

Formalización de la propiedad de Progreso para el λ -cálculo simplemente tipado

Cristian Sottile

20 de octubre de 2023

1 Avances

En este mes estuve siguiendo el libro PLFA de Wadler para interiorizarme con el asistente. Comparto el repositorio en el que estuve copiando las definiciones e implementando los ejercicios propuestos. Estoy en el capítulo de Cuantificadores, me quedan los capítulos de Decidibilidad y de Listas, y luego planeo empezar con la implementación del λ -cálculo simplemente tipado para poder probar la propiedad de progreso. No me topé con mayores dificultades aprendiendo Agda. Sí fui anotando algunas dudas que aprovecho para copiar acá.

2 Consultas

2.1 Ignorando parámetros sin éxito

Aclaro que usé este formato inusual de definir las funciones en el `where` solo para ver si funcionaba.

```
1 u-distrib-x :  $\forall \{A\ B\ C : \text{Set}\} \rightarrow (A \times B) \cup C \leq (A \cup C) \times (B \cup C)$ 
2 u-distrib-x {A} {B} {C} = record { to = to' ; from = from' ; from◦to = from◦to' }
3 where
4   to' :  $(A \times B) \cup C \rightarrow (A \cup C) \times (B \cup C)$ 
5   to' (inj1 ( a , b )) = ( inj1 a , inj1 b )
6   to' (inj2 c ) = ( inj2 c , inj2 c )
7   from' :  $(A \cup C) \times (B \cup C) \rightarrow (A \times B) \cup C$ 
8   from' ( inj1 a , inj1 b ) = inj1 ( a , b )
9   from' ( _ , inj2 c ) = inj2 c
10  from' ( inj2 c , _ ) = inj2 c
11  -- ¿Por qué me marca esto? Ah, debe ser porque no sabe si _ es inj1 o inj2. Y
12  -- cuando especifico inj1 a , inj2 c ... Ah, no. Porque entonces es un
13  -- problema también no saber si el _ de ( inj2 c , _ ) es inj1 o inj2.
14  -- ¿Quizá hay algo con que el segundo elemento pueda depender del primero?
15  -- PREGUNTAR A MIGUEL
16  from◦to' :  $(x : (A \times B) \cup C) \rightarrow \text{from}' (\text{to}' x) \equiv x$ 
17  from◦to' (inj1 ( a , b )) = refl
18  from◦to' (inj2 c ) = refl
```

2.2 Normalización de goals pero no de términos

```
3 -- Preguntar a Miguel: cuándo reduce  $\lambda x.f x$  a  $f$  como para dar el goal  $f \equiv f$ 
2 -- Considerando que pedirle a Agda que normalice  $\lambda x \rightarrow f x$  da el mismo término
1  $\eta\text{-}\rightarrow : \forall \{A B : \text{Set}\} (f : A \rightarrow B) \rightarrow (\lambda (x : A) \rightarrow f x) \equiv f$ 
128  $\eta\text{-}\rightarrow f = \{\}$ 
1
1 - 5.5k 06-Connectives.agda Agda (a)(y)(K) Git:main Mod
20 :  $f \equiv f$ 
```