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 20.

Sea $\texttt{GOTO}_1$ el modelo de computación cuyos procedimientos mecánicos son los programas `GOTO` en los que **sólo aparece una variable** (es decir, **todas** las instrucciones de los programas contienen la **misma variable**). Probar que:

1. La función $f(x) = 2x$ $(\forall x \in \mathbb{N})$ **no** es $\texttt{GOTO}_1$-computable.
2. El modelo de computación `GOTO` es **estríctamente más potente** (es decir, *calcula* más funciones) que el modelo de computación $\texttt{GOTO}_1$.
3. Los modelos de computación $\texttt{GOTO}_l$ y $\texttt{GOTO}_1$ ¿**son equivalentes**? (es decir, ¿ambos *calculan* las mismas funciones?). Razónese la respuesta.
4. Sea $\texttt{GOTO}_{1,f}$ el modelo de computación cuyos procedimientos mecánicos son los programas de $\texttt{GOTO}_1$ cuyas instrucciones condicionales deben cumplir las restricciones propias de $\texttt{GOTO}_f$. Los modelos de computación $\texttt{GOTO}_l$ y $\texttt{GOTO}_{1,f}$ ¿**son equivalentes**? Razónese la respuesta.

SOLUCIÓN:

1. Para demostrar que la función $f(x) = 2x$ $(\forall x \in \mathbb{N})$ **no** es $\texttt{GOTO}_1$-computable, vamos a ver qué ha de verificar las funciones 1-arias calculadas por programas del modelo $\texttt{GOTO}_1$.

   Sea P un programa del modelo $\texttt{GOTO}_1$. Vamos a distinguir casos según sea el **tipo de variable** que contiene dicho programa.

   - Supongamos que la única variable que aparece en P es una **variable de entrada** o bien una **variable de trabajo/auxiliar**. En ese caso, al no aparecer la variable de salida, si el programa **para** entonces ha de devolver el valor 0. En consecuencia, ese programa no puede calcular la función f ya que, por ejemplo, nunca puede devolver el valor 2.
   - Supongamos que la única variable que aparece en P es la **variable de salida** Y. En ese caso, al no aparecer ninguna variable de entrada, si el programa **para** entonces **siempre** ha de devolver el **mismo** valor (para **cualquier** dato de entrada. En consecuencia, ese programa no puede calcular la función f ya que, por ejemplo, nunca podrá devolver dos valores distintos (con independencia de los valores de entrada suministrados).

2. Es obvio que la función $f(x) = 2x$ $(\forall x \in \mathbb{N})$ es `GOTO` . No obstante, en el apartado anterior se ha probado que dicha función **no** es $\texttt{GOTO}_1$-computable. Por tanto, existen funciones `GOTO` s que, en cambio, **no** son $\texttt{GOTO}_1$-computables. En consecuencia, el modelo de computación `GOTO` es **estríctamente más potente** que el modelo de computación $\texttt{GOTO}_1$.

3. En primer lugar, obsérvese que la función vacía es $\texttt{GOTO}_1$-computable ya que dicha función es calculada por el siguiente programa del modelo $\texttt{GOTO}_1$

$$P \equiv \begin{cases} [A] \ \texttt{X} \leftarrow \texttt{X} + \texttt{1} \\ \quad\ \ \texttt{IF}\ X \neq 0\ \texttt{GOTO}\ A \end{cases}$$

Ahora bien, toda función calculada por programas del modelo $\texttt{GOTO}_l$ (e, incluso, del modelo $\texttt{GOTO}_f$) son **totales** y, por tanto, la función vacía no es $\texttt{GOTO}_l$-computable ni $\texttt{GOTO}_f$-computable. En consecuencia, los modelos de computación $\texttt{GOTO}_l$ y $\texttt{GOTO}_1$ **no son equivalentes**, al igual que los modelos de computación $\texttt{GOTO}_f$ y $\texttt{GOTO}_1$.

4. Vamos a demostrar que los modelos de computación $\texttt{GOTO}_l$ y $\texttt{GOTO}_{1,f}$ **son equivalentes**. Para ello, si dado un modelo de computación $\texttt{M}$ denotamos por $\mathcal{F}(\texttt{M})$ la clase de funciones que son computables en el modelo $\texttt{M}$, entonces hemos de probar que $\mathcal{F}(\texttt{GOTO}_l) = \mathcal{F}(\texttt{GOTO}_{1,f})$.

   – Veamos que $\mathcal{F}(\texttt{GOTO}_l)$ **es** la clase de todas las funciones constantes. En efecto: en el apartado 3 del ejercicio 18 se probó que la clase de funciones 1-arias que son $\texttt{GOTO}_l$-computables coincide con la clase de funciones 1-arias constantes. Este resultado se extiende, de manera natural: para cada $k \geq 1$ la clase de funciones k-arias que son $\texttt{GOTO}_l$-computables coincide con la clase de funciones k-arias que son constantes.

   – Veamos que $\mathcal{F}(\texttt{GOTO}_{1,f})$ también **es** la clase de todas las funciones constantes. En efecto, basta tener presente las consideraciones siguientes:

     (a) Toda función $\texttt{GOTO}_{1,f}$-computable es una función $\texttt{GOTO}_f$-computable y, en consecuencia (apartado 1 del ejercicio 19), es total.

     (b) Toda función $\texttt{GOTO}_{1,f}$-computable f es una función $\texttt{GOTO}_1$-computable y, en consecuencia (de la prueba del apartado 1 de este ejercicio) existirá un número natural a tal que el valor de f, si existe, ha de coincidir con a; es decir, para cualquier número natural x, si $f(x)\downarrow$ entonces $f(x) = a$.

De lo que antecede, se deduce que $\mathcal{F}(\texttt{GOTO}_l) = \mathcal{F}(\texttt{GOTO}_{1,f})$ y, en consecuencia, los modelos de computación $\texttt{GOTO}_l$ y $\texttt{GOTO}_{1,f}$ **son equivalentes**.

– –