

# Comandos y tácticas en Coq

Manuel Soto Romero

Víctor Zamora Gutiérrez

## 1. Comandos

- **Section-End** - Estos comandos se utilizan para encapsular partes de nuestro archivo en una sección.
- **Definition** - Se utiliza para definir funciones. Solo lo vamos a usar para funciones no recursivas.
- **Fixpoint** - Se utiliza para definir funciones recursivas.
- **Notation** - Introduce notación para nuestras funciones y estructuras.
- **Check** - Se utiliza para checar el tipo de una expresión.
- **Lemma, Theorem, Proposition, Example** - Se utiliza para iniciar el modo de prueba.
- **Proof-Qed** - Para encapsular una prueba. **Proof** no es necesario pero es buena práctica ponerlo.

## 2. Tácticas

- **unfold d**  
Reescribe **d** de acuerdo a su definición.
- **intro, intros**  
Introduce las cuantificaciones y antecedentes de implicaciones a las hipótesis. **intro** introduce solo la primera cuantificación o antecedente, mientras que **intros** introduce todos los posibles.
- **apply H**  
Aplica la hipótesis **H** a la conclusión. También se puede utilizar con teoremas previamente definidos.
- **rewrite H**  
Reescribe utilizando la hipótesis o teorema **H**.
- **reflexivity**  
Prueba igualdades evidentes (por ejemplo,  $X = X$  y  $0 + X = X$ ).

- **inversion H**

Deriva todas las hipótesis necesarias para que ocurra H, a partir de un constructor.

- **left, right**

Para elegir qué lado de un  $\backslash/$  probar.

- **trivial**

Prueba cosas como  $A = A$  o algo que está en las hipótesis.

- **assert, cut**

Para agregar nuevas hipótesis. Por ejemplo **assert**  $(0 = 0)$  introduce una nueva hipótesis  $H : 0 = 0$  a nuestro conjunto. Estas tácticas agregan submetas que hay que probar para poder hacer la prueba (pues no se puede agregar hipótesis sin probarlas). **assert** pide probar la hipótesis antes de seguir con la prueba actual, mientras que **cut** nos deja posponer la prueba de esta.

- **split**

Para partir  $\wedge$  y  $\leftrightarrow$

- **discriminate**

Sirve para probar que dos cosas son *sintácticamente* distintas. Por ejemplo, podemos probar la meta  $0 <> S\ n$  con **discriminate**

- **destruct**

Para partir en varios casos (por ejemplo, si tenemos una lista, podemos usar **destruct** para partir en los casos  $[]$  y  $a::l$ ). También sirve para instanciar existenciales.

- **simpl**

Reduce lo más que pueda una meta. Por ejemplo, si aplicamos **simpl** a la meta  $0 + n = n$ , nos queda la meta  $n = n$ . Esta táctica usa las definiciones para realizar la simplificación.

- **contradict H**

Contradice la hipótesis H y pide probar su negación en la meta.

Hay muchas más tácticas. Para una lista extensiva, visitar: <https://coq.inria.fr/refman/tactic-index.html>