# Cumulative Inductive Types in Coq

Amin Timany[1] Matthieu Sozeau[2] Bart Jacobs[1]

imec-Distrinet, KU Leuven, Belgium

Inria Paris & IRIF, France

TYPES'17 – May 31, 2017

Budapest, Hungary

- In higher order dependent type theories:
  - Types are also terms and hence have a type
  - Type of all types, as it should be the type of itself, leads to paradoxes, like Girard's paradox
  - Thus, we have a countably infinite hierarchy of universes (types of types):

    $$\mathtt{Type}_0, \mathtt{Type}_1, \mathtt{Type}_2, \ldots$$

    where:

    $$\mathtt{Type}_0 : \mathtt{Type}_1, \mathtt{Type}_1 : \mathtt{Type}_2, \ldots$$

- In higher order dependent type theories:
  - Types are also terms and hence have a type
  - Type of all types, as it should be the type of itself, leads to paradoxes, like Girard's paradox
  - Thus, we have a countably infinite hierarchy of universes (types of types):

    $$\texttt{Type}_0, \texttt{Type}_1, \texttt{Type}_2, \ldots$$

    where:

    $$\texttt{Type}_0 : \texttt{Type}_1, \texttt{Type}_1 : \texttt{Type}_2, \ldots$$

- Such a system is cumulative if for any type $T$ and $i$:

$$T : \texttt{Type}_i \Rightarrow T : \texttt{Type}_{i+1}$$

- Example: Predicative Calculus of Inductive Constructions (pCIC), the logic of the proof assistant Coq

- pCIC has recently been extended with universe polymorphism
  - Definitions can be polymorphic in universe levels, e.g., categories:

```
Record Category@{i j} : Type@{max(i+1, j+1)} :=
  { Obj : Type@{i};
    Hom : Obj → Obj → Type@{j}; ... }.
```

- pCIC has recently been extended with universe polymorphism
  - Definitions can be polymorphic in universe levels, e.g., categories:

```
Record Category@{i j} : Type@{max(i+1, j+1)} :=
  { Obj : Type@{i};
    Hom : Obj → Obj → Type@{j}; ... }.
```

  - To keep consistent, universe polymorphic definitions come with constraints, e.g., category of categories:

```
Definition Cat@{i j k l} :=
  {| Obj := Category@{k l};
     Hom := fun C D ⇒ Functor@{k l k l} C D; ... |}
  : Category@{i j}.
```

    with constraints:

$$k < i \quad \text{and} \quad l < i$$

- For universe polymorphic inductive types, e.g., `Category`, copies are considered
- With no cumulativity (subtyping), i.e., `Category@{i j}` $\preceq$ `Category@{k l}` implies `i` $=$ `k` and `j` $=$ `l`

- For universe polymorphic inductive types, e.g., `Category`, copies are considered
- With no cumulativity (subtyping), i.e., `Category@{i j}` $\preceq$ `Category@{k l}` implies `i` $=$ `k` and `j` $=$ `l`
- This means `Cat@{i j k l}` is the category of all categories at `{k l}` and *not lower* [1]

[1] There are however categories isomorphic to the categories in lower levels.

- Constraints on statements about universe polymorphic inductive definitions restrict to which copies they apply
- For `Cat@{i j k l}` the fact that it has exponentials has constraints $\texttt{j} = \texttt{k} = \texttt{l}$

- Constraints on statements about universe polymorphic inductive definitions restrict to which copies they apply
- For `Cat@{i j k l}` the fact that it has exponentials has constraints `j = k = l`
- In particular:

```
Definition Type_Cat@{i j} :=
  {| Obj := Type@{j};
     Hom := fun A B => A -> B; ... |} : Category@{i j}.
```

  with constraints: $j < i$
- It is *not* an object of any copy of `Cat` with exponentials!

- Constraints on statements about universe polymorphic inductive definitions restrict to which copies they apply
- For `Cat@{i j k l}` the fact that it has exponentials has constraints `j` = `k` = `l`
- In particular:

```
Definition Type_Cat@{i j} :=
  {| Obj := Type@{j};
     Hom := fun A B ⇒ A → B; ... |} : Category@{i j}.
```

  with constraints: $j < i$
- It is *not* an object of any copy of `Cat` with exponentials!
- Yoneda embedding can't be simply defined as the exponential transpose of the *hom* functor

- Inductive typs in pCIC:

IND

$$\frac{A \in Ar(s) \qquad \Gamma \vdash A : s' \qquad \Gamma, X : A \vdash C_i : s \qquad C_i \in Co(X)}{\Gamma \vdash \mathsf{Ind}(X : A)\{C_1, \ldots, C_n\} : A}$$

$Ar(s)$ is the set of types of the form: $\Pi \vec{x} : \vec{M}.\ s$

$Co(X)$ is the set of types of the form: $\Pi \vec{x} : \vec{M}.\ X\ \vec{m}$

- Inductive typs in pCIC:

$$\frac{\textsc{Ind}\quad A \in Ar(s) \qquad \Gamma \vdash A : s' \qquad \Gamma, X : A \vdash C_i : s \qquad C_i \in Co(X)}{\Gamma \vdash \mathsf{Ind}(X : A)\{C_1, \ldots, C_n\} : A}$$

$$Ar(s) \;\text{ is the set of types of the form: }\; \Pi \vec{x} : \vec{M}.\ s$$

$$Co(X) \;\text{ is the set of types of the form: }\; \Pi \vec{x} : \vec{M}.\ X\ \vec{m}$$

No Parameters (`A` in `vec A n`) are considered in this rule.

```
Inductive vec (T : Type) : nat → Type := nil : vec T 0
  | cons : forall n, T → vec T n → vec T (S n).
```

- Inductive typs in pCIC:

  $$\frac{A \in Ar(s) \qquad \Gamma \vdash A : s' \qquad \Gamma, X : A \vdash C_i : s \qquad C_i \in Co(X)}{\Gamma \vdash \mathsf{Ind}(X : A)\{C_1, \ldots, C_n\} : A}\ \textsc{Ind}$$

  $$Ar(s) \ \text{ is the set of types of the form: } \ \Pi \vec{x} : \vec{M}.\ s$$

  $$Co(X) \ \text{ is the set of types of the form: } \ \Pi \vec{x} : \vec{M}.\ X\ \vec{m}$$

  No Parameters (`A` in `vec A n`) are considered in this rule.

  ```
  Inductive vec (T : Type) : nat → Type := nil : vec T 0
    | cons : forall n, T → vec T n → vec T (S n).
  ```

- Some cumulativity rules in pCIC:

  $$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash B : s \qquad A \preceq B}{\Gamma \vdash t : B}\ \textsc{Conv}$$

  $$\frac{i \leq j}{\mathtt{Type}_i \preceq \mathtt{Type}_j}\ \textsc{C-Type} \qquad \frac{A \simeq A' \qquad B \preceq B'}{\Pi x : A.\ B \preceq \Pi x : A'.\ B'}\ \textsc{C-Prod}$$

- Predicative Calculus of Cumulative Inductive Types (pCuIC):

$$\textsc{C-Ind}\quad \frac{\begin{array}{c} I \equiv (\mathsf{Ind}(X : \Pi\vec{x} : \vec{N}.\ s)\{\Pi\vec{x_1} : \vec{M}_1.\ X\ \vec{m_1}, \ldots, \Pi\vec{x_n} : \vec{M}_n.\ X\ \vec{m_n}\}) \\ I' \equiv (\mathsf{Ind}(X : \Pi\vec{x} : \vec{N'}.\ s')\{\Pi\vec{x_1} : \vec{M'_1}.\ X\ \vec{m'_1}, \ldots, \Pi\vec{x_n} : \vec{M'_n}.\ X\ \vec{m'_n}\}) \\ \forall i.\ N_i \preceq {N'}_i \qquad \forall i, j.\ (M_i)_j \preceq (M'_i)_j \\ length(\vec{m}) = length(\vec{x}) \qquad \forall i.\ X\ \vec{m}_i \simeq X\ \vec{m'_i} \end{array}}{I\ \vec{m} \preceq I'\ \vec{m}}$$

- Predicative Calculus of Cumulative Inductive Types (pCuIC):

C-Ind

$$\frac{\begin{array}{c} I \equiv (\mathrm{Ind}(X : \Pi\vec{x} : \vec{N}.\ s)\{\Pi\vec{x_1} : \vec{M}_1.\ X\ \vec{m_1}, \ldots, \Pi\vec{x_n} : \vec{M}_n.\ X\ \vec{m_n}\}) \\ I' \equiv (\mathrm{Ind}(X : \Pi\vec{x} : \vec{N'}.\ s')\{\Pi\vec{x_1} : \vec{M'_1}.\ X\ \vec{m'_1}, \ldots, \Pi\vec{x_n} : \vec{M'_n}.\ X\ \vec{m'_n}\}) \\ \forall i.\ N_i \preceq N'_i \qquad \forall i, j.\ (M_i)_j \preceq (M'_i)_j \\ length(\vec{m}) = length(\vec{x}) \qquad \forall i.\ X\ \vec{m}_i \simeq X\ \vec{m'_i} \end{array}}{I\ \vec{m} \preceq I'\ \vec{m}}$$

- Predicative Calculus of Cumulative Inductive Types (pCuIC):

C-Ind

$$\frac{\begin{array}{c} I \equiv (\mathsf{Ind}(X : \Pi\vec{x} : \vec{N}.\ s)\{\Pi\vec{x_1} : \vec{M}_1.\ X\ \vec{m}_1, \ldots, \Pi\vec{x_n} : \vec{M}_n.\ X\ \vec{m}_n\}) \\ I' \equiv (\mathsf{Ind}(X : \Pi\vec{x} : \vec{N'}.\ s')\{\Pi\vec{x_1} : \vec{M'_1}.\ X\ \vec{m'_1}, \ldots, \Pi\vec{x_n} : \vec{M'_n}.\ X\ \vec{m'_n}\}) \\ \forall i.\ N_i \preceq N'_i \qquad \forall i,j.\ (M_i)_j \preceq (M'_i)_j \\ length(\vec{m}) = length(\vec{x}) \qquad \forall i.\ X\ \vec{m}_i \simeq X\ \vec{m'_i} \end{array}}{I\ \vec{m} \preceq I'\ \vec{m}}$$

- Predicative Calculus of Cumulative Inductive Types (pCuIC):

C-Ind

$$\frac{\begin{array}{c} I \equiv (\mathsf{Ind}(X : \Pi\vec{x} : \vec{N}.\ s)\{\Pi\vec{x_1} : \vec{M}_1.\ X\ \vec{m}_1, \ldots, \Pi\vec{x_n} : \vec{M}_n.\ X\ \vec{m}_n\}) \\ I' \equiv (\mathsf{Ind}(X : \Pi\vec{x} : \vec{N}'.\ s')\{\Pi\vec{x_1} : \vec{M}'_1.\ X\ \vec{m}'_1, \ldots, \Pi\vec{x_n} : \vec{M}'_n.\ X\ \vec{m}'_n\}) \\ \forall i.\ N_i \preceq N'_i \qquad \forall i, j.\ (M_i)_j \preceq (M'_i)_j \\ \mathit{length}(\vec{m}) = \mathit{length}(\vec{x}) \qquad \forall i.\ X\ \vec{m}_i \simeq X\ \vec{m}'_i \end{array}}{I\ \vec{m} \preceq I'\ \vec{m}}$$

- Predicative Calculus of Cumulative Inductive Types (pCuIC):

C-Ind

$$\frac{\begin{array}{c} I \equiv (\mathsf{Ind}(X : \Pi\vec{x} : \vec{N}.\ s)\{\Pi\vec{x_1} : \vec{M}_1.\ X\ \vec{m_1}, \ldots, \Pi\vec{x_n} : \vec{M}_n.\ X\ \vec{m_n}\}) \\ I' \equiv (\mathsf{Ind}(X : \Pi\vec{x} : \vec{N'}.\ s')\{\Pi\vec{x_1} : \vec{M}'_1.\ X\ \vec{m'_1}, \ldots, \Pi\vec{x_n} : \vec{M}'_n.\ X\ \vec{m'_n}\}) \\ \forall i.\ N_i \preceq N'_i \qquad \forall i, j.\ (M_i)_j \preceq (M'_i)_j \\ \mathit{length}(\vec{m}) = \mathit{length}(\vec{x}) \qquad \forall i.\ X\ \vec{m}_i \simeq X\ \vec{m'_i} \end{array}}{I\ \vec{m} \preceq I'\ \vec{m}}$$

- Predicative Calculus of Cumulative Inductive Types (pCuIC):

C-Ind

$$\frac{\begin{array}{c} I \equiv (\mathsf{Ind}(X : \Pi\vec{x} : \vec{N}.\ s)\{\Pi\vec{x_1} : \vec{M}_1.\ X\ \vec{m_1}, \ldots, \Pi\vec{x_n} : \vec{M}_n.\ X\ \vec{m_n}\}) \\ I' \equiv (\mathsf{Ind}(X : \Pi\vec{x} : \vec{N'}.\ s')\{\Pi\vec{x_1} : \vec{M'_1}.\ X\ \vec{m'_1}, \ldots, \Pi\vec{x_n} : \vec{M'_n}.\ X\ \vec{m'_n}\}) \\ \forall i.\ N_i \preceq {N'}_i \qquad \forall i, j.\ (M_i)_j \preceq (M'_i)_j \\ length(\vec{m}) = length(\vec{x}) \qquad \forall i.\ X\ \vec{m}_i \simeq X\ \vec{m'_i} \end{array}}{I\ \vec{m} \preceq I'\ \vec{m}}$$

- Predicative Calculus of Cumulative Inductive Types (pCuIC):

$$\frac{\begin{array}{c} I \equiv (\mathrm{Ind}(X : \Pi\vec{x} : \vec{N}.\ s)\{\Pi\vec{x_1} : \vec{M}_1.\ X\ \vec{m}_1, \ldots, \Pi\vec{x_n} : \vec{M}_n.\ X\ \vec{m}_n\}) \\ I' \equiv (\mathrm{Ind}(X : \Pi\vec{x} : \vec{N}'.\ s')\{\Pi\vec{x_1} : \vec{M}'_1.\ X\ \vec{m}'_1, \ldots, \Pi\vec{x_n} : \vec{M}'_n.\ X\ \vec{m}'_n\}) \\ \forall i.\ N_i \preceq N'_i \qquad \forall i,j.\ (M_i)_j \preceq (M'_i)_j \\ length(\vec{m}) = length(\vec{x}) \qquad \forall i.\ X\ \vec{m}_i \simeq X\ \vec{m}'_i \end{array}}{I\ \vec{m} \preceq I'\ \vec{m}}\ \textsc{C-Ind}$$

- Example:

$$\texttt{Category@}\{\texttt{i j}\} \equiv \mathrm{Ind}(X : \texttt{Type}_{max(i+1,j+1)})\{\Pi o : \texttt{Type}_i.\Pi h : o \to o \to \texttt{Type}_j.\cdots\}$$

- By C-Ind:

$$\texttt{i} \leq \texttt{k and j} \leq \texttt{l} \Rightarrow \texttt{Category@}\{\texttt{i j}\} \preceq \texttt{Category@}\{\texttt{k l}\}$$

- Predicative Calculus of Cumulative Inductive Types (pCuIC):

$$\textsc{C-Ind}\quad \frac{\begin{array}{c} I \equiv (\mathsf{Ind}(X : \Pi\vec{x} : \vec{N}.\ s)\{\Pi\vec{x_1} : \vec{M}_1.\ X\ \vec{m_1}, \ldots, \Pi\vec{x_n} : \vec{M}_n.\ X\ \vec{m_n}\}) \\ I' \equiv (\mathsf{Ind}(X : \Pi\vec{x} : \vec{N'}.\ s')\{\Pi\vec{x_1} : \vec{M'_1}.\ X\ \vec{m'_1}, \ldots, \Pi\vec{x_n} : \vec{M'_n}.\ X\ \vec{m'_n}\}) \\ \forall i.\ N_i \preceq {N'}_i \qquad \forall i, j.\ (M_i)_j \preceq (M'_i)_j \\ \mathit{length}(\vec{m}) = \mathit{length}(\vec{x}) \qquad \forall i.\ X\ \vec{m_i} \simeq X\ \vec{m'_i} \end{array}}{I\ \vec{m} \preceq I'\ \vec{m}}$$

- Example:

$$\texttt{Category@\{i j\}} \equiv \mathsf{Ind}(X : \texttt{Type}_{max(i+1,j+1)})\{\Pi o : \texttt{Type}_i.\Pi h : o \to o \to \texttt{Type}_j. \cdots\}$$

- By C-Ind:

$$\texttt{i} \leq \texttt{k} \text{ and } \texttt{j} \leq \texttt{l} \Rightarrow \texttt{Category@\{i j\}} \preceq \texttt{Category@\{k l\}}$$

- Notice C-Ind does not consider parameters or sort of the inductive type

- Example:

$$\texttt{list@\{i\}}\ (\texttt{A} : \texttt{Type}_i) \equiv \mathsf{Ind}(X : \texttt{Type}_i)\{X, A \to X \to X\}$$

- By C-Ind:

$$\texttt{list@\{i\}}\ \texttt{A} \preceq \texttt{list@\{j\}}\ \texttt{A}$$

- Example:

$$\texttt{list@}\{\texttt{i}\}\ (\texttt{A} : \texttt{Type}_i) \equiv \mathsf{Ind}(X : \texttt{Type}_i)\{X, A \to X \to X\}$$

- By C-Ind:

$$\texttt{list@}\{\texttt{i}\}\ \texttt{A} \preceq \texttt{list@}\{\texttt{j}\}\ \texttt{A} \qquad (\textbf{regardless of } \texttt{i} \textbf{ and } \texttt{j})$$

- Example:

$$\texttt{list@}\{\texttt{i}\}\ (\texttt{A} : \texttt{Type}_i) \equiv \mathsf{Ind}(X : \texttt{Type}_i)\{X, A \to X \to X\}$$

- By C-Ind:

$$\texttt{list@}\{\texttt{i}\}\ \texttt{A} \preceq \texttt{list@}\{\texttt{j}\}\ \texttt{A} \qquad \textbf{(regardless of } \texttt{i} \textbf{ and } \texttt{j)}$$

- In pCuIC we consider *fully applied* inductive types $I\ \vec{m}$ and $I'\ \vec{m}$ convertible if they are mutually subtypes

$$\textsc{Conv-Ind}\ \frac{I\ \vec{m} \preceq I'\ \vec{m} \qquad I'\ \vec{m} \preceq I\ \vec{m}}{I\ \vec{m} \simeq I'\ \vec{m}}$$

- Example:

$$\texttt{list@\{i\}}\ (\texttt{A} : \texttt{Type}_i) \equiv \mathsf{Ind}(X : \texttt{Type}_i)\{X, A \to X \to X\}$$

- By C-Ind:

$$\texttt{list@\{i\} A} \preceq \texttt{list@\{j\} A} \qquad \textbf{(regardless of } \texttt{i} \textbf{ and } \texttt{j)}$$

- In pCuIC we consider *fully applied* inductive types $I\ \vec{m}$ and $I'\ \vec{m}$ convertible if they are mutually subtypes

$$\textsc{Conv-Ind}\ \frac{I\ \vec{m} \preceq I'\ \vec{m} \qquad I'\ \vec{m} \preceq I\ \vec{m}}{I\ \vec{m} \simeq I'\ \vec{m}}$$

- Examples:

$$\texttt{i} = \texttt{k} \text{ and } \texttt{j} = \texttt{l} \Rightarrow \texttt{Category@\{i j\}} \simeq \texttt{Category@\{k l\}}$$

$$\texttt{list@\{i\} A} \simeq \texttt{list@\{j\} A} \qquad \textbf{(regardless of } \texttt{i} \textbf{ and } \texttt{j)}$$

# Demo

This is implemented in Coq!

# Theoretical justification

- In set theoretic models of pCIC $[\![\cdot]\!] : \mathit{Terms}_{pCIC} \rightarrow ZF$[2]
- For subtyping $A \preceq B$ we have $[\![A]\!] \subseteq [\![B]\!]$
- Inductive types interpreted using least fixpoints of **monotone**[3] functions
- This justifies both C-Ind and Conv-Ind

[2]ZF with suitable axioms, e.g., inaccessible cardinals or Grothendieck universes, to model pCIC universes

[3]Due to strict positivity condition

Thanks