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 18.

Sea $\texttt{GOTO}_l$ el modelo de computación cuyos procedimientos mecánicos son los programas `GOTO` que **no** contienen ninguna **instrucción condicional** (tales programas `GOTO` se denominan **programas lineales**). Probar que:

1. Para cada $k \in \mathbb{N}$ se verifica:
   (a) La función constante de aridad 1, $C_k(x) = k$ $(\forall k \in \mathbb{N})$, es $\texttt{GOTO}_l$-computable.
   (b) Si P es un programa $\texttt{GOTO}_l$ de longitud k entonces se verifica que $P(x) \leq k$, para cada $x \in \mathbb{N}$.
   (c) Si P es un programa $\texttt{GOTO}_l$ que calcula la función constante de aridad 1, C_k, entonces se verifica que $|P| \geq k$.
2. La función $f(x) = x + 1$, para cada $x \in \mathbb{N}$, **no** es $\texttt{GOTO}_l$-computable.
3. Las funciones $\texttt{GOTO}_l$-computables de aridad 1 son, exactamente, las funciones constantes.
4. El modelo de computación `GOTO` es **estríctamente más potente** (es decir, *calcula* más funciones) que el modelo de computación $\texttt{GOTO}_l$.

SOLUCIÓN:

1. Para establecer el apartado (a), consideremos el programa P_k que consta, exactamente, de k instrucciones tales que todas ellas son $Y \leftarrow Y$ (si $k = 0$ entonces P_k es el programa vacío). Obviamente, se trata de un programa de $\texttt{GOTO}_l$ que, ademas, calcula la función constante C_k de aridad 1.

   El apartado (b) se resolvió, explícitamente, en el ejercicio 7, utilizando una prueba por inducción.

   Para establecer el apartado (c), consideremos un programa P de $\texttt{GOTO}_l$ que calcula la función constante C_k. Entonces se tiene lo siguiente:

   – El programa P verifica que $P(x) = k$, para cada $x \in \mathbb{N}$.
   – Del apartado (b) se deduce que $P(x) \leq |P|$, para cada $x \in \mathbb{N}$.

   De las dos relaciones anteriores se concluye que $k \leq |P|$.

2. Supongamos que existiera un programa P de $\texttt{GOTO}_l$ tal que calcula la función total $f(x) = x + 1$. Supongamos que la longitud de P es $k \in \mathbb{N}$. Entonces:

   – Por una parte, $P(x) = x + 1$, para cada $x \in \mathbb{N}$.
   – Por otra, del apartado (b) se tiene que $P(x) \leq |P|$, para cada $x \in \mathbb{N}$.

Puesto que $|P| = k$, de las dos relaciones anteriores se deduce que $x + 1 \leq k$, para cada $x \in \mathbb{N}$. En particular, para $x = k$ se tendría que $k + 1 \leq k$. Lo que, obviamente, es una contradicción.

3. Se trata de demostrar que las funciones $\mathtt{GOTO}_l$-computables de aridad 1 son, exactamente, las funciones constantes. Puesto que en el apartado 1 (a) se ha probado que toda función constante 1-aria es $\mathtt{GOTO}_l$-computable, será suficiente demostrar que cualquier función 1-aria $\mathtt{GOTO}_l$-computable debe ser una función constante. Para ello, se considera la fórmula $\theta(k)$ definida para cada $k \in \mathbb{N}$, como sigue:

$$\theta(k) \equiv [\forall P \ (P \text{ programa } \mathtt{GOTO}_l \ \wedge |P| = k \rightarrow \exists a \in \mathbb{N} \ (\forall x \in \mathbb{N} \ (P(x) = a)))]$$

Se ha de probar que la fórmula $\theta(k)$ es verdadera para cada número natura k. Probémoslo por inducción sobre k.

   – **Caso base:** $k = 0$.

     Si P es un programa $\mathtt{GOTO}_l$ cuya longitud es 0 entonces P es el programa vacío. Por tanto, se verifica que $\forall x \in \mathbb{N} \ (P(x) = 0)$. Por tanto, en este caso, basta considerar $a = 0$.

   – **Paso inductivo: Sea $k \in \mathbb{N}$ tal que la fórmula $\theta(k)$ es verdadera. Veamos que la fórmula $\theta(k+1)$ también es verdadera**.

     Para ello, sea P un programa $\mathtt{GOTO}_l$ cuya longitud es $k + 1$. Sea I_{k+1} la última instrucción de P y sea Q el programa obtenido a partir de P eliminando la instrucción I_{k+1}. Entonces, Q es un programa $\mathtt{GOTO}_l$ cuya longitud es k. Teniendo presente que la fórmula $\theta(k)$ es verdadera, existirá un número natural a' tal que $\forall x \in \mathbb{N} \ (Q(x) = a')$. En tales circunstancias, hemos de probar la existencia de un número natural a tal que $\forall x \in \mathbb{N} \ (P(x) = a)$. Probémoslo seguidamente.

     Puesto que I_{k+1} no es una instrucción condicional resultará que el programa P **para** sobre cualquier dato de entrada. Más aún:

     * Si la instrucción I_{k+1} no contiene a la variable de salida Y, entonces $\forall x \in \mathbb{N} \ (P(x) = Q(x) = a')$. En este caso, basta considerar $a = a'$.
     * Si la instrucción I_{k+1} es $Y \rightarrow Y + 1$, entonces $\forall x \in \mathbb{N} \ (P(x) = Q(x) + 1 = a' + 1)$. En este caso, basta considerar $a = a' + 1$.
     * Si la instrucción I_{k+1} es $Y \rightarrow Y - 1$ y $a' = 0$, entonces $\forall x \in \mathbb{N} \ (P(x) = Q(x) = 0)$. En este caso, basta considerar $a = 0$.
     * Si la instrucción I_{k+1} es $Y \rightarrow Y - 1$ y $a' > 0$, entonces $\forall x \in \mathbb{N} \ (P(x) = Q(x) - 1 = a' - 1)$. En este caso, basta considerar $a = a' - 1$.

     De lo que antecede se deduce que la fórmula $\theta(k+1)$ es verdadera.

   Esto que completa la demostración de que $\forall k \in \mathbb{N} \ (\theta(k))$.

4. En primer lugar, obsérvese que todo programa $\mathtt{GOTO}_l$ es, en particular, un programa `GOTO`. Por tanto, toda función $\mathtt{GOTO}_l$-computable será, así mismo, `GOTO`-computable.

   Ahora bien, la función $f(x) = x + 1$, para cada $x \in \mathbb{N}$, es, obviamente, `GOTO`-computable y, del apartado 2 se deduce que **no** es $\mathtt{GOTO}_l$-computable.

Por tanto, existen funciones `GOTO`-computables que **no** son $\texttt{GOTO}_l$-computables. Es decir, el modelo de computación `GOTO` es **estríctamente más potente** que el modelo de computación $\texttt{GOTO}_l$.

— —