Introducción Práctica a Co-Inducción en Coq

Aplicación a pruebas y estructuras de datos infinitas

Ismael Correa

CC7125

2022-2

Motivación

Estructuras de datos infinitas

En lenguajes funcionales *lazy* las estructuras de datos infinitas son frecuentes. Nos gustaría contar con facilidades similares en Coq.

Problemas

- En Coq solo contamos con recursión bien fundada, por lo que no podemos hacer uso de `Fixpoints` para esto.

Solución en Coq

Tipos Co-Inductivos y funciones Co-Recursivas

Para nuestro beneficio, lo anterior es posible de hacer en Coq haciendo uso de tipos Co-Inductivos y funciones Co-Recursivas.

El ejemplo que usaremos para estudiar esto es el de listas potencialmente infinitas, las cuales corresponden con, por ejemplo, el *output stream* de un proceso potencialmente no terminante.

Condiciones de Guarda

Para llamados co-recursivos

La interacción entre *pattern matching* de constructores y mecanismos de *unwinding*, sumado al requerimento de terminación, resulta en el siguiente principio implementado en Coq:

Condición de Guarda

Todo llamado recursivo de una función declarada como co-fixpoint, debe ser el argumento directo de un termino no vacío compuesto solo de constructores.

Resumen Práctico

Comparación inductivos y co-inductivos

- Para ambos tipos de datos:
  - Constructores y pattern matching pueden ser usados de manera similar,
- Para tipos inductivos:
  - Recursión es solo usada para *consumir* elementos del tipo,
  - Argumentos de llamados recursivos solo pueden ser sub-componentes de constructores,
- Para tipos co-inductivos:
  - Co-recursión es solo usada para *producir* elementos del tipo,
  - Llamados co-recursivos solo pueden *producir* sub-componentes de constructores.

Resumen Práctico

Pruebas por co-inducción

- Use la táctica `cofix` para introducir valores co-inductivos,
- Agrega una nueva hipótesis al contexto con el mismo tipo que el *goal*,
- La nueva hipótesis solo puede ser usada para *rellenar* sub-componentes de un constructor (*Guarded condition*),
- Una correcta aplicación del principio anterior puede ser *chequeado* haciendo uso del comando `Guarded`.