

# Predicate logic

# Limitations of propositional logic

Propositional logic only allows us to reason about **completed statements** about things, not about the things themselves.

## Example

Some chicken cannot fly  
All chicken are birds  
-----  
Some birds cannot fly

this reasoning can not  
be expressed in  
propositional logic

## Example

Every player except the winner loses a match

# Unary predicate (example)

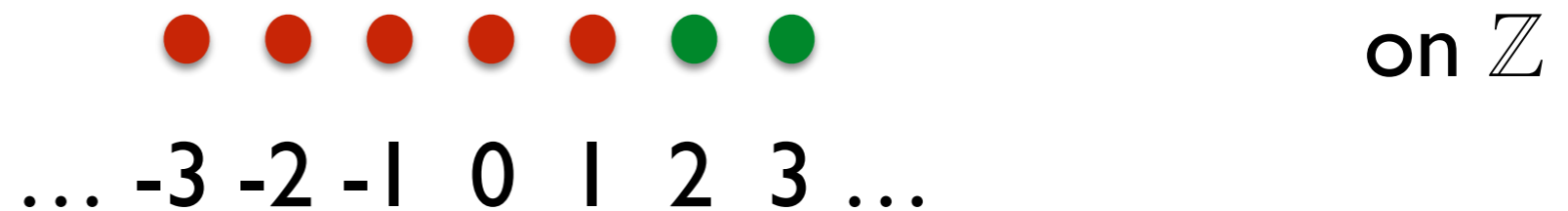
Consider the statement  $2m > 3$ .

Whether this statement is true or false depends on the value of  $m$  (and on the domain of values).

# Unary predicate (example)

Consider the statement  $2m > 3$ .

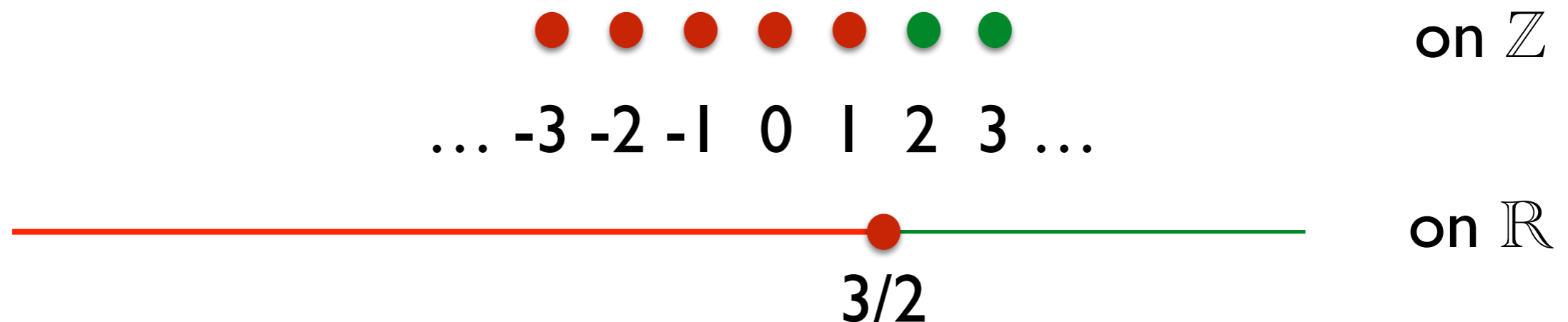
Whether this statement is true or false depends on the value of  $m$  (and on the domain of values).



# Unary predicate (example)

Consider the statement  $2m > 3$ .

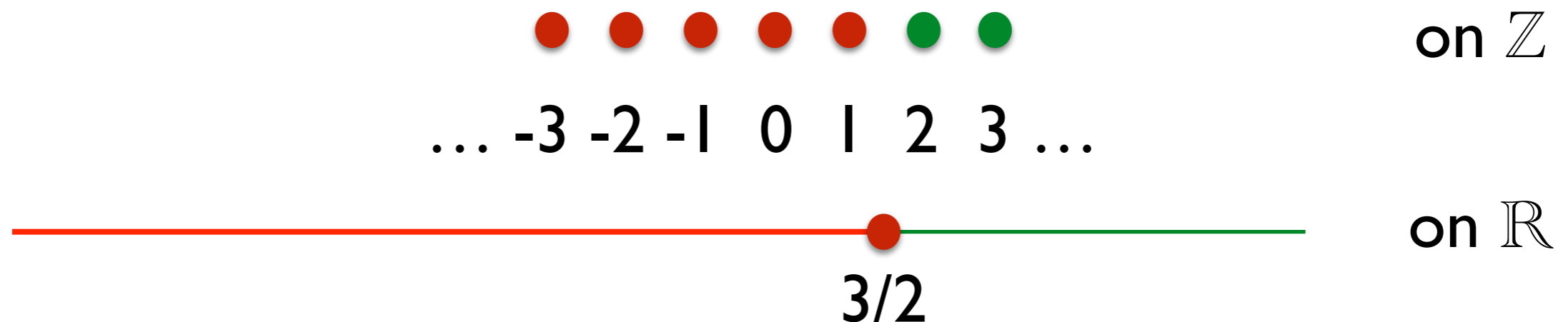
Whether this statement is true or false depends on the value of  $m$  (and on the domain of values).



# Unary predicate (example)

Consider the statement  $2m > 3$ .

Whether this statement is true or false depends on the value of  $m$  (and on the domain of values).



Note:  $2m > 3 \stackrel{\text{val}}{=} m > 3/2$  on  $\mathbb{Z}$  and  $\mathbb{R}$

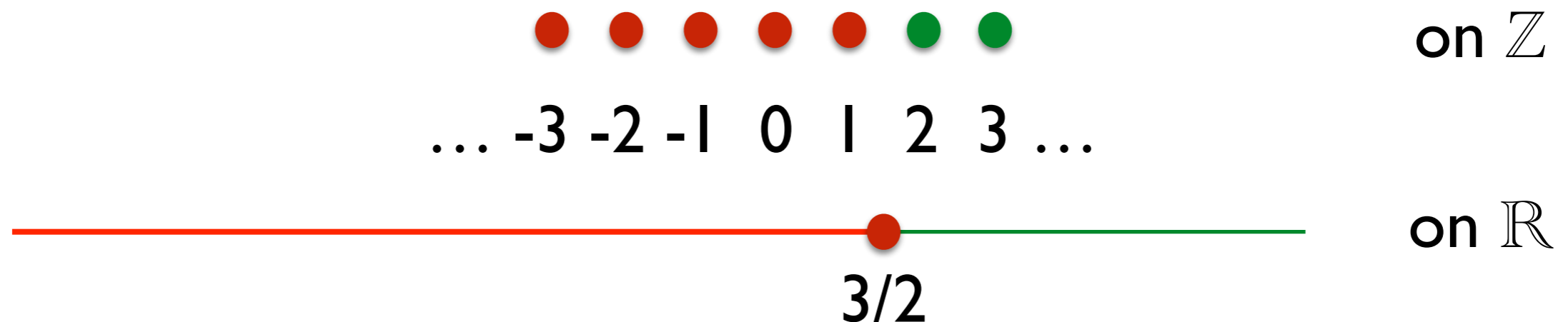
$2m > 3 \stackrel{\text{val}}{=} m \geq 2$  on  $\mathbb{Z}$  but not on  $\mathbb{R}$

# Unary predicate (example)

Consider the statement  $2m > 3$ .

a unary  
relation

Whether this statement is true or false depends on the value of  $m$  (and on the domain of values).

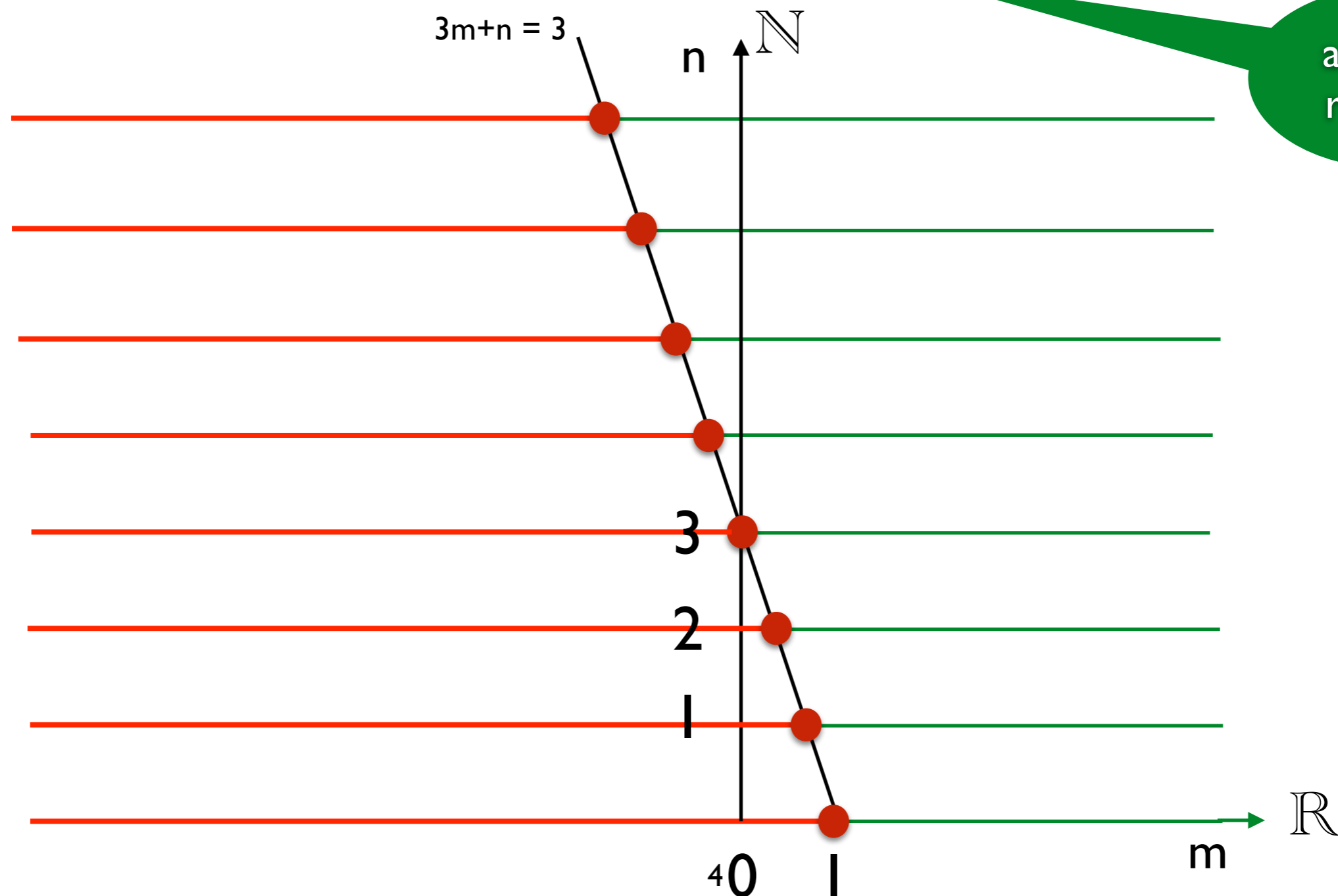


Note:  $2m > 3 \stackrel{\text{val}}{=} m > 3/2$  on  $\mathbb{Z}$  and  $\mathbb{R}$

$2m > 3 \stackrel{\text{val}}{=} m \geq 2$  on  $\mathbb{Z}$  but not on  $\mathbb{R}$

# Binary predicate (example)

The statement  $3m+n > 3$  is a binary predicate on  $\mathbb{R} \times \mathbb{N}$ .



a binary  
relation

# Predicates

In general, an  $n$ -ary predicate is an  $n$ -ary relation.

If it is on a domain  $D$ , then it's a relation  $P(x_1, \dots, x_n) \subseteq D^n$  or equivalently a function  $P: D^n \rightarrow \{0, 1\}$ .

# Predicates

In general, an n-ary predicate is an n-ary relation.

If it is on a domain  $D$ , then it's a relation  $P(x_1, \dots, x_n) \subseteq D^n$  or equivalently a function  $P: D^n \rightarrow \{0, 1\}$ .

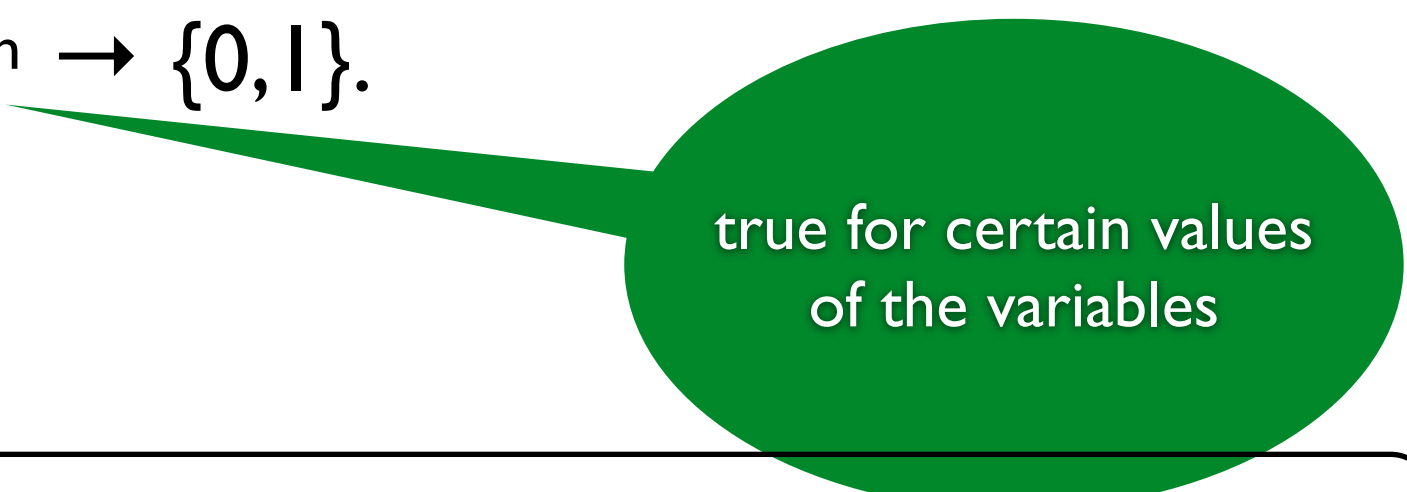


true for certain values  
of the variables

# Predicates

In general, an n-ary predicate is an n-ary relation.

If it is on a domain  $D$ , then it's a relation  $P(x_1, \dots, x_n) \subseteq D^n$  or equivalently a function  $P: D^n \rightarrow \{0, 1\}$ .



true for certain values  
of the variables

We can turn a predicate, into a proposition in three ways:

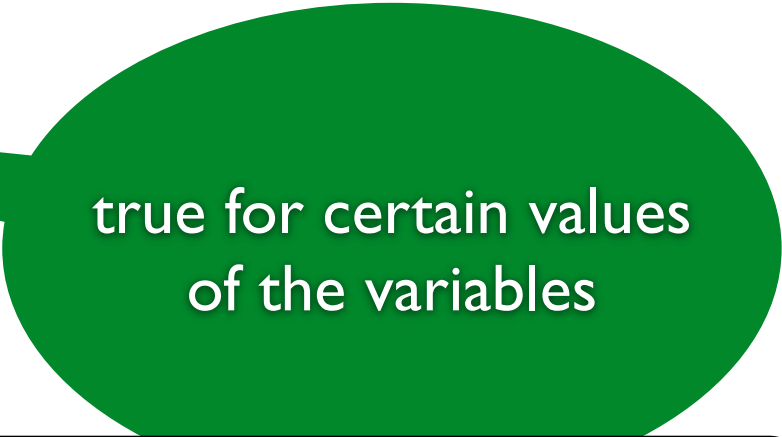
1. By assigning values to the variables.
2. By universal quantification.
3. By existential quantification.

# Predicates

In general, an  $n$ -ary predicate is an  $n$ -ary relation.

If it is on a domain  $D$ , then it's a relation  $P(x_1, \dots, x_n) \subseteq D^n$  or equivalently a function  $P: D^n \rightarrow \{0, 1\}$ .


$$2m > 3$$



true for certain values  
of the variables

We can turn a predicate, into a proposition in three ways:

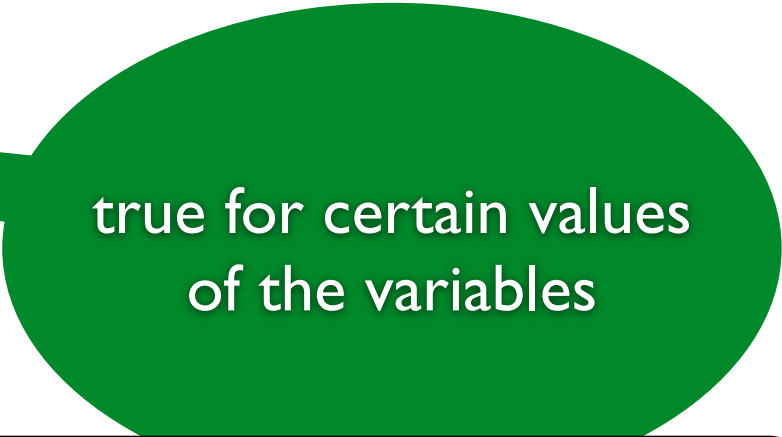
1. By assigning values to the variables.
2. By universal quantification.
3. By existential quantification.

# Predicates

In general, an  $n$ -ary predicate is an  $n$ -ary relation.

If it is on a domain  $D$ , then it's a relation  $P(x_1, \dots, x_n) \subseteq D^n$  or equivalently a function  $P: D^n \rightarrow \{0, 1\}$ .

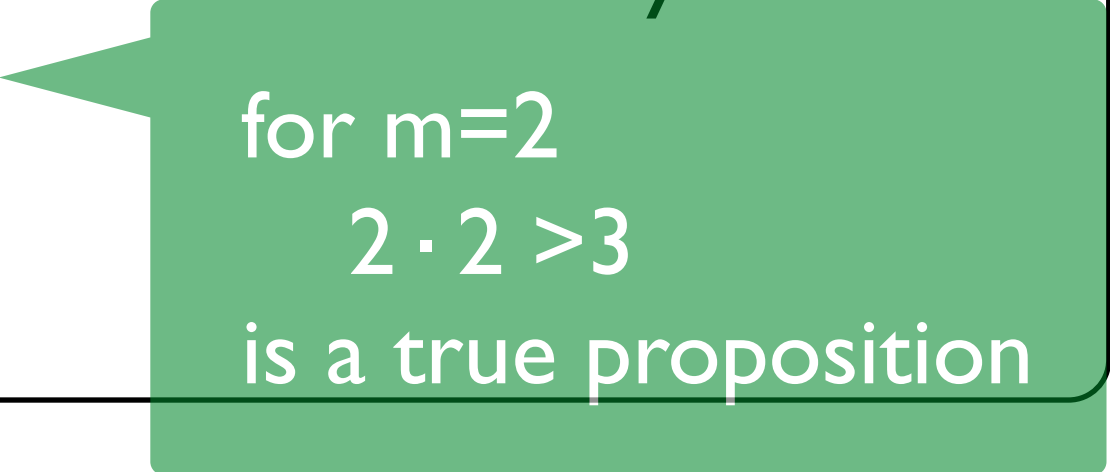

$$2m > 3$$



true for certain values  
of the variables

We can turn a predicate, into a proposition in three ways:

1. By assigning values to the variables.
2. By universal quantification.
3. By existential quantification.



for  $m=2$   
 $2 \cdot 2 > 3$   
is a true proposition

# Universal quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can be turned into a proposition by universal quantification:

# Universal quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can be turned into a proposition by universal quantification:

For all  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

# Universal quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can be turned into a proposition by universal quantification:

For all  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

false, e.g.  
for  $m = 1$

# Universal quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can be turned into a proposition by universal quantification:

For all  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

false, e.g.  
for  $m = 1$

Notation:  $\forall_m [m \in \mathbb{Z} : 2m > 3]$

# Universal quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can be turned into a proposition by universal quantification:

For all  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

false, e.g.  
for  $m = 1$

universal  
quantifier

Notation:

$\forall m [m \in \mathbb{Z} : 2m > 3]$

# Universal quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can be turned into a proposition by universal quantification:

For all  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

false, e.g.  
for  $m = 1$

Notation:

$\forall_m [m \in \mathbb{Z} : 2m > 3]$

universal  
quantifier

domain  
(predicate)

# Universal quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can be turned into a proposition by universal quantification:

For all  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

false, e.g.  
for  $m = 1$

Notation:

$\forall_m [m \in \mathbb{Z} : 2m > 3]$

universal  
quantifier

domain  
(predicate)

predicate

# Universal quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can be turned into a proposition by universal quantification:

For all  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

false, e.g.  
for  $m = 1$

Notation:

$\forall_m [m \in \mathbb{Z} : 2m > 3]$

universal  
quantifier

predicate

domain  
(predicate)

In general:

$\forall_x [P(x) : Q(x)]$  for “all  $x$  satisfying  $P$  satisfy  $Q$ ”

# Universal quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can be turned into a proposition by universal quantification:

For all  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

false, e.g.  
for  $m = 1$

Notation:

$\forall_m [m \in \mathbb{Z} : 2m > 3]$

universal  
quantifier

domain  
(predicate)

predicate

other standard (!)  
notation:

$\forall x (P(x) \Rightarrow Q(x))$

$\forall x. P(x) \Rightarrow Q(x)$

In general:

$\forall_x [P(x) : Q(x)]$  for “all  $x$  satisfying  $P$  satisfy  $Q$ ”

# Existential quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can also be turned into a proposition by existential quantification:

# Existential quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can also be turned into a proposition by existential quantification:

There exists  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

# Existential quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can also be turned into a proposition by existential quantification:

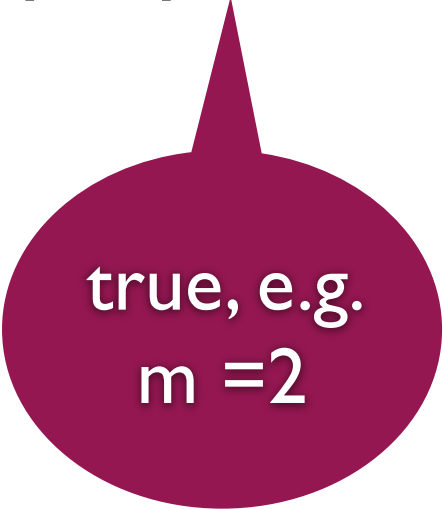
There exists  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

true, e.g.  
 $m = 2$

# Existential quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can also be turned into a proposition by existential quantification:

There exists  $m$  in  $\mathbb{Z}$ ,  $2m > 3$



true, e.g.  
 $m = 2$

Notation:  $\exists_m [m \in \mathbb{Z} : 2m > 3]$

# Existential quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can also be turned into a proposition by existential quantification:

true, e.g.  
 $m = 2$

There exists  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

existential  
quantifier

Notation:

$\exists_m [m \in \mathbb{Z} : 2m > 3]$

# Existential quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can also be turned into a proposition by existential quantification:

true, e.g.  
 $m = 2$

There exists  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

Notation:

$\exists_m [m \in \mathbb{Z} : 2m > 3]$

existential  
quantifier

domain  
(predicate)

# Existential quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can also be turned into a proposition by existential quantification:

true, e.g.  
 $m = 2$

There exists  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

Notation:

$\exists_m [m \in \mathbb{Z} : 2m > 3]$

existential  
quantifier

domain  
(predicate)

predicate

# Existential quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can also be turned into a proposition by existential quantification:

true, e.g.  
 $m = 2$

There exists  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

Notation:

$\exists_m [m \in \mathbb{Z} : 2m > 3]$

existential  
quantifier

domain  
(predicate)

predicate

In general:

$\exists_x [P(x) : Q(x)]$  for

“there exists  $x$  satisfying  $P$  that satisfies  $Q$ ”

# Existential quantification

The unary predicate  $2m > 3$  on  $\mathbb{Z}$  can also be turned into a proposition by existential quantification:

true, e.g.  
 $m = 2$

There exists  $m$  in  $\mathbb{Z}$ ,  $2m > 3$

Notation:

$\exists_m [m \in \mathbb{Z} : 2m > 3]$

existential  
quantifier

domain  
(predicate)

predicate

other standard (!)  
notation:

$\exists x (P(x) \wedge Q(x))$

$\exists x. P(x) \wedge Q(x)$

In general:

$\exists_x [P(x) : Q(x)]$  for

“there exists  $x$  satisfying  $P$  that satisfies  $Q$ ”

# Quantification

The binary predicate  $3m+n > 3$  on  $\mathbb{R} \times \mathbb{N}$  can also be turned into a proposition by quantification:

# Quantification

The binary predicate  $3m+n > 3$  on  $\mathbb{R} \times \mathbb{N}$  can also be turned into a proposition by quantification:



in 8  
possible  
ways

# Quantification

The binary predicate  $3m+n > 3$  on  $\mathbb{R} \times \mathbb{N}$  can also be turned into a proposition by quantification:



in 8  
possible  
ways

One way is:  $\exists_m [m \in \mathbb{R} : \forall_n [n \in \mathbb{N} : 3m + n > 3]]$

# Quantification

The binary predicate  $3m+n > 3$  on  $\mathbb{R} \times \mathbb{N}$  can also be turned into a proposition by quantification:

in 8 possible ways

One way is:  $\exists_m [m \in \mathbb{R} : \forall_n [n \in \mathbb{N} : \underline{3m + n > 3}]]$

binary predicate

# Quantification

The binary predicate  $3m+n > 3$  on  $\mathbb{R} \times \mathbb{N}$  can also be turned into a proposition by quantification:

in 8 possible ways

One way is:

$$\exists_m [m \in \mathbb{R} : \forall_n [n \in \mathbb{N} : 3m + n > 3]]$$

unary predicate

binary predicate

# Quantification

The binary predicate  $3m+n > 3$  on  $\mathbb{R} \times \mathbb{N}$  can also be turned into a proposition by quantification:

in 8 possible ways

One way is:

$$\exists_m [m \in \mathbb{R} : \forall_n [n \in \mathbb{N} : 3m + n > 3]]$$

unary predicate

binary predicate

proposition,  
nullary predicate

# Quantification

The binary predicate  $3m+n > 3$  on  $\mathbb{R} \times \mathbb{N}$  can also be turned into a proposition by quantification:

in 8 possible ways

One way is:

$$\exists_m [m \in \mathbb{R} : \forall_n [n \in \mathbb{N} : 3m + n > 3]]$$

unary predicate

binary predicate

proposition,  
nullary predicate

other standard (!) notation:

$$\exists m (m \in \mathbb{R} \wedge \forall n (n \in \mathbb{N} \Rightarrow 3m+n > 3))$$

# Additional Notation Rules

# Additional Notation Rules

We write  $\forall_x [P]$  for  $\forall_x [T : P]$

# Additional Notation Rules

We write  $\forall_x [P]$  for  $\forall_x [T : P]$

also for  $\exists$

# Additional Notation Rules

We write  $\forall_x [P]$  for  $\forall_x [T : P]$

also for  $\exists$

We also write  $\exists_m, \forall_n [(m,n) \in \mathbb{R} \times \mathbb{N} : 3m + n > 3]$

for  $\exists_m [m \in \mathbb{R} : \forall_n [n \in \mathbb{N} : 3m + n > 3]]$

# Additional Notation Rules

We write  $\forall_x [P]$  for  $\forall_x [T : P]$

also for  $\exists$

We also write  $\exists_m, \forall_n [(m,n) \in \mathbb{R} \times \mathbb{N} : 3m + n > 3]$   
for  $\exists_m [m \in \mathbb{R} : \forall_n [n \in \mathbb{N} : 3m + n > 3]]$

And even  $\exists_{m,n} [(m,n) \in \mathbb{R} \times \mathbb{N} : 3m + n > 3]$   
for  $\exists_m [m \in \mathbb{R} : \exists_n [n \in \mathbb{N} : 3m + n > 3]]$

# Additional Notation Rules

We write  $\forall_x [P]$  for  $\forall_x [T : P]$

also for  $\exists$

We also write  $\exists_m, \forall_n [(m,n) \in \mathbb{R} \times \mathbb{N} : 3m + n > 3]$

for  $\exists_m [m \in \mathbb{R} : \forall_n [n \in \mathbb{N} : 3m + n > 3]]$

And even  $\exists_{m,n} [(m,n) \in \mathbb{R} \times \mathbb{N} : 3m + n > 3]$

for  $\exists_m [m \in \mathbb{R} : \exists_n [n \in \mathbb{N} : 3m + n > 3]]$

but only for the same  
quantifier!

# Quantification - task

Let  $P$  be the set of all tennis players.

Let  $w \in P$  be the winner.

For  $p, q \in P$ , write  $p \neq q$  for “ $p$  and  $q$  are different players”.

Let  $M$  be the set of all matches.

For  $p \in P$  and  $m \in M$ , write  $L(p,m)$  for  
“player  $p$  loses match  $m$ ”.

Write the following sentence as a formula with predicates and quantifiers:

Every player except the winner loses a match.

# Quantification - task

Let  $P$  be the set of all tennis players.

Let  $w \in P$  be the winner.

Thanks to Bas Luttik

For  $p, q \in P$ , write  $p \neq q$  for “ $p$  and  $q$  are different players”.

Let  $M$  be the set of all matches.

For  $p \in P$  and  $m \in M$ , write  $L(p,m)$  for  
“player  $p$  loses match  $m$ ”.

Write the following sentence as a formula with predicates and quantifiers:

Every player except the winner loses a match.

# Equivalences with quantifiers

# Renaming bound variables

## Bound variables

$$\forall_x [P:Q] \stackrel{val}{=} \forall_y [P[y/x]:Q[y/x]]$$

$$\exists_x [P:Q] \stackrel{val}{=} \exists_y [P[y/x]:Q[y/x]]$$

if  $y$  does not occur in  
 $P$  or  $Q$  (not even in  $\forall y, \exists y$ )

# Domain splitting

# Domain splitting

## Domain splitting

$$\forall_x [P \vee Q : R] \stackrel{val}{=} \forall_x [P : R] \wedge \forall_x [Q : R]$$

$$\exists_x [P \vee Q : R] \stackrel{val}{=} \exists_x [P : R] \vee \exists_x [Q : R]$$

# Domain splitting

## Domain splitting

$$\forall_x [P \vee Q : R] \stackrel{val}{=} \forall_x [P : R] \wedge \forall_x [Q : R]$$

$$\exists_x [P \vee Q : R] \stackrel{val}{=} \exists_x [P : R] \vee \exists_x [Q : R]$$

Examples:

# Domain splitting

## Domain splitting

$$\forall_x [P \vee Q : R] \stackrel{val}{=} \forall_x [P : R] \wedge \forall_x [Q : R]$$

$$\exists_x [P \vee Q : R] \stackrel{val}{=} \exists_x [P : R] \vee \exists_x [Q : R]$$

## Examples:

$$\begin{aligned} & \forall_x [x \leq 1 \vee x \geq 5 : x^2 - 6x + 5 \geq 0] \\ \stackrel{val}{=} & \forall_x [x \leq 1 : x^2 - 6x + 5 \geq 0] \wedge \forall_x [x \geq 5 : x^2 - 6x + 5 \geq 0] \end{aligned}$$

# Domain splitting

## Domain splitting

$$\forall_x [P \vee Q : R] \stackrel{val}{=} \forall_x [P : R] \wedge \forall_x [Q : R]$$

$$\exists_x [P \vee Q : R] \stackrel{val}{=} \exists_x [P : R] \vee \exists_x [Q : R]$$

## Examples:

$$\begin{aligned} & \forall_x [x \leq 1 \vee x \geq 5 : x^2 - 6x + 5 \geq 0] \\ \stackrel{val}{=} & \forall_x [x \leq 1 : x^2 - 6x + 5 \geq 0] \wedge \forall_x [x \geq 5 : x^2 - 6x + 5 \geq 0] \end{aligned}$$

$$\begin{aligned} & \exists_k [0 \leq k \leq n : k^2 \leq 10] \\ \stackrel{val}{=} & \exists_k [0 \leq k \leq n - 1 \vee k = n : k^2 \leq 10] \\ \stackrel{val}{=} & \exists_k [0 \leq k \leq n - 1 : k^2 \leq 10] \vee \exists_k [k = n : k^2 \leq 10] \end{aligned}$$

# Equivalences with quantifiers

## One-element domain

$$\forall_x [x = n : Q] \stackrel{val}{=} Q[n/x]$$

$$\exists_x [x = n : Q] \stackrel{val}{=} Q[n/x]$$

# Equivalences with quantifiers

## One-element domain

$$\forall_x [x = n : Q] \stackrel{val}{=} Q[n/x]$$

$$\exists_x [x = n : Q] \stackrel{val}{=} Q[n/x]$$

**Example:**

$$\forall_x [x = 3 : 2 \cdot x \geq 1] \stackrel{val}{=} 2 \cdot 3 \geq 1$$

# Equivalences with quantifiers

## One-element domain

$$\forall_x [x = n : Q] \stackrel{val}{=} Q[n/x]$$

$$\exists_x [x = n : Q] \stackrel{val}{=} Q[n/x]$$

**Example:**

$$\forall_x [x = 3 : 2 \cdot x \geq 1] \stackrel{val}{=} 2 \cdot 3 \geq 1$$

## Empty domain

$$\forall_x [F : Q] \stackrel{val}{=} T$$

$$\exists_x [F : Q] \stackrel{val}{=} F$$

# Equivalences with quantifiers

## One-element domain

$$\forall_x [x = n : Q] \stackrel{val}{=} Q[n/x]$$

$$\exists_x [x = n : Q] \stackrel{val}{=} Q[n/x]$$

**Example:**

$$\forall_x [x = 3 : 2 \cdot x \geq 1] \stackrel{val}{=} 2 \cdot 3 \geq 1$$

“All Marsians are green”

## Empty domain

$$\forall_x [F : Q] \stackrel{val}{=} T$$

$$\exists_x [F : Q] \stackrel{val}{=} F$$

# Domain weakening

**Intuition:** The following are equivalent

$$\begin{array}{lll} \forall_x [x \in D : A(x)] & \text{and} & \forall_x [x \in D \Rightarrow A(x)] \\ \exists_x [x \in D : A(x)] & \text{and} & \exists_x [x \in D \wedge A(x)] \end{array}$$

The same can be done to parts of the domain

# Domain weakening

**Intuition:** The following are equivalent

$$\begin{array}{ll} \forall_x [x \in D : A(x)] & \text{and} \quad \forall_x [x \in D \Rightarrow A(x)] \\ \exists_x [x \in D : A(x)] & \text{and} \quad \exists_x [x \in D \wedge A(x)] \end{array}$$

The same can be done to parts of the domain

## Domain weakening

$$\begin{array}{ll} \forall_x [P \wedge Q : R] & \stackrel{val}{=} \forall_x [P : Q \Rightarrow R] \\ \exists_x [P \wedge Q : R] & \stackrel{val}{=} \exists_x [P : Q \wedge R] \end{array}$$

# Domain weakening

**Intuition:** The following are equivalent

$$\begin{array}{ll} \forall_x [x \in D : A(x)] & \text{and} \quad \forall_x [x \in D \Rightarrow A(x)] \\ \exists_x [x \in D : A(x)] & \text{and} \quad \exists_x [x \in D \wedge A(x)] \end{array}$$

The same can be done to parts of the domain

Domain weakening

$$\begin{array}{l} \forall_x [P \wedge Q : R] \stackrel{val}{=} \forall_x [P : Q \Rightarrow R] \\ \exists_x [P \wedge Q : R] \stackrel{val}{=} \exists_x [P : Q \wedge R] \end{array}$$

$$P \wedge Q \stackrel{val}{\models} P$$

# De Morgan with quantifiers

## De Morgan

$$\neg \forall x [P : Q] \stackrel{val}{=} \exists x [P : \neg Q]$$
$$\neg \exists x [P : Q] \stackrel{val}{=} \forall x [P : \neg Q]$$

# De Morgan with quantifiers

## De Morgan

$$\neg \forall x [P : Q] \stackrel{val}{=} \exists x [P : \neg Q]$$
$$\neg \exists x [P : Q] \stackrel{val}{=} \forall x [P : \neg Q]$$

not for all = at least for one not

not exists = for all not

# De Morgan with quantifiers

## De Morgan

$$\neg \forall x [P : Q] \stackrel{val}{=} \exists x [P : \neg Q]$$
$$\neg \exists x [P : Q] \stackrel{val}{=} \forall x [P : \neg Q]$$

not for all = at least for one not

not exists = for all not

Hence:  $\neg \forall = \exists \neg$  and  $\neg \exists = \forall \neg$

# De Morgan with quantifiers

## De Morgan

$$\neg \forall x [P : Q] \stackrel{val}{=} \exists x [P : \neg Q]$$

$$\neg \exists x [P : Q] \stackrel{val}{=} \forall x [P : \neg Q]$$

not for all = at least for one not

not exists = for all not

Hence:  $\neg \forall = \exists \neg$  and  $\neg \exists = \forall \neg$

It holds further that:

$$\neg \forall x \neg = \exists x \neg \neg = \exists x$$

$$\neg \exists x \neg = \forall x \neg \neg = \forall x$$

# Substitution

meta rule

Simple

$$\frac{\phi \stackrel{val}{=} \psi}{\phi[\xi/P] \stackrel{val}{=} \psi[\xi/P]}$$

Sequential

$$\frac{\phi \stackrel{val}{=} \psi}{\phi[\xi/P][\eta/Q] \stackrel{val}{=} \psi[\xi/P][\eta/Q]}$$

Simultaneous

$$\frac{\phi \stackrel{val}{=} \psi}{\phi[\xi/P, \eta/Q] \stackrel{val}{=} \psi[\xi/P, \eta/Q]}$$

EVERY occurrence of  
P is substituted!

holds also for  
quantified formulas!

# Substitution

meta rule

Simple

$$\frac{\phi \stackrel{val}{=} \psi}{\phi[\xi/P] \stackrel{val}{=} \psi[\xi/P]}$$

Sequential

$$\frac{\phi \stackrel{val}{=} \psi}{\phi[\xi/P][\eta/Q] \stackrel{val}{=} \psi[\xi/P][\eta/Q]}$$

Simultaneous

$$\frac{\phi \stackrel{val}{=} \psi}{\phi[\xi/P, \eta/Q] \stackrel{val}{=} \psi[\xi/P, \eta/Q]}$$

EVERY occurrence of  
P is substituted!

# The rule of Leibniz

Leibniz

$$\phi \stackrel{val}{=} \psi$$

---

$$C[\phi] \stackrel{val}{=} C[\psi]$$

meta rule

formula that has  
 $\phi$  as a sub formula

single occurrence is  
replaced!

holds also for  
quantified formulas!

# The rule of Leibniz

meta rule

Leibniz

$$\phi \stackrel{val}{=} \psi$$

---

$$C[\phi] \stackrel{val}{=} C[\psi]$$

formula that has  
 $\phi$  as a sub formula

single occurrence is  
replaced!

# Other equivalences with quantifiers

## Exchange trick

$$\forall_x [P:Q] \stackrel{val}{=} \forall_x [\neg Q:\neg P]$$

$$\exists_x [P:Q] \stackrel{val}{=} \exists_x [Q:P]$$

# Other equivalences with quantifiers

## Exchange trick

$$\forall_x [P:Q] \stackrel{val}{=} \forall_x [\neg Q:\neg P]$$

$$\exists_x [P:Q] \stackrel{val}{=} \exists_x [Q:P]$$

No wonder as

$$\forall_x [P:Q] \stackrel{val}{=} \forall_x [P \Rightarrow Q]$$

$$\exists_x [P:Q] \stackrel{val}{=} \exists_x [P \wedge Q]$$

# Other equivalences with quantifiers

## Exchange trick

$$\forall_x [P:Q] \stackrel{val}{=} \forall_x [\neg Q:\neg P]$$

$$\exists_x [P:Q] \stackrel{val}{=} \exists_x [Q:P]$$

No wonder as

$$\forall_x [P:Q] \stackrel{val}{=} \forall_x [P \Rightarrow Q]$$

$$\exists_x [P:Q] \stackrel{val}{=} \exists_x [P \wedge Q]$$

## Term splitting

$$\forall_x [P:Q \wedge R] \stackrel{val}{=} \forall_x [P:Q] \wedge \forall_x [P:R]$$

$$\exists_x [P:Q \vee R] \stackrel{val}{=} \exists_x [P:Q] \vee \exists_x [P:R]$$

# Other equivalences with quantifiers

## Monotonicity of quantifiers

$$\forall_x [P:Q \Rightarrow R] \Rightarrow (\forall_x [P:Q] \Rightarrow \forall_x [P:R]) \stackrel{val}{=} T$$

$$\forall_x [P:Q \Rightarrow R] \Rightarrow (\exists_x [P:Q] \Rightarrow \exists_x [P:R]) \stackrel{val}{=} T$$

# Other equivalences with quantifiers

## Monotonicity of quantifiers

$$\forall_x [P:Q \Rightarrow R] \Rightarrow (\forall_x [P:Q] \Rightarrow \forall_x [P:R]) \stackrel{val}{=} T$$

$$\forall_x [P:Q \Rightarrow R] \Rightarrow (\exists_x [P:Q] \Rightarrow \exists_x [P:R]) \stackrel{val}{=} T$$

tautologies

# Other equivalences with quantifiers

## Monotonicity of quantifiers

$$\forall_x [P:Q \Rightarrow R] \Rightarrow (\forall_x [P:Q] \Rightarrow \forall_x [P:R]) \stackrel{val}{=} T$$

$$\forall_x [P:Q \Rightarrow R] \Rightarrow (\exists_x [P:Q] \Rightarrow \exists_x [P:R]) \stackrel{val}{=} T$$

tautologies

**Lemma EI:**  $P \stackrel{val}{=} Q$  iff  $P \Leftrightarrow Q$  is a tautology.

# Other equivalences with quantifiers

## Monotonicity of quantifiers

$$\forall_x [P:Q \Rightarrow R] \Rightarrow (\forall_x [P:Q] \Rightarrow \forall_x [P:R]) \stackrel{val}{=} T$$

$$\forall_x [P:Q \Rightarrow R] \Rightarrow (\exists_x [P:Q] \Rightarrow \exists_x [P:R]) \stackrel{val}{=} T$$

tautologies

**Lemma EI:**  $P \stackrel{val}{=} Q$  iff  $P \Leftrightarrow Q$  is a tautology.

**Lemma W4:**  $P \stackrel{val}{\models} Q$  iff  $P \Rightarrow Q$  is a tautology.

# Other equivalences with quantifiers

## Monotonicity of quantifiers

$$\forall x [P:Q \Rightarrow R] \Rightarrow (\forall x [P:Q] \Rightarrow \forall x [P:R]) \stackrel{val}{=} T$$

$$\forall x [P:Q \Rightarrow R] \Rightarrow (\exists x [P:Q] \Rightarrow \exists x [P:R]) \stackrel{val}{=} T$$

tautologies

**Lemma EI:**  $P \stackrel{val}{=} Q$  iff  $P \Leftrightarrow Q$  is a tautology.

**Lemma W4:**  $P \stackrel{val}{\models} Q$  iff  $P \Rightarrow Q$  is a tautology.

still hold (in predicate logic)

# Other equivalences with quantifiers

## Monotonicity of quantifiers

$$\forall_x [P:Q \Rightarrow R] \Rightarrow (\forall_x [P:Q] \Rightarrow \forall_x [P:R]) \stackrel{val}{=} T$$

$$\forall_x [P:Q \Rightarrow R] \Rightarrow (\exists_x [P:Q] \Rightarrow \exists_x [P:R]) \stackrel{val}{=} T$$

tautologies

**Lemma EI:**  $P \stackrel{val}{=} Q$  iff  $P \Leftrightarrow Q$  is a tautology.

**Lemma W4:**  $P \stackrel{val}{\models} Q$  iff  $P \Rightarrow Q$  is a tautology.

still hold (in predicate logic)

**Lemma W5:** If  $Q \stackrel{val}{\models} R$  then  $\forall_x [P:Q] \stackrel{val}{\models} \forall_x [P:R]$ .