rightarrow$ and $\leftarrow$.

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 Law

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:

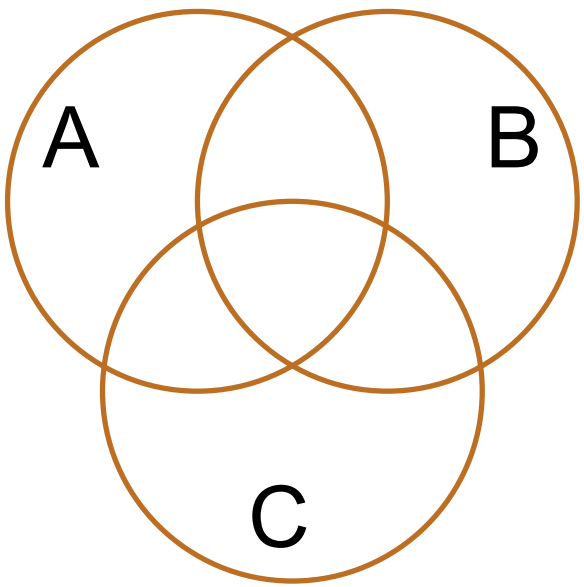
$x \in (A \cup B)^C$	$\equiv \neg(x \in A \cup B)$	Def of Comp
	$\equiv \neg(x \in A \vee x \in B)$	Def of Union
	$\equiv \neg(x \in A) \wedge \neg(x \in B)$	De Morgan
	$\equiv x \in A^C \wedge x \in B^C$	Def of Comp
	$\equiv x \in A^C \cap B^C$	Def of Intersection

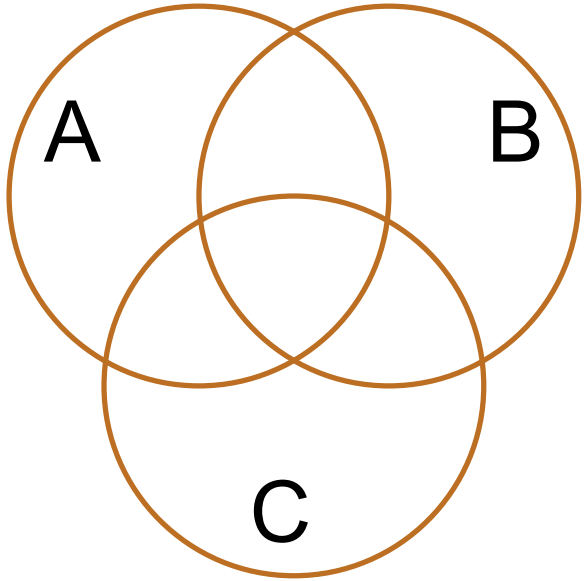
Calculation blocks are often easier to read like this rather than as English text

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

Distributive Laws

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





Distributive Law

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

Proof: Let x be an arbitrary object.

The stated biconditional holds since:

$x \in A \cap (B \cup C)$

$\equiv (x \in A) \wedge (x \in B \cup C)$ Def of Intersection

$\equiv (x \in A) \wedge ((x \in B) \vee (x \in C))$ Def of Union

$\equiv ((x \in A) \wedge (x \in B)) \vee ((x \in A) \wedge (x \in C))$ Distributive

$\equiv (x \in A \cap B) \vee (x \in A \cap C)$ Def of Intersection

$\equiv x \in (A \cap B) \cup (A \cap C)$ Def of Union

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

The Meta Theorem

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

Example: $\neg(A \vee B) \equiv \neg A \wedge \neg B$ **becomes**

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

Example: $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$ **becomes**

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

The Meta Theorem Proof Template

Meta-Theorem: Translate any Propositional Logic equivalence into "=" relationship between sets by replacing $\cup$ with $\vee$, $\cap$ with $\wedge$, and $\cdot^C$ with $\neg$.

"Proof": Let x be an arbitrary object.

The stated bi-condition holds since:

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

$\equiv$ apply Propositional Logic equivalence

$\equiv$ replace propositional logic with set ops

$\equiv x \in$ right side

Since x was arbitrary, we have shown, by definition, that 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)=?

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**

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

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

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**

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

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

Cartesian Product

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

- $\mathbb{R} \times \mathbb{R}$ 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)}.

Cartesian Product

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

- $\mathbb{R} \times \mathbb{R}$ 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)}.

What is $A \times \emptyset$?

Cartesian Product

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

- $\mathbb{R} \times \mathbb{R}$ 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)}.

$$\begin{aligned} A \times \emptyset &= \{x : \exists a\, \exists b\, (a \in A \wedge b \in \emptyset \wedge x = (a, b))\} \\ &= \{x : \exists a\, \exists b\, (a \in A \wedge \mathbf{F} \wedge x = (a, b))\} \\ &= \{x : \mathbf{F}\} = \emptyset \end{aligned}$$

More Set Builder Notation

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

- **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 exception to the rule that
    unbound variables are implicitly ∀-quantified

Another Expression on the Left Side

- The set of all squares can be defined as

$$S := \{ n^2 : n \in \mathbb{Z} \}$$

- Formally, this is the set of all x such that $\mathrm{x} = n^2$ for some $n \in \mathbb{Z}$.

- Informally, go through all $n \in \mathbb{Z}$, and for each of them, add n^2 to the set.
  - can think of this as a "map" operation

More Set Builder Notation

$$S := \{\, x \in U : P(x) \,\}$$ "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.**

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

- **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.

Since one of these must hold, we have proven F in general.

Uh... that's not good.

Recall: Formal Proofs

- In principle, formal proofs are the standard for what it means to be "proven" in mathematics
  - almost all math (and theory CS) done in Predicate Logic

- But they can be tedious and impractical
  - e.g., applications of commutativity and associativity
  - Russell & Whitehead's formal proof that 1+1 = 2 is *several hundred pages* long

    we allow ourselves to cite "Arithmetic", "Algebra", etc.

Domain-Restriction to Sets

Often want to prove facts about all elements of a set

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

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 set-restricted variables improves *clarity*

Sets of Numbers

- Define some familiar sets of numbers

$$\mathbb{E} = \{n \in \mathbb{Z} \mid \exists k\ (n = 2k)\}$$
$$\mathbb{O} = \{n \in \mathbb{Z} \mid \exists k\ (n = 2k + 1)\}$$

  - previously, we defined these as predicates

Recall: Even and Odd

Prove "The square of every even integer is even."

Formally, prove $\forall x\,(\text{Even}(x) \rightarrow \text{Even}(x^2))$

Proof: Let **a** be an arbitrary integer.

Suppose **a** is even. Then, by definition, $\mathbf{a} = 2\mathbf{b}$ for some integer b. Squaring both sides, we get $\mathbf{a^2} = 4\mathbf{b^2} = 2(2\mathbf{b^2})$. So $\mathbf{a^2}$ is, by definition, is even.

Since **a** was arbitrary, we have shown that the square of every even number is even. ■

Even and Odd As Sets

Prove "The square of every even integer is even."

Formally, prove $\forall x \in \mathbb{E}\ (x^2 \in \mathbb{E})$

Proof: Let **a** be an arbitrary **even** integer.

~~Suppose **a** is even.~~ Then, by definition, $\mathbf{a} = 2\mathbf{b}$ for some integer b. Squaring both sides, we get $\mathbf{a^2} = 4\mathbf{b^2} = 2(2\mathbf{b^2})$. So $\mathbf{a^2}$ is, by definition, is even.

Since **a** was arbitrary, we have shown that the square of every even number is even. ■

The structure of the proof *follows* the structure of the claim.

Recall: Even and Odd

Prove "The sum of any two odd numbers is even."

Formally, prove $\forall x \, \forall y \, ((\text{Odd}(x) \wedge \text{Odd}(y)) \rightarrow \text{Even}(x+y))$

Proof: Let x and y be arbitrary integers.

Suppose that both are odd. Then, we have x = 2a+1 for some integer a and y = 2b+1 for some integer b. Their sum is x+y = (2a+1) + (2b+1) = 2a+2b+2 = 2(a+b+1), so x+y is, by definition, even.

Since x and y were arbitrary, the sum of any two odd integers is even. ■

Recall: Even and Odd

Prove "The sum of any two odd numbers is even."

Formally, prove $\forall x \in \mathbb{O}, \forall y \in \mathbb{O}\ (x + y \in \mathbb{E})$

Proof: Let x and y be arbitrary **odd** integers.

~~Suppose that both are odd.~~ Then, we have x = 2a+1 for some integer a and y = 2b+1 for some integer b. Their sum is x+y = (2a+1) + (2b+1) = 2a+2b+2 = 2(a+b+1), so x+y is, by definition, even.

Since x and y were arbitrary, the sum of any two odd integers is even. ■

Another Odd One

"The square of the sum of any even and odd is congruent to 1 mod 4"

Formally, prove $\forall x \in \mathbb{E}, \forall y \in \mathbb{O}\ ((x + y)^2 \equiv_4 1)$

Proof:

Let x be an arbitrary **even** and y an arbitrary **odd**.

Since x is even and y is odd, we have x = 2j for some integer j, and y = 2k + 1 for some integer k. We can now see that

$$\begin{aligned}(x+y)^2 &= (2j + 2k+1)^2 \\ &= (2(j+k) + 1)^2 \\ &= 4(j+k)^2 + 4(j+k) + 1\end{aligned}$$

This shows that $4 \mid (x+y)^2 - 1$ by definition of divides, which means that $(x+y)^2 \equiv_4 1$ by definition of congruent.

Since x and y were arbitrary, we have proven the claim. ■

Recall: 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** $n \geq 0$**.**

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 $x+2 \in S$

In comparison to earlier definitions:

- $\mathbb{N} := \{x \in \mathbb{Z} \mid x \geq 0\}$
- $\mathbb{E} := \{x \in \mathbb{N} \mid \exists k\ (x = 2k)\}$

these definitions are constructive.

Recursive Definitions 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

Recall: 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 ∈ S
Recursive: If x ∈ S, then x+2 ∈ 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.

Formally, $\forall x \in S\ (3 \mid x)$

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: P(x+y), i.e., 3 | 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:

   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:

   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 $x + 15 \in T$

- Now, two base cases and two *recursive* cases

Claim: Every element of T is also in S.

Formally, $\forall x \in T\ (x \in S)$

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

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

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

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

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

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

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

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

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

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

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

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

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:

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

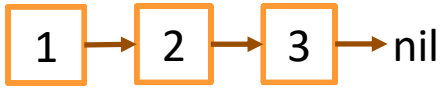
Basis: $6 \in S;\ 15 \in S$	**Basis:** $6 \in T;\ 15 \in T$
Recursive: if $x, y \in S$, then $x + y \in S$	**Recursive:** if $x \in T$, then $x + 6 \in T$ and $x + 15 \in T$

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:

   Since P(x) holds, we have x ∈ S. From the recursive step of S, since 6 ∈ S, we can see that x + 6 ∈ S, so P(x+6) is true, and since 15 ∈ S, we can see that x + 15 ∈ S, so P(x+15) is true.
5. Therefore P(x) for all x ∈ T by induction.

Lists of Integers



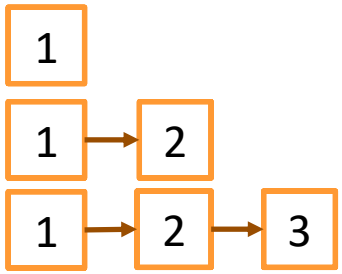
- **Basis:** $\text{nil} \in \textbf{List}$
- **Recursive step:**

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

  then $a :: L \in \textbf{List}$

Examples:

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



Functions on Lists

Length:

$$\text{len}(\text{nil}) := 0$$
$$\text{len}(a :: L) := \text{len}(L) + 1 \qquad \textbf{for any } L \in \textbf{List and } a \in \mathbb{Z}$$

Concatenation:

$$\text{concat}(\text{nil}, R) := R \qquad \textbf{for any } R \in \textbf{List}$$
$$\text{concat}(a :: L, R) := a :: \text{concat}(L, R) \qquad \textbf{for any } L, R \in \textbf{List and any } a \in \mathbb{Z}$$

Structural Induction

Basis: nil ∈ List

Recursive step:
if L ∈ List **and** a ∈ ℤ,
then a :: L ∈ 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: $\text{concat}(S, \text{nil}) = S$ **for all** $S \in$ **List**

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

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

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

Concatenation:

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

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

Let $P(S)$ **be** "$\text{concat}(S, \text{nil}) = S$" .
We will prove $P(S)$ **for all** $S \in$ **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: $\text{concat}(S, \text{nil}) = S$ **for all** $S \in$ **List**

Let $P(S)$ **be** "$\text{concat}(S, \text{nil}) = S$" .
We will prove $P(S)$ **for all** $S \in$ **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** $P(\text{nil})$.

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

Inductive Step:

Concatenation:

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

Goal: For any $a \in \mathbb{Z}$, **show** $P(a :: L)$, i.e., $\text{concat}(a :: L, \text{nil}) = a :: L$

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

Let $P(S)$ **be** "$\text{concat}(S, \text{nil}) = S$" .
We will prove $P(S)$ **for all** $S \in$ **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** $P(\text{nil})$.

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

Inductive Step:

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 $L \in$ **List**.

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

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

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

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

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

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

Base Case (nil): **Let** $R \in$ **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,** $P(\text{nil})$ **holds.**

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

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

Base Case (nil): **Let** $R \in$ **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** $P(\text{nil})$.

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

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

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

Base Case (nil): **Let** $R \in$ **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** $P(\text{nil})$.

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

Inductive Step:

Goal: For any $a \in \mathbb{Z}$, **and** $R \in$ **List, show** $P(a :: L)$, **i.e.,**
$\text{len}(\text{concat}(a :: L, R)) = \text{len}(a :: L) + \text{len}(R)$

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

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

Base Case (nil): **Let** $R \in$ **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** $P(\text{nil})$.

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

Inductive Step: Let $a \in \mathbb{Z}$ **and** $R \in$ **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: $\text{len}(\text{concat}(S, R)) = \text{len}(S) + \text{len}(R)$ **for all** $S, R \in$ **List**

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

Base Case (nil): **Let** $R \in$ **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** $P(\text{nil})$**.**

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

Inductive Step: Let $a \in \mathbb{Z}$ **and** $R \in$ **List be arbitrary. Then, we have**

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

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

By induction, we have shown the claim holds for all $S \in$ **List**.

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

We can write this equivalently as...

$$\forall R \in \text{List}, \forall S \in \text{List}, \forall T \in \text{List}\ (\text{concat}(\text{concat}(R,S),T) = \text{concat}(R,\text{concat}(S,T)))$$

$$\equiv \forall S \in \text{List}, \forall T \in \text{List}, \forall R \in \text{List}\ (\text{concat}(\text{concat}(R,S),T) = \text{concat}(R,\text{concat}(S,T)))$$

$$\equiv \forall T \in \text{List}, \forall S \in \text{List}, \forall R \in \text{List}\ (\text{concat}(\text{concat}(R,S),T) = \text{concat}(R,\text{concat}(S,T)))$$

... or any other order we want.

Induction proves only the outer-most "∀", but we can *pick* which variable we want to do induction on.

The body of the outer-most "∀" still includes more "∀"s...

$$\forall R \in \text{List}\ \underbrace{\left(\forall S \in \text{List}, \forall T \in \text{List}\ \left(\text{concat}(\text{concat}(R,S),T) = \text{concat}\left(R,\text{concat}(S,T)\right)\right)\right)}_{P(R)}$$

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

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

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

Base Case (nil):

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

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

Base Case (nil): **Let** S, T **be arbitrary lists.**

Concatenation:

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

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

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

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

concat(concat(nil, S), T)
= concat(S, T) **def of** concat
= concat(nil, concat(S), T) **def of** concat

which is P(nil)**.**

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

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

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

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

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

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

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

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

Inductive Step:

Goal: Show that P(a :: L) **is true for any** a ∈ ℤ**, i.e.,**
concat(concat(a :: L, S), T) = concat(a :: L, concat(S, T)) **for any** S, T

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

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

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

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

Inductive Step: Let a ∈ ℤ **and** S, T ∈ **List be arbitrary.**

Goal: Show that P(a :: L) **is true for any** a ∈ ℤ**, i.e.,**
concat(concat(a :: L, S), T) = concat(a :: L, concat(S, T)) **for any** S, T

Concatenation:

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

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

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

Base Case (nil): **Let** S, T **be arbitrary lists. Then, we can see that** concat(nil, concat(S, T)) = concat(S, T) = concat(concat(nil, S), T)**, 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(concat(L, S), T) = concat(L, concat(S, T)) **for all** S, T ∈ **List**.

Inductive Step: Let a ∈ ℤ **and** S, T ∈ **List be arbitrary. Then, we have**

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

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

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

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

Base Case (nil): **Let** S, T **be arbitrary lists. Then, we can see that** concat(nil, concat(S, T)) = concat(S, T) = concat(concat(nil, S), T)**, 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(concat(L, S), T) = concat(L, concat(S, T)) **for all** S, T ∈ **List**.

Inductive Step: Let a ∈ ℤ **and** S, T ∈ **List be arbitrary. Then, we have**

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

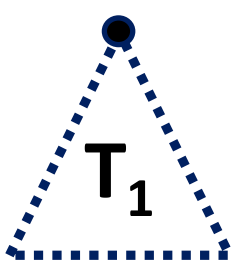
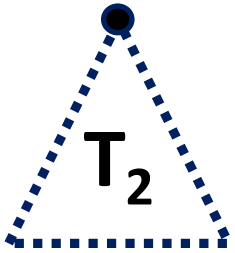
Since L **was arbitrary, we have shown** P(a :: L).
By induction, we have shown the claim holds for all R ∈ **List**.

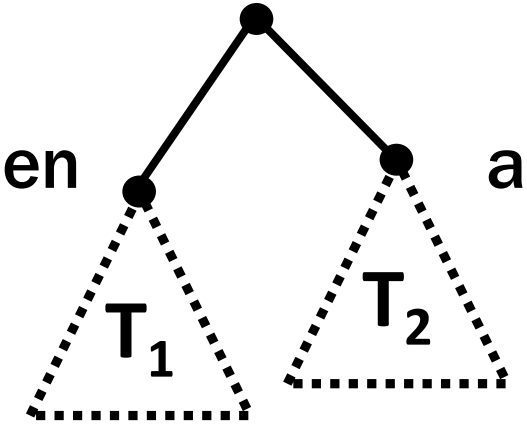
Rooted Binary Trees

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

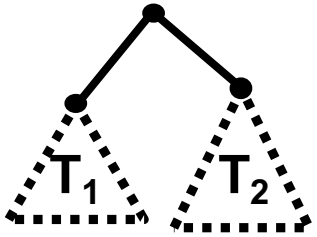
Rooted Binary Trees

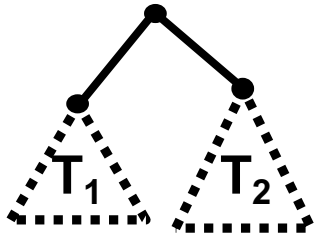
- **Basis:** • is a rooted binary tree
- **Recursive step:**

If  and  are rooted binary trees,

then  also is a rooted binary tree.

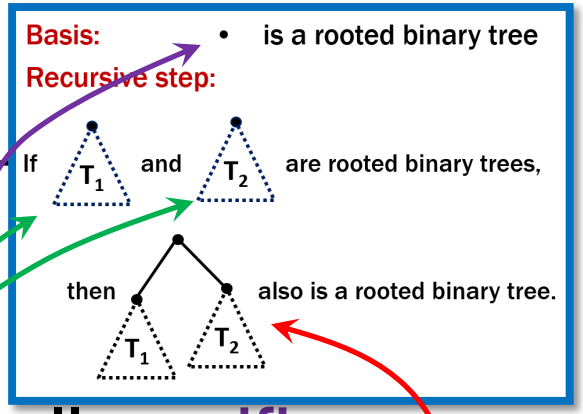
Defining Functions on Rooted Binary Trees

- size(•) := 1
- size 

:= 1 + size(T_1) + size(T_2)
- height(•) := 0
- height 

:= 1 + max{height(T_1), height(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 $\forall\, x \in S, P(x)$

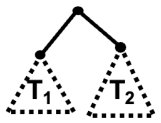
Claim: For every rooted binary tree **T**, $\text{size}(\mathbf{T}) \leq 2^{\text{height}(\mathbf{T})+1} - 1$

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

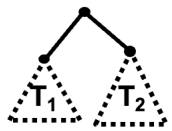
---

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

size(•) ::= 1

size () ::= 1 + size(T_1) + size(T_2)

height(•) ::= 0

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

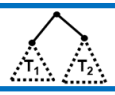
Claim: **For every rooted binary tree T,** $\text{size}(T) \leq 2^{\text{height}(T)+1} - 1$

---

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

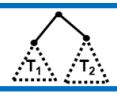
Claim: For every rooted binary tree T, $\text{size}(T) \leq 2^{\text{height}(T)+1} - 1$

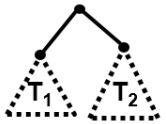
---

1. Let P(T) be "$\text{size}(T) \leq 2^{\text{height}(T)+1}-1$". We prove P(T) for all rooted binary trees T by structural induction.
2. Base Case: $\text{size}(\bullet)=1$, $\text{height}(\bullet)=0$, 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., $\text{size}(T_k) \leq 2^{\text{height}(T_k)+1} - 1$ for k=1,2
4. Inductive Step: Goal: Prove P().

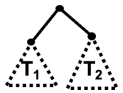
Claim: For every rooted binary tree T, $\text{size}(T) \le 2^{\text{height}(T)+1} - 1$

---

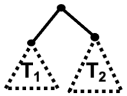
1. Let P(T) be "$\text{size}(T) \le 2^{\text{height}(T)+1}-1$". We prove P(T) for all rooted binary trees T by structural induction.
2. Base Case: $\text{size}(\bullet)=1$, $\text{height}(\bullet)=0$, and $2^{0+1}-1=2^1-1=1$ so P(•) is true.
3. Inductive Hypothesis: Suppose that P(T_1) and P(T_2) are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

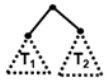
size()

size(•) ::= 1

size () ::= 1 + size(T_1) + size(T_2)

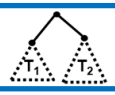
height(•) ::= 0

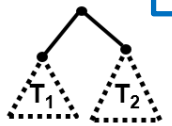
height () ::= 1 + max{height(T_1), height(T_2)}

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

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

---

1. Let P(T) be "$\text{size}(T) \le 2^{\text{height}(T)+1}-1$". We prove P(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(•) 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., $\text{size}(T_k) \le 2^{\text{height}(T_k)+1} - 1$ for k=1,2
4. Inductive Step: **Goal: Prove P().**

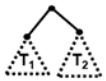
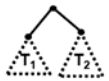
By def, size() $= 1+\text{size}(T_1)+\text{size}(T_2)$

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

by IH for T_1 and T_2

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

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

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

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 $wa \in \Sigma^*$

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 $a \neq c$

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

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.**

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,**

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

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

x • ε := x
x • wa := (x • w)a

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: 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

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

Let a ∈ Σ **and** x ∈ Σ* **be arbitrary. Then,**

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

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

Let a ∈ Σ **and** x ∈ Σ* **be arbitrary. Then,**

len(x•wa)	= len((x•w)a)	**def of •**
	= len(x•w)+1	**def of len**
	= len(x)+len(w)+1	**by IH**
	= len(x)+len(wa)	**def of len**

Since x **was arbitrary, we have shown** P(wa).

x • ε := x
x • wa := (x • w)a

len(ε) := 0
len(wa) := len(w) + 1

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 $x \in$

We prove $P(y)$ **for all** $y \in \Sigma^*$ **by structural indu**

Does this look familiar?

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:

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

$$\begin{aligned} \text{len}(x \bullet wa) &= \text{len}((x \bullet w)a) && \textbf{def of } \bullet \\ &= \text{len}(x \bullet w)+1 && \textbf{def of len} \\ &= \text{len}(x)+\text{len}(w)+1 && \textbf{by IH} \\ &= \text{len}(x)+\text{len}(wa) && \textbf{def of len} \end{aligned}$$

Since x **was arbitrary, we have shown** $P(wa)$.

By induction, we have shown the claim holds for all $y \in \Sigma^*$.

Recall: $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **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** $a \in \mathbb{Z}$ **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** $L \in \text{List}$, **i.e.,** $\text{len}(\text{concat}(L, R)) = \text{len}(L) + \text{len}(R)$ **for all** $R \in$ **List**.

Inductive Step:

Let $a \in \mathbb{Z}$ **and** $R \in$ **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 $L \in$ **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.**

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

Let $a \in \Sigma$ **be arbitrary. Then, we can calculate**

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.**

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: Let $a \in \Sigma$ **be arbitrary. Then, we can calculate**

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

Thus, $P(wa)$ **is true for every** $a \in \Sigma$.

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.**

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: Let $a \in \Sigma$ **be arbitrary. Then, we can calculate**

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

Thus, $P(wa)$ **is true for every** $a \in \Sigma$.

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