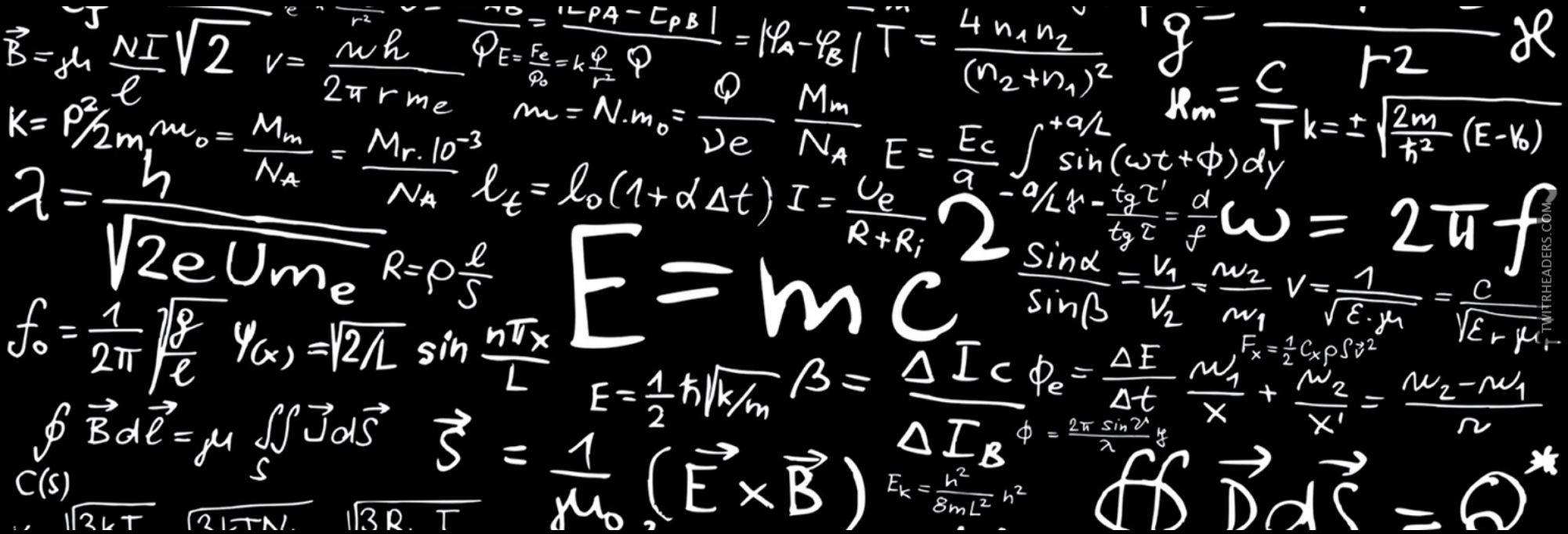
Introducción a la complejidad computacional

definida sobre anillos arbitrarios

Matías San Martín (*masanmartin@uc.cl*)

18 de junio de 2016



Fuente: *http://www.utmmcss.com/*

¿Por qué otro modelo?

- Continuo vs discreto.
- Intuición interiorizada del análisis.
- Buscar mayor naturalidad para problemas relacionados a la física.
- Otro enfoque para buscar responder la pregunta $P = NP$.

Anillo

Definición (Anillo)

El triple $(R, +, \cdot)$, donde $+$ y $\cdot$ son operaciones binarias sobre R, se dice un anillo si cumple que:

- La operación $+$ es asociativa, conmutativa, tiene elemento neutro y tiene elemento inverso (aditivo) sobre R.
- La operación $\cdot$ es asociativa sobre R y distributiva (por ambos lados) con respecto a la operación $+$, sobre R.

Ejemplos

- $(\mathbb{Z}, +, \cdot)$ con la suma y producto usual.
- $(\mathbb{Z}_4, +, \cdot)$ con la suma y producto usual módulo 4.
- $\left(\mathcal{M}_2(\mathbb{R}), +, \cdot\right)$ donde $\mathcal{M}_2(\mathbb{R})$ son las matrices reales de 2×2 con la suma y producto usual de matrices.

Campo

Definición (Campo)

El triple $(R, +, \cdot)$, donde $+$ y $\cdot$ son operaciones binarias sobre R, se dice un campo si cumple que:

- $(R, +, \cdot)$ es un anillo.
- La operación $\cdot$ es conmutativa, tiene elemento neutro y tiene elemento inverso (multiplicativo) sobre R (salvo el 0).

Ejemplos

- $(\mathbb{Q}, +, \cdot)$ y $(\mathbb{R}, +, \cdot)$ campos ordenados.
- $(\mathbb{Z}_2, +, \cdot)$ y $(\mathbb{C}, +, \cdot)$ campos no ordenados.

¿Por qué sobre anillos?

Sea $P = \{x_1, \ldots, x_n\}$ y sea $\varphi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ una fórmula en $L(P)$ que está en CNF, donde cada C_i es una cláusula bien definida.

Intentaremos de relacionar las fórmulas en CNF con polinomios en $\mathbb{Z}_2$. Sea $L \subseteq \left(\mathbb{Z}_2[P]\right)^m$ el lenguaje definido como:

$$L = \{(f_1, \ldots, f_m) \in \left(\mathbb{Z}_2[P]\right)^m : \text{ existe } \xi \in \left(\mathbb{Z}_2\right)^n \text{ tal que } f_1(\xi) = f_2(\xi) = \cdots = f_m(\xi) = 0\}.$$

Y consideremos la siguiente transformación para variables $p, q \in P$:

$$\begin{aligned} \neg p &\mapsto p + 1 \mod 2 \\ p \vee q &\mapsto p + q + p \cdot q \mod 2 \\ p \wedge q &\mapsto p \cdot q \mod 2 \\ p \oplus q &\mapsto p + q \mod 2 \end{aligned}$$

De CNF-SAT a L

En CNF-SAT buscamos que:

$$\begin{aligned} C_1 &\equiv \top \\ C_2 &\equiv \top \\ &\vdots \\ C_m &\equiv \top \end{aligned}$$

En L buscamos que:

$$\begin{aligned} f_1 &= 0 \\ f_2 &= 0 \\ &\vdots \\ f_m &= 0 \end{aligned}$$

Notamos que $C_i \equiv \top$ si y solo si $C_i \oplus \top \equiv \bot$.

Por lo tanto, transformaremos $C_i \oplus \top \equiv \bot$ en $f_i + 1 = 0$ en $\mathbb{Z}_2$.

Pequeño ejemplo

Sea $\varphi = \underbrace{(p \vee \neg q)}_{C_1} \wedge \underbrace{(\neg p \vee q)}_{C_2} \wedge \underbrace{(p \vee q)}_{C_3}$.

Usando la transformación, llegamos a que:

$$\begin{aligned} C_1 &\mapsto q + p \cdot q + 1 \\ C_2 &\mapsto p + q \cdot p + 1 \\ C_3 &\mapsto p + q + p \cdot q \end{aligned}$$

Y lo transformaremos en:

$$f_1(p,q) = q + p \cdot q, \qquad f_2(p,q) = p + q \cdot p$$
$$\text{y} \quad f_3(p,q) = p + q + p \cdot q + 1.$$

Llegando a que $\varphi \in \mathit{CNF\text{-}SAT}$ si y solo si $(f_1, f_2, f_3) \in L$.

Problema de decisión *HN*

El problema L definido anteriormente es conocido como la formulación de decisión del problema Nullstellensatz de Hilbert (sobre $\mathbb{Z}_2$).

Definición (Hilbert's Nullstellensatz sobre R)

Sea R un anillo. Definimos el conjunto $HN_R^{m,n}$ como:

$$HN_R^{m,n} = \{(f_1, \ldots, f_m) \in \left(R[x_1, \ldots, x_n]\right)^m : \text{ existe } \xi \in R^n \text{ tal que } f_1(\xi) = f_2(\xi) = \cdots = f_m(\xi) = 0\}.$$

¿Qué pasa si $R = \mathbb{Z}$?
¿Y si $R = \mathbb{R}$ con $m = n = 1$?
¿Y si $R = \mathbb{C}$ con $m = n = 1$?
¿Y si $R = \mathbb{C}$ y todas las funciones son lineales?

¿Qué queremos hacer?

- "Generalizar" la máquina de Turing.
- Clases de complejidad: P_R y NP_R.
- Reducciones polinomiales.

Teorema

Para cualquier campo $(R, =)$ se tiene que HN_R es NP_R-completo.

Corolario

Para cualquier campo $(R, =)$ se tiene que

$$P_R = NP_R \text{ si y solo si } HN_R \text{ está en } P_R.$$

Método de Newton

El método "por excelencia" del análisis numérico.

Mostraremos este método para encontrar raíces aproximadas de un polinomio de una variable en $\mathbb{C}$.

Definición (Función de Newton)

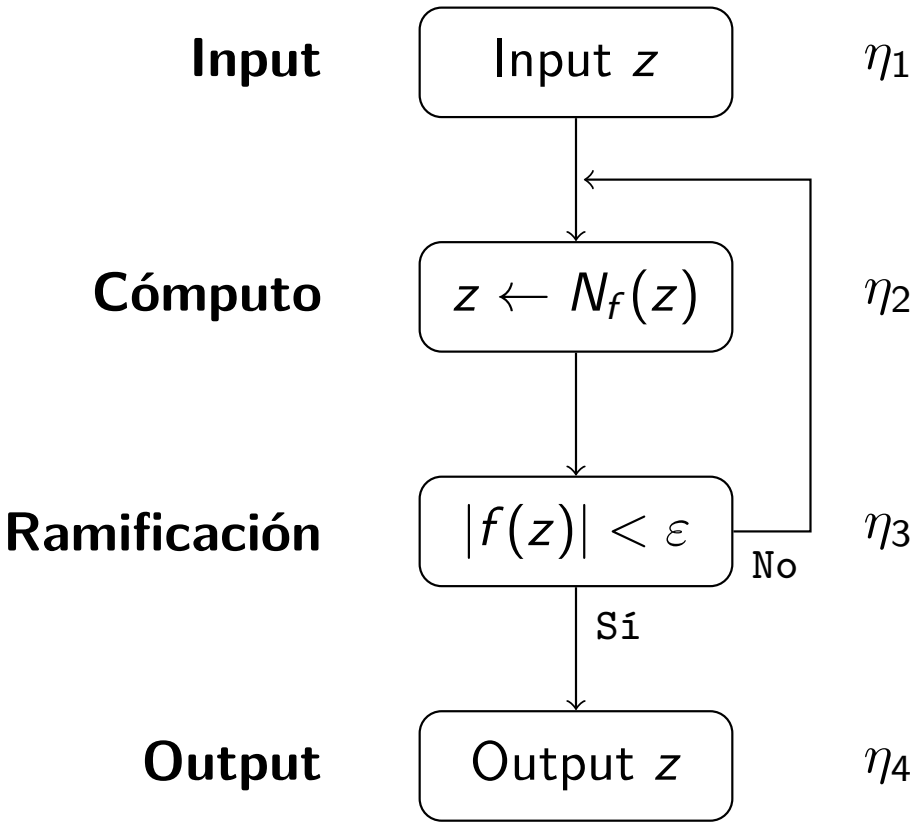
Sea $f : \mathbb{C} \to \mathbb{C}$ un polinomio. Definimos $N_f : \mathbb{C} \to \mathbb{C}$ como:

$$N_f(z) = z - \frac{f(z)}{f'(z)},$$

para todo $z \in \mathbb{C}$, donde $f'(z) \neq 0$.

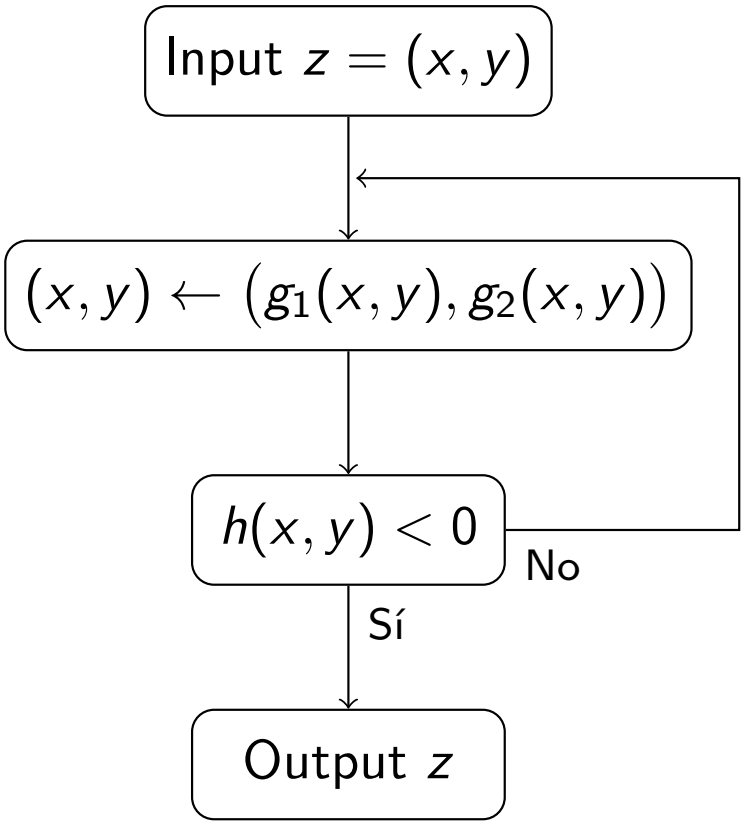
Dado un punto inicial $z_0 \in \mathbb{C}$, podemos definir la órbita de z_0 como la sucesión $(z_0, z_1, \ldots, z_k, \ldots)$ donde $z_k = N_f^k(z_0)$ y podemos establecer una condición de detención, como $|f(z_k)| < \varepsilon$ para algún $\varepsilon > 0$.

Método de Newton



Método de Newton

¿Sobre qué anillo está definido este algoritmo?



Trataremos a $\mathbb{C}$ como $\mathbb{R}^2$ con la biyección natural, es decir,

$$x + yi \mapsto (x, y).$$

Consideramos:

$$\begin{aligned} g_1(x, y) &= \mathrm{Re}\big(N_f(x + yi)\big), \\ g_2(x, y) &= \mathrm{Im}\big(N_f(x + yi)\big), \\ h(x, y) &= f_1(x, y)^2 + f_2(x, y)^2 - \varepsilon^2, \end{aligned}$$

donde

$$f(x + yi) = f_1(x, y) + f_2(x, y)i.$$

Máquinas de dimensión finita

Dado: un anillo R, dotado de un orden $\leq$.

Definición (Máquinas de dimensión finita)

Para una máquina de dimensión finita (MDF) M sobre R, usaremos tres espacios de trabajo: el espacio de input $\mathcal{I}_M = R^n$, el espacio de estados $\mathcal{S}_M = R^m$ y el espacio de output $\mathcal{O}_M = R^\ell$, donde $n, m, \ell \in \mathbb{N}$.
M consiste en un grafo dirigido y conexo (finito) con cuatro tipo de nodos y transformaciones asociadas:

- **Nodo de input**: $I : \mathcal{I}_M \to \mathcal{S}_M$ y un único nodo siguiente β_1.
- **Nodo de cómputo** η: $g_\eta : \mathcal{S}_M \to \mathcal{S}_M$ y un único nodo siguiente β_η.
- **Nodo de ramificación** η: $h_\eta : \mathcal{S}_M \to R$ y dos nodos salientes: para la arista `Sí`, es el nodo β_η^+, asociado con la condición $h_\eta(z) \geq 0$ y la arista `No` con el nodo β_η^-, asociado a $h_\eta(z) < 0$.
- **Nodo de output** η: $O_\eta : \mathcal{S}_M \to \mathcal{O}_M$ y no tiene nodo siguiente.

Funcionamiento de una máquina de dimensión finita

Sea $\mathcal{N} = \{1, \ldots, N\}$ los nodos de M, donde 1 corresponde al nodo de input y N al único de output, sin pérdida de generalidad.
Se define la función de computabilidad (o de funcionamiento) H como:

$$H : \mathcal{N} \times \mathcal{S}_M \quad \to \quad \mathcal{N} \times \mathcal{S}_M$$

$$(\text{nodo}, \text{estado}) \quad \mapsto \quad \begin{pmatrix} \text{nodo} \\ \text{siguiente} \end{pmatrix}, \begin{pmatrix} \text{estado} \\ \text{siguiente} \end{pmatrix}$$

- **Input**: $H(1, z) = (\beta_1, z)$.
- **Cómputo**: $H(\eta, z) = \big(\beta_\eta, g_\eta(z)\big)$.
- **Ramificación**: $H(\eta, z) = \begin{cases} \big(\beta_\eta^+, z\big) & \text{si } h_\eta(z) \geq 0 \\ \big(\beta_\eta^-, z\big) & \text{si } h_\eta(z) < 0 \end{cases}$.
- **Output**: $H(N, z) = (N, z)$.

Función de Input-Output, tiempo de detención

Dado $x \in \mathcal{I}_M$, definimos $x^0 = I(x) \in \mathcal{S}_M$ y $z^0 = (1, I(x)) \in \mathcal{N} \times \mathcal{S}_M$. Podemos definir la sucesión de configuraciones $z^0, z^1, \ldots$, donde $z^k = (\eta^k, x^k) = H^k(z^0)$.

Decimos que la máquina se detiene en un tiempo T si es que $z^T = (N, u)$ para algún $u \in \mathcal{S}_M$. El tiempo de detención $\tau_M(x)$ es el menor valor T tal que M se detiene en tiempo T con input x.

Definimos la función de Input-Output Φ_M como $\Phi_M(x) = O(x^T) \in \mathcal{O}_M$ donde $T = \tau_M(x)$.

Si no existe dicho T, decimos que $\tau_M(x) = \infty$ y $\Phi_M(x)$ no está definido. Sea $\Omega_M = \{x \in \mathcal{I}_M : \tau_M(x) < \infty\}$, entonces las funciones $T_M : \Omega_M \to \mathbb{N}^+$ con $T_M = \tau_M\big|_{\Omega_M}$ y $\Phi_M : \Omega_M \to \mathcal{O}_M$ estarían bien definidas.

Máquinas universales

¿Puede existir una máquina universal, similar a la definida por Turing?

Supongamos que podemos definir (bien) una codificación para cualquier máquina M sobre R, es decir, que $\langle M \rangle \in R^k$ para algún $k \in \mathbb{N}$, está bien definido.

Supongamos que existe esta máquina universal sobre R. Sea U una MDF sobre R que es capaz de simular cualquier otra MDF sobre R. De la definición, tenemos que $\mathcal{I}_U = R^p$ para algún p.

¿Qué pasa si $k > p$?

Necesitamos que el espacio de input no esté acotado.

Supongamos que R es un anillo conmutativo con unidad.

Definición (R^*)

Definimos R^* como el conjunto de todas las sucesiones finitas sobre R, o bien, como la unión disjunta:

$$R^* = \bigcup_{n \in \mathbb{N}} R^n.$$

Definición (R_∞)

Definimos R_∞ como el conjunto de los elementos x de la forma:

$$x = (\ldots\ ,\ x_{-2}\ ,\ x_{-1}\ ,\ x_0\ .\ x_1\ ,\ x_2\ ,\ \ldots)$$

donde $x_i \in R$ para todo $i \in \mathbb{Z}$, $x_k = 0$ para $|k|$ suficientemente grande y . es un marcador entre x_0 y x_1.

Operador de Shift

El espacio R_∞ tiene asociada una operación shift, de forma natural.

Definimos las funciones σ_ℓ y σ_r como las operaciones shift hacia la izquierda y derecha, respectivamente.

Para $x \in R_\infty$ se tiene que:

$$\sigma_\ell(x)_k = x_{k+1} \qquad \text{y} \qquad \sigma_r(x)_k = x_{k-1}.$$

Lo que hacen estas funciones es correr todo el vector bi-infinito en una posición hacia la derecha o la izquierda, de esta forma, se cambia de posición el marcador . en el vector.

¿Cómo podemos generalizar una función para que trabaje en R_∞?

Máquinas sobre R

Definición (Máquina sobre R)

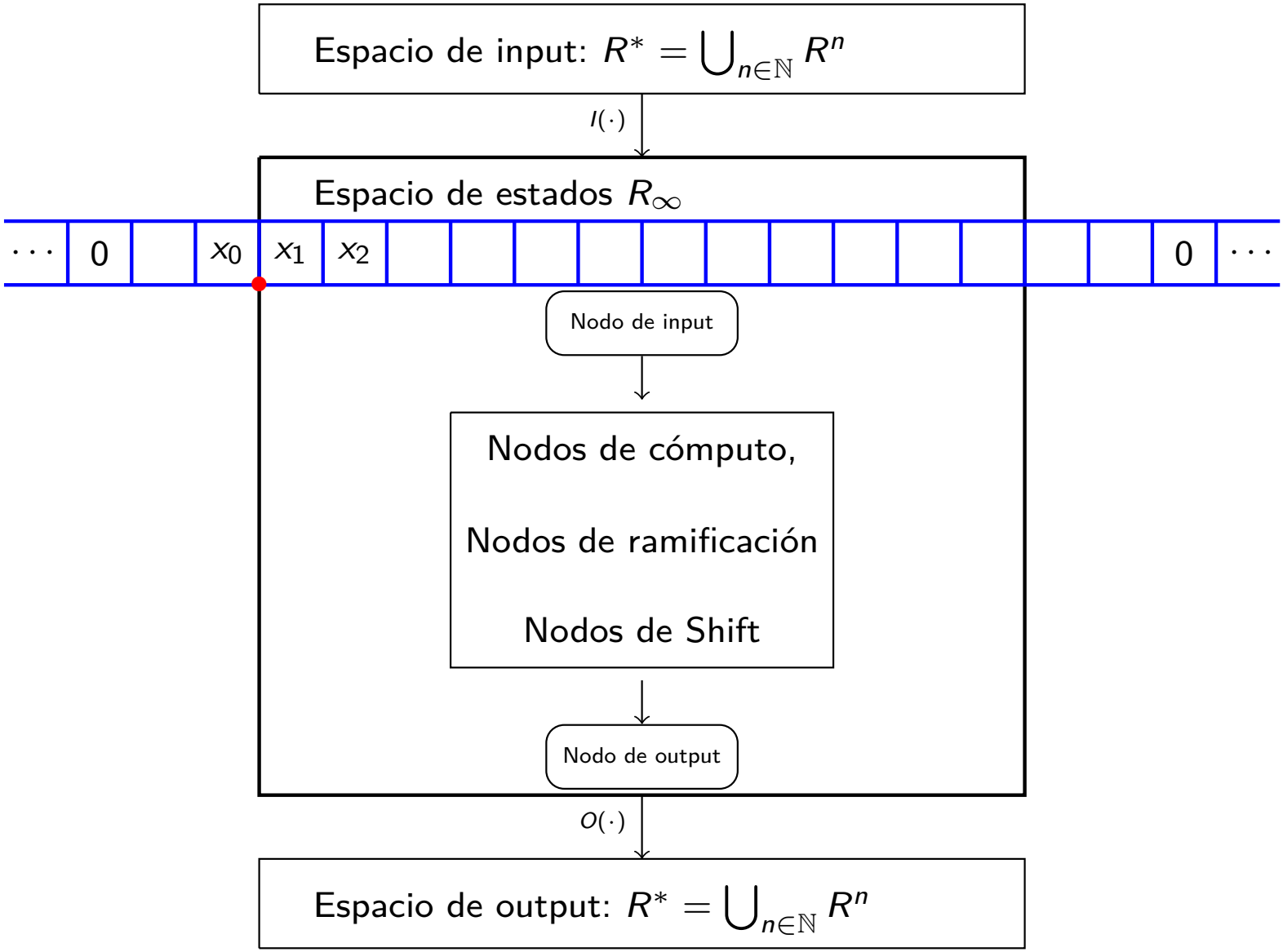
Una máquina M sobre R es un grafo dirigido y conexo con cinco tipos de nodos: los mismos cuatro de antes y nodos shift. Además, cada nodo shift η tiene asociada una función $g_\eta \in \{\sigma_\ell, \sigma_R\}$ y un único nodo siguiente β_η. Los espacios de input y output ahora son R^* y el espacio de estados es R_∞. Redefinimos las funciones I_M y O_M como:

$$I_M(x) = (\dots, 0, 0, \hat{n}.x_1, x_2, \dots, x_n, 0, 0, \dots), \qquad \text{para } x \in R^n,$$

donde $n \in \mathbb{N}$ y $\hat{n}$ corresponde a una sucesión de n 1s, y

$$O_M(\dots, x_0.x_1, x_2, \dots, x_k, \dots) = \begin{cases} 0 \in R^0 & \text{si } k = 0 \\ (x_1, \dots, x_k) \in R^k & \text{si no} \end{cases},$$

donde $k = \text{mín}\{i \in \mathbb{N} : x_{-i} = 0\}$.
El resto de los objetos son generalizados de manera (casi) directa.


Espacio de input: $R^* = \bigcup_{n\in\mathbb{N}} R^n$
$I(\cdot)$
Espacio de estados R_∞
$\cdots$ 0 x_0 x_1 x_2 0 $\cdots$
Nodo de input
Nodos de cómputo,
Nodos de ramificación
Nodos de Shift
Nodo de output
$O(\cdot)$
Espacio de output: $R^* = \bigcup_{n\in\mathbb{N}} R^n$

Problemas de decisión

Definición (Problema de decisión)

Un problema de decisión sobre R, corresponde a un conjunto $S \subseteq R^*$. Se responde a la pregunta de si $x \in S$ o no, para cualquier $x \in R^*$.

La complejidad de un problema de decisión S, está dada por la complejidad de su función característica, la que está definida para todo $x \in R$ como:

$$\chi s(x) = \begin{cases} 1 & \text{si } x \in S \\ 0 & \text{si no} \end{cases}$$

Costo de ejecución

La complejidad de una función es medida por una función de costos óptima.

Diremos que una máquina M sobre R trabaja en tiempo polinomial, si existen constantes $c, q \in \mathbb{N}^+$ tal que:

$$\mathrm{costo}_M(x) \leq c \cdot \big(\mathrm{size}(x)\big)^q, \qquad \text{para todo } x \in R^*.$$

¿Cómo podemos definir las funciones $\mathrm{costo}_M(\cdot)$ y $\mathrm{size}(\cdot)$?

La clase de complejidad P_R

Análogo a la teoría clásica:

Definición (Funciones polinomiales)

Una funcion $\varphi : R^* \to R^*$ se dice computable en tiempo polinomial (sobre R), si φ es computable por una máquina sobre R en tiempo polinomial.

Definición (Clase de complejidad P_R)

Un problema de decisión $S \subseteq R^*$ está en la clase P sobre R, denotado por $S \in P_R$, si su función característica es computable en tiempo polinomial.

La clase EXP_R se puede definir de manera completamente análoga a P_R.

La clase de complejidad NP_R

Definición (Clase de complejidad NP_R)

Un problema de decisión $S \subseteq R^*$ está en la clase NP sobre R, denotado por $S \in NP_R$, si existe una máquina M sobre R con $\mathcal{I}_M = R^* \times R^*$ y constantes $c, q \in \mathbb{N}^+$, tal que:

(1) Si $x \in S$, entonces existe algún $w \in R^*$ tal que $\Phi_M(x, w) = 1$ y

$$\text{costo}_M(x, w) \leq c \cdot \big(\text{size}(x)\big)^q.$$

(2) Si $x \notin S$, entonces no hay ningún $w \in R^*$ tal que $\Phi_M(x, w) = 1$.

Proposición

Para cada anillo R, se tiene que $P_R \subseteq NP_R$

Reducciones en tiempo polinomial

Definición (Reducción polinomial)

Decimos que un problema de decisión $S \subseteq R^*$ es p-reducible a otro problema de decisión $S' \subseteq R^*$, denotado como $S \hookrightarrow_p S'$, si existe una función $\varphi : R^* \to R^*$ computable en tiempo polinomial, tal que

$$\varphi(S) \subseteq S' \qquad \text{y} \qquad \varphi(R^* \backslash S) \subseteq R^* \backslash S'.$$

Esta definición es lo mismo que tener para todo $x \in R^*$ que:

$$x \in S \qquad \text{si y solo si} \qquad \varphi(x) \in S'.$$

Se dice que S' es "al menos tan difícil" como S.

Proposición

Si $S \hookrightarrow_p S'$ y $S' \in NP_R$, entonces $S \in NP_R$.

Idea de demostración del teorema

Teorema

Para cualquier campo $(R, =)$ se tiene que HN_R es NP_R-completo.

Para que HN_R sea NP_R-hard: Sea R un campo cualquiera y sea $L \in NP_R$. Esto significa que existe una máquina M no determinista sobre R, que decide a L. Queremos demostrar que existe una p-reducción $\varphi : R^* \to R^*$ tal que para todo $x \in R^*$:

$$x \in L \qquad \text{si y solo si} \qquad \varphi(x) \in HN_R.$$

Para esto, se utilizan las ecuaciones de registro (que no hemos definido) asociadas a la máquina M que decide a L, definiendo un sistema de ecuaciones que será factible si y solo si $x \in L$ (trabaja con el certificado de NP_R).

¿Cómo demostrar que $HN_R \in NP_R$?

Estado actual sobre $P = NP$

R	Órden	Costo	HN dec.	$HN \in NP$	$NP \subseteq EXP$	$P = NP$
$\mathbb{Z}$	¡	unitario	No	Yes	No	No
$\mathbb{Z}$	¡	bit	No	No	Yes	?
$\mathbb{Z}$	$=$	unitario	No	Yes	No	No
$\mathbb{Z}$	$=$	bit	No	No	Yes	No
$\mathbb{Q}$	¡	bit	?	Yes	Yes	No
$\mathbb{R}$	¡	unitario	Yes	Yes	Yes	?
$\mathbb{C}$	$=$	unitario	Yes	Yes	Yes	?
$\mathbb{Z}_2$	$=$	unitario	Yes	Yes	Yes	?

Lo último

Teorema

Se tiene que $P_{\mathbb{C}} = NP_{\mathbb{C}}$ si y solo si $P_{\mathbb{K}} = NP_{\mathbb{K}}$, donde $\mathbb{K}$ es cualquier campo algebraicamente cerrado con característica 0.

Problema abierto

¿Existe algún resultado parecido al teorema anterior, pero sobre campos con característica distinta de cero? (digamos 2) ¿Y sobre campos ordenados?

Bibliografía

- BLUM, L.; SHUB, M.; SMALE, S. 1989. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. Bulletin of the American Mathematical Society, **21**(1).

- BLUM, L. 2004. Computing over the reals: Where Turing Meets Newton. Notices of the American Mathematical Society, **51**(9).

- KALLA, P. 2014. Lectures on Complexity and Accuracy in Numeric Computations, The University of Utah.

- BLUM, L [et al.], 1998. Complexity and Real Computation. Primera edición, Springer Science & Business Media.