All Coq Rules in One Place

Xiaohong Chen

March 11, 2020

Abstract

This document summarizes all the proof rules of the Coq proof assistant, as listed in `https://coq.inria.fr/distrib/current/refman/language/cic.html`.

1 Syntax

Let us fix a countably infinite set V of *variables*, denoted $x, y, \ldots$. Let us fix a countably infinite set C of *constants*, denoted $c, d, \ldots$.

Definition 1.1. We define the set *Term* to be the smallest set that satisfies the following conditions:

1. $\mathsf{SProp}, \mathsf{Prop}, \mathsf{Set} \in \mathit{Term}$; $\mathsf{Type}(i) \in \mathit{Term}$ for every $i \in \mathbb{N}$.
2. $V \subseteq \mathit{Term}$.
3. $C \subseteq \mathit{Term}$.
4. If $x \in V$ and $T, U \in \mathit{Term}$, then $\forall x : T, U \in \mathit{Term}$.
5. If $x \in V$ and $T, u \in \mathit{Term}$, then $\lambda x : T.u \in \mathit{Term}$.
6. If $t, u \in \mathit{Term}$, then $(t\,u) \in \mathit{Term}$, called *application*.
7. If $x \in V$ and $t, T, u \in \mathit{Term}$, then $\mathsf{let}\, x \coloneqq t : T \,\mathsf{in}\, u \in \mathit{Term}$.

where $\forall x : T, U$ binds x to U and $\lambda x : T.u$ binds x to u. We use $\mathrm{FV}(T) \subseteq V$ to denote the set of free variables in $T \in \mathit{Term}$. For $T, U \in \mathit{Term}$ and $x \in V$, we use $T[U/x]$ to denote the result of substituting U for x in T, where α-renaming happens implicitly to prevent variable capture.

Definition 1.2. We define the set $\mathit{Sort} = \{\mathsf{SProp}, \mathsf{Prop}, \mathsf{Set}\} \cup \{\mathsf{Type}(i) \mid i \in \mathbb{N}\}$. Note that $\mathit{Sort} \subseteq \mathit{Term}$. Elements in *Sort* are called *sorts* and denoted as s, possibly with subscripts.

Definition 1.3. A *local assumption* is written $x : T$, where $x \in V$ and $T \in \mathit{Term}$. A *local definition* is written $x \coloneqq u : T$, where $x \in V$ and $u, T \in \mathit{Term}$. In both cases, we call x the *declared variable*. A *local context* is an ordered list of local assumptions and local definitions, such that the declared variables are all distinct. We use Γ, possibly with subscripts, to denote local contexts.

Notation 1.4. We use the notation $[x : T \,;\, y \coloneqq u : U \,;\, z : V]$ to denote the local context that consists of the local assumption $x : T$, the local definition $y \coloneqq u : U$ and the local assumption $z : V$, with the implicit requirement that x, y, z are all distinct. The empty local context is written as $[]$. Let Γ be a local context. We write $x \in \Gamma$ to mean that x is declared in Γ. We write $(x : T) \in \Gamma$ to mean that the local assumption $x : T$ is in Γ, or that the local definition $x \coloneqq u : T$ is in Γ for some $u \in \mathit{Term}$. We write $(x \coloneqq u : T) \in \Gamma$ to mean that the local definition $x \coloneqq u : T$ is in Γ. We write $\Gamma :: (x : T)$ to denote the local context that enriches Γ with $x : T$, with the implicit requirement that $x \notin \Gamma$. Similarly, we write $\Gamma :: (x \coloneqq u : T)$ to denote the local context that enriches Γ with $x \coloneqq u : T$, with the implicit requirement that $x \notin \Gamma$. We write $\Gamma_1 \,;\, \Gamma_2$ to mean the local context obtained by concatenating Γ_1 and Γ_2, with the implicit requirement that all variables declared in Γ_2 are not declared in Γ_1.

Definition 1.5. A *global assumption* is written $(x : T)$, where $c \in C$ and $T \in \mathit{Term}$. A *global definition* is written $c \coloneqq u : T$, where $c \in C$ and $u, T \in \mathit{Term}$. In both cases, we call c the *declared constant.* A *global environment* is an ordered list of global assumptions and global definitions. We use E, possibly with subscripts, to denote global environments.

Notation 1.6. We use the notation $c_1 : T\ ;\ c_2 \coloneqq u : U\ ;\ c_3 : V$ to denote the local context that consists of the global assumption $c_1 : T$, the global definition $c_2 \coloneqq u : U$ and the global assumption $c_3 : V$. The empty global context is written as $[]$. Let E be a local context. We write $c \in E$ to mean that c is declared in E. We write $(c : T) \in E$ to mean that the global assumption $c : T$ is in E, or that the global definition $c \coloneqq u : T$ is in E for some $u \in \mathit{Term}$. We write $(c \coloneqq u : T) \in E$ to mean that the global definition $c \coloneqq u : T$ is in E. We write $E\ ;\ c : T$ to denote the global context that enriches E with $c : T$. Similarly, we write $E\ ;\ c \coloneqq u : T$ to denote the global context that enriches E with $(c \coloneqq u : T)$.

Notation 1.7. We write $E[\Gamma] \vdash u : T$ to mean that u is well-typed with type T in global environment E and local environment Γ. We write $\mathcal{WF}(E)[\Gamma]$ to mean that the global environment E is well-formed and Γ is a valid local context in E.

Definition 1.8. A term u is *well-typed* in a global environment E if there is a local environment Γ and type T such that $E[\Gamma] \vdash u : T$ is derivable with the rules below.

2 Coq Typing Rules