MODELOS DE COMPUTACIÓN Y COMPLEJIDAD

Grado en Ingeniería Informática. Tecnologías Informáticas

ETS Ingeniería Informática. Universidad de Sevilla (Curso 2021-2022)

Problemas de FUNCIONES GOTO-COMPUTABLES

EJERCICIO 16.

Sea $f : \mathbb{N}^{k+1} \to \mathbb{N}$ una función total $(k+1)$-aria, con $k \geq 0$. Se define la suma acotada de f, que notaremos Σ_f, así: $\Sigma_f(\vec{x}, y) = \Sigma_{z \leq y} f(\vec{x}, z)$, para cada k-tupla de números naturales $\vec{x}$ y cada número natural y. Probar que si f es `GOTO`-computable, entonces la función suma acotada Σ_f también lo es.

SOLUCIÓN:

Vamos a ver que la función Σ_f (de aridad $k+1$) se puede describir mediante la recursión primitiva de dos ciertas funciones g (de aridad k) y h (de aridad $k+2$), de tal manera que estas funciones g y h sean `GOTO`-computables.

Para ello, si $\Sigma_f = \mathcal{R}(g, h)$ entonces deberá verificarse las siguientes condiciones:

(a) $\Sigma_f(\vec{x}, 0) = g(\vec{x})$, para cada k-tupla $\vec{x}$.

(b) $\Sigma_f(\vec{x}, y+1) = h(\vec{x}, y, \Sigma_f(\vec{x}, y))$, para cada k-tupla $\vec{x}$ y cada número natural $y \in \mathbb{N}$.

Ahora bien, de la propia definición de Σ_f se tiene que:

(a) $\Sigma_f(\vec{x}, 0) = \Sigma_{z \leq 0} f(\vec{x}, z) = f(\vec{x}, 0)$, para cada k-tupla $\vec{x}$.

(b) $\Sigma_f(\vec{x}, y+1) = \Sigma_{z \leq y+1} f(\vec{x}, z) = \Sigma_{z \leq y} f(\vec{x}, z) + f(\vec{x}, y+1)$, para cada k-tupla $\vec{x}$ y cada número natural $y \in \mathbb{N}$.

Por tanto, las funciones g y h deberán verificar lo siguiente:

(a) $g(\vec{x}) = f(\vec{x}, 0)$, para cada k-tupla $\vec{x}$.

(b) $h(\vec{x}, y, \Sigma_f(\vec{x}, y)) = \Sigma_{z \leq y} f(\vec{x}, z) + f(\vec{x}, y+1) = \Sigma_f(\vec{x}, y) + f(\vec{x}, y+1)$, para cada k-tupla $\vec{x}$ y cada número natural $y \in \mathbb{N}$.

En consecuencia:

(a) Teniendo presente que la función f es `GOTO`-computable y que $g(\vec{x}) = f(\vec{x}, 0)$, deducimos que la función g es `GOTO`-computable.

(b) Teniendo presente que la función f es `GOTO`-computable y que $h(\vec{x}, y, t)) = t + f(\vec{x}, y+1)$, deducimos que la función h es `GOTO`-computable.

Finalmente, de la `GOTO`-computabilidad de las funciones g y h, así como del hecho de que Σ_f está definida por recursión primitiva a partir de g y de h, concluimos que la función Σ_f es `GOTO`-computable.

– –