Predicates

A predicate is a function whose coDomain is $\{\text{True}, \text{False}\}$

Predicates

A predicate is a function whose coDomain is $\{\text{True}, \text{False}\}$
Could be $B \times B \rightarrow B$, such as with propositional operators

Predicates

A predicate is a function whose coDomain is $\{\text{True}, \text{False}\}$
Could be $B \times B \to B$, such as with propositional operators
Could be $R \times R \to B$, such as $>$

Predicates

A predicate is a function whose coDomain is $\{\text{True}, \text{False}\}$
Could be $B \times B \rightarrow B$, such as with propositional operators
Could be $R \times R \rightarrow B$, such as $>$
Nouns: Could be $Person \times Person \rightarrow B$, such as the "Mother" predicate

Predicates

A predicate is a function whose coDomain is $\{\text{True}, \text{False}\}$
Could be $B \times B \rightarrow B$, such as with propositional operators
Could be $R \times R \rightarrow B$, such as $>$
Nouns: Could be $Person \times Person \rightarrow B$, such as the "Mother" predicate

Other examples:

Predicates

A predicate is a function whose coDomain is $\{\text{True}, \text{False}\}$
Could be $B \times B \to B$, such as with propositional operators
Could be $R \times R \to B$, such as $>$
Nouns: Could be $Person \times Person \to B$, such as the "Mother" predicate

Other examples:
Adjectives: Brown(desk)

Predicates

A predicate is a function whose coDomain is $\{\text{True}, \text{False}\}$
Could be $B \times B \rightarrow B$, such as with propositional operators
Could be $R \times R \rightarrow B$, such as $>$
Nouns: Could be $Person \times Person \rightarrow B$, such as the "Mother" predicate

Other examples:
Adjectives: Brown(desk)
Color (desk1, brown)

Predicates

A predicate is a function whose coDomain is $\{\text{True}, \text{False}\}$
Could be $B \times B \to B$, such as with propositional operators
Could be $R \times R \to B$, such as $>$
Nouns: Could be $Person \times Person \to B$, such as the "Mother" predicate

Other examples:
Adjectives: Brown(desk)
Color (desk1, brown)
Big(Ant3, ants)
Big(Elephant1, elephants)

Predicates

A predicate is a function whose coDomain is $\{\text{True}, \text{False}\}$
Could be $B \times B \rightarrow B$, such as with propositional operators
Could be $R \times R \rightarrow B$, such as $>$
Nouns: Could be $Person \times Person \rightarrow B$, such as the "Mother" predicate

Other examples:
Adjectives: Brown(desk)
Color (desk1, brown)
Big(Ant3, ants)
Big(Elephant1, elephants)
Size(Elephant1, elephants, big)

Predicates

A predicate is a function whose coDomain is $\{\text{True}, \text{False}\}$
Could be $B \times B \rightarrow B$, such as with propositional operators
Could be $R \times R \rightarrow B$, such as $>$
Nouns: Could be $Person \times Person \rightarrow B$, such as the "Mother" predicate

Other examples:
Adjectives: Brown(desk)
Color (desk1, brown)
Big(Ant3, ants)
Big(Elephant1, elephants)
Size(Elephant1, elephants, big)
Size(Jumbo, Elephant, Small) $\wedge$ Size(Jumbo, Mammal, Big)

Predicates

A predicate is a function whose coDomain is $\{\text{True}, \text{False}\}$
Could be $B \times B \rightarrow B$, such as with propositional operators
Could be $R \times R \rightarrow B$, such as $>$
Nouns: Could be $Person \times Person \rightarrow B$, such as the "Mother" predicate

Other examples:
Adjectives: Brown(desk)
Color (desk1, brown)
Big(Ant3, ants)
Big(Elephant1, elephants)
Size(Elephant1, elephants, big)
Size(Jumbo, Elephant, Small) $\wedge$ Size(Jumbo, Mammal, Big)

Verbs:
Borders(Bolivia, Peru)

Predicates

A predicate is a function whose coDomain is $\{\text{True}, \text{False}\}$
Could be $B \times B \rightarrow B$, such as with propositional operators
Could be $R \times R \rightarrow B$, such as $>$
Nouns: Could be $Person \times Person \rightarrow B$, such as the "Mother" predicate

Other examples:
Adjectives: Brown(desk)
Color (desk1, brown)
Big(Ant3, ants)
Big(Elephant1, elephants)
Size(Elephant1, elephants, big)
Size(Jumbo, Elephant, Small) $\wedge$ Size(Jumbo, Mammal, Big)

Verbs:
Borders(Bolivia, Peru)
Plays(Basketball, James)
Plays(Soccer, Messi)

Quantifiers

Universal quantifier: $\forall$ (forall or every).

Quantifiers

Universal quantifier: $\forall$ (forall or every).
Every element of some set satisfies some predicate:

Quantifiers

Universal quantifier: $\forall$ (forall or every).
Every element of some set satisfies some predicate:
Specify the set and the predicate:
$\forall x \in \mathcal{I} : (x^2 \geq 0)$

Quantifiers

Universal quantifier: $\forall$ (forall or every).
Every element of some set satisfies some predicate:
Specify the set and the predicate:
$\forall x \in \mathcal{I} : (x^2 \geq 0)$

Existential quantifier: $\exists$ (for some, there exists ... such that).

Quantifiers

Universal quantifier: $\forall$ (forall or every).
Every element of some set satisfies some predicate:
Specify the set and the predicate:
$\forall x \in \mathcal{I} : (x^2 \geq 0)$

Existential quantifier: $\exists$ (for some, there exists ... such that).
Specify the set and the predicate:
$\exists x \in \mathcal{I} : (x > 3) \wedge (x < 7)$

Quantifiers

Universal quantifier: $\forall$ (forall or every).
Every element of some set satisfies some predicate:
Specify the set and the predicate:
$\forall x \in \mathcal{I} : (x^2 \geq 0)$

Existential quantifier: $\exists$ (for some, there exists ... such that).
Specify the set and the predicate:
$\exists x \in \mathcal{I} : (x > 3) \wedge (x < 7)$

$\mathcal{U} = \{x_1, x_2, x_3\}$
$\forall x \in \mathcal{U} : p(x) \equiv (p(x_1) \wedge p(x_2) \wedge p(x_3))$

Quantifiers

Universal quantifier: $\forall$ (forall or every).
Every element of some set satisfies some predicate:
Specify the set and the predicate:
$\forall x \in \mathcal{I} : (x^2 \geq 0)$

Existential quantifier: $\exists$ (for some, there exists ... such that).
Specify the set and the predicate:
$\exists x \in \mathcal{I} : (x > 3) \land (x < 7)$

$\mathcal{U} = \{x_1, x_2, x_3\}$
$\forall x \in \mathcal{U} : p(x) \equiv (p(x_1) \land p(x_2) \land p(x_3))$

$\neg(p(x_1) \land p(x_2) \land p(x_3)) \equiv \neg(\forall x \in \mathcal{U} : p(x))$

Quantifiers

Universal quantifier: $\forall$ (forall or every).
Every element of some set satisfies some predicate:
Specify the set and the predicate:
$\forall x \in \mathcal{I} : (x^2 \geq 0)$

Existential quantifier: $\exists$ (for some, there exists ... such that).
Specify the set and the predicate:
$\exists x \in \mathcal{I} : (x > 3) \wedge (x < 7)$

$\mathcal{U} = \{x_1, x_2, x_3\}$
$\forall x \in \mathcal{U} : p(x) \equiv (p(x_1) \wedge p(x_2) \wedge p(x_3))$

$$\begin{aligned} \neg(p(x_1) \wedge p(x_2) \wedge p(x_3)) &\equiv \neg(\forall x \in \mathcal{U} : p(x)) \\ &\equiv \neg p(x_1) \vee \neg p(x_2) \vee \neg p(x_3) \end{aligned}$$

Quantifiers

Universal quantifier: $\forall$ (forall or every).
Every element of some set satisfies some predicate:
Specify the set and the predicate:
$\forall x \in \mathcal{I} : (x^2 \geq 0)$

Existential quantifier: $\exists$ (for some, there exists ... such that).
Specify the set and the predicate:
$\exists x \in \mathcal{I} : (x > 3) \wedge (x < 7)$

$\mathcal{U} = \{x_1, x_2, x_3\}$
$\forall x \in \mathcal{U} : p(x) \equiv (p(x_1) \wedge p(x_2) \wedge p(x_3))$

$$\begin{aligned}
\neg(p(x_1) \wedge p(x_2) \wedge p(x_3)) &\equiv \neg(\forall x \in \mathcal{U} : p(x)) \\
&\equiv \neg p(x_1) \vee \neg p(x_2) \vee \neg p(x_3) \\
&\equiv \exists x \in \mathcal{U} : \neg p(x)
\end{aligned}$$

Quantifiers

Universal quantifier: $\forall$ (forall or every).
Every element of some set satisfies some predicate:
Specify the set and the predicate:
$\forall x \in \mathcal{I} : (x^2 \geq 0)$

Existential quantifier: $\exists$ (for some, there exists ... such that).
Specify the set and the predicate:
$\exists x \in \mathcal{I} : (x > 3) \wedge (x < 7)$

$\mathcal{U} = \{x_1, x_2, x_3\}$
$\forall x \in \mathcal{U} : p(x) \equiv (p(x_1) \wedge p(x_2) \wedge p(x_3))$

$$\begin{aligned} \neg(p(x_1) \wedge p(x_2) \wedge p(x_3)) &\equiv \neg(\forall x \in \mathcal{U} : p(x)) \\ &\equiv \neg p(x_1) \vee \neg p(x_2) \vee \neg p(x_3) \\ &\equiv \exists x \in \mathcal{U} : \neg p(x) \end{aligned}$$

Placement of the $\neg$ operator is very important!

$\neg \exists x \in P : IsCarOwner(x)$
$\exists x \in P : \neg IsCarOwner(x)$
$\neg \exists x \in P : \neg IsCarOwner(x)$

Quantifiers

Universal quantifier: $\forall$ (forall or every).
Every element of some set satisfies some predicate:
Specify the set and the predicate:
$\forall x \in \mathcal{I} : (x^2 \geq 0)$

Existential quantifier: $\exists$ (for some, there exists ... such that).
Specify the set and the predicate:
$\exists x \in \mathcal{I} : (x > 3) \wedge (x < 7)$

$\mathcal{U} = \{x_1, x_2, x_3\}$
$\forall x \in \mathcal{U} : p(x) \equiv (p(x_1) \wedge p(x_2) \wedge p(x_3))$

$$\begin{aligned} \neg(p(x_1) \wedge p(x_2) \wedge p(x_3)) &\equiv \neg(\forall x \in \mathcal{U} : p(x)) \\ &\equiv \neg p(x_1) \vee \neg p(x_2) \vee \neg p(x_3) \\ &\equiv \exists x \in \mathcal{U} : \neg p(x) \end{aligned}$$

Placement of the $\neg$ operator is very important!
$\neg \exists x \in P : \mathit{IsCarOwner}(x)$ It is not the case that someone has a car.
$\exists x \in P : \neg \mathit{IsCarOwner}(x)$
$\neg \exists x \in P : \neg \mathit{IsCarOwner}(x)$

Quantifiers

Universal quantifier: $\forall$ (forall or every).
Every element of some set satisfies some predicate:
Specify the set and the predicate:
$\forall x \in \mathcal{I} : (x^2 \geq 0)$

Existential quantifier: $\exists$ (for some, there exists ... such that).
Specify the set and the predicate:
$\exists x \in \mathcal{I} : (x > 3) \wedge (x < 7)$

$\mathcal{U} = \{x_1, x_2, x_3\}$
$\forall x \in \mathcal{U} : p(x) \equiv (p(x_1) \wedge p(x_2) \wedge p(x_3))$

$$\begin{aligned}\neg(p(x_1) \wedge p(x_2) \wedge p(x_3)) &\equiv \neg(\forall x \in \mathcal{U} : p(x)) \\ &\equiv \neg p(x_1) \vee \neg p(x_2) \vee \neg p(x_3) \\ &\equiv \exists x \in \mathcal{U} : \neg p(x)\end{aligned}$$

Placement of the $\neg$ operator is very important!

$\neg \exists x \in P : IsCarOwner(x)$	It is not the case that someone has a car.
$\exists x \in P : \neg IsCarOwner(x)$	Someone does not have a car.
$\neg \exists x \in P : \neg IsCarOwner(x)$	

Quantifiers

Universal quantifier: $\forall$ (forall or every).
Every element of some set satisfies some predicate:
Specify the set and the predicate:
$\forall x \in \mathcal{I} : (x^2 \geq 0)$

Existential quantifier: $\exists$ (for some, there exists ... such that).
Specify the set and the predicate:
$\exists x \in \mathcal{I} : (x > 3) \wedge (x < 7)$

$\mathcal{U} = \{x_1, x_2, x_3\}$
$\forall x \in \mathcal{U} : p(x) \equiv (p(x_1) \wedge p(x_2) \wedge p(x_3))$

$$\begin{aligned} \neg(p(x_1) \wedge p(x_2) \wedge p(x_3)) &\equiv \neg(\forall x \in \mathcal{U} : p(x)) \\ &\equiv \neg p(x_1) \vee \neg p(x_2) \vee \neg p(x_3) \\ &\equiv \exists x \in \mathcal{U} : \neg p(x) \end{aligned}$$

Placement of the $\neg$ operator is very important!

$\neg \exists x \in P : \mathit{IsCarOwner}(x)$	It is not the case that someone has a car.
$\exists x \in P : \neg \mathit{IsCarOwner}(x)$	Someone does not have a car.
$\neg \exists x \in P : \neg \mathit{IsCarOwner}(x)$	No one is without a car.

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : BioMother(x, y) \leftrightarrow (Birthed(x, y) \wedge (FemaleSex(x))$

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : BioMother(x, y) \leftrightarrow (Birthed(x, y) \wedge (FemaleSex(x))$
$\forall x : \forall y : \forall z : Offspring(x, y, z) \leftrightarrow (BioMother(y, x) \wedge BioFather(z, x))$

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : BioMother(x, y) \leftrightarrow (Birthed(x, y) \wedge (FemaleSex(x))$
$\forall x : \forall y : \forall z : Offspring(x, y, z) \leftrightarrow (BioMother(y, x) \wedge BioFather(z, x))$

The use of $\leftrightarrow$ here allows us to define the predicate, using other predicates.

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : \mathit{BioMother}(x, y) \leftrightarrow (\mathit{Birthed}(x, y) \wedge (\mathit{FemaleSex}(x))$
$\forall x : \forall y : \forall z : \mathit{Offspring}(x, y, z) \leftrightarrow (\mathit{BioMother}(y, x) \wedge \mathit{BioFather}(z, x))$

The use of $\leftrightarrow$ here allows us to define the predicate, using other predicates.
In particular, note that the right hand side allows us to determine the value of the Offspring predicate *on every single input*.

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : BioMother(x, y) \leftrightarrow (Birthed(x, y) \wedge (FemaleSex(x))$
$\forall x : \forall y : \forall z : Offspring(x, y, z) \leftrightarrow (BioMother(y, x) \wedge BioFather(z, x))$

The use of $\leftrightarrow$ here allows us to define the predicate, using other predicates.
In particular, note that the right hand side allows us to determine the value of the Offspring predicate *on every single input*.

Some examples:
Every integer has a square that is greater or equal to 0:
$\forall x \in \mathcal{I} : (x^2 > 0)$

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : BioMother(x, y) \leftrightarrow (Birthed(x, y) \wedge (FemaleSex(x))$
$\forall x : \forall y : \forall z : Offspring(x, y, z) \leftrightarrow (BioMother(y, x) \wedge BioFather(z, x))$

The use of $\leftrightarrow$ here allows us to define the predicate, using other predicates.
In particular, note that the right hand side allows us to determine the value of the Offspring predicate *on every single input*.

Some examples:
Every integer has a square that is greater or equal to 0:
$\forall x \in \mathcal{I} : (x^2 > 0)$

Every integer has a square that is a positive integer:
$\forall x \in \mathcal{I} : \exists y \in \mathcal{N} : y = x^2$

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : BioMother(x, y) \leftrightarrow (Birthed(x, y) \wedge (FemaleSex(x))$
$\forall x : \forall y : \forall z : Offspring(x, y, z) \leftrightarrow (BioMother(y, x) \wedge BioFather(z, x))$

The use of $\leftrightarrow$ here allows us to define the predicate, using other predicates.
In particular, note that the right hand side allows us to determine the value of the Offspring predicate *on every single input*.

Some examples:
Every integer has a square that is greater or equal to 0:
$\forall x \in \mathcal{I} : (x^2 > 0)$

Every integer has a square that is a positive integer:
$\forall x \in \mathcal{I} : \exists y \in \mathcal{N} : y = x^2$ The choice if y here might depend on the value of x already fixed.

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : BioMother(x, y) \leftrightarrow (Birthed(x, y) \wedge (FemaleSex(x))$
$\forall x : \forall y : \forall z : Offspring(x, y, z) \leftrightarrow (BioMother(y, x) \wedge BioFather(z, x))$

The use of $\leftrightarrow$ here allows us to define the predicate, using other predicates.
In particular, note that the right hand side allows us to determine the value of the Offspring predicate *on every single input*.

Some examples:
Every integer has a square that is greater or equal to 0:
$\forall x \in \mathcal{I} : (x^2 > 0)$

Every integer has a square that is a positive integer:
$\forall x \in \mathcal{I} : \exists y \in \mathcal{N} : y = x^2$ The choice if y here might depend on the value of x already fixed.

Ordering of quantifiers matters! (This is false!)
$\exists y \in \mathcal{N} : \forall x \in \mathcal{I} : (y = x^2)$

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : BioMother(x, y) \leftrightarrow (Birthed(x, y) \wedge (FemaleSex(x))$
$\forall x : \forall y : \forall z : Offspring(x, y, z) \leftrightarrow (BioMother(y, x) \wedge BioFather(z, x))$

The use of $\leftrightarrow$ here allows us to define the predicate, using other predicates.
In particular, note that the right hand side allows us to determine the value of the Offspring predicate *on every single input*.

Some examples:
Every integer has a square that is greater or equal to 0:
$\forall x \in \mathcal{I} : (x^2 > 0)$

Every integer has a square that is a positive integer:
$\forall x \in \mathcal{I} : \exists y \in \mathcal{N} : y = x^2$ The choice if y here might depend on the value of x already fixed.

Ordering of quantifiers matters! (This is false!)
$\exists y \in \mathcal{N} : \forall x \in \mathcal{I} : (y = x^2)$

Define predicate isPerfectSquare:
$\forall x \in \mathcal{I} : \text{IsPerfectSquare}(x) \leftrightarrow (\exists y \in \boldsymbol{I} : x = y^2)$

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : BioMother(x, y) \leftrightarrow (Birthed(x, y) \wedge (FemaleSex(x))$
$\forall x : \forall y : \forall z : Offspring(x, y, z) \leftrightarrow (BioMother(y, x) \wedge BioFather(z, x))$

The use of $\leftrightarrow$ here allows us to define the predicate, using other predicates.
In particular, note that the right hand side allows us to determine the value of the Offspring predicate *on every single input*.

Some examples:
Every integer has a square that is greater or equal to 0:
$\forall x \in \mathcal{I} : (x^2 > 0)$

Every integer has a square that is a positive integer:
$\forall x \in \mathcal{I} : \exists y \in \mathcal{N} : y = x^2$ The choice if y here might depend on the value of x already fixed.

Ordering of quantifiers matters! (This is false!)
$\exists y \in \mathcal{N} : \forall x \in \mathcal{I} : (y = x^2)$

Define predicate isPerfectSquare:
$\forall x \in \mathcal{I} : \text{IsPerfectSquare}(x) \leftrightarrow (\exists y \in \boldsymbol{I} : x = y^2)$

There is a special member of $\mathcal{N}$ that is smaller than all other members:
$\exists y \in \mathcal{N} : \forall x \in \mathcal{N} : (y \leq x)$

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : BioMother(x, y) \leftrightarrow (Birthed(x, y) \wedge (FemaleSex(x))$
$\forall x : \forall y : \forall z : Offspring(x, y, z) \leftrightarrow (BioMother(y, x) \wedge BioFather(z, x))$

The use of $\leftrightarrow$ here allows us to define the predicate, using other predicates.
In particular, note that the right hand side allows us to determine the value of the Offspring predicate *on every single input*.

Some examples:
Every integer has a square that is greater or equal to 0:
$\forall x \in \mathcal{I} : (x^2 > 0)$

Every integer has a square that is a positive integer:
$\forall x \in \mathcal{I} : \exists y \in \mathcal{N} : y = x^2$ The choice if y here might depend on the value of x already fixed.

Ordering of quantifiers matters! (This is false!)
$\exists y \in \mathcal{N} : \forall x \in \mathcal{I} : (y = x^2)$

Define predicate isPerfectSquare:
$\forall x \in \mathcal{I} : \text{IsPerfectSquare}(x) \leftrightarrow (\exists y \in \boldsymbol{I} : x = y^2)$

There is a special member of $\mathcal{N}$ that is smaller than all other members:
$\exists y \in \mathcal{N} : \forall x \in \mathcal{N} : (y \leq x)$

When all quantifiers are of the same kind, ordering does not matter:
$(\forall x : \forall y : p(x, y)) \equiv (\forall y : \forall x : p(x, y))$.

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : BioMother(x, y) \leftrightarrow (Birthed(x, y) \wedge (FemaleSex(x))$
$\forall x : \forall y : \forall z : Offspring(x, y, z) \leftrightarrow (BioMother(y, x) \wedge BioFather(z, x))$

The use of $\leftrightarrow$ here allows us to define the predicate, using other predicates.
In particular, note that the right hand side allows us to determine the value of the Offspring predicate *on every single input*.

Some examples:
Every integer has a square that is greater or equal to 0:
$\forall x \in \mathcal{I} : (x^2 > 0)$

Every integer has a square that is a positive integer:
$\forall x \in \mathcal{I} : \exists y \in \mathcal{N} : y = x^2$ The choice if y here might depend on the value of x already fixed.

Ordering of quantifiers matters! (This is false!)
$\exists y \in \mathcal{N} : \forall x \in \mathcal{I} : (y = x^2)$

Define predicate isPerfectSquare:
$\forall x \in \mathcal{I} : \text{IsPerfectSquare}(x) \leftrightarrow (\exists y \in \boldsymbol{I} : x = y^2)$

There is a special member of $\mathcal{N}$ that is smaller than all other members:
$\exists y \in \mathcal{N} : \forall x \in \mathcal{N} : (y \leq x)$

When all quantifiers are of the same kind, ordering does not matter:
$(\forall x : \forall y : p(x, y)) \equiv (\forall y : \forall x : p(x, y))$. Both mean that p holds for all pairs (x, y).

Multiple Quantifiers

If a predicate has more than 1 variable, we might apply more than 1 quantifier.
$\forall x : \forall y : BioMother(x, y) \leftrightarrow (Birthed(x, y) \wedge (FemaleSex(x))$
$\forall x : \forall y : \forall z : Offspring(x, y, z) \leftrightarrow (BioMother(y, x) \wedge BioFather(z, x))$

The use of $\leftrightarrow$ here allows us to define the predicate, using other predicates.
In particular, note that the right hand side allows us to determine the value of the Offspring predicate *on every single input*.

Some examples:
Every integer has a square that is greater or equal to 0:
$\forall x \in \mathcal{I} : (x^2 > 0)$

Every integer has a square that is a positive integer:
$\forall x \in \mathcal{I} : \exists y \in \mathcal{N} : y = x^2$ The choice if y here might depend on the value of x already fixed.

Ordering of quantifiers matters! (This is false!)
$\exists y \in \mathcal{N} : \forall x \in \mathcal{I} : (y = x^2)$

Define predicate isPerfectSquare:
$\forall x \in \mathcal{I} : \text{IsPerfectSquare}(x) \leftrightarrow (\exists y \in \boldsymbol{I} : x = y^2)$

There is a special member of $\mathcal{N}$ that is smaller than all other members:
$\exists y \in \mathcal{N} : \forall x \in \mathcal{N} : (y \leq x)$

When all quantifiers are of the same kind, ordering does not matter:
$(\forall x : \forall y : p(x, y)) \equiv (\forall y : \forall x : p(x, y))$. Both mean that p holds for all pairs (x, y).
$(\exists x : \exists y : p(x, y)) \equiv (\exists y : \exists x : p(x, y))$.

Defining Formulas

Formulas

- If p is an n-argument predicate, and each of $a_1, \ldots, a_n$ is either an element of a domain $\mathcal{U}$, or a variable over a domain, then $p(a_1, \ldots, a_n)$ is a formula. By convention, a predicate with $n = 0$ arguments is a simple Boolean variable.

Defining Formulas

Formulas

- If p is an n-argument predicate, and each of $a_1, \ldots, a_n$ is either an element of a domain $\mathcal{U}$, or a variable over a domain, then $p(a_1, \ldots, a_n)$ is a formula. By convention, a predicate with $n = 0$ arguments is a simple Boolean variable.
- If α is a formula and β is a formula, then $\neg\alpha$, $(\alpha \vee \beta)$ and $(\alpha \wedge \beta)$ are formulas.

Defining Formulas

Formulas

- If p is an n-argument predicate, and each of $a_1, \ldots, a_n$ is either an element of a domain $\mathcal{U}$, or a variable over a domain, then $p(a_1, \ldots, a_n)$ is a formula. By convention, a predicate with $n = 0$ arguments is a simple Boolean variable.
- If α is a formula and β is a formula, then $\neg\alpha$, $(\alpha \vee \beta)$ and $(\alpha \wedge \beta)$ are formulas.
- If α is a formula and x is a variable over a domain, then $(\exists x : \alpha)$ and $(\forall x : \alpha)$ are formulas.

More Examples: Character Arrays

Let $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$ be sequences of characters.
$\prec$: predicate meaning x comes before y lexicographically. E.g. $b \prec d \equiv$ True.
$\mathcal{I}_n$ is the first n integers.

More Examples: Character Arrays

Let $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$ be sequences of characters.
$\prec$: predicate meaning x comes before y lexicographically. E.g. $b \prec d \equiv$ True.
$\mathcal{I}_n$ is the first n integers.

State that some letter appears in the same position in the sequences A and B.

More Examples: Character Arrays

Let $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$ be sequences of characters.
$\prec$: predicate meaning x comes before y lexicographically. E.g. $b \prec d \equiv \text{True}$.
$\mathcal{I}_n$ is the first n integers.

State that some letter appears in the same position in the sequences A and B.
$\exists i \in \mathcal{I}_n : a_i = b_i$

More Examples: Character Arrays

Let $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$ be sequences of characters.
$\prec$: predicate meaning x comes before y lexicographically. E.g. $b \prec d \equiv$ True.
$\mathcal{I}_n$ is the first n integers.

State that some letter appears in the same position in the sequences A and B.
$\exists i \in \mathcal{I}_n : a_i = b_i$

Every pair of elements of A are equal to each other:

More Examples: Character Arrays

Let $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$ be sequences of characters.
$\prec$: predicate meaning x comes before y lexicographically. E.g. $b \prec d \equiv$ True.
$\mathcal{I}_n$ is the first n integers.

State that some letter appears in the same position in the sequences A and B.
$\exists i \in \mathcal{I}_n : a_i = b_i$

Every pair of elements of A are equal to each other:
$\forall i \in \mathcal{I}_n : \forall j \in \mathcal{I}_n : a_i = a_j$.

More Examples: Character Arrays

Let $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$ be sequences of characters.
$\prec$: predicate meaning x comes before y lexicographically. E.g. $b \prec d \equiv$ True.
$\mathcal{I}_n$ is the first n integers.

State that some letter appears in the same position in the sequences A and B.
$\exists i \in \mathcal{I}_n : a_i = b_i$

Every pair of elements of A are equal to each other:
$\forall i \in \mathcal{I}_n : \forall j \in \mathcal{I}_n : a_i = a_j$.
$\forall i \in \mathcal{I}_n : a_i = a_1$

More Examples: Character Arrays

Let $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$ be sequences of characters.
$\prec$: predicate meaning x comes before y lexicographically. E.g. $b \prec d \equiv$ True.
$\mathcal{I}_n$ is the first n integers.

State that some letter appears in the same position in the sequences A and B.
$\exists i \in \mathcal{I}_n : a_i = b_i$

Every pair of elements of A are equal to each other:
$\forall i \in \mathcal{I}_n : \forall j \in \mathcal{I}_n : a_i = a_j$.
$\forall i \in \mathcal{I}_n : a_i = a_1$

What does the following formula mean (in English)?
$\forall i \in \mathcal{I}_{n-1} : a_i \prec a_{i+1}$

More Examples: Graphs

Undirected Graph $G = (V, E)$:
V is a set of vertices, and
$E \subseteq V \times V$ is a relation on V, called the set of edges.

More Examples: Graphs

Undirected Graph $G = (V, E)$:
V is a set of vertices, and
$E \subseteq V \times V$ is a relation on V, called the set of edges.
Because it is undirected, $(x, y) \in E \leftrightarrow (y, x) \in E$.

More Examples: Graphs

Undirected Graph $G = (V, E)$:
V is a set of vertices, and
$E \subseteq V \times V$ is a relation on V, called the set of edges.
Because it is undirected, $(x, y) \in E \leftrightarrow (y, x) \in E$.

A path of length k from $x \in V$ to $y \in V$ is a sequence of vertices,
$x = x_0, x_1, \ldots, x_k = y$, such that $\forall i \in \{1, \ldots, k\} : (x_{i-1}, x_i) \in E$

More Examples: Graphs

Undirected Graph $G = (V, E)$:
V is a set of vertices, and
$E \subseteq V \times V$ is a relation on V, called the set of edges.
Because it is undirected, $(x, y) \in E \leftrightarrow (y, x) \in E$.

A path of length k from $x \in V$ to $y \in V$ is a sequence of vertices,
$x = x_0, x_1, \ldots, x_k = y$, such that $\forall i \in \{1, \ldots, k\} : (x_{i-1}, x_i) \in E$

Suppose $V = \{a, b, c, d\}$ and $E = \{(a, b), (b, c), (a, d)\}$
Is the following formula True or False?
$\forall x \in V : \exists y \in V : (x, y) \in E$

More Examples: Graphs

Undirected Graph $G = (V, E)$:
V is a set of vertices, and
$E \subseteq V \times V$ is a relation on V, called the set of edges.
Because it is undirected, $(x, y) \in E \leftrightarrow (y, x) \in E$.

A path of length k from $x \in V$ to $y \in V$ is a sequence of vertices,
$x = x_0, x_1, \ldots, x_k = y$, such that $\forall i \in \{1, \ldots, k\} : (x_{i-1}, x_i) \in E$

Suppose $V = \{a, b, c, d\}$ and $E = \{(a, b), (b, c), (a, d)\}$
Is the following formula True or False?
$\forall x \in V : \exists y \in V : (x, y) \in E$

What about the following formula?
$\exists x \in V : \forall y \in V : (x, y) \in E$

More Examples: Graphs

Undirected Graph $G = (V, E)$:
V is a set of vertices, and
$E \subseteq V \times V$ is a relation on V, called the set of edges.
Because it is undirected, $(x, y) \in E \leftrightarrow (y, x) \in E$.

A path of length k from $x \in V$ to $y \in V$ is a sequence of vertices,
$x = x_0, x_1, \ldots, x_k = y$, such that $\forall i \in \{1, \ldots, k\} : (x_{i-1}, x_i) \in E$

Suppose $V = \{a, b, c, d\}$ and $E = \{(a, b), (b, c), (a, d)\}$
Is the following formula True or False?
$\forall x \in V : \exists y \in V : (x, y) \in E$

What about the following formula?
$\exists x \in V : \forall y \in V : (x, y) \in E$

Assume $\forall x \in V : (x, x) \notin E$
Give a formula for the following English sentence:
Every pair of vertices has a path of length 2.

More Examples: Graphs

Undirected Graph $G = (V, E)$:
V is a set of vertices, and
$E \subseteq V \times V$ is a relation on V, called the set of edges.
Because it is undirected, $(x, y) \in E \leftrightarrow (y, x) \in E$.

A path of length k from $x \in V$ to $y \in V$ is a sequence of vertices, $x = x_0, x_1, \ldots, x_k = y$, such that $\forall i \in \{1, \ldots, k\} : (x_{i-1}, x_i) \in E$

Suppose $V = \{a, b, c, d\}$ and $E = \{(a, b), (b, c), (a, d)\}$
Is the following formula True or False?
$\forall x \in V : \exists y \in V : (x, y) \in E$

What about the following formula?
$\exists x \in V : \forall y \in V : (x, y) \in E$

Assume $\forall x \in V : (x, x) \notin E$
Give a formula for the following English sentence:
Every pair of vertices has a path of length 2.

$\forall x \in V : \forall y \in V : \exists Z \in V : ((x, z) \in E) \wedge ((y, z) \in E)$

Example: Recursive Definition of Path

$$\forall x \in V : \forall y \in V : Edge(x, y) \leftrightarrow (x, y) \in E$$

Example: Recursive Definition of Path

$\forall x \in V : \forall y \in V : Edge(x, y) \leftrightarrow (x, y) \in E$

Let Path(x, y) be a predicate for the property that there is a sequence of 0, 1, 2 or more edges connecting x and y.
Give a recursive definition of Path, using predicate Edge:

Example: Recursive Definition of Path

$\forall x \in V : \forall y \in V : Edge(x, y) \leftrightarrow (x, y) \in E$

Let Path(x, y) be a predicate for the property that there is a sequence of 0, 1, 2 or more edges connecting x and y.
Give a recursive definition of Path, using predicate Edge:

$\forall x \in V : \forall y \in V : \mathrm{Path}(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$

Example: Recursive Definition of Path

$\forall x \in V : \forall y \in V : Edge(x, y) \leftrightarrow (x, y) \in E$

Let Path(x, y) be a predicate for the property that there is a sequence of 0, 1, 2 or more edges connecting x and y.
Give a recursive definition of Path, using predicate Edge:

$\forall x \in V : \forall y \in V : \mathrm{Path}(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Path of length 0: $Path(x, x) \leftrightarrow (x = x) \vee \ldots$

Example: Recursive Definition of Path

$\forall x \in V : \forall y \in V : Edge(x, y) \leftrightarrow (x, y) \in E$

Let Path(x, y) be a predicate for the property that there is a sequence of 0, 1, 2 or more edges connecting x and y.
Give a recursive definition of Path, using predicate Edge:

$\forall x \in V : \forall y \in V : \mathrm{Path}(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Path of length 0: $Path(x, x) \leftrightarrow (x = x) \vee \ldots$
Path of length 1: $Path(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$

Example: Recursive Definition of Path

$\forall x \in V : \forall y \in V : Edge(x, y) \leftrightarrow (x, y) \in E$

Let Path(x, y) be a predicate for the property that there is a sequence of 0, 1, 2 or more edges connecting x and y.
Give a recursive definition of Path, using predicate Edge:

$\forall x \in V : \forall y \in V : \mathrm{Path}(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Path of length 0: $Path(x, x) \leftrightarrow (x = x) \vee \ldots$
Path of length 1: $Path(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Take $z = y$: $Edge(x, y) \wedge Path(y, y)$

Example: Recursive Definition of Path

$\forall x \in V : \forall y \in V : Edge(x, y) \leftrightarrow (x, y) \in E$

Let Path(x, y) be a predicate for the property that there is a sequence of 0, 1, 2 or more edges connecting x and y.
Give a recursive definition of Path, using predicate Edge:

$\forall x \in V : \forall y \in V : \mathrm{Path}(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Path of length 0: $Path(x, x) \leftrightarrow (x = x) \vee \ldots$
Path of length 1: $Path(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Take $z = y$: $Edge(x, y) \wedge Path(y, y)$
Path of length 2 (x, r, y): $Path(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$

Example: Recursive Definition of Path

$\forall x \in V : \forall y \in V : Edge(x, y) \leftrightarrow (x, y) \in E$

Let Path(x, y) be a predicate for the property that there is a sequence of 0, 1, 2 or more edges connecting x and y.
Give a recursive definition of Path, using predicate Edge:

$\forall x \in V : \forall y \in V : \mathrm{Path}(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Path of length 0: $Path(x, x) \leftrightarrow (x = x) \vee \ldots$
Path of length 1: $Path(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Take $z = y$: $Edge(x, y) \wedge Path(y, y)$
Path of length 2 (x, r, y): $Path(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Take $z = r$: $Edge(x, r) \wedge Path(r, y)$

Example: Recursive Definition of Path

$\forall x \in V : \forall y \in V : Edge(x, y) \leftrightarrow (x, y) \in E$

Let Path(x, y) be a predicate for the property that there is a sequence of 0, 1, 2 or more edges connecting x and y.
Give a recursive definition of Path, using predicate Edge:

$\forall x \in V : \forall y \in V : \mathrm{Path}(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Path of length 0: $Path(x, x) \leftrightarrow (x = x) \vee \ldots$
Path of length 1: $Path(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Take $z = y$: $Edge(x, y) \wedge Path(y, y)$
Path of length 2 (x, r, y): $Path(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Take $z = r$: $Edge(x, r) \wedge Path(r, y)$
$Path(r, y) \leftrightarrow (r = y) \vee \exists z \in V : Edge(r, z) \wedge Path(z, y)$

Example: Recursive Definition of Path

$\forall x \in V : \forall y \in V : Edge(x, y) \leftrightarrow (x, y) \in E$

Let Path(x, y) be a predicate for the property that there is a sequence of 0, 1, 2 or more edges connecting x and y.
Give a recursive definition of Path, using predicate Edge:

$\forall x \in V : \forall y \in V : \mathrm{Path}(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Path of length 0: $Path(x, x) \leftrightarrow (x = x) \vee \ldots$
Path of length 1: $Path(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Take $z = y$: $Edge(x, y) \wedge Path(y, y)$
Path of length 2 (x, r, y): $Path(x, y) \leftrightarrow (x = y) \vee \exists z \in V : Edge(x, z) \wedge Path(z, y)$
Take $z = r$: $Edge(x, r) \wedge Path(r, y)$
$Path(r, y) \leftrightarrow (r = y) \vee \exists z \in V : Edge(r, z) \wedge Path(z, y)$
Take $z = y$: $Edge(r, y) \wedge Path(y, y)$

Exercise

Let $V = \{a, b, c, d\}$
Case a) Edge(a,b) = Edge(b,c) = Edge(b,d) = TRUE
Case b) Edge(a,b) = Edge(b,c) = Edge(c,d) = TRUE
$\forall x \in V : \forall y \in V : (x \neq y) \rightarrow Edge(x, y)$

Exercise

Let $V = \{a, b, c, d\}$
Case a) Edge(a,b) = Edge(b,c) = Edge(b,d) = TRUE
Case b) Edge(a,b) = Edge(b,c) = Edge(c,d) = TRUE
$\forall x \in V : \forall y \in V : (x \neq y) \rightarrow Edge(x, y)$

$\forall x \in V : \forall y \in V : Path(x, y)$