Limitations of propositional logic

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

Consider an instance of Aristotle’s syllogism:

- Some chickens cannot fly
- All chickens are birds
- Some birds cannot fly

Such a reasoning cannot be described in propositional logic.

Other example:

Every player except the winner loses a match.

Unary predicate (example)

Consider the statement $4m < 5$.

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

\[
\begin{array}{cccccccc}
\text{on } \mathbb{Z} & -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \\
\text{on } \mathbb{R} & & & & & & & & & \\
\end{array}
\]

NB:

- $4m < 5$ $\equiv$ $m < \frac{5}{4}$ (on $\mathbb{R}$ and $\mathbb{Z}$)
- $\equiv$ $m \leq 1$ (on $\mathbb{Z}$, but not on $\mathbb{R}$).

$4m < 5$ is a unary predicate

Binary predicates (example)

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

\[
\begin{array}{cccccccc}
\text{on } \mathbb{Z} & -1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
\text{on } \mathbb{R} & & & & & & & & & \\
\end{array}
\]
Binary predicates (example)

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

With $m=0, n=2$ (false prop.)

We express the statement “for all $m \in \mathbb{Z}$ it holds that $4m < 5$” as

$$\forall m \in \mathbb{Z} : 4m < 5$$

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

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

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

$$\forall x [P(x) : Q(x)]$$

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 $m = 0$ or $m = 1$.)

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

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

for “there exists $x$ satisfying $P$ that also satisfies $Q$.”

Quantifying many-place predicates

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

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

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 $n_0 = \lceil 3 - 3m_0 \rceil + 1$.

Explanation:

‘Not all vertical lines through the $m$-axis are entirely green.”
∀₀∈ℝ : ∃₁∈ℕ : 3₀ + ₁ > 3 \quad \text{val} \quad \text{True}

Explanation:
“On every vertical line through the m-axis at least one point is green.”

∃ₚ∈ℝ : ∀₁∈ℕ : 3ₚ + ₁ > 3 \quad \text{val} \quad \text{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 N

∃ₚ∈ℝ : ∃₁∈ℕ : 3ₚ + ₁ > 3 \quad \text{val} \quad \text{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.)

∀₁∈ℕ : ∀ₚ∈ℝ : 3ₚ + ₁ > 3 \quad \text{val} \quad \text{False}

Explanation:
“Not all horizontal lines through the n-axis are entirely green.”
∀ₙ[ₙ ∈ ℤ : ∃ₘ[ₘ ∈ ℜ : 3ₘ + ₙ > 3]] \text{ val } \text{ True}

Explanation:
“All horizontal lines through the \( n \)-axis have a green point.”

∃ₙ[ₙ ∈ ℤ : ∀ₘₙ[ₘ ∈ ℜ : 3ₘₙ + ₙ > 3]] \text{ val } \text{ False}

Explanation:
“There is no horizontal line through the \( n \)-axis that is entirely green.”

∃ₙ[ₙ ∈ ℤ : ∃ₘₙ[ₘ ∈ ℜ : 3ₘₙ + ₙ > 3]] \text{ val } \text{ True}

Explanation:
“There exists a horizontal line through the \( n \)-axis with a green point.”

A word on notation

We often write
\[ ∀ₓ[P] \quad \text{for} \quad ∀ₓ[\text{True} : P] \]
(i.e., we omit the domain when it is \( \text{True} \)).

Furthermore, we will often write
\[ ∀ₘ∃ₙ[(ₘ, ₙ) ∈ ℜ × ℤ : 3ₘₙ + ₙ > 3] \]
\[ \text{instead of} \quad ∀ₘₙ[ₘ ∈ ℜ : ∃ₙ[ₙ ∈ ℤ : 3ₘₙ + ₙ > 3]] \]
\[ ∀ₘₙ[(ₘ, ₙ) ∈ ℜ × ℤ : 3ₘₙ + ₙ > 3] \]
\[ \text{instead of} \quad ∀ₘ∀ₙ[(ₘ, ₙ) ∈ ℜ × ℤ : 3ₘₙ + ₙ > 3] \]

NB: we only contract multiple occurrences of \( \text{the same} \) quantifier (i.e., \( ∀ₘ∃ₙ \) becomes \( ∀ₘₙ \), but \( ∀ₘ∀ₙ \) is not contracted).
Exercise

Let \( P \) be the set of all tennis players; Serena is one of them.
Write \( p \neq q \) for '\( p \) and \( q \) are different players'.
Let \( M \) be the set of all matches.
Write \( L(p, m) \) for '\( p \) loses match \( m \)'.

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

Every player except Serena loses a match.

Solution:

\[
\forall x \left[ x \in P \land x \neq \text{Serena} : \exists y \left[ y \in M : L(x, y) \right] \right]
\]

Equivalence of 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, 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

\[
x = 0 \land y \leq 2
\]

are equivalent.

Renaming bound variables

Example:

\[
\forall m \left[ m \in \mathbb{N} : m + n > 6 \right]
\]

(true if \( n > 6 \); false if \( n \leq 6 \))

\[
\forall k \left[ k \in \mathbb{N} : k + n > 6 \right]
\]

(true if \( n > 6 \); false if \( n \leq 6 \))

\[
\forall m \neq \forall n \left[ n \in \mathbb{N} : n + n > 6 \right]
\]

(always false)

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

\[
\forall x \left[ P \land Q : Q[y \ for \ x] \right] \quad \forall y \left[ P[y \ for \ x] : Q[y \ for \ x] \right]
\]

provided that \( y \) does not occur in \( P \) or \( Q \)
(not even as part of \( \forall y \) or \( \exists y \))

Domain Splitting

\[
\forall x \left[ x \leq -2 \lor x \geq 1 : x^2 \geq 1 \right] \quad \forall y \left[ y \leq -2 : y^2 \geq 1 \right] \land \forall x \left[ x \geq 1 : x^2 \geq 1 \right]
\]

\[
\exists x \left[ x \leq -2 \lor x \geq 1 : x^2 = 1 \right] \quad \exists y \left[ y \leq -2 : y^2 = 1 \right] \lor \exists x \left[ x \geq 1 : x^2 = 1 \right]
\]

Domain Splitting

\[
\forall x \left[ P \lor Q : R \right] \quad \forall x \left[ P : R \right] \land \forall x \left[ Q : R \right]
\]

\[
\exists x \left[ P \lor Q : R \right] \quad \exists x \left[ P : R \right] \lor \exists x \left[ Q : R \right]
\]
One-or zero-element domains

One-element domain
\[ \forall x [x = n : P] \equiv P[n \text{ for } x] \]
\[ \exists x [x = n : P] \equiv P[n \text{ for } x] \]

Example: \[ \forall m [m = 3 : 4 \cdot m < 5] \equiv 4 \cdot 3 < 5. \]

Empty domain
\[ \forall x [\text{False} : P] \equiv \text{True} \]
\[ \exists x [\text{False} : P] \equiv \text{False} \]

“All gnomes are green!”

Domain weakening

NB: \[ P \land Q \equiv P \] (all \( x \) satisfying \( P \land Q \) also satisfy \( P \)).

Examples:
\[ \forall x [x \in \mathbb{Z} \land x > 1 : x^2 > 1] \]
\[ \equiv \forall x [x \in \mathbb{Z} : x > 1 \Rightarrow x^2 > 1] \]
\[ \exists k [0 \leq x \land x < 4 : x^2 > 8] \]
\[ \equiv \exists x [0 \leq x : x < 4 \land x^2 > 8] \]

Domain weakening
\[ \forall x [P \land Q : R] \equiv \forall x [P : R] \]
\[ \exists x [P \land Q : R] \equiv \exists x [P : Q \land R] \]

De Morgan for quantifiers

\( \forall \) is generalised \( \land \)
\( \exists \) is generalised \( \lor \)

De Morgan
\[ \neg \forall x [P : Q] \equiv \exists x [P : \neg Q] \]
\[ \neg \exists x [P : Q] \equiv \forall x [P : \neg Q] \]

not all = there exists one for which not
not one = all not

So:
\[ \neg \forall = \exists \neg; \text{ and } \quad \neg \exists = \forall \neg. \]

We also have:
\[ \neg \forall = \exists \neg; \text{ and } \quad \neg \exists = \forall \neg. \]

Substitution

The Substitution Rule is valid for (quantified) predicates.

Example:
We have the following valid equivalence:
\[ \forall x [P \land Q : R] \equiv \forall x [P : Q \Rightarrow R] \]

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:
\[ \forall x [Q \land \neg P : R \Rightarrow S] \equiv \forall x [Q : \neg P \Rightarrow (R \Rightarrow S)] . \]
Leibniz’s Rule is valid for (quantified) predicates.

Example
We have the following valid equivalence:

\[ \exists_y [y = 2 : x \geq y] \quad \vdash \quad x \geq 2 \]

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

\[ \forall_x [x \in D : \exists_y [y = 2 : x \geq y]] \quad \vdash \quad \forall_x [x \in D : x \geq 2] \]

Calculation with quantifiers (example)

We show with a calculation that

\[ \forall_x [P \land R : S] \land \forall_x [Q \land R : S] \quad \vdash \quad \neg \exists_x [P \lor Q : \neg (R \Rightarrow S)] \]

\[ \quad \vdash \quad \neg \exists_x [P \lor Q : (R \Rightarrow S)] \]

\[ \quad \vdash \quad \{ \text{De Morgan} \} \]

\[ \quad \forall_x [P \lor Q : R \Rightarrow S] \]

\[ \quad \vdash \quad \{ \text{Double Negation} \} \]

\[ \quad \forall_x [P \lor Q : S] \land \forall_x [Q : R \Rightarrow S] \]

\[ \quad \vdash \quad \{ \text{Domain Weakening (2 ? ?)} \} \]

\[ \quad \forall_x [P \land R : S] \land \forall_x [Q \land R : S] \]