Predicate Logic

Lectures 2–3 (Chapters 8-10)

TU/e Technische Universiteit Eindhoven University of Technology

Valid Logical Reasoning

If Alfred is a duck, then Alfred can swim
Alfred is a duck

Alfred can swim

a = 'Alfred is a duck'
b = 'Alfred can swim'

$$a \Rightarrow b \wedge a \stackrel{val}{\models} b$$

TU/e Technische Universiteit Eindhoven University of Technology

Invalid Logical Reasoning

If Alfred is a duck, then Alfred can swim
Alfred can swim

Alfred is a duck

a = 'Alfred is a duck'
b = 'Alfred can swim'

$$a \Rightarrow b \wedge b \stackrel{val}{\not\models} a$$

TU/e Technische Universiteit Eindhoven University of Technology

Valid Logical Reasoning?

Every animal with feathers is a bird
Alfred is an animal with feathers

Alfred is a bird

a = 'Every animal with feathers is a bird'
b = 'Alfred is an animal with feathers'
c = 'Alfred is a bird'

$$a \wedge b \stackrel{val}{\not\models} c$$

TU/e Technische Universiteit Eindhoven University of Technology

Predicate Logic

Propositional logic only describes how to reason about complete statements about things; it does not describe how to reason about things themselves.

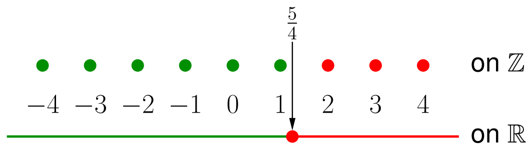
Predicate logic will extend propositional logical with:

- names for individuals;
- predicates to express properties of individuals;
- quantifiers to quantify over individuals.

Unary predicate (example)

Consider the statement $4m < 5$.

Whether the statement is true or false depends on the value of m:



$4m < 5$ is a unary predicate

Equivalence of predicates

Predicates $P(x)$ and $Q(x)$ are equivalent if

> ***for all*** *x we have that $P(x)$ evaluates to true* ***if, and only if,*** *$Q(x)$ evaluates to true.*

Example

$$4m < 5 \stackrel{val}{==} m < \frac{5}{4} \qquad \text{(on } \mathbb{R} \text{ and } \mathbb{Z}\text{)}$$
$$\stackrel{val}{==} m \leq 1 \qquad \text{(on } \mathbb{Z}\text{, but not on } \mathbb{R}\text{)} \; .$$

Universal quantification

We express the statement

> *"for all $m \in \mathbb{Z}$ it holds that $4m < 5$"*

as

$$\forall_m [m \in \mathbb{Z} : 4m < 5]$$

Universal quantifier $\forall_m$ turns the predicate $4m < 5$ into a proposition.

(This proposition is false; take, e.g., $m = 2, 3, \ldots$.)

Universal quantification

In general, if $P(x)$ and $Q(x)$ are predicates, then we write

$$\forall_x[\underbrace{P(x)}_{\text{domain}} : \underbrace{Q(x)}_{\text{predicate}}]$$

for

"all x satisfying P satisfy Q."

Existential quantification

We express the statement

"there exists $m \in \mathbb{Z}$ such that $4m < 5$"

as

$$\exists_m[m \in \mathbb{Z} : 4m < 5]$$

Existential quantifier $\exists_m$ turns the predicate $4m < 5$ into a proposition.

(This proposition is true; take $m = 1, 0, -1, \ldots$.)

Existential quantification

In general, if $P(x)$ and $Q(x)$ are predicates, then we write

$$\exists_x[\underbrace{P(x)}_{\text{domain}} : \underbrace{Q(x)}_{\text{predicate}}]$$

for

"there exists x satisfying P that also satisfies Q."

Domain of Quantification

What are the truth values associated with the following propositions?

- $\exists_x[x \in \mathbb{R} : x^2 = 1] \stackrel{val}{=\!=} \texttt{True}$
- $\exists_x[x \in \mathbb{R} \wedge x > 0 : x^2 = 1] \stackrel{val}{=\!=} \texttt{True}$
- $\exists_x[x \in \mathbb{R} \wedge x > 2 : x^2 = 1] \stackrel{val}{=\!=} \texttt{False}$
- $\forall_x[x \in \mathbb{R} : x^2 > 1] \stackrel{val}{=\!=} \texttt{False}$
- $\forall_x[x \in \mathbb{R} \wedge x > 0 : x^2 > 1] \stackrel{val}{=\!=} \texttt{False}$
- $\forall_x[x \in \mathbb{R} \wedge x > 2 : x^2 > 1] \stackrel{val}{=\!=} \texttt{True}$

One- or zero-element domains

Empty domain

$\forall_x[\texttt{False} : P] \stackrel{val}{==} \texttt{True}$

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

"All gnomes are green!"

One-element domain

$\forall_x[x = n : P] \stackrel{val}{==} P[n \text{ for } x]$

$\exists_x[x = n : P] \stackrel{val}{==} P[n \text{ for } x]$

Example: $\forall_m[m = 3 : 4 \cdot m < 5] \stackrel{val}{==} 4 \cdot 3 < 5.$

Domain weakening

NB: $P \wedge Q \stackrel{val}{\models} P$ (all x satisfying $P \wedge Q$ also satisfy P).

Examples:

$$\exists_x[x \in \mathbb{R} \wedge x > 0 : x^2 = 1]$$
$$\stackrel{val}{==} \exists_x[x \in \mathbb{R} : x > 0 \wedge x^2 = 1]$$

$$\forall_x[x \in \mathbb{Z} \wedge x > 1 : x^2 > 1]$$
$$\stackrel{val}{==} \forall_x[x \in \mathbb{Z} : x > 1 \Rightarrow x^2 > 1]$$

Domain weakening

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

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

Domain Splitting

$$\forall_x[x \leq -2 \vee x \geq 1 : x^2 > 1]$$
$$\stackrel{val}{==} \forall_x[x \leq -2 : x^2 > 1] \wedge \forall_x[x \geq 1 : x^2 > 1]$$

$$\exists_x[x \leq -2 \vee x \geq 1 : x^2 = 1]$$
$$\stackrel{val}{==} \exists_x[x \leq -2 : x^2 = 1] \vee \exists_x[x \geq 1 : x^2 = 1]$$

Domain Splitting

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

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

De Morgan for quantifiers

$\forall$ is generalised $\wedge$ }
$\exists$ is generalised $\vee$ } see book

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 all = there exists one for which not

not one = all not

So:

$\neg\forall = \exists\neg$; and

$\neg\exists = \forall\neg$.

We also have:

$\neg\forall\neg = \exists\neg\neg = \exists$; and

$\neg\exists\neg = \forall\neg\neg = \forall$.

Substitution

The Substitution Rule is valid for (quantified) predicates.

Example:

We have the following valid equivalence:

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

and hence, by the Substitution Rule, if we substitute Q for P, $\neg P$ for Q, and $R \Rightarrow S$ for R, then we get another valid equivalence:

$$\forall_x[\, Q \wedge \neg P : R \Rightarrow S \,] \stackrel{val}{=\!=} \forall_x[\, Q : \neg P \Rightarrow (\, R \Rightarrow S \,)] \ .$$

Leibniz

Leibniz's Rule is valid for (quantified) predicates.

Example

We have the following valid equivalence:

$$\exists_y[y = 2 : x \geq y] \stackrel{val}{=\!=} x \geq 2 \ ,$$

and hence, by Leibniz's Rule, we make another valid equivalence by applying it in a bigger context:

$$\forall_x[x \in \mathbf{D} : \exists_y[y = 2 : x \geq y] \,] \stackrel{val}{=\!=} \forall_x[x \in \mathbf{D} : x \geq 2 \,] \ .$$

Calculation with quantifiers (example)

We show with a calculation that

$$\forall_x[P \wedge R : S] \wedge \forall_x[Q \wedge R : S] \stackrel{val}{=\!=} \neg\exists_x[P \vee Q : \neg(R \Rightarrow S)]:$$

$\neg\exists_x[P \vee Q : \neg(R \Rightarrow S)]$

$\stackrel{val}{=\!=}$ { De Morgan }

$\forall_x[P \vee Q : \neg\neg(R \Rightarrow S)]$

$\stackrel{val}{=\!=}$ { Double Negation }

$\forall_x[P \vee Q : R \Rightarrow S]$

$\stackrel{val}{=\!=}$ { Domain Splitting }

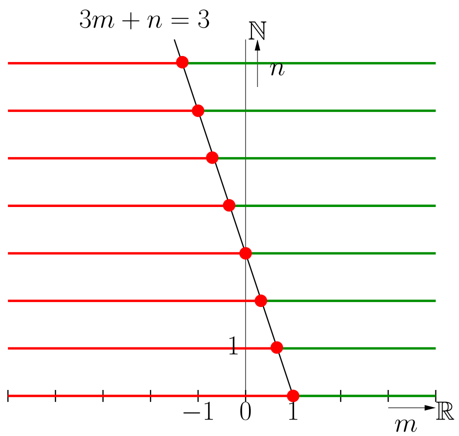
$\forall_x[P : R \Rightarrow S] \wedge \forall_x[Q : R \Rightarrow S]$

$\stackrel{val}{=\!=}$ { Domain Weakening (2×) }

$\forall_x[P \wedge R : S] \wedge \forall_x[Q \wedge R : S]$.

Binary predicates (example)

The statement $3m + n > 3$ is a binary predicate.



Equivalence of binary predicates

Predicates $P(x, y)$ and $Q(x, y)$ are equivalent if

> ***for all*** *x, y we have that $P(x, y)$ evaluates to true* ***if, and only if,*** *$Q(x, y)$ evaluates to true.*

Example

Let $x, y \in \mathbb{Z}$.
The predicates

$$\neg(x = 0 \Rightarrow y > 2)$$

and

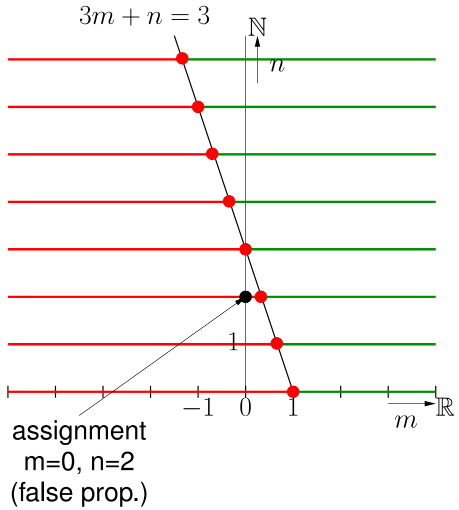
$$x = 0 \wedge y \leq 2$$

are equivalent.

$$\neg(x = 0 \Rightarrow y > 2)$$
$\stackrel{val}{==}$ { Implication }
$$\neg(\neg(x = 0) \vee y > 2)$$
$\stackrel{val}{==}$ { De Morgan }
$$\neg\neg(x = 0) \wedge \neg(y > 2)$$
$\stackrel{val}{==}$ { Double Negation, Mathematics }
$$x = 0 \wedge y \leq 2$$

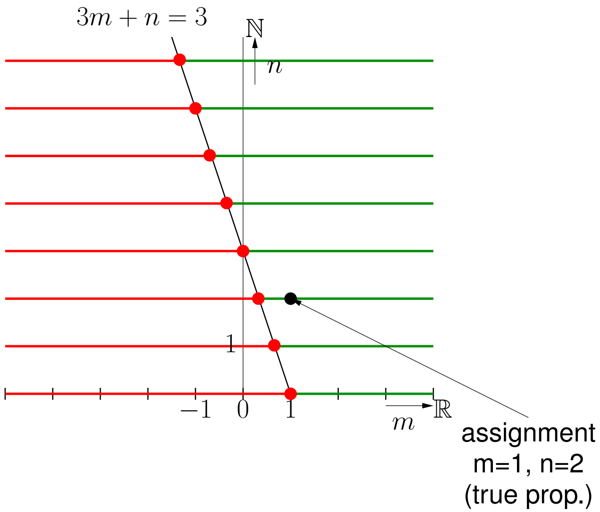
Binary predicates (example)

The statement $3m + n > 3$ is a binary predicate.



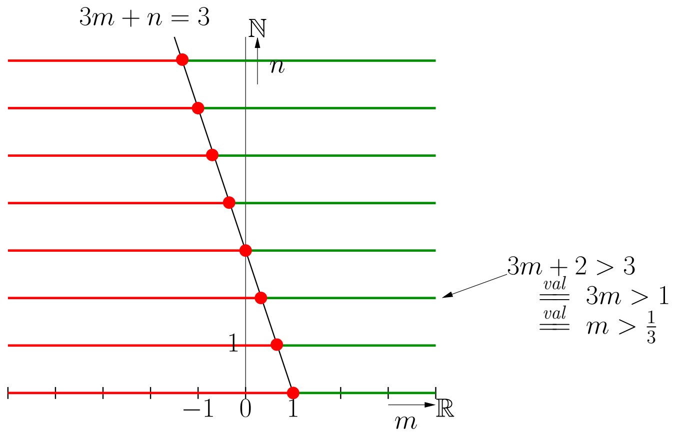
Binary predicates (example)

The statement $3m + n > 3$ is a binary predicate.



Binary predicates (example)

The statement $3m + n > 3$ is a binary predicate.



Renaming bound variables

Example:

$\forall_m [\, m \in \mathbb{N} : m + n > 6]$ (true if $n > 6$; false if $n \leq 6$)

$\overset{val}{=} \forall_k [\, k \in \mathbb{N} : k + n > 6]$ (true if $n > 6$; false if $n \leq 6$)

$\overset{val}{\neq} \forall_n [\, n \in \mathbb{N} : n + n > 6]$ (always false)

Bound variables may be renamed,
but the binding structure may not change!

Bound variable
$\forall_x[P : Q] \overset{val}{=} \forall_y[P[y \text{ for } x] : Q[y \text{ for } x]]$
$\exists_x[P : Q] \overset{val}{=} \exists_y[P[y \text{ for } x] : Q[y \text{ for } x]]$

provided that y does not occur in P or Q
(not even as part of $\forall_y$ or $\exists_y$)

Quantifying many-place predicates

We can turn the binary predicate $3m + n > 3$ into a proposition by quantification:

$$\underbrace{\forall_m[m \in \mathbb{R} : \underbrace{\exists_n[n \in \mathbb{N} : \underbrace{3m + n > 3}_{\text{binary predicate}}]}_{\text{unary predicate}}]}_{\text{proposition}}$$

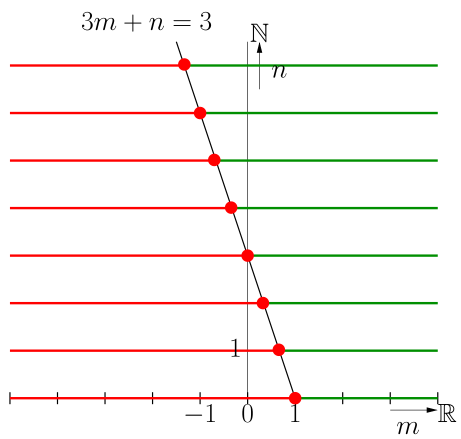
This proposition is true! Informal argument:

1. Consider an arbitrary $m_0 \in \mathbb{R}$.
2. Does there exist $n \in \mathbb{N}$ such that $3m_0 + n > 3$?
3. Yes! We can, for instance, take $n_0 = \lceil 3 - 3m_0 \rceil + 1$.

Quantifying many-place predicates

1. $\forall_m[m \in \mathbb{R} : \forall_n[n \in \mathbb{N} : 3m + n > 3]] \overset{val}{=}$ `False`
2. $\forall_m[m \in \mathbb{R} : \exists_n[n \in \mathbb{N} : 3m + n > 3]] \overset{val}{=}$ `True`
3. $\exists_m[m \in \mathbb{R} : \forall_n[n \in \mathbb{N} : 3m + n > 3]] \overset{val}{=}$ `True`
4. $\exists_m[m \in \mathbb{R} : \exists_n[n \in \mathbb{N} : 3m + n > 3]] \overset{val}{=}$ `True`
5. $\forall_n[n \in \mathbb{N} : \forall_m[m \in \mathbb{R} : 3m + n > 3]] \overset{val}{=}$ `False`
6. $\forall_n[n \in \mathbb{N} : \exists_m[m \in \mathbb{R} : 3m + n > 3]] \overset{val}{=}$ `True`
7. $\exists_n[n \in \mathbb{N} : \forall_m[m \in \mathbb{R} : 3m + n > 3]] \overset{val}{=}$ `False`
8. $\exists_n[n \in \mathbb{N} : \exists_m[m \in \mathbb{R} : 3m + n > 3]] \overset{val}{=}$ `True`

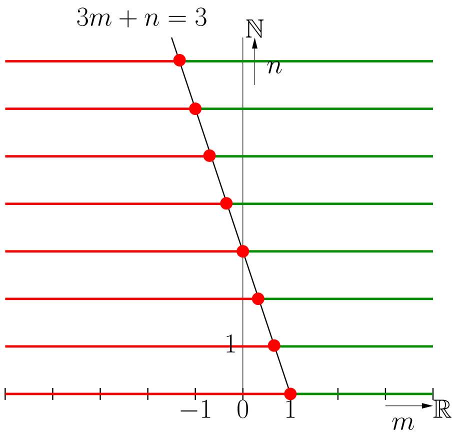
$\forall_m[m \in \mathbb{R} : \forall_n[n \in \mathbb{N} : 3m + n > 3]] \overset{val}{=}$ `False`



Explanation:

'Not all vertical lines through the m-axis are entirely green.'

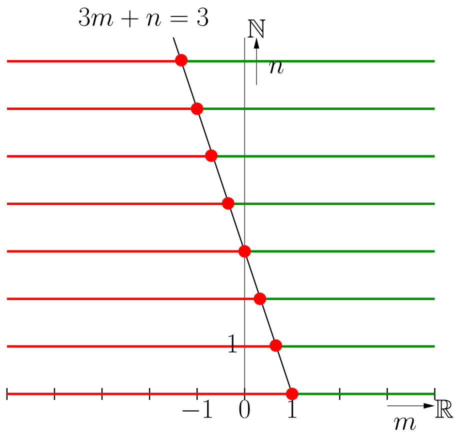
$\forall_m[m \in \mathbb{R} : \exists_n[n \in \mathbb{N} : 3m+n>3]] \overset{val}{=\!=} \texttt{True}$



Explanation:

"On every vertical line through the m-axis at least one point is green."

$\exists_m[m \in \mathbb{R} : \forall_n[n \in \mathbb{N} : 3m+n>3]] \overset{val}{=\!=} \texttt{True}$

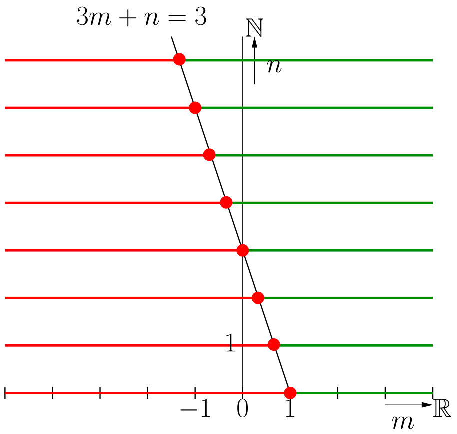


Explanation:

"There exists a vertical line through the m-axis (e.g., the line $m=2$) such that all points are green."

NB: we only consider points whose n-coordinate is in $\mathbb{N}$

$\exists_m[m \in \mathbb{R} : \exists_n[n \in \mathbb{N} : 3m+n>3]] \overset{val}{=\!=} \texttt{True}$

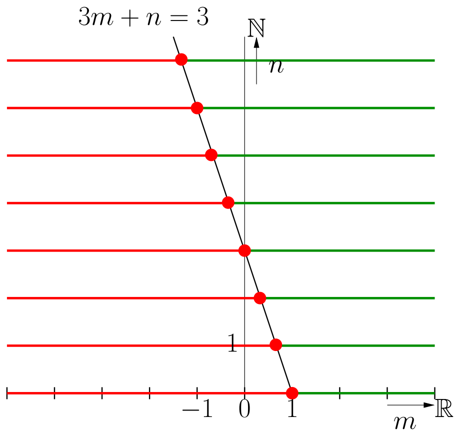


Explanation:

"There exists a vertical line through the m-axis with a green point."

(In fact, we have already seen that *all* vertical lines have this property.)

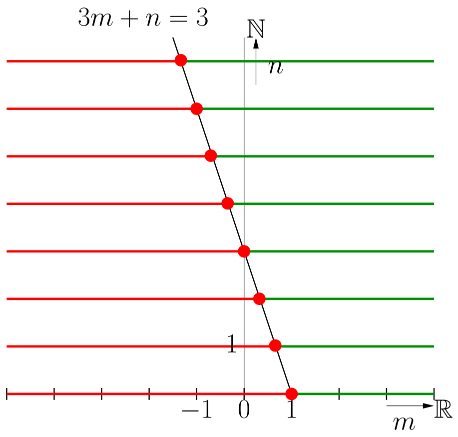
$\forall_n[n \in \mathbb{N} : \forall_m[m \in \mathbb{R} : 3m+n>3]] \overset{val}{=\!=} \texttt{False}$



Explanation:

"Not all horizontal lines through the n-axis are entirely green."

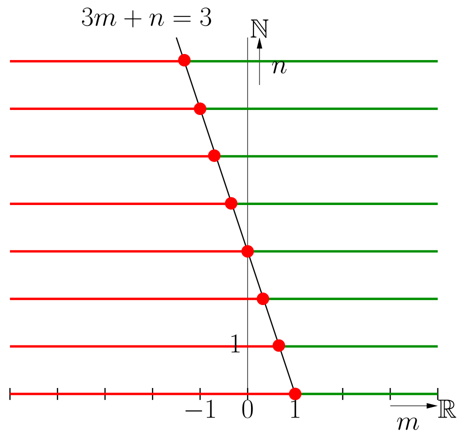
$\forall_n[n \in \mathbb{N} : \exists_m[m \in \mathbb{R} : 3m + n > 3]] \overset{val}{==} \texttt{True}$



Explanation:

"All horizontal lines through the n-axis have a green point."

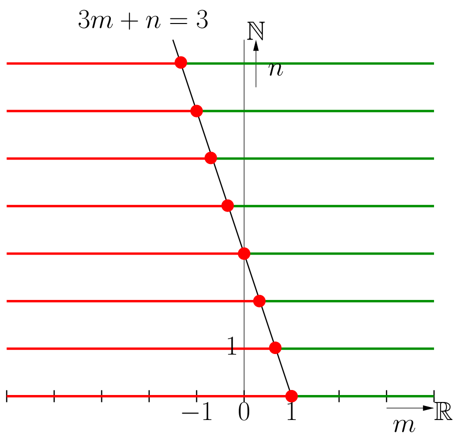
$\exists_n[n \in \mathbb{N} : \forall_m[m \in \mathbb{R} : 3m + n > 3]] \overset{val}{==} \texttt{False}$



Explanation:

"There is no horizontal line through the n-axis that is entirely green."

$\exists_n[n \in \mathbb{N} : \exists_m[m \in \mathbb{R} : 3m + n > 3]] \overset{val}{==} \texttt{True}$



Explanation:

"There exists a horizontal line through the n-axis with a green point."

A word on notation

We often write

$\forall_x[P]$ for $\forall_x[\texttt{True} : P]$

(i.e., we omit the domain when it is $\texttt{True}$).

Furthermore, we will often write

$\forall_m \exists_n[(m, n) \in \mathbb{R} \times \mathbb{N} : 3m + n > 3]]$
instead of $\forall_m[m \in \mathbb{R} : \exists_n[n \in \mathbb{N} : 3m + n > 3]]$

$\forall_{m,n}[(m, n) \in \mathbb{R} \times \mathbb{N} : 3m + n > 3]$
instead of $\forall_m \forall_n[(m, n) \in \mathbb{R} \times \mathbb{N} : 3m + n > 3]$

NB: we only contract multiple occurrences of *the same* quantifier (i.e., $\forall_m \forall_n$ becomes $\forall_{m,n}$, but $\forall_m \exists_n$ is not contracted).