# The Use of the Coinduction Hypothesis in Coinductive Proofs

Anton Setzer
Swansea University
TYPES 2016, Novi Sad, Serbia

23 May 2016

# Transfer from Type Theory to Ordinary Mathematics

- When proving a property $\forall n : \mathbb{N}.\varphi(n)$ we don't use directly that $\mathbb{N}$ is least set closed under 0 and $\mathrm{S}$.
  - We don't define
$$A := \{n \in \mathbb{N} \mid \varphi(n)\}$$
    and show that $A$ is closed under 0 and $\mathrm{S}$.
- Instead we use proofs by induction.
  - Proofs by induction are essentially **recursive proofs** with **restrictions** on the use of the **IH**.
- **Goal of Talk**
  - Define proofs by coinduction as corecursive proofs with restrictions on the use of the **co-IH**.
  - Introduce methodology for carrying out proofs in **ordinary mathematics**.
  - Based on theory of coinductive proofs in type theory.

# Desired Coinductive Proof

- Consider an unlabelled Transition system:

$$\begin{tikzcd} * \arrow[loop above] & 0 \arrow[r, bend left] & 1 \arrow[r, bend left] & 2 \arrow[r, bend left] & \cdots \end{tikzcd}$$

- Textbook proof:
  Define $R := \{(*, n) \mid n \in \mathbb{N}\}$.
  Show that $R$ is a bisimulation relation, i.e. closed under elimination.
- A proof of $\forall n : \mathbb{N}. * \sim n$ by **coinduction** should be as follows:
  - We show $\forall n : \mathbb{N}. * \sim n$ by coinduction on $\sim$.
    - Assume $* \longrightarrow x$. We need to find $y$ s.t. $n \longrightarrow y$ and $x \sim y$. Choose $y = n + 1$. By **co-IH** $* \sim n + 1$.
    - Assume $n \longrightarrow y$. We need to find $x$ s.t. $* \longrightarrow x$ and $x \sim y$. Choose $x = *$. By **co-IH** $* \sim n + 1$.
- In essence same proof as textbook proof, but hopefully easier to teach and use.

# Iteration

- $\mathbb{N}$ is defined inductively by the introduction rules

$$\begin{array}{lcl} 0 & : & \mathbb{N} \\ \mathrm{S} & : & \mathbb{N} \to \mathbb{N} \end{array}$$

  - So we have an $\mathbb{N}$-algebra

$$(\mathbb{N}, 0, \mathrm{S}) : (X : \mathrm{Set}) \times X \times (X \to X)$$

  - Minimality of $(\mathbb{N}, 0, \mathrm{S})$ means:
  - Assume another $\mathbb{N}$-algebra $(X, z, s)$, i.e.

$$\begin{array}{lcl} z & : & X \\ s & : & X \to X \end{array}$$

  - Then there exist a **unique homomorphism** $g : (\mathbb{N}, 0, \mathrm{S}) \to (X, z, s)$, i.e.

$$\begin{array}{lcl} g : \mathbb{N} \to X & & \\ g\ 0 & = & z \\ g\ (\mathrm{S}\ n) & = & s\ (g\ n) \end{array}$$

# Iteration

- This means we can define uniquely

$$
\begin{array}{lcll}
g : \mathbb{N} \to X & & & \\
g\ 0 & = & x & \text{for some } x : X \\
g\ (\mathrm{S}\ n) & = & x' & \text{for some } x' : X \text{ depending on } (g\ n)
\end{array}
$$

- This is the principle of **unique iteration**.
- Definition by **pattern matching**.
- Can be strengthened to principle of **unique primitive recursion**:
- We can define uniquely

$$
\begin{array}{lcll}
g : \mathbb{N} \to X & & & \\
g\ 0 & = & x & \text{for some } x : X \\
g\ (\mathrm{S}\ n) & = & x' & \text{for some } x' : X \text{ depending on } n,\ g\ n
\end{array}
$$

# Coiteration and Primitive Corecursion

- Dually, **coinductive sets** are given by their elimination rules i.e. by **observations** or **eliminators**. Consider $\mathrm{Stream}$ given coinductively by

$$\begin{array}{lcl} \mathrm{head} & : & \mathrm{Stream} \to \mathbb{N} \\ \mathrm{tail} & : & \mathrm{Stream} \to \mathrm{Stream} \end{array}$$

  We obtain a $\mathrm{Stream}$-coalgebra

$$(\mathrm{Stream}, \mathrm{head}, \mathrm{tail}) : (X : \mathrm{Set}) \times (X \to \mathbb{N}) \times (X \to X)$$

- That $(\mathrm{Stream}, \mathrm{head}, \mathrm{tail})$ is maximal can be given by:
  - Assume another $\mathrm{Stream}$-coalgebra $(X, h, t)$: $\begin{array}{lcl} h & : & X \to \mathbb{N} \\ t & : & X \to X \end{array}$
  - Then there exist a **unique homomorphism** $g : (X, h, t) \to (\mathrm{Stream}, \mathrm{head}, \mathrm{tail})$, i.e.:

$$\begin{array}{lcl} \multicolumn{3}{l}{g : X \to \mathrm{Stream}} \\ \mathrm{head}\ (g\ x) & = & h\ x \\ \mathrm{tail}\ \ (g\ x) & = & g\ (t\ x) \end{array}$$

# Unique Coiteration

- Means we can define uniquely

$$
\begin{array}{lcll}
g : X \to \mathrm{Stream} & & & \\
\mathrm{head}\ (g\ x) & = & n & \text{for some } n : \mathbb{N} \text{ depending on } x \\
\mathrm{tail}\ \ (g\ x) & = & g\ x' & \text{for some } x' : X \text{ depending on } x
\end{array}
$$

  This is the principle of **unique coiteration**.
- Definition by **copattern matching**.
- Can be extended to the principle of **unique primitive corecursion:**
- We can define uniquely

$$
\begin{array}{lcll}
g : X \to \mathrm{Stream} & & & \\
\mathrm{head}\ (g\ x) & = & n & \text{for some } n : \mathbb{N} \text{ depending on } x \\
\mathrm{tail}\ \ (g\ x) & = & g\ x' & \text{for some } x' : X \text{ depending on } x \\
 & & \text{or} & \\
 & = & s & \text{for some } s : \mathrm{Stream} \text{ depending on } x
\end{array}
$$

# Comparison

- When using iteration the **instances** of $g$ we can use is **restricted**, but we can **apply an arbitrary function to it**.
- When using coiteration we can **choose any instance** $a$ **of** $g$, but **cannot apply** any function to $(g\ a)$.
- **Product** used in primitive recursion is **dualised** to **disjoint union** in primitive corecursion.

‘

# Induction

- Induction is in type theory **dependent primitive recursion**.
  - We don't know how to dualise this because that would require something like "codependent primitive corecursion".
- The rôle of induction is to have a principle which is essentially an **introduction principle**, which allows to **prove** the **uniqueness** of the functions defined by iteration and primitive recursion.
- So we need a proof principle for coinduction which is **introductory** in nature and allows to prove the **uniqueness** of the functions defined by **coiteration** and **primitive corecursion**.
- **Remark:** Uniqueness means that we have a **final coalgebra**.
  - Therefore **equality is undecidable**,
  - **type checking** becomes undecidable.
  - For **weakly final coalgebras** as e.g. in Agda we need to omit this conditions.

## Coinduction

- Uniqueness in coiteration is equivalent to the principle:
  **Bisimulation implies equality**
- Bisimulation on $\mathrm{Stream}$ is the largest relation $\sim$ on $\mathrm{Stream}$ s.t.

$$s \sim s' \to \mathrm{head}\ s = \mathrm{head}\ s' \wedge \mathrm{tail}\ s \sim \mathrm{tail}\ s'$$

- Largest can be expressed as $\sim$ being an **indexed coinductively defined set**.
- Primitive corecursion over $\sim$ means:
  We can prove

$$\forall s, s'. X\ s\ s' \to s \sim s'$$

  by showing the corecursive steps

$$\begin{array}{lll} \forall s, s. \quad X\ s\ s' & \to & \mathrm{head}\ s = \mathrm{head}\ s' \\ \forall s, s. \quad X\ s\ s' & \to & X\ (\mathrm{tail}\ s)\ (\mathrm{tail}\ s') \vee \mathrm{tail}\ s \sim \mathrm{tail}\ s' \end{array}$$

# Schema of Coinduction

- Combining
  - bisimulation implies equality
  - bisimulation can be shown corecursively

  we obtain the following principle of **coinduction**:
- We can prove
  $$\forall s, s'.X\ s\ s' \to s = s'$$
  by showing
  $$\begin{array}{lcrcl} \forall s, s'.X\ s\ s' & \to & \mathrm{head}\ s & = & \mathrm{head}\ s' \\ \forall s, s'.X\ s\ s' & \to & \mathrm{tail}\ s & = & \mathrm{tail}\ s' \end{array}$$
  where $\mathrm{tail}\ s = \mathrm{tail}\ s'$ can be derived
  - directly or
  - from a proof of
    $$X\ (\mathrm{tail}\ s)\ (\mathrm{tail}\ s')$$
    invoking the **co-induction-hypothesis** (which can be only used directly)
    $$X\ (\mathrm{tail}\ s)\ (\mathrm{tail}\ s') \to \mathrm{tail}\ s = \mathrm{tail}\ s'$$

# Coinduction Hypothesis

- When using iteration/primitive recursion/induction
  - there are **restrictions on the instances of the IH** to be used.
  - in case of iteration/primitive recursion we can **apply arbitrary functions** to the IH
  - in case of induction can use **use arbitrary reasoning steps** to obtain the statement from the IH.
- When using coiteration/primitive corecursion/coinduction
  - there are **no restrictions on the instances of the coIH** to be used.
  - however can **use the coIH only directly**.

# Example

- Define by **primitive corecursion**

$$
\begin{array}{lcl}
s : \mathrm{Stream} & & \\
\mathrm{head}\ s & = & 0 \\
\mathrm{tail}\ \ s & = & s \\
\mathrm{cons} : \mathbb{N} \to \mathrm{Stream} \to \mathrm{Stream} & & \\
\mathrm{head}\ (\mathrm{cons}\ n\ s) & = & n \\
\mathrm{tail}\ \ (\mathrm{cons}\ n\ s) & = & s
\end{array}
\qquad
\begin{array}{lcl}
s' : \mathbb{N} \to \mathrm{Stream} & & \\
\mathrm{head}\ (s'\ n) & = & 0 \\
\mathrm{tail}\ \ (s'\ n) & = & s'\ (n+1)
\end{array}
$$

- We show $\forall n : \mathbb{N}. s = s'\ n$ by **coinduction**:

$$
\begin{array}{lclcl}
\mathrm{head}\ s & = & 0 & = & \mathrm{head}\ (s'\ n) \\
\mathrm{tail}\ s & = & s \overset{\text{co-IH}}{=} s'\ (n+1) & = & \mathrm{tail}\ (s'\ n)
\end{array}
$$

- We show $\mathrm{cons}\ 0\ s = s$ by **coinduction**:

$$
\begin{array}{lclcll}
\mathrm{head}\ (\mathrm{cons}\ 0\ s) & = & 0 & = & \mathrm{head}\ s & \\
\mathrm{tail}\ \ (\mathrm{cons}\ 0\ s) & = & s & = & \mathrm{tail}\ \ s & \text{(no use of co-IH)}
\end{array}
$$

# Equivalence

- (Co)iteration, primitive (co)recursion, (co)induction can be generalise to (in the sense of Dybjer/AS) restricted indexed (co)inductively defined sets, which can be reduced to
  **Petersson Synek Trees (PST)**
  (= fixed points of **indexed containers**).

**Theorem**

*The following is equivalent for PST-(co)algebras:*

1. *The principle of unique (co)iteration.*
2. *The principle of unique primitive (co)recursion.*
3. *The principle of (co)iteration + (co)induction.*
4. *The principle of primitive (co)recursion + (co)induction.*

# Duality

[1]

| | |
|---|---|
| Inductive Definition | Coinductive Definition |
| Determined by Introduction | Determined by Observation/Elimination |
| Iteration | Coiteration |
| Pattern matching | Copattern matching |
| Primitive Recursion | Primitive Corecursion |
| Induction | Coinduction |
| Induction-Hypothesis | Coinduction-Hypothesis |

[1] This table is essentially due to Peter Hancock.

# Coinduction over Coinductively Defined Predicates

- When carrying out practical proofs of properties of a coinductively defined set $\mathrm{I}$, one often **doesn't prove equalities** (Question by Schwichtenberg at PCC 2015).
- Instead one proves that a predicate $\mathrm{P}$ over the coinductively defined set holds.
- Such predicates are often defined as an **indexed coinductively defined set**, indexed over $\mathrm{I}$ which follow the coinductive definition of $\mathrm{I}$.
  - Examples are bisimulation (indexed over a pair of elements of $\mathrm{I}$), or the predicate $\mathrm{CoEven}$ on $\mathbb{N}^\infty$ (see my PCC 2015 talk)
- Proofs of such kind of predicates can be done by **primitive corecursion over the indexed coinductively defined set**.
- A proof by corecursion can be considered as a proof by coinduction.
- We consider as example the predicate of increasing streams.

# Example IncStream

- Define coinductively

$$\begin{array}{lll} \mathrm{IncStream} : \mathrm{Stream} \to \mathrm{Set} \ \text{by} & & \\ \forall s : \mathrm{Stream}. & \mathrm{IncStream}\ s \to & \mathrm{head}\ (\mathrm{tail}\ s) < \mathrm{head}\ s \\ \forall s : \mathrm{Stream}. & \mathrm{IncStream}\ s \to & \mathrm{IncStream}\ (\mathrm{tail}\ s) \end{array}$$

- Define $\_{+}\mathrm{str}\_ : \mathrm{Stream} \to \mathrm{Stream} \to \mathrm{Stream}$ by primitive corecursion:

$$\begin{array}{llllll} \mathrm{head} & (s +\mathrm{str}\ s') & = & \mathrm{head}\ s + & & \mathrm{head}\ s' \\ \mathrm{tail} & (s +\mathrm{str}\ s') & = & \mathrm{tail}\ s & +\mathrm{str} & \mathrm{head}\ s' \end{array}$$

- **Remark:** We deviate from the abstract by defining $\mathrm{IncStream}$ as a predicate on $\mathrm{Stream}$ rather than a directly defined indexed coinductive set.

# Example IncStream

- We prove

$$\forall s, s'.\mathrm{IncStream}\ s \to \mathrm{IncStream}\ s' \to \mathrm{IncStream}\ (s +\mathrm{str}\ s')$$

by coinduction on $\mathrm{IncStream}\ (s +\mathrm{str}\ s')$:

- We need to show $\mathrm{head}\ (\mathrm{tail}\ (s +\mathrm{str}\ s')) < \mathrm{head}\ (s +\mathrm{str}\ s')$:

$$\begin{array}{rcl} \mathrm{head}\ (\mathrm{tail}\ (s +\mathrm{str}\ s')) & = & \mathrm{head}\ (\mathrm{tail}\ s) + \mathrm{head}\ (\mathrm{tail}\ s') \\ & < & \mathrm{head}\ s + \mathrm{head}\ s' \\ & = & \mathrm{head}\ (s +\mathrm{str}\ s') \end{array}$$

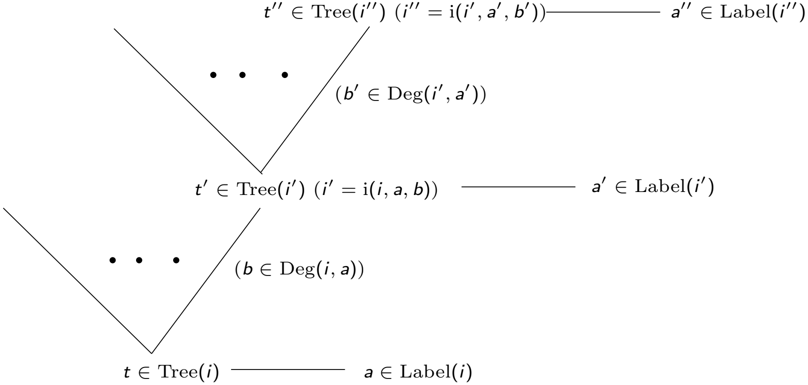
- We need to show $\mathrm{IncStream}\ (\mathrm{tail}\ (s +\mathrm{str}\ s'))$:

$\mathrm{tail}\ (s +\mathrm{str}\ s') = \mathrm{tail}\ s +\mathrm{str}\ \mathrm{tail}\ s'$
by co-IH $\mathrm{IncStream}\ (\mathrm{tail}\ s +\mathrm{str}\ \mathrm{tail}\ s')$
therefore $\mathrm{IncStream}\ (\mathrm{tail}\ (s +\mathrm{str}\ s'))$

# Conclusion

- Coiteration, primitive corecursion, coinduction are the duals of iteration, primitive recursion, induction.
- Coinduction is primitive corecursion over indexed coinductively defined sets
  - In case of bisimulation we obtain equality for final coalgebras.
- In iteration/primitive recursion/induction, the instances of the IH used are restricted, but the result can be used in arbitrary functions and formulas.
- In coiteration/primitive corecursion/coinduction, the instances of the IH used are unrestricted, but the result can be only be used directly.

# Generalisation: Petersson-Synek Trees (or Fixed Points of Containers)



# Petersson-Synek Trees (PST)

- Strictly positive inductive definitions can be reduced to the PSTs
- Inductive PSTs are the data types

$$
\begin{array}{l}
\mathrm{data\ Tree} : \mathrm{I} \to \mathrm{Set\ where} \\
\quad \mathrm{C} : (i : \mathrm{I}) \\
\qquad \to (a : \mathrm{Label}\ i) \\
\qquad \to ((b : \mathrm{Deg}\ i\ a) \to \mathrm{Tree}\ (\mathrm{j}\ i\ a\ b)) \\
\qquad \to \mathrm{Tree}\ i
\end{array}
$$

- Coinductive PSTs are defined follows:

$$
\begin{array}{lll}
\multicolumn{3}{l}{\mathrm{coalg\ Tree}^{\infty} : \mathrm{I} \to \mathrm{Set\ where}} \\
\quad \mathrm{label} & : & (i : \mathrm{I}) \to \mathrm{Tree}^{\infty}\ i \to \mathrm{Label}\ i \\
\quad \mathrm{subtree} & : & (i : \mathrm{I}) \\
& & \to (t : \mathrm{Tree}^{\infty}\ i) \\
& & \to (b : \mathrm{Deg}\ i\ (\mathrm{label}\ i\ t)) \\
& & \to \mathrm{Tree}^{\infty}\ (\mathrm{j}\ i\ (\mathrm{label}\ i\ t)\ b)
\end{array}
$$

# Schema for Primitive Corecursion

- Consider

$$
\begin{array}{l}
\mathrm{coalg}\ \mathrm{Tree}^{\infty} : \mathrm{I} \to \mathrm{Set}\ \mathrm{where} \\
\quad \begin{array}{lll}
\mathrm{label} & : & (i : \mathrm{I}) \to \mathrm{Tree}^{\infty}\ i \to \mathrm{Label}\ i \\
\mathrm{subtree} & : & (i : \mathrm{I}) \to (t : \mathrm{Tree}^{\infty}\ i) \to (b : \mathrm{Deg}\ i\ (\mathrm{label}\ i\ t)) \\
 & & \to \mathrm{Tree}^{\infty}\ (\mathrm{j}\ i\ (\mathrm{label}\ i\ t)\ b)
\end{array}
\end{array}
$$

- We can define a function

$$
\begin{array}{lclcl}
\multicolumn{5}{l}{f : (i : \mathrm{I}) \to X\ i \to \mathrm{Tree}^{\infty}\ i} \\
\mathrm{label}\ i\ (f\ \ i\ x) & = & a'\ i\ x & : & \mathrm{Label}\ i \\
\mathrm{subtree}\ i\ (f\ \ i\ x)\ b & = & t'\ i\ x\ b & : & \mathrm{Tree}^{\infty}\ i' \text{ with } i' := \mathrm{j}\ i\ a'\ b
\end{array}
$$

where $a'\ i\ x : \mathrm{Label}\ i$
and $(t'\ i\ x\ b)$ can be defined
  - as an element of $\mathrm{Tree}^{\infty}\ i'$ defined before
  - or corecursively defined as $\mathrm{subtree}\ i\ (f\ \ i\ x)\ b = f\ \ i'\ x'$
    for some $x' : X\ i'$.
    Here $f(i', x')$ will be called the **corecursion hypothesis**.

# Schema for Coinduction

- Assume

$$\begin{array}{lcl} J & : & \mathrm{Set} \\ \widehat{i} & : & J \to \mathrm{I} \\ x_0, x_1 & : & (j : J) \to \mathrm{Tree}^{\infty}\ (\widehat{i}\ j) \end{array}$$

We can show $\forall j : J . x_0\ j = x_0\ j'$ coinductively by showing
  - $\mathrm{label}\ (\widehat{i}\ j)\ (x_0\ j)$ and $\mathrm{label}\ (\widehat{i}\ j)\ (x_1\ j)$ are equal
  - and for all $b$ that
    $\mathrm{subtree}\ (\widehat{i}\ j)\ (x_0\ j)\ b$ and $(\mathrm{subtree}\ (\widehat{i}\ j)\ (x_0\ j)\ b$ are equal,
    where we can use either the fact that
    - this was shown before,
    - or we can use the **coinduction-hypothesis**, which means using the fact
      $\mathrm{subtree}\ (\widehat{i}\ j)\ (x_0\ j)\ b = x_0\ j'$ and $\mathrm{subtree}\ (\widehat{i}\ j)\ (x_1\ j)\ b = x_1\ j'$ for some $j' : J$.

# $\mathbb{N}^\infty$

$$\begin{array}{l}\text{coalg } \mathbb{N}^\infty : \text{Set where}\\ \quad \text{shape} : \mathbb{N}^\infty \to (\mathsf{0} + \mathrm{S}\ \mathbb{N}^\infty)\end{array}$$

- $\mathbb{N}^\infty$ can be reduced to non-indexed PSTs:

$$\begin{array}{llcl}\text{coalg } \mathbb{N}^\infty : \text{Set where} & & & \\ \quad \text{label} & : & & \mathbb{N}^\infty \to \{\mathsf{0}, \mathrm{S}\} \\ \quad \text{subtree} & : & & (n : \mathbb{N}^\infty) \to \text{Deg (label } n) \to \mathbb{N}^\infty \\ \text{where} \ \ \text{Deg } \mathsf{0} & = & & \emptyset \\ \qquad\quad\ \ \text{Deg S} & = & & \{*\}\end{array}$$

- Define $+$ by primitive corecursion

$$\begin{array}{l}\_ + \_ : \mathbb{N}^\infty \to \mathbb{N}^\infty \to \mathbb{N}^\infty \\ \text{shape } (n + m) = \text{case (shape } m) \text{ of} \\ \qquad\qquad\qquad\qquad \left\{ \begin{array}{lcl} \mathsf{0} & \longrightarrow & \text{shape } n \\ \mathrm{S}\ m' & \longrightarrow & \mathrm{S}\ (n + m') \end{array} \right\}\end{array}$$

# CoEven, CoOdd

- We define simultaneously coinductively

$$\begin{aligned}
&\mathrm{CoEven} : \mathbb{N}^{\infty} \to \mathrm{Set} \\
&\mathrm{CoEven}\ n \to \mathrm{CoEvenCond}\ (\mathrm{shape}\ n)
\end{aligned}$$

$$\begin{aligned}
&\mathrm{CoOdd} : \mathbb{N}^{\infty} \to \mathrm{Set} \\
&\mathrm{CoOdd}\ n \to \mathrm{CoOddCond}\ (\mathrm{shape}\ n)
\end{aligned}$$

where

$\mathrm{CoEvenCond}$ 0 is true
$\mathrm{CoEvenCond}\ (\mathrm{S}\ m) = \mathrm{CoOdd}\ m$

$\mathrm{CoOddCond}$ 0 doesn't hold
$\mathrm{CoOddCond}\ (\mathrm{S}\ m) = \mathrm{CoEven}\ m$

# CoEven, CoOdd as PSTs

- Define CoEven, CoOdd as one PST indexed over $\mathrm{I} := \{\mathrm{CoEven}, \mathrm{CoOdd}\} \times \mathbb{N}^{\infty} \times \mathbb{N}^{\infty}$

$$
\begin{array}{l}
\mathrm{coalg\ CoEvenOdd} : \mathrm{I} \to \mathrm{Set\ where} \\
\quad \begin{array}{lcl}
\mathrm{label} & : & (i : \mathrm{I}) \to \mathrm{CoEvenOdd}\ i \to \mathrm{Label}; \\
\mathrm{subtree} & : & (i : \mathrm{I}) \to (p : \mathrm{CoEvenOdd}\ i) \to \mathrm{Deg}\ i\ (\mathrm{label}\ i\ p) \\
& & \to \mathrm{CoEvenOdd}\ (\mathrm{j}\ i)
\end{array}
\end{array}
$$

where

$$
\begin{array}{lcl}
\mathrm{Label}\ c\ n\ m & = & \begin{cases} \emptyset & \text{if shape } m = 0 \text{ and } c = \mathrm{CoOdd} \\ \{*\} & \text{otherwise} \end{cases} \\
\mathrm{Deg}\ c\ n\ m & = & \begin{cases} \emptyset & \text{if shape } m = 0 \text{ and } c = \mathrm{CoEven} \\ \{*\} & \text{otherwise} \end{cases} \\
\mathrm{j}\ (\mathrm{CoEven}\ n\ m) & = & \mathrm{CoOdd}\ n\ (\mathrm{pred}\ m) \\
\mathrm{j}\ (\mathrm{CoOdd}\ n\ m) & = & \mathrm{CoEven}\ n\ (\mathrm{pred}\ m)
\end{array}
$$

# Closure of $\mathrm{CoEven}$ under $+$

- We show simultaneously

$$\forall n, m : \mathbb{N}^{\infty}.\mathrm{CoEven}\ n \rightarrow \mathrm{CoEven}\ m \rightarrow \mathrm{CoEven}\ (n+m)$$
$$\forall n, m : \mathbb{N}^{\infty}.\mathrm{CoEven}\ n \rightarrow \mathrm{CoOdd}\ m \rightarrow \mathrm{CoOdd}\ (n+m)$$

  by coinduction on $\mathrm{CoEven}$, $\mathrm{CoOdd}$
  - Assume $n, m$, $(\mathrm{CoEven}\ n)$, $(\mathrm{CoEven}\ m)$.
    For showing $(\mathrm{CoEven}\ (n+m))$ we have to show
    $\mathrm{CoEvenCond}\ (\mathrm{shape}\ (n+m))$.
    - If $\mathrm{shape}\ m = \mathrm{zero}$ then $\mathrm{shape}\ (n+m) = \mathrm{shape}\ n$ and by $(\mathrm{CoEven}\ n)$ we have $(\mathrm{CoEvenCond}\ (\mathrm{shape}\ n)$.
    - If $\mathrm{shape}\ m = \mathrm{S}\ m'$ then $\mathrm{shape}\ (n+m) = \mathrm{S}\ (n+m')$, $\mathrm{CoEvenCond}\ (\mathrm{shape}\ (n+m)) = \mathrm{CoOdd}\ (n+m')$ which follows by the **coIH** and $\mathrm{CoOdd}(m')$.
  - The proof of the second condition follows similarly