Inductive and Coinductive Reasoning with Isabelle/HOL – Introduction

Andrei Popescu

University of Sheffield

Part of the POPL 2023 tutorial
"Isabelle/HOL: Foundations, Induction, and Coinduction"
given jointly with Dmitriy Traytel

16 January, 2023

Inductive and Coinductive Reasoning with or without Isabelle/HOL – Introduction

Andrei Popescu

University of Sheffield

Part of the POPL 2023 tutorial
"Isabelle/HOL: Foundations, Induction, and Coinduction"
given jointly with Dmitriy Traytel

16 January, 2023

Motivation

What is a (well-formed) definition?

1. Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x = x + 1$

What is a (well-formed) definition?

1. Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x = x + 1$
2. Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x =$ if $(x = 0)$ then 1 else $f(x - 1) * 2$

What is a (well-formed) definition?

1. Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x = x + 1$
2. Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x =$ if $(x = 0)$ then 1 else $f(x - 1) * 2$
3. Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x = 1 + f\ x$

What is a (well-formed) definition?

1. Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x = x + 1$
2. Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x =$ if $(x = 0)$ then 1 else $f(x - 1) * 2$
3. Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x = 1 + f\ x$

Does there exist a unique function f with that property?

What is a (well-formed) definition?

1. Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x = x + 1$ ✓
2. Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x =$ if $(x = 0)$ then 1 else $f(x - 1) * 2$ ✓
3. Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x = 1 + f\ x$ ✗

Does there exist a unique function f with that property?

What is a (well-formed) definition?

Assume Stream is the set of all infinite sequences $[x_0, x_1, \ldots]$ of natural numbers and $x \# xs$ means consing (prepending) number x to stream xs.

What is a (well-formed) definition?

Assume Stream is the set of all infinite sequences $[x_0, x_1, \ldots]$ of natural numbers and $x\#xs$ means consing (prepending) number x to stream xs.

1. Define zeros : Stream by
zeros = $0 \#$ zeros

What is a (well-formed) definition?

Assume Stream is the set of all infinite sequences $[x_0, x_1, \ldots]$ of natural numbers and $x \# xs$ means consing (prepending) number x to stream xs.

1. Define zeros : Stream by
zeros = $0 \#$ zeros

2. Define plus : Stream → Stream → Stream by
plus $(x \# xs)$ $(y \# ys)$ = $(x + y) \#$ (plus xs ys)

What is a (well-formed) definition?

Assume Stream is the set of all infinite sequences $[x_0, x_1, \ldots]$ of natural numbers and $x\#xs$ means consing (prepending) number x to stream xs.

1. Define zeros : Stream by
zeros = $0 \#$ zeros

2. Define plus : Stream → Stream → Stream by
plus $(x\#xs)$ $(y\#ys)$ = $(x + y) \#$ (plus xs ys)

3. Define f : Stream → Stream → Stream by
f $(x\#xs)$ $(y\#ys)$ = $(x * y) \#$ (plus (f $(x\#xs)$ zeros) (f xs $(y\#ys)$)).

What is a (well-formed) definition?

Assume Stream is the set of all infinite sequences $[x_0, x_1, \ldots]$ of natural numbers and $x\#xs$ means consing (prepending) number x to stream xs.

1. Define zeros : Stream by
zeros = $0 \#$ zeros

2. Define plus : Stream → Stream → Stream by
plus $(x\#xs)$ $(y\#ys)$ = $\ldots$

3. Define f : Stream → Stream → Stream by
f $(x\#xs)$ $(y\#ys)$ = $(x * y)$ $\#$ (plus (f $(x\#xs)$ zeros) (f xs $(y\#ys)$)).

What is a (well-formed) definition?

Assume Stream is the set of all infinite sequences $[x_0, x_1, \ldots]$ of natural numbers and $x\#xs$ means consing (prepending) number x to stream xs.

1. Define zeros : Stream by
zeros = $0 \#$ zeros

2. Define plus : Stream → Stream → Stream by
plus $(x\#xs)$ $(y\#ys)$ = $(x + y) \#$ (plus xs ys)

3. Define f : Stream → Stream → Stream by
f $(x\#xs)$ $(y\#ys)$ = $(x * y) \#$ (plus (f $(x\#xs)$ zeros) (f xs $(y\#ys)$)).

What is a (well-formed) definition?

Assume Stream is the set of all infinite sequences $[x_0, x_1, \ldots]$ of natural numbers and $x\#xs$ means consing (prepending) number x to stream xs.

1. Define zeros : Stream by
zeros = $0 \#$ zeros

2. Define plus : Stream → Stream → Stream by
plus $(x\#xs)$ $(y\#ys)$ = $(x + y) \#$ (plus xs ys)

3. Define f : Stream → Stream → Stream by
f $(x\#xs)$ $(y\#ys)$ = $(x * y) \#$ (plus (f $(x\#xs)$ zeros) (f xs $(y\#ys)$)).

4. Define $f : \mathbb{N} \to$ Stream

$$f\ x = \begin{cases} \text{zeros} & \text{if } x \leq 1 \\ f\ (x\,/\,2) & \text{if } x > 1 \text{ and } x \text{ even} \\ x \# f\ (3 * x + 1) & \text{if } x > 1 \text{ and } x \text{ odd} \end{cases}$$

What is a (well-formed) definition?

Assume Stream is the set of all infinite sequences $[x_0, x_1, \ldots]$ of natural numbers and $x\#xs$ means consing (prepending) number x to stream xs.

1. Define zeros : Stream by
zeros = $0 \#$ zeros

2. Define plus : Stream → Stream → Stream by
plus $(x\#xs)$ $(y\#ys)$ = $(x + y) \#$ (plus xs ys)

3. Define f : Stream → Stream → Stream by
f $(x\#xs)$ $(y\#ys)$ = $(x * y) \#$ (plus (f $(x\#xs)$ zeros) (f xs $(y\#ys)$)).

4. Define $f : \mathbb{N} \to$ Stream

$$f\ x = \begin{cases} \text{zeros} & \text{if } x \leq 1 \\ f\ (x\,/\,2) & \text{if } x > 1 \text{ and } x \text{ even} \\ x \# f\ (3 * x + 1) & \text{if } x > 1 \text{ and } x \text{ odd} \end{cases}$$

Does there exist a unique item (zeros, plus, f) with that property?

What is a (well-formed) definition?

Assume Stream is the set of all infinite sequences $[x_0, x_1, \ldots]$ of natural numbers and $x \# xs$ means consing (prepending) number x to stream xs.

1. Define zeros : Stream by ✓
zeros = $0 \#$ zeros

2. Define plus : Stream → Stream → Stream by ✓
plus $(x\#xs)$ $(y\#ys)$ = $(x + y) \#$ (plus xs ys)

3. Define f : Stream → Stream → Stream by ✓
f $(x\#xs)$ $(y\#ys)$ = $(x * y) \#$ (plus (f $(x\#xs)$ zeros) (f xs $(y\#ys)$)).

4. Define $f : \mathbb{N} \to$ Stream ✓

$$f\ x = \begin{cases} \text{zeros} & \text{if } x \le 1 \\ f\ (x\,/\,2) & \text{if } x > 1 \text{ and } x \text{ even} \\ x \# f\ (3 * x + 1) & \text{if } x > 1 \text{ and } x \text{ odd} \end{cases}$$

Does there exist a unique item (zeros, plus, f) with that property?

What is a (well-formed) definition?

Assume Stream is the set of all infinite sequences $[x_0, x_1, \ldots]$ of natural numbers and $x \# xs$ means consing (prepending) number x to stream xs.

1. Define zeros : Stream by ✓
zeros = $0 \#$ zeros

2. Define plus : Stream → Stream → Stream by ✓
plus $(x \# xs)$ $(y \# ys)$ = $\ldots$

3. Define f : Stream → Stream → Stream by ✓
$f\ (x \# xs)\ (y \# ys) = (x * y) \# (\text{plus}\ (f\ (x \# xs)\ \text{zeros})\ (f\ xs\ (y \# ys)))$.

4. Define $f : \mathbb{N} \to$ Stream ✓

$$f\ x = \begin{cases} \text{zeros} & \text{if } x \leq 1 \\ f\ (x\,/\,2) & \text{if } x > 1 \text{ and } x \text{ even} \\ x \# f\ (3 * x + 1) & \text{if } x > 1 \text{ and } x \text{ odd} \end{cases}$$

Does there exist a unique item (zeros, plus, f) with that property?

What is a (well-formed) definition?

Assume Stream is the set of all infinite sequences $[x_0, x_1, \ldots]$ of natural numbers and $x \# xs$ means consing (prepending) number x to stream xs.

1. Define zeros : Stream by ✓
zeros = $0 \#$ zeros

2. Define plus : Stream → Stream → Stream by ✓
plus $(x \# xs)$ $(y \# ys)$ = $\ldots$

3. Define f : Stream → Stream → Stream by **?**
$f\ (x \# xs)\ (y \# ys) = (x * y) \#$ (plus $(f\ (x \# xs)$ zeros$)\ (f\ xs\ (y \# ys)))$.

4. Define $f : \mathbb{N} \to$ Stream ✓

$$f\ x = \begin{cases} \text{zeros} & \text{if } x \leq 1 \\ f\ (x \,/\, 2) & \text{if } x > 1 \text{ and } x \text{ even} \\ x \# f\ (3 * x + 1) & \text{if } x > 1 \text{ and } x \text{ odd} \end{cases}$$

Does there exist a unique item (zeros, plus, f) with that property?

What is a (well-formed) definition?

1. Define $A \subseteq \mathbb{N}$ by $A = \{x \in \mathbb{N} \mid \exists y.\, x = y * 2\}$

What is a (well-formed) definition?

1. Define $A \subseteq \mathbb{N}$ by $A = \{x \in \mathbb{N} \mid \exists y.\ x = y * 2\}$
2. Define $A \subseteq \mathbb{N}$ by the following rules:
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \in A$ then $n + 2 \in A$

What is a (well-formed) definition?

1. Define $A \subseteq \mathbb{N}$ by $A = \{x \in \mathbb{N} \mid \exists y.\, x = y * 2\}$
2. Define $A \subseteq \mathbb{N}$ by the following rules:
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \in A$ then $n + 2 \in A$
3. Define $A \subseteq \mathbb{N}$ by the following rules:
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \notin A$ then $n + 2 \in A$

What is a (well-formed) definition?

1. Define $A \subseteq \mathbb{N}$ by $A = \{x \in \mathbb{N} \mid \exists y.\, x = y * 2\}$

2. Define $A \subseteq \mathbb{N}$ by the following rules:
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \in A$ then $n + 2 \in A$

3. Define $A \subseteq \mathbb{N}$ by the following rules:
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \notin A$ then $n + 2 \in A$

4. Define $A \subseteq$ Stream by the following rules:
   - zeros $\in A$
   - If $x \in \mathbb{N}$, x even and $xs \in A$, then $x \# xs \in A$

What is a (well-formed) definition?

1. Define $A \subseteq \mathbb{N}$ by $A = \{x \in \mathbb{N} \mid \exists y.\, x = y * 2\}$

2. Define $A \subseteq \mathbb{N}$ by the following rules:

   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \in A$ then $n + 2 \in A$

3. Define $A \subseteq \mathbb{N}$ by the following rules:

   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \notin A$ then $n + 2 \in A$

4. Define $A \subseteq$ Stream by the following rules:

   - zeros $\in A$
   - If $x \in \mathbb{N}$, x even and $xs \in A$, then $x \# xs \in A$

Does there exist a unique set A with that property?

What is a (well-formed) definition?

1. Define $A \subseteq \mathbb{N}$ by $A = \{x \in \mathbb{N} \mid \exists y.\, x = y * 2\}$ ✓

2. Define $A \subseteq \mathbb{N}$ by the following rules:
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \in A$ then $n + 2 \in A$

3. Define $A \subseteq \mathbb{N}$ by the following rules:
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \notin A$ then $n + 2 \in A$

4. Define $A \subseteq$ Stream by the following rules:
   - zeros $\in A$
   - If $x \in \mathbb{N}$, x even and $xs \in A$, then $x \# xs \in A$

Does there exist a unique set A with that property?

What is a (well-formed) definition?

1. Define $A \subseteq \mathbb{N}$ by $A = \{x \in \mathbb{N} \mid \exists y.\, x = y * 2\}$ ✓

2. Define $A \subseteq \mathbb{N}$ inductively by the following rules:
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \in A$ then $n + 2 \in A$

3. Define $A \subseteq \mathbb{N}$ inductively by the following rules:
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \notin A$ then $n + 2 \in A$

4. Define $A \subseteq$ Stream inductively by the following rules:
   - zeros $\in A$
   - If $x \in \mathbb{N}$, x even and $xs \in A$, then $x \# xs \in A$

Does there exist a unique set A with that property?

If not, maybe we need to complete the definition – be more specific!

What is a (well-formed) definition?

1. Define $A \subseteq \mathbb{N}$ by $A = \{x \in \mathbb{N} \mid \exists y.\, x = y * 2\}$ ✓

2. Define $A \subseteq \mathbb{N}$ inductively by the following rules: ✓
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \in A$ then $n + 2 \in A$

3. Define $A \subseteq \mathbb{N}$ inductively by the following rules: x
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \notin A$ then $n + 2 \in A$

4. Define $A \subseteq$ Stream inductively by the following rules: ✓
   - zeros $\in A$
   - If $x \in \mathbb{N}$, x even and $xs \in A$, then $x \# xs \in A$

Does there exist a unique set A with that property?

If not, maybe we need to complete the definition – be more specific!

What is a (well-formed) definition?

1. Define $A \subseteq \mathbb{N}$ by $A = \{x \in \mathbb{N} \mid \exists y.\, x = y * 2\}$ ✓

2. Define $A \subseteq \mathbb{N}$ inductively by the following rules: ✓
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \in A$ then $n + 2 \in A$

3. Define $A \subseteq \mathbb{N}$ inductively by the following rules: x
   - $0 \in A$
   - For all $n \in \mathbb{N}$, if $n \notin A$ then $n + 2 \in A$

4. Define $A \subseteq$ Stream coinductively by the following rules: ✓
   - zeros $\in A$
   - If $x \in \mathbb{N}$, x even and $xs \in A$, then $x \# xs \in A$

Does there exist a unique set A with that property?

If not, maybe we need to complete the definition – be more specific!

Why do we care?

Obviously mathematicians want definitions that are rigorous, correct, meaningful and readable.

Definitional mechanisms are central to proof assistants.

Good definitions are the key to productive proof developments.

Proof assistants

ACL2 Agda Coq HOL4 HOL Light HOL-ω

Isabelle Lean Mizar Nuprl Matita PVS

Software systems that "assist" at

- formalizing mathematics
- verifying software and hardware systems

Prominent examples:

- formally proved Kepler's conjecture, Four Color theorem, Gödel's Incompleteness theorems, Odd Order theorem
- verified OS kernel (seL4), C compiler (CompCert), ML compiler (CakeML), web browser (Quark)

What do we want from definitions/specifications

Proof assistants offer facilities for definitions and proofs.

A definitional mechanism in a proof assistant should be expressive, allowing us to define what we want!

What do we want from definitions/specifications

Proof assistants offer facilities for definitions and proofs.

A definitional mechanism in a proof assistant should be expressive, allowing us to define what we want!
We should be able to express the intended concepts

- as directly as possible, without detours

What do we want from definitions/specifications

Proof assistants offer facilities for definitions and proofs.

A definitional mechanism in a proof assistant should be expressive, allowing us to define what we want!
We should be able to express the intended concepts

- as directly as possible, without detours
  E.g., if we want to define the Collatz function $f : \mathbb{N} \to \mathsf{Stream}$

$$f\ x = \begin{cases} \mathsf{zeros} & \text{if } x \leq 1 \\ f\ (x\ /\ 2) & \text{if } x > 1 \text{ and } x \text{ even} \\ x \ \#\ f\ (3 * x + 1) & \text{if } x > 1 \text{ and } x \text{ odd} \end{cases}$$

What do we want from definitions/specifications

Proof assistants offer facilities for definitions and proofs.

A definitional mechanism in a proof assistant should be expressive, allowing us to define what we want!
We should be able to express the intended concepts

- as directly as possible, without detours
  E.g., if we want to define the Collatz function $f : \mathbb{N} \to \text{Stream}$

  $$f\ x = \begin{cases} \text{zeros} & \text{if } x \leq 1 \\ f\ (x\ /\ 2) & \text{if } x > 1 \text{ and } x \text{ even} \\ x \mathbin{\#} f\ (3 * x + 1) & \text{if } x > 1 \text{ and } x \text{ odd} \end{cases}$$

  ... we should be able to do so without having to "patch" things and define auxiliary infrastructure.

- at the desired level of abstraction

What do we want from definitions/specifications

Proof assistants offer facilities for definitions and proofs.

A definitional mechanism in a proof assistant should be expressive, allowing us to define what we want!
We should be able to express the intended concepts

- as directly as possible, without detours
  E.g., if we want to define the Collatz function $f : \mathbb{N} \to \text{Stream}$
  $$f\ x = \begin{cases} \text{zeros} & \text{if } x \leq 1 \\ f\ (x\,/\,2) & \text{if } x > 1 \text{ and } x \text{ even} \\ x \# f\ (3 * x + 1) & \text{if } x > 1 \text{ and } x \text{ odd} \end{cases}$$
  ... we should be able to do so without having to "patch" things and define auxiliary infrastructure.

- at the desired level of abstraction
  E.g., if we want unordered trees, the proof assistant should not force us to encode them as ordered trees.

What do we want from definitions/specifications

After defining a concept, we should have at our disposal rules for reasoning about this concept.

E.g., it wouldn't helpful being able to define

$$f\ x = \begin{cases} \text{zeros} & \text{if } x \leq 1 \\ f\ (x\ /\ 2) & \text{if } x > 1 \text{ and } x \text{ even} \\ x \# f\ (3 * x + 1) & \text{if } x > 1 \text{ and } x \text{ odd} \end{cases}$$

(which involves a fancy combination of recursion and corecursion), but not getting suitable rules for reasoning about f.

> Proof assistants strive to achieve definition and proof expressiveness and automation, so that their users are productive.

But the users should not be so "productive" that they prove False. :-)

Important to keep definitions consistent, i.e., forbid the writing of inconsistent definitions.

What do we want from definitions/specifications

After defining a concept, we should have at our disposal rules for reasoning about this concept.

E.g., it wouldn't helpful being able to define

$$f\ x = \begin{cases} \text{zeros} & \text{if } x \leq 1 \\ f\ (x\ /\ 2) & \text{if } x > 1 \text{ and } x \text{ even} \\ x \# f\ (3 * x + 1) & \text{if } x > 1 \text{ and } x \text{ odd} \end{cases}$$

(which involves a fancy combination of recursion and corecursion), but not getting suitable rules for reasoning about f.

Proof assistants strive to achieve definition and proof expressiveness and automation, so that their users are productive.

But the users should not be so “productive” that they prove False. :-)

Important to keep definitions consistent, i.e., forbid the writing of inconsistent definitions.
E.g., is the above scheme for combining recursion with corecursion sound? How can we be sure?

What do we want from definitions/specifications

If a proof assistant based on a total-function logic allows a definition like

"Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x = 1 + f\ x$"

... then False is immediately derivable, so everything becomes provable.

Proof assistants try to prevent such situations via

1. syntactic checks, e.g., force definitions to be "guarded" or "positive"
   - error-prone (trivial bugs can introduce inconsistencies)

What do we want from definitions/specifications

If a proof assistant based on a total-function logic allows a definition like

"Define $f : \mathbb{N} \to \mathbb{N}$ by $f\ x = 1 + f\ x$"

... then False is immediately derivable, so everything becomes provable.

Proof assistants try to prevent such situations via

1. syntactic checks, e.g., force definitions to be "guarded" or "positive"
   - error-prone (trivial bugs can introduce inconsistencies)
   - too rigid (can reject many obviously valid definitions)
2. semantic reductions: make sense of the recursive and inductive definitions in terms of more basic (non-recursive) primitives.
   A non-recursive definition, no matter how complex, is obviously consistent!

In practice, each proof assistant provides a combination of these two, in various proportions.

Overview

In this tutorial...

I'll teach a foundation of (co)induction and (co)recursion following the semantic approach

- favored by HOL-based proof assistants such as HOL4, HOL Light and Isabelle/HOL
- developed substantially in Isabelle/HOL in recent years

In this tutorial...

I'll teach a foundation of (co)induction and (co)recursion following the semantic approach

- favored by HOL-based proof assistants such as HOL4, HOL Light and Isabelle/HOL
- developed substantially in Isabelle/HOL in recent years

I'll use examples and exercises in Isabelle/HOL.

In this tutorial...

I'll teach a foundation of (co)induction and (co)recursion following the semantic approach

- favored by HOL-based proof assistants such as HOL4, HOL Light and Isabelle/HOL
- developed substantially in Isabelle/HOL in recent years

I'll use examples and exercises in Isabelle/HOL.

The foundation itself is independent from proof assistant technology, <u>and I'll present it independently</u>.

Some Conventions and Notations

Given two sets A and B, $A \to B$ denotes the set of functions from A to B. So that, for example, $f : A \to B$ is the same as $f \in A \to B$.

For multiple-argument functions, we prefer the curried forms, e.g., $f : A \to B \to \text{Bool}$, to the uncurried forms, e.g., $f : A \times B \to \text{Bool}$.

We'll sometimes use lambda notation. E.g., a numeric function in $\mathbb{N} \to \mathbb{N}$ that adds 5 can be written as $\lambda x.\, x + 5$.

Given two sets A and B, $A \to B$ denotes the set of functions from A to B. So that, for example, $f : A \to B$ is the same as $f \in A \to B$.

For multiple-argument functions, we prefer the curried forms, e.g., $f : A \to B \to \text{Bool}$, to the uncurried forms, e.g., $f : A \times B \to \text{Bool}$.

We'll sometimes use lambda notation. E.g., a numeric function in $\mathbb{N} \to \mathbb{N}$ that adds 5 can be written as $\lambda x.\, x + 5$. (Mathematicians sometimes write this as $x \mapsto x + 5$.)

We write $\lambda a, b \ldots$ for $\lambda a.\, \lambda b \ldots$. E.g., the function in $\mathbb{N} \to \mathbb{N} \to \mathbb{N}$ that adds two numbers can be written as $\lambda x, y.\, x + y$.

Bool = $\{\top, \bot\}$ is the set of Boolean values:
$\top$ means "true", $\bot$ means "false".

Predicates

Bool = $\{\top, \bot\}$ is the set of Boolean values:
$\top$ means "true", $\bot$ means "false".

Let A be a set. A mathematical property of the elements of A, a.k.a a predicate on A, corresponds to a function $P : A \to \text{Bool}$.

Predicates

Bool = $\{\top, \bot\}$ is the set of Boolean values:
$\top$ means "true", $\bot$ means "false".

Let A be a set. A mathematical property of the elements of A, a.k.a a predicate on A, corresponds to a function $P : A \to$ Bool.

We identify such properties with functions to Bool.
E.g., given $P : A \to$ Bool and $a \in A$, we say "$P\ a$ holds", or simply "$P\ a$", to mean that $P\ a = \top$.

Predicates

Bool = $\{\top, \bot\}$ is the set of Boolean values:
$\top$ means "true", $\bot$ means "false".

Let A be a set. A mathematical property of the elements of A, a.k.a a predicate on A, corresponds to a function $P : A \to$ Bool.

We identify such properties with functions to Bool.
E.g., given $P : A \to$ Bool and $a \in A$, we say "$P\ a$ holds", or simply "$P\ a$", to mean that $P\ a = \top$.

And similarly for multiple-argument predicates, a.k.a. relations.
E.g., given $P : A \to B \to C \to$ Bool, $a \in A$, $b \in B$ and $c \in C$, we say "$P\ a\ b\ c$ holds", or simply "$P\ a\ b\ c$", to mean that $P\ a\ b\ c = \top$.