Teoría de la Complejidad Computacional
Tema IV: Clases de complejidad computacional

Mario de J. Pérez Jiménez

Grupo de Investigación en Computación Natural
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

marper@us.es

http://www.cs.us.es/~marper/

Máster Universitario en Matemáticas
Curso 2015-2016

Índice del tema IV

* Complejidad en tiempo.
* Decidibilidad en tiempo polinomial.
* La clase P.
* Decidibilidad no determinista en tiempo polinomial.
* La clase NP.

Complejidad en tiempo: problemática

¿Como definir la *complejidad computacional* de un problema abstracto (respecto de una medida)?

- Idea para una definición local:
  - ⋆ Se consideran todas las soluciones mecánicas del problema abstracto.
  - ⋆ Se calcula el coste de cada solución a partir de la medida de complejidad.
  - ⋆ Se elige la solución más económica.
- No es posible una definición *local* (teorema de aceleración).
- Estudio de forma global a través de las *clases de complejidad*.

Una *clase de complejidad* estará especificada por:

- ∗ Un *modelo de computación*.
- ∗ Un *modo de computación*.
- ∗ Los *recursos* a contabilizar.
- ∗ Una *cota superior* de los recursos permitidos.

Complejidad en tiempo de una MTD

Definición: Sean M una MTD y $\mathcal{C} = (C_x, C_1, \ldots, C_t)$ una computación de parada de M con entrada $x \in \Sigma^*$. Entonces diremos que t es el **tiempo de ejecución** de dicha computación.

Definición: Una MTD, M, **trabaja en tiempo acotado por** una función total $f : \mathbf{N} \rightarrow \mathbf{N}$, si existe $c > 0$ tal que para cada $x \in \Sigma^{(n)}$, el tiempo de ejecución de M con entrada x es, a lo sumo, $c \cdot f(n)$ (asintóticamente).

(También se dirá que M es de **coste en tiempo acotado por** f)

Definición: Sea M una MTD. La **complejidad en tiempo** de M es la función parcial $t_M : \mathbf{N} - \to \mathbf{N}$ definida así:

- $t_M(n) =$ mayor tiempo de ejecución de M sobre cualquier palabra de longitud n (si existe dicho máximo).
- No definida, en caso contrario.

Nota: Si $\{M_e : \ e \in \mathbf{N}\}$ es una enumeración efectiva de las MTD, entonces $\{t_{M_e} : \ e \in \mathbf{N}\}$ es una medida de complejidad.

Definición: Una función $f : \Sigma^* - \to \Sigma^*$ es **Turing–computable en tiempo acotado** por una función computable total 1-aria g, si existe una MTD que calcula la función f y trabaja en tiempo acotado por g.

Nota: Una función es **Turing–computable en tiempo polinomial** si es **Turing–computable** en tiempo acotado por un polinomio.

Complejidad en tiempo de un programa GOTO

Definición: Sea P un programa GOTO y $x = (x_1, \ldots, x_n) \in \mathbb{N}^n$. Supongamos que la computación $\mathcal{C} = (C_x, C_1, \ldots, C_t)$ es de parada. Entonces diremos que t es el **tiempo de ejecución** de P con entrada x.

Definición: Un programa GOTO, P, **trabaja en tiempo acotado por** una función total $f : \mathbf{N} \to \mathbf{N}$, si existe $c > 0$ tal que para cada $x \in \mathbb{N}^{(n)}$, el tiempo de ejecución de P con entrada x es, a lo sumo, $c \cdot f(n)$ (asintóticamente).

(También se dirá que P es de *coste en tiempo acotado por* f)

Definición: Sea P un programa GOTO. La **complejidad en tiempo** de P es la función parcial $t_P : \mathbf{N} - \to \mathbf{N}$ definida así:

- $t_P(n) =$ mayor tiempo de ejecución de P sobre una n–tupla $(x_1, \ldots, x_n) \in \mathbb{N}^n$ (si existe dicho máximo).
- No definida, en caso contrario.

Codificaciones

Definición: Un conjunto A es **codificable** a partir de un conjunto B si existen funciones inyectivas computables,

$$f : A \to B \quad \text{y} \quad g : B - \to A$$

tales que $g(f(x)) = x$, para cada $x \in A$.

- f es la función **codificadora**.
- g es la función **decodificadora**.

Teorema: Sean $\Sigma = \{s_1, \dots, s_n\}$ y $\Gamma = \{t_1, \dots, t_r\}$ alfabetos tales que $r \geq 2$, $s_1 < \dots < s_n$ y $t_1 < \dots < t_r$. Entonces, Σ^* es codificable a partir de Γ^* mediante una función computable, f, en *tiempo lineal* (es decir, existe $c \in \mathbf{N}$ tal que $|f(\gamma)| = c \cdot |\gamma|$, para cada $\gamma \in \Sigma^*$).

Nota: Todo problema abstracto se puede representar usando el alfabeto $\Sigma = \{0, 1\}$, a partir de codificaciones *adecuadas*.

Aceptación y decisión de lenguajes

Mediante MTD (ver tema II).

Mediante programas `GOTO`.

- Programas `GOTO` que trabajan sobre el alfabeto $\Sigma = \{0, 1\}$.
- Sea P un programa `GOTO`:
  - P **acepta** un dato $u \in \Sigma^*$ si $P(u) \downarrow= 1$.
  - P **rechaza** un dato $u \in \Sigma^*$ si $P(u) \downarrow\neq 1$.
  - P **decide** un lenguaje $L \subseteq \Sigma^*$ si para $u \in \Sigma^*$:
    - ⋆ Si $u \in L$, entonces P acepta u.
    - ⋆ Si $u \notin L$, entonces P rechaza u.

    (P es un **programa de decisión** de L)
  - P **acepta** un lenguaje $L \subseteq \Sigma^*$ si para $u \in \Sigma^*$:
    - ⋆ Si $u \in L$, entonces P acepta u.
    - ⋆ Si $u \notin L$, entonces $P(u) \uparrow$.

    (P es un **programa de aceptación** de L)

Definición: El lenguaje asociado a un problema de decisión $X = (E_X, \theta_X)$ es $L_X = \{u \in E_X : \ \theta_X(u) = 1\}$.

Definición: Un *problema de decisión* es *resoluble algorítmicamente* o *decidible* si existe una MTD (o un programa `GOTO`) que *decide* el lenguaje asociado.

Definición: Un problema de decisión es *parcialmente decidible* si existe una MTD (o un programa `GOTO`) que *acepta* el lenguaje asociado.

Decidibilidad de un lenguaje $\equiv$ *Resolubilidad algorítmica* de un problema

Decidibilidad en tiempo polinomial

Definición: Una MTD, M, es de *coste en tiempo polinomial* si existe un polinomio $p(n)$ tal que $t_M(n) \in O(p(n))$.

- Son equivalentes:
  - ⋆ M es de coste en tiempo polinomial.
  - ⋆ Existe $k \in \mathbf{N}$ tal que $t_M(n) \in O(n^k)$.
  - ⋆ Existen $k \in \mathbf{N}, c \in \mathbb{R}^+ - \{0\}$ y $n_0 \in \mathbf{N}$ tales que $\forall u \in \Sigma^* \ (|u| \geqslant n_0 \wedge M \downarrow u \Longleftrightarrow t_M(|u|) \leqslant c \cdot |u|^k)$.

Definición: Un programa GOTO, P, es de *coste en tiempo polinomial* si existe un polinomio $p(n)$ tal que $t_P(n) \in O(p(n))$.

Definición: Un lenguaje L es *decidible en tiempo polinomial* si existe una MTD (o un programa GOTO) de coste en tiempo polinomial que decide L.

Definición: Un problema de decisión es *resoluble algorítmicamente en tiempo polinomial* si existe una MTD (o un programa GOTO) de coste en tiempo polinomial que decide el lenguaje asociado al problema.

Clases de complejidad en tiempo

Sea f una función total computable 1-aria:

Definición: $\mathbf{TIME}(f)$ es la clase de todos los lenguajes decidibles (problemas resolubles algorítmicamente) por MTDs que trabajan en tiempo acotado por f.

La clase P:

* $\mathbf{P} = \bigcup\limits_{k \in \mathbf{N}} \mathbf{TIME}(n^k)$.

P: clase de los lenguajes decididos por MTDs en tiempo polinomial.

P: clase de los problemas resolubles por MTDs en tiempo polinomial.

Decidibilidad no determinista

Definición: Una MTND, M, *decide* un lenguaje $L \subseteq \Sigma^*$ si para cada $x \in \Sigma^*$: ($x \in L$ sii *existe, al menos, una computación de aceptación* de $M(x)$).

Definición: Sea M una MTND y $x \in \Sigma^*$. Si $\mathcal{C} = (C_x, C_1, \ldots, C_t)$ es una computación finita de $M(x)$, entonces t es el *tiempo de ejecución* de $\mathcal{C}$.

Definición: Sea M una MTND. La *complejidad en tiempo* de M es la función total $t_M : \mathbf{N} \to \mathbf{N}$ definida así:

$$t_M(n) = \text{máx}\{t^*_M(x) : \ x \in \Sigma^* \wedge |x| = n\}$$

en donde

$$t^*_M(x) = \begin{cases} \min\{t(\mathcal{C}) : \ \mathcal{C} \text{ es computación de aceptación de } x\} & \text{si existe dicho mínimo} \\ 0 & \text{e.c.o.c.} \end{cases}$$

Definición: Una MTND, M, *trabaja en tiempo acotado por* $f(n)$ si existe $c > 0$ tal que $t_M(n) \leq c \cdot f(n)$ (asintóticamente).

Definición: Una MTND es de *coste en tiempo polinomial* si trabaja en tiempo acotado por un polinomio.

La clase NP

Sea f una función total computable 1-aria:

Definición: NTIME(f) es la clase de todos los lenguajes decidibles (problemas resolubles) por MTNDs que trabajan en tiempo acotado por f.

* Obviamente, $\mathbf{TIME}(f) \subseteq \mathbf{NTIME}(f)$.

Definición: $\mathbf{NP} = \bigcup_{k \in \mathbf{N}} \mathbf{NTIME}(n^k)$.

NP: clase de los lenguajes decidibles por MTNDs en tiempo polinomial.

NP: clase de los problemas resolubles algorítmicamente por MTNDs en tiempo polinomial.

* Se verifica que $\mathbf{P} \subseteq \mathbf{NP}$.