2. Inductive Predicates

Some Informal Examples of Inductive Definitions

Informal example 1

The predicate P on natural numbers is defined inductively by the following rules:

- $P\,0$ holds;
- if $P\,n$ holds, then $P\,(n+2)$ holds.

What predicate is this?

Informal example 1

The predicate P on natural numbers is defined inductively by the following rules:

- $P\ 0$ holds;
- if $P\ n$ holds, then $P\ (n + 2)$ holds.

What predicate is this?

Informal example 1: the $even$ predicate

The predicate $even$ on natural numbers is defined inductively by the following rules:

- $even\ 0$ holds;
- if $even\ n$ holds, then $even\ (n+2)$ holds.

What predicate is this?

But why does this capture the notion of even number?

Informal example 1: the $even$ predicate

The predicate $even$ on natural numbers is defined inductively by the following rules:

- $even\,0$ holds;
- if $even\,n$ holds, then $even\,(n+2)$ holds.

What predicate is this?

But why does this capture the notion of even number?

For example, why does $even\,4$ hold, but $even\,3$ not?

Informal example 2

Given a set A, let $\mathsf{List}(A)$ be the set of lists $[a_1, \ldots, a_n]$ with elements in A. We write $[]$ for the empty list and $a\#as$ for the list obtained by consing a to as.

The binary relation R on $\mathsf{List}(A)$ is defined inductively by the rules:

- $R\,[]\,as$ holds;
- if $R\ as\ as'$ holds, then $R\ as\ (a\#as')$ holds;
- if $R\ as\ as'$ holds, then $R\ (a\#as)\ (a\#as')$ holds.

What relation is this?

Informal example 2

Given a set A, let List(A) be the set of lists $[a_1, \ldots, a_n]$ with elements in A. We write $[]$ for the empty list and $a\#as$ for the list obtained by consing a to as.

The binary relation R on List(A) is defined inductively by the rules:

- $R\,[]\,as$ holds;
- if $R\,as\,as'$ holds, then $R\,as\,(a\#as')$ holds;
- if $R\,as\,as'$ holds, then $R\,(a\#as)\,(a\#as')$ holds.

What relation is this?

$subl\,as\,as'$ holds if and only if as is a sublist (subsequence) of as'

Informal example 2

Given a set A, let $\text{List}(A)$ be the set of lists $[a_1, \ldots, a_n]$ with elements in A. We write $[]$ for the empty list and $a\#as$ for the list obtained by consing a to as.

The binary relation R on $\text{List}(A)$ is defined inductively by the rules:

- $R\,[]\;as$ holds;
- if $R\;as\;as'$ holds, then $R\;as\;(a\#as')$ holds;
- if $R\;as\;as'$ holds, then $R\;(a\#as)\;(a\#as')$ holds.

What relation is this?

$subl\;as\;as'$ holds if and only if as is a sublist (subsequence) of as' in that, if as' has the form $[a'_0, \ldots, a'_{n-1}]$, then there exist $k \geq 0$ and $0 \leq j_0 < \ldots < j_{k-1} \leq n-1$ such that $as = [a'_{j_0}, \ldots, a'_{j_{k-1}}]$.

Informal example 2: the $subl$ relation

Given a set A, let List(A) be the set of lists $[a_1, \ldots, a_n]$ with elements in A. We write $[]$ for the empty list and $a\#as$ for the list obtained by consing a to as.

The binary relation $subl$ on List(A) is defined inductively by the rules:

- $subl\ []\ as$ holds;
- if $subl\ as\ as'$ holds, then $subl\ as\ (a\#as')$ holds;
- if $subl\ as\ as'$ holds, then $subl\ (a\#as)\ (a\#as')$ holds.

What relation is this?

$subl\ as\ as'$ holds if and only if as is a sublist (subsequence) of as' in that, if as' has the form $[a'_0, \ldots, a'_{n-1}]$, then there exist $k \geq 0$ and $0 \leq j_0 < \ldots < j_{k-1} \leq n-1$ such that $as = [a'_{j_0}, \ldots, a'_{j_{k-1}}]$.

Can we prove, e.g., $subl\ [a]\ [a, b]$, but $\neg\ subl\ [a, b]\ [a]$?

Informal example 2: the $subl$ relation

Given a set A, let List(A) be the set of lists $[a_1, \ldots, a_n]$ with elements in A. We write $[]$ for the empty list and $a\#as$ for the list obtained by consing a to as.

The binary relation $subl$ on List(A) is defined inductively by the rules:

- $subl\ []\ as$ holds;
- if $subl\ as\ as'$ holds, then $subl\ as\ (a\#as')$ holds;
- if $subl\ as\ as'$ holds, then $subl\ (a\#as)\ (a\#as')$ holds.

What relation is this?

$subl\ as\ as'$ holds if and only if as is a sublist (subsequence) of as' in that, if as' has the form $[a'_0, \ldots, a'_{n-1}]$, then there exist $k \geq 0$ and $0 \leq j_0 < \ldots < j_{k-1} \leq n-1$ such that $as = [a'_{j_0}, \ldots, a'_{j_{k-1}}]$.

Can we prove, e.g., $subl\ [a]\ [a, b]$, but $\neg\ subl\ [a, b]\ [a]$?

Can we prove the equivalence with the above alternative description?

Informal example 2: the $subl$ relation

Given a set A, let List(A) be the set of lists $[a_1, \ldots, a_n]$ with elements in A. We write $[]$ for the empty list and $a\#as$ for the list obtained by consing a to as.

The binary relation $subl$ on List(A) is defined inductively by the rules:

- $subl\ []\ as$ holds;
- if $subl\ as\ as'$ holds, then $subl\ as\ (a\#as')$ holds;
- if $subl\ as\ as'$ holds, then $subl\ (a\#as)\ (a\#as')$ holds.

Another way to write this inductive definition (where the labels "Nil", "ConsR" and "Cons" are names we give to the rules for convenience):

$$\frac{\cdot}{subl\ []\ as}\ \text{(Nil)} \qquad\qquad \frac{subl\ as\ as'}{subl\ as\ (a\#as')}\ \text{(ConsR)}$$

$$\frac{subl\ as\ as'}{subl\ (a\#as)\ (a\#as')}\ \text{(Cons)}$$

Informal example 3

Given a set A, let LazyList(A) be the set of "lazy lists" (finite or infinite lists) with elements in A – they have the form $[a_1, a_2, \ldots, a_n]$ or $[a_1, a_2, \ldots]$. We write $a\#as$ for the lazy list obtained by consing a to as.

Informal example 3

Given a set A, let LazyList(A) be the set of "lazy lists" (finite or infinite lists) with elements in A – they have the form $[a_1, a_2, \ldots, a_n]$ or $[a_1, a_2, \ldots]$. We write $a\#as$ for the lazy list obtained by consing a to as.

The binary relation R on LazyList(A), is defined inductively by the following rules:

$$\frac{\cdot}{R\ []\ as}\ \text{(Nil)} \qquad \frac{R\ as\ as'}{R\ as\ (a\#as')}\ \text{(ConsR)}$$

$$\frac{R\ as\ as'}{R\ (a\#as)\ (a\#as')}\ \text{(Cons)}$$

Informal example 3

Given a set A, let LazyList(A) be the set of "lazy lists" (finite or infinite lists) with elements in A – they have the form $[a_1, a_2, \ldots, a_n]$ or $[a_1, a_2, \ldots]$. We write $a\#as$ for the lazy list obtained by consing a to as.

The binary relation R on LazyList(A), is defined inductively by the following rules:

$$\frac{\cdot}{R\;[]\;as}\;\text{(Nil)} \qquad\qquad \frac{R\;as\;as'}{R\;as\;(a\#as')}\;\text{(ConsR)}$$

$$\frac{R\;as\;as'}{R\;(a\#as)\;(a\#as')}\;\text{(Cons)}$$

What relation is this?

Informal example 3: the $sublll$ predicate

Given a set A, let LazyList(A) be the set of "lazy lists" (finite or infinite lists) with elements in A – they have the form $[a_1, a_2, \ldots, a_n]$ or $[a_1, a_2, \ldots]$. We write $a\#as$ for the lazy list obtained by consing a to as.

The binary relation $sublll$ on LazyList(A), is defined inductively by the following rules:

$$\frac{\cdot}{sublll\ []\ as}\ (\text{Nil}) \qquad \frac{sublll\ as\ as'}{sublll\ as\ (a\#as')}\ (\text{ConsR})$$

$$\frac{sublll\ as\ as'}{sublll\ (a\#as)\ (a\#as')}\ (\text{Cons})$$

What relation is this?

Is it the sub-lazylist relation, in that $sublll\ as\ as'$ holds iff as consists of the elements located on some positions in as' (preserving the order)?

Informal example 3: the $sublll$ predicate

Given a set A, let LazyList(A) be the set of "lazy lists" (finite or infinite lists) with elements in A – they have the form $[a_1, a_2, \ldots, a_n]$ or $[a_1, a_2, \ldots]$. We write $a\#as$ for the lazy list obtained by consing a to as.

The binary relation $sublll$ on LazyList(A), is defined inductively by the following rules:

$$\frac{\cdot}{sublll\ []\ as}\ (\text{Nil}) \qquad \frac{sublll\ as\ as'}{sublll\ as\ (a\#as')}\ (\text{ConsR})$$

$$\frac{sublll\ as\ as'}{sublll\ (a\#as)\ (a\#as')}\ (\text{Cons})$$

What relation is this?

Is it the sub-lazylist relation, in that $sublll\ as\ as'$ holds iff as consists of the elements located on some positions in as' (preserving the order)?

No! The inductive definition restricts as to be finite.

Informal example 3: the subll predicate

Given a set A, let LazyList(A) be the set of "lazy lists" (finite or infinite lists) with elements in A – they have the form $[a_1, a_2, \ldots, a_n]$ or $[a_1, a_2, \ldots]$. We write $a\#as$ for the lazy list obtained by consing a to as.

The binary relation subll on LazyList(A), is defined inductively by the following rules:

$$\frac{\cdot}{\mathit{subll}\ []\ as}\ \text{(Nil)} \qquad \frac{\mathit{subll}\ as\ as'}{\mathit{subll}\ as\ (a\#as')}\ \text{(ConsR)}$$

$$\frac{\mathit{subll}\ as\ as'}{\mathit{subll}\ (a\#as)\ (a\#as')}\ \text{(Cons)}$$

What relation is this?

Is it the sub-lazylist relation, in that $\mathit{subll}\ as\ as'$ holds iff as consists of the elements located on some positions in as' (preserving the order)?

No! The inductive definition restricts as to be finite.

What we need here is a coinductive definition...

Informal example 3: the $subll$ predicate

Given a set A, let LazyList(A) be the set of "lazy lists" (finite or infinite lists) with elements in A – they have the form $[a_1, a_2, \ldots, a_n]$ or $[a_1, a_2, \ldots]$. We write $a\#as$ for the lazy list obtained by consing a to as.

The binary relation $subll$ on LazyList(A), is defined coinductively by the following rules:

$$\frac{\cdot}{subll\ []\ as}\ \text{(Nil)} \qquad \frac{subllR\ as\ as'}{subllR\ as\ (a\#as')}\ \text{(ConsR)}$$

$$\frac{subllR\ as\ as'}{subllR\ (a\#as)\ (a\#as')}\ \text{(Cons)}$$

What relation is this?

Is it the sub-lazylist relation, in that $subll\ as\ as'$ holds iff as consists of the elements located on some positions in as' (preserving the order)? No! The inductive definition restricts as to be finite.

What we need here is a coinductive definition...

Informal example 3: the $subll$ predicate

Given a set A, let LazyList(A) be the set of "lazy lists" (finite or infinite lists) with elements in A – they have the form $[a_1, a_2, \ldots, a_n]$ or $[a_1, a_2, \ldots]$. We write $a\#as$ for the lazy list obtained by consing a to as.

The binary relation $subll$ on LazyList(A), is defined coinductively by the following rules:

$$\frac{\cdot}{subll\ []\ as}\ \text{(Nil)} \qquad\qquad \frac{subllR\ as\ as'}{subllR\ as\ (a\#as')}\ \text{(ConsR)}$$

$$\frac{subllR\ as\ as'}{subllR\ (a\#as)\ (a\#as')}\ \text{(Cons)}$$

What relation is this?

Is it the sub-lazylist relation, in that $subll\ as\ as'$ holds iff as consists of the elements located on some positions in as' (preserving the order)?

No! The inductive definition restricts as to be finite.

What we need here is a coinductive definition...

Next, we'll make the notions of inductive and coinductive definition rigorous.

Foundation of (Co)Induction

$(A, \leq)$ is said to be a partially ordered set when

- A is a set
- $\leq$ is a binary relation on A that is

  reflexive: $x \leq x$
  transitive: $x \leq y$ and $y \leq z$ imply $x \leq z$
  anti-symmetric: $x \leq y$ and $y \leq x$ imply $x = z$

Let $(A, \leq)$ be a partially ordered set, let $X \subseteq A$ and $a \in A$. We say that:

- a is the greatest element of X if $a \in X$ and $\forall x \in X.\ x \leq a$;
- a is the least element of X if $a \in X$ and $\forall x \in X.\ a \leq x$.

Let $(A, \leq)$ be a partially ordered set.

Given $X \subseteq A$, we define:

- Lower(X), the set of lower bounds of X, to be $\{a \in A \mid \forall x \in X.\, a \leq x\}$.

Let $(A, \leq)$ be a partially ordered set.

Given $X \subseteq A$, we define:

- Lower(X), the set of lower bounds of X, to be $\{a \in A \mid \forall x \in X.\, a \leq x\}$.
  If it exists, the greatest element of Lower(X) is called the infimum of X and is denoted by $\bigwedge X$.

Let $(A, \leq)$ be a partially ordered set.

Given $X \subseteq A$, we define:

- Lower(X), the set of lower bounds of X, to be $\{a \in A \mid \forall x \in X.\, a \leq x\}$.
  If it exists, the greatest element of Lower(X) is called the infimum of X and is denoted by $\bigwedge X$.
- Upper(X), the set of upper bounds of X, to be $\{a \in A \mid \forall x \in X.\, x \leq a\}$.

Let $(A, \leq)$ be a partially ordered set.

Given $X \subseteq A$, we define:

- Lower(X), the set of lower bounds of X, to be $\{a \in A \mid \forall x \in X.\, a \leq x\}$.
  If it exists, the greatest element of Lower(X) is called the infimum of X and is denoted by $\bigwedge X$.
- Upper(X), the set of upper bounds of X, to be $\{a \in A \mid \forall x \in X.\, x \leq a\}$.
  If it exists, the least element of Upper(X) is called the supremum of X and is denoted by $\bigvee X$.

$(A, \leq)$ is said to be a complete lattice if infima $\bigwedge X$ and suprema $\bigvee X$ exist for all $X \subseteq A$.

Let $(A, \leq)$ be a partially ordered set.

Given $X \subseteq A$, we define:

- Lower(X), the set of lower bounds of X, to be $\{a \in A \mid \forall x \in X.\, a \leq x\}$.
  If it exists, the greatest element of Lower(X) is called the infimum of X and is denoted by $\bigwedge X$.
- Upper(X), the set of upper bounds of X, to be $\{a \in A \mid \forall x \in X.\, x \leq a\}$.
  If it exists, the least element of Upper(X) is called the supremum of X and is denoted by $\bigvee X$.

$(A, \leq)$ is said to be a complete lattice if infima $\bigwedge X$ and suprema $\bigvee X$ exist for all $X \subseteq A$.

Exercise. 1. Prove that, if they exist, $\bigvee\varnothing$ and $\bigwedge\varnothing$ are the least and greatest elements of A.
2. Prove that, if $\bigvee X$ exists and is in X, then it is the greatest element of X. Dually, if $\bigwedge X$ exists and is in X, then it is the least element of X.

Fixpoints, pre-fixpoints and post-fixpoints

Fix a partially ordered set $(A, \leq)$ and a function $F : A \to A$.

An element $a \in A$ is called:

- a fixpoint (fixed point) of F if $F\,a = a$

Fixpoints, pre-fixpoints and post-fixpoints

Fix a partially ordered set $(A, \leq)$ and a function $F : A \to A$.

An element $a \in A$ is called:

- a <u>fixpoint</u> (fixed point) of F if $F\,a = a$
- a <u>pre-fixpoint</u> of F if $F\,a \leq a$

Fixpoints, pre-fixpoints and post-fixpoints

Fix a partially ordered set $(A, \leq)$ and a function $F : A \to A$.

An element $a \in A$ is called:

- a <u>fixpoint</u> (fixed point) of F if $F\,a = a$
- a <u>pre-fixpoint</u> of F if $F\,a \leq a$
- a <u>post-fixpoint</u> of F if $a \leq F\,a$

Fixpoints, pre-fixpoints and post-fixpoints

Fix a partially ordered set $(A, \leq)$ and a function $F : A \to A$.

An element $a \in A$ is called:

- a fixpoint (fixed point) of F if $F\,a = a$
- a pre-fixpoint of F if $F\,a \leq a$
- a post-fixpoint of F if $a \leq F\,a$

Note: fixpoint = pre-fixpoint + post-fixpoint

Fixpoints, pre-fixpoints and post-fixpoints

Fix a partially ordered set $(A, \leq)$ and a function $F : A \to A$.

An element $a \in A$ is called:

- a fixpoint (fixed point) of F if $F\,a = a$
- a pre-fixpoint of F if $F\,a \leq a$
- a post-fixpoint of F if $a \leq F\,a$

Note: fixpoint = pre-fixpoint + post-fixpoint

The function F is said to be monotonic if it preserves the order:
$a \leq b$ implies $F\,a \leq F\,b$ for all $a, b \in A$.

The fixpoint theorem of Knaster and Tarski

Theorem (Knaster-Tarski, short version). Any monotonic function on a complete lattice has a least and a greatest fixpoint.

The fixpoint theorem of Knaster and Tarski

Theorem (Knaster-Tarski, full version). Let $(A, \leq)$ be a complete lattice and $F : A \to A$ a monotonic function.

1. Let $\mathsf{I}_F = \bigwedge\{a \mid F\,a \leq a\}$ (the infimum of the set of pre-fixpoints). Then I_F is the least fixpoint of F and the least pre-fixpoint of F.
2. Let $\mathsf{J}_F = \bigvee\{a \mid a \leq F\,a\}$ (the supremum of the set of post-fixpoints). Then J_F is the greatest fixpoint and the greatest post-fixpoint of F.

Theorem (Knaster-Tarski, full version). Let $(A, \leq)$ be a complete lattice and $F : A \to A$ a monotonic function.

1. Let $\mathsf{I}_F = \bigwedge\{a \mid F\,a \leq a\}$ (the infimum of the set of pre-fixpoints). Then I_F is the least fixpoint of F and the least pre-fixpoint of F.
2. Let $\mathsf{J}_F = \bigvee\{a \mid a \leq F\,a\}$ (the supremum of the set of post-fixpoints). Then J_F is the greatest fixpoint and the greatest post-fixpoint of F.

Proof. Let $X = \{a \mid F\,a \leq a\}$.
We have $F\,\mathsf{I}_F \in \mathsf{Lower}(X)$.
Indeed, given $a \in X$:
– on the one hand, we have $\mathsf{I}_F \leq a$, which implies $F\,\mathsf{I}_F \leq F\,a$;
– on the other hand, we have $F\,a \leq a$;
– the last two give us $F\,\mathsf{I}_F \leq a$.
Hence $F\,\mathsf{I}_F \leq \mathsf{I}_F$, which means $\mathsf{I}_F \in X$.
Hence I_F is the least pre-fixpoint of F.
But we also have $F\,(F\,\mathsf{I}_F) \leq F\,\mathsf{I}_F$, i.e., $F\,\mathsf{I}_F \in X$, hence $\mathsf{I}_F \leq F\,\mathsf{I}_F$.
Hence $F\,\mathsf{I}_F = \mathsf{I}_F$, making I_F a fixpoint, and also the least fixpoint of F.
... and the fact about greatest (post-)fixpoints is dual.

Example: The powerset complete lattice

$(\mathcal{P}(A), \leq)$

- $\mathcal{P}(A)$ is the powerset (set of all sets) of a set A
- the order $\leq$ is inclusion, $\subseteq$

Example: The powerset complete lattice

$(\mathcal{P}(A), \leq)$

- $\mathcal{P}(A)$ is the powerset (set of all sets) of a set A
- the order $\leq$ is inclusion, $\subseteq$

Exercise. Show that this forms a complete lattice, where infima are intersections and suprema are unions.

Example: the complete lattices of predicates / relations

$(A \to \text{Bool}, \leq)$ – the complete lattice of predicates on A.

- The order $\leq$ is defined by $P \leq Q$ iff $\forall a \in A.\ P\ a \longrightarrow Q\ a$
- Infima and suprema are given by $\forall$ and $\exists$.
  Namely, for $X \subseteq (A \to \text{Bool})$: $\bigwedge X = \lambda a.\ \forall P \in X.\ P\ a$
  $\bigvee X = \lambda a.\ \exists P \in X.\ P\ a$
- The least and greatest elements are $\lambda a.\ \bot$ and $\lambda a.\ \top$

Exercise. Show that this is isomorphic to $(\mathcal{P}(A), \subseteq)$.

Example: the complete lattices of predicates / relations

$(A \to \mathsf{Bool}, \leq)$ – the complete lattice of predicates on A.

- The order $\leq$ is defined by $P \leq Q$ iff $\forall a \in A.\ P\ a \longrightarrow Q\ a$
- Infima and suprema are given by $\forall$ and $\exists$.
  Namely, for $X \subseteq (A \to \mathsf{Bool})$: $\bigwedge X = \lambda a.\ \forall P \in X.\ P\ a$
  $\bigvee X = \lambda a.\ \exists P \in X.\ P\ a$
- The least and greatest elements are $\lambda a.\ \bot$ and $\lambda a.\ \top$

Exercise. Show that this is isomorphic to $(\mathcal{P}(A), \subseteq)$.

And similarly for relations of any arity

Example: the complete lattices of predicates / relations

$(A \to \mathsf{Bool}, \leq)$ – the complete lattice of predicates on A.

- The order $\leq$ is defined by $P \leq Q$ iff $\forall a \in A.\ P\ a \longrightarrow Q\ a$
- Infima and suprema are given by $\forall$ and $\exists$.
  Namely, for $X \subseteq (A \to \mathsf{Bool})$: $\bigwedge X = \lambda a.\ \forall P \in X.\ P\ a$
  $\bigvee X = \lambda a.\ \exists P \in X.\ P\ a$
- The least and greatest elements are $\lambda a.\ \bot$ and $\lambda a.\ \top$

Exercise. Show that this is isomorphic to $(\mathcal{P}(A), \subseteq)$.

And similarly for relations of any arity, for example:

$(A \to B \to \mathsf{Bool}, \leq)$ – the complete lattice of relations between A and B.

- The order $\leq$ is defined by $P \leq Q$ iff $\forall a \in A, b \in B.\ P\ a\ b \longrightarrow Q\ a\ b$
- Infima and suprema are given by $\forall$ and $\exists$.
  Namely, for $X \subseteq (A \to B \to \mathsf{Bool})$: $\bigwedge X = \lambda a, b.\ \forall P \in X.\ P\ a\ b$
  $\bigvee X = \lambda a, b.\ \exists P \in X.\ P\ a\ b$
- The least and greatest elements are $\lambda a, b.\ \bot$ and $\lambda a, b.\ \top$

Example: the complete lattices of predicates / relations

$(A \to \text{Bool}, \leq)$ – the complete lattice of predicates on A.

- The order $\leq$ is defined by $P \leq Q$ iff $\forall a \in A.\ P\ a \longrightarrow Q\ a$
- Infima and suprema are given by $\forall$ and $\exists$.
  Namely, for $X \subseteq (A \to \text{Bool})$: $\bigwedge X = \lambda a.\ \forall P \in X.\ P\ a$
  $\bigvee X = \lambda a.\ \exists P \in X.\ P\ a$
- The least and greatest elements are $\lambda a.\ \bot$ and $\lambda a.\ \top$

Exercise. Show that this is isomorphic to $(\mathcal{P}(A), \subseteq)$.

And similarly for relations of any arity, for example:

$(A \to B \to \text{Bool}, \leq)$ – the complete lattice of relations between A and B.

- The order $\leq$ is defined by $P \leq Q$ iff $\forall a \in A, b \in B.\ P\ a\ b \longrightarrow Q\ a\ b$
- Infima and suprema are given by $\forall$ and $\exists$.
  Namely, for $X \subseteq (A \to B \to \text{Bool})$: $\bigwedge X = \lambda a, b.\ \forall P \in X.\ P\ a\ b$
  $\bigvee X = \lambda a, b.\ \exists P \in X.\ P\ a\ b$
- The least and greatest elements are $\lambda a, b.\ \bot$ and $\lambda a, b.\ \top$

Exercise. Show that this is isomorphic to $(\mathcal{P}(A \times B), \subseteq)$.

Back to Our Examples of (Co)Inductive Definitions

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \rightarrow \mathrm{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \mathsf{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ (\mathsf{Zero}) \qquad \frac{even\ n}{even\ (n+2)}\ (\mathsf{Suc})$$

"Inductively" means: smallest predicate closed under the given rules.

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ (\text{Zero}) \qquad \frac{even\ n}{even\ (n+2)}\ (\text{Suc})$$

"Inductively" means: smallest predicate closed under the given rules.
More precisely: We define $even = \mathsf{I}_F$, where
$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

"Inductively" means: smallest predicate closed under the given rules.

More precisely: We define $even = \mathsf{I}_F$, where

$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:

$F\ P\ =\ \lambda m.\ m = 0\ \lor\ (\exists n.\ m = n + 2\ \land\ P\ n)$

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

"Inductively" means: smallest predicate closed under the given rules.

More precisely: We define $even = \mathsf{I}_F$, where

$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:

$F\ P\ =\ \lambda m.\ m = 0\ \vee\ (\exists n.\ m = n + 2\ \wedge\ P\ n)$

> Thus, $F\ P$ essentially applies the rules to P, i.e.,
> $F\ P$ holds for exactly those items m that are produced
> by applying the rules to items for which P holds.

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ (\text{Zero}) \qquad \frac{even\ n}{even\ (n+2)}\ (\text{Suc})$$

"Inductively" means: smallest predicate closed under the given rules.

More precisely: We define $even = \mathsf{I}_F$, where

$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:

$F\ P\ =\ \lambda m.\ m = 0\ \lor\ (\exists n.\ m = n + 2\ \land\ P\ n)$

Thus, $F\ P$ essentially applies the rules to P, i.e., $F\ P$ holds for exactly those items m that are produced by applying the rules to items for which P holds.

F is monotonic, so I_F exists by Knaster-Tarski.

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ (\text{Zero}) \qquad \frac{even\ n}{even\ (n+2)}\ (\text{Suc})$$

"Inductively" means: smallest predicate closed under the given rules.

More precisely: We define $even = \mathsf{I}_F$, where

$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:

$F\ P\ =\ \lambda m.\ m = 0\ \lor\ (\exists n.\ m = n + 2\ \land\ P\ n)$

Thus, $F\ P$ essentially applies the rules to P, i.e., $F\ P$ holds for exactly those items m that are produced by applying the rules to items for which P holds.

$even$ is a pre-fixpoint of F,
i.e., $F\ even \le even$

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

"Inductively" means: smallest predicate closed under the given rules.
More precisely: We define $even = \mathsf{I}_F$, where
$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:
$F\ P\ =\ \lambda m.\ m = 0\ \lor\ (\exists n.\ m = n + 2\ \land\ P\ n)$

Thus, $F\ P$ essentially applies the rules to P, i.e.,
$F\ P$ holds for exactly those items m that are produced
by applying the rules to items for which P holds.

$even$ is a pre-fixpoint of F,
i.e., $F\ even \leq even$

This means $\forall m.\ m = 0 \lor (\exists n.\ m = n + 2\ \land\ even\ n) \longrightarrow even\ m$

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ (\text{Zero}) \qquad \frac{even\ n}{even\ (n+2)}\ (\text{Suc})$$

"Inductively" means: <u>smallest predicate closed under the given rules</u>.

More precisely: We define $even = \mathsf{I}_F$, where

$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:

$F\ P\ =\ \lambda m.\ m = 0\ \vee\ (\exists n.\ m = n + 2\ \wedge\ P\ n)$

> Thus, $F\ P$ essentially applies the rules to P, i.e.,
> $F\ P$ holds for exactly those items m that are produced
> by applying the rules to items for which P holds.

$even$ is a pre-fixpoint of F,
i.e., $F\ even \leq even$

This means $\forall m.\ m = 0 \vee (\exists n.\ m = n + 2\ \wedge\ even\ n) \longrightarrow even\ m$

i.e., $even\ 0$ and $\forall n.\ even\ n \longrightarrow even\ (n+2)$

Making sense of the inductive specification of *even*

The predicate $even : \mathbb{N} \to \mathsf{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad\qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

"Inductively" means: smallest predicate closed under the given rules.
More precisely: We define $even = \mathsf{I}_F$, where
$F : (\mathbb{N} \to \mathsf{Bool}) \to (\mathbb{N} \to \mathsf{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \mathsf{Bool}$:
$F\ P\ =\ \lambda m.\ m = 0\ \vee\ (\exists n.\ m = n + 2\ \wedge\ P\ n)$

Thus, $F\ P$ essentially applies the rules to P, i.e., $F\ P$ holds for exactly those items m that are produced by applying the rules to items for which P holds.

$even$ is a pre-fixpoint of F,
i.e., $F\ even \leq even$

This means $\forall m.\ m = 0 \vee (\exists n.\ m = n + 2\ \wedge\ even\ n) \longrightarrow even\ m$
i.e., $even\ 0$ and $\forall n.\ even\ n \longrightarrow even\ (n+2)$
which simply means that the rules (Zero) and (Suc) are valid.

Making sense of the inductive specification of *even*

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

"Inductively" means: smallest predicate closed under the given rules.
More precisely: We define $even = \mathsf{I}_F$, where
$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:
$F\ P\ =\ \lambda m.\ m = 0\ \lor\ (\exists n.\ m = n + 2\ \land\ P\ n)$

Thus, $F\ P$ essentially applies the rules to P, i.e.,
$F\ P$ holds for exactly those items m that are produced
by applying the rules to items for which P holds.

$even$ is a pre-fixpoint of F,
i.e., $F\ even \leq even$

This means $\forall m.\ m = 0 \lor (\exists n.\ m = n + 2\ \land\ even\ n) \longrightarrow even\ m$
i.e., $even\ 0$ and $\forall n.\ even\ n \longrightarrow even\ (n+2)$
which simply means that the rules (Zero) and (Suc) are valid.

(Zero) and (Suc) are called introduction rules for $even$, because they allow to prove that $even$ holds (for certain items).

Making sense of the inductive specification of even

The predicate $\mathit{even} : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{\mathit{even}\ 0}\ (\text{Zero}) \qquad \frac{\mathit{even}\ n}{\mathit{even}\ (n+2)}\ (\text{Suc})$$

"Inductively" means: smallest predicate closed under the given rules.
More precisely: We define $\mathit{even} = \mathsf{I}_F$, where
$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:
$F\ P\ =\ \lambda m.\ m = 0\ \vee\ (\exists n.\ m = n + 2\ \wedge\ P\ n)$

> Thus, $F\ P$ essentially applies the rules to P, i.e.,
> $F\ P$ holds for exactly those items m that are produced
> by applying the rules to items for which P holds.

even is a fixpoint of F, in particular a post-fixpoint,
i.e., $\mathit{even} \leq F\ \mathit{even}$.

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ (\text{Zero}) \qquad \frac{even\ n}{even\ (n+2)}\ (\text{Suc})$$

"Inductively" means: smallest predicate closed under the given rules.
More precisely: We define $even = \mathsf{I}_F$, where
$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:
$F\ P\ =\ \lambda m.\ m = 0\ \lor\ (\exists n.\ m = n + 2\ \land\ P\ n)$

Thus, $F\ P$ essentially applies the rules to P, i.e.,
$F\ P$ holds for exactly those items m that are produced
by applying the rules to items for which P holds.

$even$ is a fixpoint of F, in particular a post-fixpoint,
i.e., $even \leq F\ even$.

This means $\forall m.\ even\ m \longrightarrow m = 0 \lor (\exists n.\ m = n + 2\ \land\ even\ n)$

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ (\text{Zero}) \qquad \frac{even\ n}{even\ (n+2)}\ (\text{Suc})$$

"Inductively" means: smallest predicate closed under the given rules.
More precisely: We define $even = \mathsf{I}_F$, where
$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:
$F\ P\ =\ \lambda m.\ m = 0\ \lor\ (\exists n.\ m = n + 2\ \land\ P\ n)$

Thus, $F\ P$ essentially applies the rules to P, i.e., $F\ P$ holds for exactly those items m that are produced by applying the rules to items for which P holds.

$even$ is a fixpoint of F, in particular a post-fixpoint, i.e., $even \le F\ even$.

This means $\forall m.\ even\ m \longrightarrow m = 0 \lor (\exists n.\ m = n + 2\ \land\ even\ n)$
i.e., whenever $even\ m$ holds, it must have been obtained by one of the rules (Zero) and (Suc)

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

"Inductively" means: smallest predicate closed under the given rules.
More precisely: We define $even = \mathsf{I}_F$, where
$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:
$F\ P\ =\ \lambda m.\ m = 0\ \lor\ (\exists n.\ m = n + 2\ \land\ P\ n)$

Thus, $F\ P$ essentially applies the rules to P, i.e.,
$F\ P$ holds for exactly those items m that are produced
by applying the rules to items for which P holds.

$even$ is a fixpoint of F, in particular a post-fixpoint,
i.e., $even \le F\ even$.

This means $\forall m.\ even\ m \longrightarrow m = 0 \lor (\exists n.\ m = n + 2\ \land\ even\ n)$
i.e., whenever $even\ m$ holds, it must have been
obtained by one of the rules (Zero) and (Suc)
... leading to the following case distinction (elimination) rule for $even$:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2 \land even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ (\text{Zero}) \qquad \frac{even\ n}{even\ (n+2)}\ (\text{Suc})$$

"Inductively" means: smallest predicate closed under the given rules.
More precisely: We define $even = \mathsf{I}_F$, where
$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:
$F\ P\ =\ \lambda m.\ m = 0\ \lor\ (\exists n.\ m = n + 2\ \land\ P\ n)$

Thus, $F\ P$ essentially applies the rules to P, i.e.,
$F\ P$ holds for exactly those items m that are produced
by applying the rules to items for which P holds.

$even$ is the least among the pre-fixpoints of F,
i.e., for all $P : \mathbb{N} \to \text{Bool}$, $F\ P \leq P$ implies $even \leq P$

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{\cdot}{even\ 0}\ (\text{Zero}) \qquad \frac{even\ n}{even\ (n+2)}\ (\text{Suc})$$

"Inductively" means: smallest predicate closed under the given rules.
More precisely: We define $even = \mathsf{I}_F$, where
$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:
$F\ P\ =\ \lambda m.\ m = 0\ \lor\ (\exists n.\ m = n + 2\ \land\ P\ n)$

Thus, $F\ P$ essentially applies the rules to P, i.e., $F\ P$ holds for exactly those items m that are produced by applying the rules to items for which P holds.

$even$ is the least among the pre-fixpoints of F,
i.e., for all $P : \mathbb{N} \to \text{Bool}$, $F\ P \leq P$ implies $even \leq P$

This means that $even \leq P$ for all predicates P that are closed under the rules (Zero), (Suc) (i.e., $P\ 0$ holds, and $P\ n$ implies $P(n+2)$ for all n)

Making sense of the inductive specification of $even$

The predicate $even : \mathbb{N} \to \text{Bool}$ specified inductively by the following rules:

$$\frac{.}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

"Inductively" means: smallest predicate closed under the given rules.
More precisely: We define $even = \mathsf{I}_F$, where
$F : (\mathbb{N} \to \text{Bool}) \to (\mathbb{N} \to \text{Bool})$ is defined as follows, for all $P : \mathbb{N} \to \text{Bool}$:
$F\ P\ =\ \lambda m.\ m = 0\ \lor\ (\exists n.\ m = n + 2\ \land\ P\ n)$

Thus, $F\ P$ essentially applies the rules to P, i.e., $F\ P$ holds for exactly those items m that are produced by applying the rules to items for which P holds.

$even$ is the least among the pre-fixpoints of F,
i.e., for all $P : \mathbb{N} \to \text{Bool}$, $F\ P \le P$ implies $even \le P$

This means that $even \le P$ for all predicates P that are closed under the rules (Zero), (Suc) (i.e., $P\ 0$ holds, and $P\ n$ implies $P(n+2)$ for all n)
... leading to the following induction rule for $even$:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)

... and inferring various rules from this definition

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)

... and inferring various rules from this definition:

Thanks to	... we obtain
$even$ being a pre-fixpoint of F	the introduction rules (Zero), (Suc)

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition:

Thanks to	... we obtain
$even$ being a pre-fixpoint of F	the introduction rules (Zero), (Suc)
$even$ being a post-fixpoint of F	the case distinction rule (Cases)

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)

... and inferring various rules from this definition:

Thanks to	... we obtain
$even$ being a pre-fixpoint of F	the introduction rules (Zero), (Suc)
$even$ being a post-fixpoint of F	the case distinction rule (Cases)
$even$ being $\leq$ all pre-fixpoints of F	the induction rule (Induct)

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint l_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition:

Thanks to	... we obtain
$even$ being a pre-fixpoint of F	the introduction rules (Zero), (Suc)
$even$ being a post-fixpoint of F	the case distinction rule (Cases)
$even$ being $\leq$ all pre-fixpoints of F	the induction rule (Induct)

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2 \wedge even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint l_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition:

Thanks to	... we obtain
$even$ being a pre-fixpoint of F	the introduction rules (Zero), (Suc)
$even$ being a post-fixpoint of F	the case distinction rule (Cases)
$even$ being $\leq$ all pre-fixpoints of F	the induction rule (Induct)

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2 \wedge even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n \wedge P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

$even$ is also the least (pre-)fixpoint of
$G = \lambda P.\ F(even \wedge P) = \lambda P.\ F(\lambda n.\ even\ n \wedge P\ n)$

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Remember the operator for $even$:

$$F\ P\ =\ \lambda m.\ m = 0 \lor (\exists n.\ m = n + 2 \land P\ n)$$

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad\qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Remember the operator for $even$:

$$F\ P\ =\ \lambda m.\ m = 0\ \vee\ (\exists n.\ m = n + 2\ \wedge\ P\ n)$$

It is trivially monotonic!

$P \leq Q$
immediately implies
F $P \leq F\ Q$

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad\qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Remember the operator for $even$:

$$F\ P\ =\ \lambda m.\ m = 0\ \lor\ (\exists n.\ m = n + 2\ \land\ P\ n)$$

It is trivially monotonic!

$$\forall m.\ P\ m \longrightarrow Q\ m$$

immediately implies

$$\mathsf{F}\ P \leq F\ Q$$

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint l_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Remember the operator for $even$:

$$F\ P\ =\ \lambda m.\ m = 0\ \vee\ (\exists n.\ m = n + 2\ \wedge\ P\ n)$$

It is trivially monotonic!

$$\forall m.\ P\ m \longrightarrow Q\ m$$
immediately implies
$$\forall m.\ F\ P\ m \longrightarrow F\ Q\ m$$

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad\qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Remember the operator for $even$:

$$F\ P\ =\ \lambda m.\ m = 0\ \vee\ (\exists n.\ m = n + 2\ \wedge\ P\ n)$$

It is trivially monotonic!

$$\forall m.\ P\ m \longrightarrow Q\ m$$

immediately implies

$$\forall m.\ m = 0 \vee (\exists n.\ m = n + 2\ \wedge\ P\ n) \longrightarrow m = 0 \vee (\exists n.\ m = n + 2\ \wedge\ Q\ n)$$

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint l_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Remember the operator for $even$:

$$F\ P\ =\ \lambda m.\ m = 0\ \vee\ (\exists n.\ m = n + 2\ \wedge\ P\ n)$$

It is trivially monotonic!

$$\forall m.\ P\ m \longrightarrow Q\ m$$
immediately implies
$$\forall m.\ (\exists n.\ m = n + 2\ \wedge\ P\ n) \longrightarrow (\exists n.\ m = n + 2\ \wedge\ Q\ n)$$

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \text{How about...} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \text{How about...} \qquad \frac{even\ (n+2)}{even\ n}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \text{How about...} \qquad \frac{even\ (n+2)}{even\ n}\ \text{(Suc)} \qquad \checkmark$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \text{How about...} \qquad \frac{\forall m < n - 2.\ even\ m}{even\ n}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$\frac{\cdot}{even\ 0}$ (Zero) How about... $\frac{\forall m < n-2.\ even\ m}{even\ n}$ (Suc) ✓

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \text{How about...} \qquad \frac{\forall m < n - 2.\ \exists k \geq m.\ even\ k}{even\ n}\ \text{(Suc)}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \text{How about...} \qquad \frac{\forall m < n-2.\ \exists k \geq m.\ even\ k}{even\ n}\ \text{(Suc)} \quad ✓$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \text{How about...} \qquad \frac{\forall m < n - 2.\ \exists k \geq m.\ even\ k}{even\ n}\ \text{(Suc)} \quad \checkmark$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

If the rule hypotheses follow a "positive logic" ($\exists, \forall, \wedge, \vee$), then F is monotonic.

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \text{How about...} \qquad \frac{\neg\ even\ n}{even\ (n+2)}\ \text{(Suc)} \quad \mathbf{x}$$

... by defining $even$ as the least (pre-)fixpoint I_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

If the rule hypotheses follow a "positive logic" ($\exists, \forall, \wedge, \vee$), then F is monotonic.

Recipe for making sense of inductive specifications

We make sense of an inductive specification of a predicate such as

$\dfrac{\cdot}{even\ 0}$ (Zero) How about... $\dfrac{even\ n \longrightarrow even\ (n+1)}{even\ (n+2)}$ (Suc) **x**

... by defining $even$ as the least (pre-)fixpoint l_F of a monotonic operator F on predicates, where F is defined from the rules ($F\ P$ is the predicate obtained from applying the rules to the items satisfying P)
... and inferring various rules from this definition.

If the rule hypotheses follow a "positive logic" ($\exists, \forall, \wedge, \vee$), then F is monotonic.

Applying the recipe to the inductive specification of $subl$

The relation $subl : \mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool}$
specified inductively by the rules:

$$\frac{\cdot}{subl\ []\ as}\ \text{(Nil)} \qquad \frac{subl\ as\ as'}{subl\ as\ (a\#as')}\ \text{(ConsR)}$$

$$\frac{subl\ as\ as'}{subl\ (a\#as)\ (a\#as')}\ \text{(Cons)}$$

Applying the recipe to the inductive specification of $subl$

The relation $subl : \mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool}$
specified inductively by the rules:

$$\frac{\cdot}{subl\ []\ as}\ (\mathsf{Nil}) \qquad \frac{subl\ as\ as'}{subl\ as\ (a\#as')}\ (\mathsf{ConsR})$$

$$\frac{subl\ as\ as'}{subl\ (a\#as)\ (a\#as')}\ (\mathsf{Cons})$$

"Inductively" means: smallest relation closed under the given rules.

Applying the recipe to the inductive specification of $subl$

The relation $subl : \mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool}$
specified inductively by the rules:

$$\frac{\cdot}{subl\ []\ as}\ (\mathsf{Nil}) \qquad \frac{subl\ as\ as'}{subl\ as\ (a\#as')}\ (\mathsf{ConsR})$$

$$\frac{subl\ as\ as'}{subl\ (a\#as)\ (a\#as')}\ (\mathsf{Cons})$$

"Inductively" means: smallest relation closed under the given rules.
More precisely: We define $subl = \mathsf{I}_F$, where
$F : (\mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool}) \to (\mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool})$ is defined as follows, for all $R : \mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool}$:

Applying the recipe to the inductive specification of $subl$

The relation $subl : \mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool}$
specified inductively by the rules:

$$\frac{\cdot}{subl\ []\ as}\ (\mathsf{Nil}) \qquad \frac{subl\ as\ as'}{subl\ as\ (a\#as')}\ (\mathsf{ConsR})$$

$$\frac{subl\ as\ as'}{subl\ (a\#as)\ (a\#as')}\ (\mathsf{Cons})$$

"Inductively" means: smallest relation closed under the given rules.
More precisely: We define $subl = \mathsf{I}_F$, where
$F : (\mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool}) \to (\mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool})$ is defined as follows, for all $R : \mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool}$:

$$\begin{aligned} F\ R = \lambda bs, bs'.\ & \exists as.\ bs = [] \wedge bs' = as \\ & \vee \\ & \exists as, a, as'.\ bs = as \wedge bs' = a\#as' \wedge R\ as\ as' \\ & \vee \\ & \exists a, as, as'.\ bs = a\#as \wedge bs' = a\#as' \wedge R\ as\ as' \end{aligned}$$

Applying the recipe to the inductive specification of $subl$

The relation $subl : \mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool}$
specified inductively by the rules:

$$\frac{\cdot}{subl\ []\ as}\ (\mathsf{Nil}) \qquad \frac{subl\ as\ as'}{subl\ as\ (a\#as')}\ (\mathsf{ConsR})$$

$$\frac{subl\ as\ as'}{subl\ (a\#as)\ (a\#as')}\ (\mathsf{Cons})$$

"Inductively" means: smallest relation closed under the given rules.
More precisely: We define $subl = \mathsf{I}_F$, where
$F : (\mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool}) \to (\mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool})$ is defined as follows, for all $R : \mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool}$:

$$\begin{aligned} F\ R\ = \lambda bs, bs'.\ & \exists as.\ bs = [] \wedge bs' = as \\ & \vee \\ & \exists as, a, as'.\ bs = as \wedge bs' = a\#as' \wedge R\ as\ as' \\ & \vee \\ & \exists a, as, as'.\ bs = a\#as \wedge bs' = a\#as' \wedge R\ as\ as' \end{aligned}$$

Again, F is monotonic, so I_F exists by Knaster-Tarski.

Applying the recipe to the inductive specification of $subl$

Thanks to	... we obtain
$subl$ being a pre-fixpoint of F	the introduction rules (Nil), (ConsR), (Cons)
$subl$ being a post-fixpoint of F	the case distinction rule (Cases)
$subl$ being $\leq$ all pre-fixpoints of F	the induction rule (Induct)

Applying the recipe to the inductive specification of $subl$

Thanks to	... we obtain
$subl$ being a pre-fixpoint of F	the introduction rules (Nil), (ConsR), (Cons)
$subl$ being a post-fixpoint of F	the case distinction rule (Cases)
$subl$ being $\leq$ all pre-fixpoints of F	the induction rule (Induct)

$$\frac{\cdot}{subl\ []\ as}\ \text{(Nil)} \qquad \frac{subl\ as\ as'}{subl\ as\ (a\#as')}\ \text{(ConsR)}$$

$$\frac{subl\ as\ as'}{subl\ (a\#as)\ (a\#as')}\ \text{(Cons)}$$

Applying the recipe to the inductive specification of $subl$

Thanks to	... we obtain
$subl$ being a pre-fixpoint of F	the introduction rules (Nil), (ConsR), (Cons)
$subl$ being a post-fixpoint of F	the case distinction rule (Cases)
$subl$ being ≤ all pre-fixpoints of F	the induction rule (Induct)

$$\frac{\cdot}{subl\ []\ as}\ \text{(Nil)} \qquad \frac{subl\ as\ as'}{subl\ as\ (a\#as')}\ \text{(ConsR)}$$

$$\frac{subl\ as\ as'}{subl\ (a\#as)\ (a\#as')}\ \text{(Cons)}$$

$$\frac{\begin{array}{c} subl\ bs\ bs' \qquad \forall as.\ bs = [] \wedge bs' = as \longrightarrow P \\ \forall as, as', a.\ bs = as \wedge bs' = a\#as' \wedge subl\ as\ as' \longrightarrow P\ bs\ bs' \\ \forall as, as', a.\ bs = a\#as \wedge bs' = a\#as' \wedge subl\ as\ as' \longrightarrow P\ bs\ bs' \end{array}}{P\ bs\ bs'}\ \text{(Cases)}$$

Applying the recipe to the inductive specification of $subl$

Thanks to	... we obtain
$subl$ being a pre-fixpoint of F	the introduction rules (Nil), (ConsR), (Cons)
$subl$ being a post-fixpoint of F	the case distinction rule (Cases)
$subl$ being ≤ all pre-fixpoints of F	the induction rule (Induct)

$$\frac{\cdot}{subl\ []\ as}\ \text{(Nil)} \qquad \frac{subl\ as\ as'}{subl\ as\ (a\#as')}\ \text{(ConsR)}$$

$$\frac{subl\ as\ as'}{subl\ (a\#as)\ (a\#as')}\ \text{(Cons)}$$

$$\frac{\begin{array}{c} subl\ bs\ bs' \qquad\qquad \forall as.\ P\ []\ as \\ \forall as, as', a.\ subl\ as\ as' \wedge P\ as\ as' \longrightarrow P\ as\ (a\#as') \\ \forall as, as', a.\ subl\ as\ as' \wedge P\ as\ as' \longrightarrow P\ (a\#as)\ (a\#as') \end{array}}{P\ bs\ bs'}\ \text{(Induct)}$$

Applying the recipe to the inductive specification of $subl$

Thanks to	... we obtain
$subl$ being a pre-fixpoint of F	the introduction rules (Nil), (ConsR), (Cons)
$subl$ being a post-fixpoint of F	the case distinction rule (Cases)
$subl$ being ≤ all pre-fixpoints of F	the induction rule (Induct)

$$\frac{\cdot}{subl\ []\ as}\ \text{(Nil)} \qquad \frac{subl\ as\ as'}{subl\ as\ (a\#as')}\ \text{(ConsR)}$$

$$\frac{subl\ as\ as'}{subl\ (a\#as)\ (a\#as')}\ \text{(Cons)}$$

$$\frac{\begin{array}{c} subl\ bs\ bs' \qquad\qquad \forall as.\ P\ []\ as \\ \forall as, as', a.\ subl\ as\ as' \wedge P\ as\ as' \longrightarrow P\ as\ (a\#as') \\ \forall as, as', a.\ subl\ as\ as' \wedge P\ as\ as' \longrightarrow P\ (a\#as)\ (a\#as') \end{array}}{P\ bs\ bs'}\ \text{(Induct)}$$

$subl$ is also the least (pre-)fixpoint of
$G = \lambda P.\ F(subl \wedge P) = \lambda P.\ F(\lambda as, as'.\ subl\ as\ as' \wedge P\ as\ as')$.

Summary of the semantic approach to inductive definitions

We specify an inductive predicate/relation P by indicating rules involving P

Summary of the semantic approach to inductive definitions

We specify an inductive predicate/relation P by indicating rules involving P – this is not yet a definition!

Summary of the semantic approach to inductive definitions

We specify an inductive predicate/relation P by indicating rules involving P – this is not yet a definition!

We turn the specification into an actual (non-inductive!) definition by:

- extracting an operator F on predicates/relations from these rules
- showing that F is monotonic – which is trivial if the rules' premises have a "positive logic" format
- defining P as I_F, the least (pre-)fixpoint of F

Summary of the semantic approach to inductive definitions

We specify an inductive predicate/relation P by indicating rules involving P – this is not yet a definition!

We turn the specification into an actual (non-inductive!) definition by:

- extracting an operator F on predicates/relations from these rules
- showing that F is monotonic – which is trivial if the rules' premises have a "positive logic" format
- defining P as l_F, the least (pre-)fixpoint of F

Finally, from the definition of P as least (pre-)fixpoint, we infer:

- introduction rules – which coincide with the originally specified rules
- a case distinction rule
- an induction rule

The Isabelle/HOL implementation of the approach

We, the users, specify an inductive predicate/relation P by indicating rules involving P – this is not yet a definition!

She turns the specification into an actual (non-inductive!) definition by:

- extracting an operator F on predicates/relations from these rules
- showing that F is monotonic – which is trivial if the rules' premises have a "positive logic" format
- defining P as l_F, the least (pre-)fixpoint of F

Finally, from the definition of P as least (pre-)fixpoint, she infers:

- introduction rules – which coincide with the originally specified rules
- a case distinction rule
- an induction rule

Isabelle automates this approach.

The Isabelle/HOL implementation of the approach

We, the users, specify an inductive predicate/relation P by indicating rules involving P – this is not yet a definition!

She turns the specification into an actual (non-inductive!) definition by:

- extracting an operator F on predicates/relations from these rules
- showing that F is monotonic – which is trivial if the rules' premises have a "positive logic" format; if F is not obviously monotonic and Isabelle fails to prove this, users can help by providing "hints"
- defining P as I_F, the least (pre-)fixpoint of F

Finally, from the definition of P as least (pre-)fixpoint, she infers:

- introduction rules – which coincide with the originally specified rules
- a case distinction rule
- an induction rule

Isabelle automates this approach.

1. For the predicate $even$:

   (i) Infer the case distinction rule for the predicate from the introduction rules and the induction rule.

   (ii) Show that the introduction and induction rules determine the predicate uniquely, i.e., there is only one predicate satisfying them.

2. Let $(A, \leq)$ be a partially ordered set and $F : A \to A$ a monotonic function. Show that, if it exists, then the least pre-fixpoint of F is also a post-fixpoint of F.

3. What is the connection between points 1(i) and 2 above?

4. Show that the previously mentioned "optimization" of induction is correct: If $(A, \leq)$ is a complete lattice and $F : A \to A$ a monotonic function, then I_F (the smallest (pre-)fixpoint of F) is also the smallest pre-fixpoint of the operator $G = \lambda a.\ F\ (\mathsf{I}_F \wedge a)$.

5. Dualize points (2)-(4) above into statements about greatest (post-)fixpoints.

Reasoning about inductive predicates

Reasoning about inductive predicates

We'll use the inductive predicate $even : \mathbb{N} \to \mathsf{Bool}$ as running example, but the ideas apply generally.

Proving that an inductive predicate holds

Introduction rules:
$$\frac{\cdot}{\mathit{even}\ 0}\ \text{(Zero)} \qquad \frac{\mathit{even}\ n}{\mathit{even}\ (n+2)}\ \text{(Suc)}$$

Case distinction rule:

$$\frac{\mathit{even}\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ \mathit{even}\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{\mathit{even}\ m \qquad P\ 0 \qquad \forall n.\ \mathit{even}\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $\mathit{even}\ 4$ hold?

Proving that an inductive predicate holds

Introduction rules: $$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $even\ 4$ hold? Reason "backwards" using the introduction rules:

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\qquad \dfrac{even\ n}{even\ (n+2)}$ (Suc)

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $even\ 4$ hold? Reason "backwards" using the introduction rules:

- We must prove $even\ 4$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\qquad \dfrac{even\ n}{even\ (n+2)}$ (Suc)

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $even\ 4$ hold? Reason "backwards" using the introduction rules:

- We must prove $even\ 4$.
- Applying rule (Suc), suffices to prove $even\ 2$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\qquad \dfrac{even\ n}{even\ (n+2)}$ (Suc)

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n+2\ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $even\ 4$ hold? Reason "backwards" using the introduction rules:

- We must prove $even\ 4$.
- Applying rule (Suc), suffices to prove $even\ 2$.
- Applying again rule (Suc), suffices to prove $even\ 0$.

Proving that an inductive predicate holds

Introduction rules:

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $even\ 4$ hold? Reason "backwards" using the introduction rules:

- We must prove $even\ 4$.
- Applying rule (Suc), suffices to prove $even\ 2$.
- Applying again rule (Suc), suffices to prove $even\ 0$.
- And the last holds by rule (Zero).

Using the assumption that an inductive predicate holds

Introduction rules: $$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2 \ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n \ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $\neg\ even\ 3$ hold?

Using the assumption that an inductive predicate holds

Introduction rules:

$$\frac{\cdot}{\mathit{even}\ 0}\ \text{(Zero)} \qquad \frac{\mathit{even}\ n}{\mathit{even}\ (n+2)}\ \text{(Suc)}$$

Case distinction rule:

$$\frac{\mathit{even}\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ \mathit{even}\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{\mathit{even}\ m \qquad P\ 0 \qquad \forall n.\ \mathit{even}\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $\neg\ \mathit{even}\ 3$ hold? Rephrase the statement as $\mathit{even}\ 3 \longrightarrow \bot$ and again reason backwards.

Using the assumption that an inductive predicate holds: case distinction

Introduction rules: $$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n+2\ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $\neg\ even\ 3$ hold? Rephrase the statement as $even\ 3 \longrightarrow \bot$ and again reason backwards.

- Apply the case rule for $m = 3$ and $P = \bot$, reducing our goal to:

Using the assumption that an inductive predicate holds: case distinction

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $\neg\ even\ 3$ hold? Rephrase the statement as $even\ 3 \longrightarrow \bot$ and again reason backwards.

- Apply the case rule for $m = 3$ and $P = \bot$, reducing our goal to:

— $3 = 0 \longrightarrow \bot$, which is trivially true;

Using the assumption that an inductive predicate holds: case distinction

Introduction rules: $$\frac{\cdot}{\mathit{even}\ 0}\ \text{(Zero)} \qquad \frac{\mathit{even}\ n}{\mathit{even}\ (n+2)}\ \text{(Suc)}$$

Case distinction rule:

$$\frac{\mathit{even}\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n+2\ \wedge\ \mathit{even}\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{\mathit{even}\ m \qquad P\ 0 \qquad \forall n.\ \mathit{even}\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $\neg\ \mathit{even}\ 3$ hold? Rephrase the statement as $\mathit{even}\ 3 \longrightarrow \bot$ and again reason backwards.

- Apply the case rule for $m = 3$ and $P = \bot$, reducing our goal to:
  - $3 = 0 \longrightarrow \bot$, which is trivially true;
  - $\forall n.\ 3 = n+2\ \wedge\ \mathit{even}\ n \longrightarrow \bot$, which means $\mathit{even}\ 1 \longrightarrow \bot$.

Using the assumption that an inductive predicate holds: case distinction

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2 \wedge even\ n \longrightarrow P}{P} \text{ (Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n \wedge P\ n \longrightarrow P\ (n+2)}{P\ m} \text{ (Induct)}$$

Why does $\neg\ even\ 3$ hold? Rephrase the statement as $even\ 3 \longrightarrow \bot$ and again reason backwards.

- Apply the case rule for $m = 3$ and $P = \bot$, reducing our goal to:

— $3 = 0 \longrightarrow \bot$, which is trivially true;

— $\forall n.\ 3 = n + 2 \wedge even\ n \longrightarrow \bot$, which means $even\ 1 \longrightarrow \bot$.

— Apply the case rule for $m = 1$ and $P = \bot$, reducing our goal to:

Using the assumption that an inductive predicate holds: case distinction

Introduction rules:

$$\frac{\cdot}{\mathit{even}\ 0}\ \text{(Zero)} \qquad \frac{\mathit{even}\ n}{\mathit{even}\ (n+2)}\ \text{(Suc)}$$

Case distinction rule:

$$\frac{\mathit{even}\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n+2\ \wedge\ \mathit{even}\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{\mathit{even}\ m \qquad P\ 0 \qquad \forall n.\ \mathit{even}\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $\neg\ \mathit{even}\ 3$ hold? Rephrase the statement as $\mathit{even}\ 3 \longrightarrow \bot$ and again reason backwards.

- Apply the case rule for $m = 3$ and $P = \bot$, reducing our goal to:

— $3 = 0 \longrightarrow \bot$, which is trivially true;

— $\forall n.\ 3 = n+2\ \wedge\ \mathit{even}\ n \longrightarrow \bot$, which means $\mathit{even}\ 1 \longrightarrow \bot$.

— Apply the case rule for $m = 1$ and $P = \bot$, reducing our goal to:

—— $1 = 0 \longrightarrow \bot$, which is trivially true;

Using the assumption that an inductive predicate holds: case distinction

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\qquad \dfrac{even\ n}{even\ (n+2)}$ (Suc)

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $\neg\ even\ 3$ hold? Rephrase the statement as $even\ 3 \longrightarrow \bot$ and again reason backwards.

- Apply the case rule for $m = 3$ and $P = \bot$, reducing our goal to:

— $3 = 0 \longrightarrow \bot$, which is trivially true;

— $\forall n.\ 3 = n + 2\ \wedge\ even\ n \longrightarrow \bot$, which means $even\ 1 \longrightarrow \bot$.

— Apply the case rule for $m = 1$ and $P = \bot$, reducing our goal to:

—— $1 = 0 \longrightarrow \bot$, which is trivially true;

—— $\forall n.\ 1 = n + 2\ \wedge\ even\ n \longrightarrow \bot$, which is trivially true.

A closer look at a case distinction step

$$\frac{\textit{even}\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ \textit{even}\ n \longrightarrow P}{P}\ \text{(Cases)}$$

- What to prove $\textit{even}\ 3 \longrightarrow \bot$.
- Apply the case rule for $m = 3$ and $P = \bot$, reducing our goal to:
— $3 = 0 \longrightarrow \bot$
— $\forall n.\ 3 = n + 2\ \wedge\ \textit{even}\ n \longrightarrow \bot$

A closer look at a case distinction step

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2 \land even\ n \longrightarrow P}{P} \text{ (Cases)}$$

- What to prove $even\ 3 \longrightarrow \bot$.
- Apply the case rule for $m = 3$ and $P = \bot$, reducing our goal to:

— $3 = 0 \longrightarrow \bot$

— $\forall n.\ 3 = n + 2 \land even\ n \longrightarrow \bot$

We match major premise and conclusion against what we need to prove

A closer look at a case distinction step

$$\frac{\textit{even}\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ \textit{even}\ n \longrightarrow P}{P}\ \text{(Cases)}$$

- What to prove $\textit{even}\ 3 \longrightarrow \bot$.
- Apply the case rule for $m = 3$ and $P = \bot$, reducing our goal to:

— $3 = 0 \longrightarrow \bot$

— $\forall n.\ 3 = n + 2\ \wedge\ \textit{even}\ n \longrightarrow \bot$

We match major premise and conclusion against what we need to prove ... which gives us the instantiation

A closer look at a case distinction step

$$\frac{\mathit{even}\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2 \wedge \mathit{even}\ n \longrightarrow P}{P}\ \text{(Cases)}$$

- What to prove $\mathit{even}\ 3 \longrightarrow \bot$.
- Apply the case rule for $m = 3$ and $P = \bot$, reducing our goal to:

— $3 = 0 \longrightarrow \bot$

— $\forall n.\ 3 = n + 2 \wedge \mathit{even}\ n \longrightarrow \bot$

We match major premise and conclusion against what we need to prove
... which gives us the instantiation
... and we are left to prove the instances of the other premises

A closer look at a case distinction step

$$\frac{\mathit{even}\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2 \wedge \mathit{even}\ n \longrightarrow P}{P}\ \text{(Cases)}$$

- What to prove $\mathit{even}\ 3 \longrightarrow \bot$.
- Apply the case rule for $m = 3$ and $P = \bot$, reducing our goal to:

— $3 = 0 \longrightarrow \bot$

— $\forall n.\ 3 = n + 2 \wedge \mathit{even}\ n \longrightarrow \bot$

We match major premise and conclusion against what we need to prove
... which gives us the instantiation
... and we are left to prove the instances of the other premises

This is the elimination reasoning pattern.

Using the assumption that an inductive predicate holds

Introduction rules:

$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $even$ capture the notion of even number?

Using the assumption that an inductive predicate holds

Introduction rules:

$$\frac{\cdot}{even\ 0}\ (\text{Zero}) \qquad \frac{even\ n}{even\ (n+2)}\ (\text{Suc})$$

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ even\ n \longrightarrow P}{P}\ (\text{Cases})$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ (\text{Induct})$$

Why does $even$ capture the notion of even number?

Let's prove that $even\ m \longrightarrow \exists k.\ m = 2 * k$, reasoning backwards.

Using the assumption that an inductive predicate holds: induction

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Case distinction rule:

$$\dfrac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ even\ n \longrightarrow P}{P} \text{ (Cases)}$$

Induction rule:

$$\dfrac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m} \text{ (Induct)}$$

Why does $even$ capture the notion of even number?

Let's prove that $even\ m \longrightarrow \exists k.\ m = 2 * k$, reasoning backwards.

- Apply the induction rule for $P = \lambda m.\ \exists k.\ m = 2 * k$, reducing our goal to:

Using the assumption that an inductive predicate holds: induction

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\qquad \dfrac{even\ n}{even\ (n+2)}$ (Suc)

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does *even* capture the notion of even number?

Let's prove that $even\ m \longrightarrow \exists k.\ m = 2 * k$, reasoning backwards.

- Apply the induction rule for $P = \lambda m.\ \exists k.\ m = 2 * k$, reducing our goal to:

— $\exists k.\ 0 = 2 * k$

Using the assumption that an inductive predicate holds: induction

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\qquad \dfrac{even\ n}{even\ (n+2)}$ (Suc)

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $even$ capture the notion of even number?

Let's prove that $even\ m \longrightarrow \exists k.\ m = 2 * k$, reasoning backwards.

- Apply the induction rule for $P = \lambda m.\ \exists k.\ m = 2 * k$, reducing our goal to:

— $\exists k.\ 0 = 2 * k$, which is true, taking $k = 0$;

Using the assumption that an inductive predicate holds: induction

Introduction rules:
$$\frac{\cdot}{even\ 0}\ \text{(Zero)} \qquad \frac{even\ n}{even\ (n+2)}\ \text{(Suc)}$$

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \land\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \land\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $even$ capture the notion of even number?

Let's prove that $even\ m \longrightarrow \exists k.\ m = 2 * k$, reasoning backwards.

- Apply the induction rule for $P = \lambda m.\ \exists k.\ m = 2 * k$, reducing our goal to:

— $\exists k.\ 0 = 2 * k$, which is true, taking $k = 0$;

— $\forall n.\ even\ n \land (\exists k.\ n = 2 * k) \longrightarrow (\exists k.\ n + 2 = 2 * k)$

Using the assumption that an inductive predicate holds: induction

Introduction rules: $$\frac{\cdot}{even\ 0}\ (\text{Zero}) \qquad \frac{even\ n}{even\ (n+2)}\ (\text{Suc})$$

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n+2\ \wedge\ even\ n \longrightarrow P}{P}\ (\text{Cases})$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ (\text{Induct})$$

Why does $even$ capture the notion of even number?

Let's prove that $even\ m \longrightarrow \exists k.\ m = 2 * k$, reasoning backwards.

- Apply the induction rule for $P = \lambda m.\ \exists k.\ m = 2 * k$, reducing our goal to:

— $\exists k.\ 0 = 2 * k$, which is true, taking $k = 0$;

— $\forall n.\ even\ n \wedge (\exists k.\ n = 2 * k) \longrightarrow (\exists k.\ n + 2 = 2 * k)$,
which means $\forall n, k.\ even\ n \wedge n = 2 * k \longrightarrow (\exists k'.\ n + 2 = 2 * k')$

Using the assumption that an inductive predicate holds: induction

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\qquad \dfrac{even\ n}{even\ (n+2)}$ (Suc)

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n+2\ \wedge\ even\ n \longrightarrow P}{P}\ \text{(Cases)}$$

Induction rule:

$$\frac{even\ m \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{P\ m}\ \text{(Induct)}$$

Why does $even$ capture the notion of even number?

Let's prove that $even\ m \longrightarrow \exists k.\ m = 2 * k$, reasoning backwards.

- Apply the induction rule for $P = \lambda m.\ \exists k.\ m = 2 * k$, reducing our goal to:

— $\exists k.\ 0 = 2 * k$, which is true, taking $k = 0$;

— $\forall n.\ even\ n \wedge (\exists k.\ n = 2 * k) \longrightarrow (\exists k.\ n+2 = 2 * k)$,
which means $\forall n, k.\ even\ n \wedge n = 2 * k \longrightarrow (\exists k'.\ n+2 = 2 * k')$,
which is true, taking $k' = k + 1$.

Using the assumption that an inductive predicate holds: induction

Introduction rules:

$$\frac{\cdot}{even\ 0}\ (\text{Zero}) \qquad \frac{even\ n}{even\ (n+2)}\ (\text{Suc})$$

Case distinction rule:

$$\frac{even\ m \qquad m = 0 \longrightarrow P \qquad \forall n.\ m = n + 2\ \wedge\ even\ n \longrightarrow P}{P}\ (\text{Cases})$$

Induction rule:

$$\frac{\boxed{even\ m} \qquad P\ 0 \qquad \forall n.\ even\ n\ \wedge\ P\ n \longrightarrow P\ (n+2)}{\boxed{P\ m}}\ (\text{Induct})$$

Why does *even* capture the notion of even number?

Let's prove that $\boxed{even\ m} \longrightarrow \boxed{\exists k.\ m = 2 * k}$, reasoning backwards.

- Apply the induction rule for $P = \lambda m.\ \exists k.\ m = 2 * k$, reducing our goal to:

— $\exists k.\ 0 = 2 * k$, which is true, taking $k = 0$;

— $\forall n.\ even\ n \wedge (\exists k.\ n = 2 * k) \longrightarrow (\exists k.\ n + 2 = 2 * k)$,
which means $\forall n, k.\ even\ n \wedge n = 2 * k \longrightarrow (\exists k'.\ n + 2 = 2 * k')$,
which is true, taking $k' = k + 1$.

Again, the elimination reasoning pattern.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Why does $even$ capture the notion of even number?

Let's now prove the converse implication, $(\exists k.\ m = 2 * k) \longrightarrow even\ m$.
which means $\forall k.\ m = 2 * k \longrightarrow even\ m$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Why does $even$ capture the notion of even number?

Let's now prove the converse implication, $(\exists k.\ m = 2 * k) \longrightarrow even\ m$.
which means $\forall k.\ m = 2 * k \longrightarrow even\ m$.
Let $Q = \lambda k.\ \forall m.\ m = 2 * k \longrightarrow even\ m$.
The proof of $Q\ k$ is by structural induction on $k \in \mathbb{N}$ (unrelated to $even$)

— $Q\ 0$ means $m = 2 * 0 \longrightarrow even\ m$,
which means $even\ 0$, which is true by (Zero).

— $Q\ k \longrightarrow Q\ (k+1)$ means
$(\forall m.\ m = 2 * k \longrightarrow even\ m) \longrightarrow (\forall m.\ m = 2 * (k+1) \longrightarrow even\ m)$,
which means
$\forall m'.\ (\forall m.\ m = 2 * k \longrightarrow even\ m) \wedge m' = 2 * (k+1) \longrightarrow even\ m'$.
Fixing m', we have $m' = 2 * k + 2$.
Taking $m = 2 * k$, we have $even\ m$.
Applying (Suc), we obtain $even\ (m+2)$, i.e., $even\ m'$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Why does $even$ capture the notion of even number?

Let's now prove the converse implication, $(\exists k.\ m = 2 * k) \longrightarrow even\ m$.
which means $\forall k.\ m = 2 * k \longrightarrow even\ m$.
Let $Q = \lambda k.\ \forall m.\ m = 2 * k \longrightarrow even\ m$.
The proof of $Q\ k$ is by structural induction on $k \in \mathbb{N}$ (unrelated to $even$)

— $Q\ 0$ means $m = 2 * 0 \longrightarrow even\ m$,
which means $even\ 0$, which is true by (Zero).

— $Q\ k \longrightarrow Q\ (k+1)$ means
$(\forall m.\ m = 2 * k \longrightarrow even\ m) \longrightarrow (\forall m.\ m = 2 * (k+1) \longrightarrow even\ m)$,
which means
$\forall m'.\ (\forall m.\ m = 2 * k \longrightarrow even\ m) \land m' = 2 * (k+1) \longrightarrow even\ m'$.
Fixing m', we have $m' = 2 * k + 2$.
Taking $m = 2 * k$, we have $even\ m$.
Applying (Suc), we obtain $even\ (m+2)$, i.e., $even\ m'$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Why does $even$ capture the notion of even number?

Let's now prove the converse implication, $(\exists k.\ m = 2 * k) \longrightarrow even\ m$.
which means $\forall k.\ m = 2 * k \longrightarrow even\ m$.
Let $Q = \lambda k.\ \forall m.\ m = 2 * k \longrightarrow even\ m$.
The proof of $Q\ k$ is by structural induction on $k \in \mathbb{N}$ (unrelated to $even$)
— $Q\ 0$ means $m = 2 * 0 \longrightarrow even\ m$,
which means $even\ 0$, which is true by (Zero).
— $Q\ k \longrightarrow Q\ (k+1)$ means
$(\forall m.\ m = 2 * k \longrightarrow even\ m) \longrightarrow (\forall m.\ m = 2 * (k+1) \longrightarrow even\ m)$,
which means
$\forall m'.\ (\forall m.\ m = 2 * k \longrightarrow even\ m) \wedge m' = 2 * (k+1) \longrightarrow even\ m'$.
Fixing m', we have $m' = 2 * k + 2$.
Taking $m = 2 * k$, we have $even\ m$.
Applying (Suc), we obtain $even\ (m+2)$, i.e., $even\ m'$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Why does $even$ capture the notion of even number?

Let's now prove the converse implication, $(\exists k.\ m = 2 * k) \longrightarrow even\ m$.
which means $\forall k.\ m = 2 * k \longrightarrow even\ m$.
Let $Q = \lambda k.\ \forall m.\ m = 2 * k \longrightarrow even\ m$.
The proof of $Q\ k$ is by structural induction on $k \in \mathbb{N}$ (unrelated to $even$)

— $Q\ 0$ means $m = 2 * 0 \longrightarrow even\ m$,
which means $even\ 0$, which is true by (Zero).

— $Q\ k \longrightarrow Q\ (k+1)$ means
$(\forall m.\ m = 2 * k \longrightarrow even\ m) \longrightarrow (\forall m.\ m = 2 * (k+1) \longrightarrow even\ m)$,
which means
$\forall m'.\ (\forall m.\ m = 2 * k \longrightarrow even\ m) \wedge m' = 2 * (k+1) \longrightarrow even\ m'$.
Fixing m', we have $m' = 2 * k + 2$.
Taking $m = 2 * k$, we have $even\ m$.
Applying (Suc), we obtain $even\ (m+2)$, i.e., $even\ m'$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Why does $even$ capture the notion of even number?

Let's now prove the converse implication, $(\exists k.\ m = 2 * k) \longrightarrow even\ m$.
which means $\forall k.\ m = 2 * k \longrightarrow even\ m$.
Let $Q = \lambda k.\ \forall m.\ m = 2 * k \longrightarrow even\ m$.
The proof of $Q\ k$ is by structural induction on $k \in \mathbb{N}$ (unrelated to $even$)

— $Q\ 0$ means $m = 2 * 0 \longrightarrow even\ m$,
which means $even\ 0$, which is true by (Zero).

— $Q\ k \longrightarrow Q\ (k+1)$ means
$(\forall m.\ m = 2 * k \longrightarrow even\ m) \longrightarrow (\forall m.\ m = 2 * (k+1) \longrightarrow even\ m)$,
which means
$\forall m'.\ (\forall m.\ m = 2 * k \longrightarrow even\ m) \wedge m' = 2 * (k+1) \longrightarrow even\ m'$.
Fixing m', we have $m' = 2 * k + 2$.
Taking $m = 2 * k$, we have $even\ m$.
Applying (Suc), we obtain $even\ (m+2)$, i.e., $even\ m'$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Why does *even* capture the notion of even number?

Let's now prove the converse implication, $(\exists k.\ m = 2 * k) \longrightarrow even\ m$.
which means $\forall k.\ m = 2 * k \longrightarrow even\ m$.
Let $Q = \lambda k.\ \forall m.\ m = 2 * k \longrightarrow even\ m$.
The proof of $Q\ k$ is by structural induction on $k \in \mathbb{N}$ (unrelated to *even*)

— $Q\ 0$ means $m = 2 * 0 \longrightarrow even\ m$,
which means $even\ 0$, which is true by (Zero).

— $Q\ k \longrightarrow Q\ (k+1)$ means
$(\forall m.\ m = 2 * k \longrightarrow even\ m) \longrightarrow (\forall m.\ m = 2 * (k+1) \longrightarrow even\ m)$,
which means
$\forall m'.\ (\forall m.\ m = 2 * k \longrightarrow even\ m) \wedge m' = 2 * (k+1) \longrightarrow even\ m'$.
Fixing m', we have $m' = 2 * k + 2$.
Taking $m = 2 * k$, we have $even\ m$.
Applying (Suc), we obtain $even\ (m+2)$, i.e., $even\ m'$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\qquad \dfrac{even\ n}{even\ (n+2)}$ (Suc)

Why does $even$ capture the notion of even number?

Let's now prove the converse implication, $(\exists k.\ m = 2 * k) \longrightarrow even\ m$.
which means $\forall k.\ m = 2 * k \longrightarrow even\ m$.
Let $Q = \lambda k.\ \forall m.\ m = 2 * k \longrightarrow even\ m$.
The proof of $Q\ k$ is by structural induction on $k \in \mathbb{N}$ (unrelated to $even$)

— $Q\ 0$ means $m = 2 * 0 \longrightarrow even\ m$,
which means $even\ 0$, which is true by (Zero).

— $Q\ k \longrightarrow Q\ (k+1)$ means
$(\forall m.\ m = 2 * k \longrightarrow even\ m) \longrightarrow (\forall m.\ m = 2 * (k+1) \longrightarrow even\ m)$,
which means
$\forall m'.\ (\forall m.\ m = 2 * k \longrightarrow even\ m) \wedge m' = 2 * (k+1) \longrightarrow even\ m'$.
Fixing m', we have $m' = 2 * k + 2$.
Taking $m = 2 * k$, we have $even\ m$.
Applying (Suc), we obtain $even\ (m+2)$, i.e., $even\ m'$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\qquad \dfrac{even\ n}{even\ (n+2)}$ (Suc)

Why does $even$ capture the notion of even number?

Let's now prove the converse implication, $(\exists k.\ m = 2 * k) \longrightarrow even\ m$.
which means $\forall k.\ m = 2 * k \longrightarrow even\ m$.
Let $Q = \lambda k.\ \forall m.\ m = 2 * k \longrightarrow even\ m$.
The proof of $Q\ k$ is by structural induction on $k \in \mathbb{N}$ (unrelated to $even$)
— $Q\ 0$ means $m = 2 * 0 \longrightarrow even\ m$,
which means $even\ 0$, which is true by (Zero).
— $Q\ k \longrightarrow Q\ (k+1)$ means
$(\forall m.\ m = 2 * k \longrightarrow even\ m) \longrightarrow (\forall m.\ m = 2 * (k+1) \longrightarrow even\ m)$,
which means
$\forall m'.\ (\forall m.\ m = 2 * k \longrightarrow even\ m) \wedge m' = 2 * (k+1) \longrightarrow even\ m'$.
Fixing m', we have $m' = 2 * k + 2$.
Taking $m = 2 * k$, we have $even\ m$.
Applying (Suc), we obtain $even\ (m+2)$, i.e., $even\ m'$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Why does $even$ capture the notion of even number?

Let's now prove the converse implication, $(\exists k.\ m = 2 * k) \longrightarrow even\ m$.
which means $\forall k.\ m = 2 * k \longrightarrow even\ m$.
Let $Q = \lambda k.\ \forall m.\ m = 2 * k \longrightarrow even\ m$.
The proof of $Q\ k$ is by structural induction on $k \in \mathbb{N}$ (unrelated to $even$)

— $Q\ 0$ means $m = 2 * 0 \longrightarrow even\ m$,
  which means $even\ 0$, which is true by (Zero).

— $Q\ k \longrightarrow Q\ (k+1)$ means
  $(\forall m.\ m = 2 * k \longrightarrow even\ m) \longrightarrow (\forall m.\ m = 2 * (k+1) \longrightarrow even\ m)$,
  which means
  $\forall m'.\ (\forall m.\ m = 2 * k \longrightarrow even\ m) \wedge m' = 2 * (k+1) \longrightarrow even\ m'$.
  Fixing m', we have $m' = 2 * k + 2$.
  Taking $m = 2 * k$, we have $even\ m$.
  Applying (Suc), we obtain $even\ (m+2)$, i.e., $even\ m'$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Why does $even$ capture the notion of even number?

Let's now prove the converse implication, $(\exists k.\ m = 2 * k) \longrightarrow even\ m$.
which means $\forall k.\ m = 2 * k \longrightarrow even\ m$.
Let $Q = \lambda k.\ \forall m.\ m = 2 * k \longrightarrow even\ m$.
The proof of $Q\ k$ is by structural induction on $k \in \mathbb{N}$ (unrelated to $even$)

— $Q\ 0$ means $m = 2 * 0 \longrightarrow even\ m$,
which means $even\ 0$, which is true by (Zero).

— $Q\ k \longrightarrow Q\ (k+1)$ means
$(\forall m.\ m = 2 * k \longrightarrow even\ m) \longrightarrow (\forall m.\ m = 2 * (k+1) \longrightarrow even\ m)$,
which means
$\forall m'.\ (\forall m.\ m = 2 * k \longrightarrow even\ m) \wedge m' = 2 * (k+1) \longrightarrow even\ m'$.
Fixing m', we have $m' = 2 * k + 2$.
Taking $m = 2 * k$, we have $even\ m$.
Applying (Suc), we obtain $even\ (m+2)$, i.e., $even\ m'$.

Proving that an inductive predicate holds

Introduction rules: $\dfrac{\cdot}{even\ 0}$ (Zero) $\dfrac{even\ n}{even\ (n+2)}$ (Suc)

Why does $even$ capture the notion of even number?

Let's now prove the converse implication, $(\exists k.\ m = 2 * k) \longrightarrow even\ m$.
which means $\forall k.\ m = 2 * k \longrightarrow even\ m$.
Let $Q = \lambda k.\ \forall m.\ m = 2 * k \longrightarrow even\ m$.
The proof of $Q\ k$ is by structural induction on $k \in \mathbb{N}$ (unrelated to $even$)

— $Q\ 0$ means $m = 2 * 0 \longrightarrow even\ m$,
  which means $even\ 0$, which is true by (Zero).

— $Q\ k \longrightarrow Q\ (k+1)$ means
  $(\forall m.\ m = 2 * k \longrightarrow even\ m) \longrightarrow (\forall m.\ m = 2 * (k+1) \longrightarrow even\ m)$,
  which means
  $\forall m'.\ (\forall m.\ m = 2 * k \longrightarrow even\ m) \wedge m' = 2 * (k+1) \longrightarrow even\ m'$.
  Fixing m', we have $m' = 2 * k + 2$.
  Taking $m = 2 * k$, we have $even\ m$.
  Applying (Suc), we obtain $even\ (m+2)$, i.e., $even\ m'$.

Summary on reasoning about inductive predicates

An inductive predicate has introduction rules, a case distinction rule and an induction rule.

We use the introduction rules to prove that an inductive predicate holds. Examples:

- $even\ 4$
- $(\exists k.\ m = 2 * k) \longrightarrow even\ m$

We use the case distinction rule and the induction rule following the elimination reasoning pattern to prove something under the assumption that an inductive predicate holds. Examples:

- $\neg\ even\ 3$, i.e., $even\ 3 \longrightarrow \bot$
- $even\ m \longrightarrow (\exists k.\ m = 2 * k)$

1. Consider the inductive predicate $subl$ we defined before. Show the following:

- $subl\ [a, c]\ [a, b, c]$
- $\neg\ subl\ [a, b, c]\ [a, c]$
- $subl\ as\ as' \longrightarrow \mathsf{set}\ as \subseteq \mathsf{set}\ as'$, where the operator $\mathsf{set} : \mathsf{List}(A) \to \mathcal{P}(A)$ gives all the elements appearing in a list.

2. Assume that, in our informal example 3, we define $subll : \mathsf{List}(A) \to \mathsf{List}(A) \to \mathsf{Bool}$ inductively by the rules indicated there. Show that $subll\ as\ as'$ implies that as is a finite lazylist.