MA-53G Intr. a la Criptografía	**Semestre Primavera, 1996**

Cátedra No. 15 : 15 de Octubre

Transcriptor: Alejandro Hevia

1 Clases de Lenguajes

Nos abocaremos ahora a la tarea de definir más formalmente ciertos conceptos sobre los cuales descansa la criptografía moderna. Por ello, algunos conceptos básicos de complejidad como lenguajes y máquinas de Turing (determinísticas, no-determinísticas, probabilísticas) se supondrán conocidos.[1]

Definición 1 Lenguaje *Sea Σ un conjunto finito de símbolos. Diremos que Σ es un alfabeto (finito). Sea Σ^* el conjunto de todos las posibles palabras sobre el alfabeto Σ. Un* ***lenguaje*** *$\mathcal{L}$ es un conjunto de palabras sobre un alfabeto finito Σ, tal que $\mathcal{L} \subseteq \Sigma^*$.*

Definición 2 Máquina de Turing (Determinista) *Definiremos informalmente una Máquina de Turing (MT) como consistente de un* control finito de estado *(llamado usualmente* cabeza*), una* cinta de entrada *y una* cinta de salida*. Ambas cintas, la de* entrada *como la de* trabajo *consisten de celdas que pueden almacenar un sólo símbolo o letra de un alfabeto finito Σ. La cinta de trabajo tiene una cantidad infinita de celdas que pueden ser usadas por la MT. El control finito tiene dos* cabezales*, cada uno apuntando a una celda específica en cada cinta, y con la habilidad de moverse cada una hacia la izquierda o derecha. Los estados posibles del* control finito *pueden ser cualquiera de un conjunto finito Q de estados, uno de los cuales se distinguirá como* estado de aceptación *y otro como* estado inicial*. Asociado al* control finito de estados *tendremos una función de transición δ la cual, en base a las celdas actualmente apuntadas por los* cabezales *y el estado actual de la* cabeza, describe dónde mover los cabezales (izquierda o derecha) y el nuevo estado al cual entra la *cabeza*. Esta función puede también especificar un cambio en la celda actualmente apuntada por el cabezal de la cinta de trabajo (no así en la cinta de *entrada*).
La entrada de una MT consiste de una *palabra* (una concatenación finita de letras del alfaveto Σ). Esta palabra es colocada entre un símbolo especial de Σ, \$, al comienzo

[1] Para un tratamiento más profundo, el lector interesado puede consultar el texto *Computational Complexity*, *C. Papadimitriou*, Addison Wesley, 1994.

de la *cinta de entrada*, un símbolo en cada celda. Inicialmente, cada celda de la cinta de trabajo posee el símbolo *blanco* y los *cabezales* apuntan a la primera celda de la cinta de entrada y a la primera celda de la cinta de trabajo.

Definición 3 Máquina de Turing Probabilista *Una máquina de Turing probabilista M, es una MT que posee un estado especial de* petición de lanzamiento de moneda*. Cuando M entra en este estado, adquiere conocimiento sobre el resultado del lanzamiento de una moneda, el cual puede ser "cara" (1) o "sello" (0) (cada uno con probabilidad $\frac{1}{2}$). Cada lanzamiento de moneda se considera que ocurre en forma independiente de cualquier otro lanzamiento.*

Definición 4 Aceptación de Palabras
Diremos que una máquina de Turing M acepta *una palabra x de un lenguaje $\mathcal{L}$si la máquina M al recibir en su cinta de entrada la codificación* [2] *de la palabra x, eventualmente se detiene quedando en su cinta de trabajo la codificación de la palabra* aceptar.[3] *Esto último lo notaremos por $M(\,x\,) =$ acepta.*

Definición 5 Aceptación de Lenguajes
Diremos que una máquina de Turing M acepta *un lenguaje $\mathcal{L}$si ocurre que: M acepta x si y sólo si $x \in \mathcal{L}$.*

Definición 6 *Decimos que un lenguaje $\mathcal{L}$está en $\mathcal{P}$si existe una máquina de Turing M y un polinomio p tal que*

1. *$x \in L$ si y sólo si M acepta x, es decir $M(x) =$ acepta.*
2. *Al operar M en la entrada x realiza a lo más $p(\,|\,x\,|\,)$ pasos, donde $|\,x\,|$ es el número de caracteres de x.*

La clase de lenguajes $\mathcal{P}$es considerada clásicamente como la clase de los lenguajes "eficientemente computables".

Observación 1 *Notar que $\mathcal{P}$es cerrado bajo complemento, pues pertenencia en el lenguaje $\overline{\mathcal{L}}$ puede ser decidida en tiempo polinomial por la máquina de Turnig M' tal que $M'(x) =$ acepta si y sólo si $M(x) \neq$ acepta.*

[2]No entraremos aquí en el detalle de cómo debe hacerse esta codificación y consideraremos la noción intuitiva de este proceso, bajo el supuesto que este sólo realiza lo "justo y necesario".

[3]Basta que sea alguna codificación predefinida de ella.

Definición 7 *Decimos que un lenguaje* $\mathcal{L}$*pertenece a* $\mathcal{NP}$*si existe una máquina de Turing* $M\,(\,\cdot\,,\,\cdot\,)$ *y polinomios* p *y* q *tales que*

1. *Si* $x \in \mathcal{L}$*entonces* $\exists\, y,\ |y| \leq q(|x|)$ *tal que* $M(x,y) =$ *acepta.*
2. *Si* $x \notin \mathcal{L}$*entonces* $\forall\, y,\ |y| \leq q(|x|)$ *tal que* $M(x,y) \neq$ *acepta.*
3. M *en la entrada* x *se detiene en a lo más* $p(\,|x|\,)$ *pasos.*

Notar que esta definición es equivalente a la definición de $\mathcal{L}\in \mathcal{NP}$si existe una máquina de Turing no-determinista de tiempo polinomial M, la cual acepta x si y sólo si $x \in \mathcal{L}$. Aquí, la palabra y de la definición corresponde a la *adivinanza* de la máquina de Turing no-determinista.

Observación 2 *Si* $\mathcal{NP}$*es cerrado o no es un problema abierto. Su complemento se denomina* $\rfloor\wr\mathcal{NP}$*.*

Definición 8 *Decimos que un lenguaje* $\mathcal{L}$*está en* $\mathcal{BPP}$*si existe una máquina de Turing* $M\,(\,\cdot\,,\,\cdot\,)$ *y polinomios* p *y* q *tales que*

1. *Si* $x \in \mathcal{L}$*entonces* $\mathbb{P}_{\rho\in\{0,1\}^{q(|x|)}}[\,M(x,\rho) = \textit{acepta}]\ \geq \frac{2}{3}$
2. *Si* $x \notin \mathcal{L}$*entonces* $\mathbb{P}_{\rho\in\{0,1\}^{q(|x|)}}[\,M(x,\rho) = \textit{acepta}]\ \leq \frac{1}{3}$
3. *Al operar* M *sobre la entrada* x *realiza a lo más* $p(|x|)$ *pasos.*

Ejercicio 1 *Demuestre que en la definición de* $\mathcal{BPP}$*es posible reemplazar las constantes* $\frac{2}{3}$ *y* $\frac{1}{3}$ *por* $1-\epsilon$ *y* ϵ *respectivamente (con* $0 < \epsilon < \frac{1}{2}$*) y la clase* $\mathcal{BPP}$*no se modifica.*

Dem: Denotemos por BPP_ϵ al conjunto definido en forma análoga a $\mathcal{BPP}$, cambiando las constantes $\frac{2}{3}$ y $\frac{1}{3}$ por $1-\epsilon$ y ϵ respectivamente.
Supongamos que $\mathcal{L}\in\ \mathcal{BPP}_\epsilon$ y probemos que $\mathcal{L}\in\mathcal{BPP}_\delta$ donde
$0 < \delta < \epsilon < \frac{1}{2}$.
Con ello, se sigue que existe una máquina de Turing M, tal que si $x \in \mathcal{L}$

$$\mathbb{P}_{\rho\in\{0,1\}^{q(|x|)}}[\,M(x,\rho) = \textit{acepta}]\ \leq 1-\epsilon\,.$$

Luego, si $x \notin \mathcal{L}$sigue que

$$\mathbb{P}_{\rho \in \{0,1\}^{q(|x|)}}[\, M(x,\rho) = \ acepta] \ \leq \epsilon \,.$$

Sea M' otra máquina de Turing que simula k veces la máquina M y acepta si y sólo si al menos $\frac{k}{2}$ de estas simulaciones llevan a M a aceptar. Sea χ el número de veces que M acepta.

Si $x \in L$, entonces como $\chi \sim Bin(k, 1-\epsilon)$ tendremos que

$$\begin{aligned}
\mathbb{P}[\, M' \,\text{acepta}\,] &= \mathbb{P}[\chi \geq \frac{k}{2}] \\
&= 1 - \mathbb{P}[\chi < \frac{k}{2}] \\
&= 1 - \mathbb{P}[\mathbb{E}(\chi) - \chi > k\,(\frac{1}{2} - \epsilon)] \\
&\geq 1 - \mathbb{P}[|\chi - \mathbb{E}(\chi)| > k\,(\frac{1}{2} - \epsilon)] \\
&\geq 1 - \frac{\mathbb{V}ar\chi}{k^2(\frac{1}{2} - \epsilon)^2} \\
&= 1 - \frac{1}{k}\frac{\epsilon(1-\epsilon)}{(\frac{1}{2} - \epsilon)^2} \\
&= 1 - \delta
\end{aligned}$$

(donde denotamos por δ a $\frac{1}{k}\frac{\epsilon(1-\epsilon)}{(\frac{1}{2}-\epsilon)^2}$).
Si ϵ es constante, esta probabilidad puede hacerse arbitrariamente cercana a 1 al aumentar k.

Análogamente, si $x \notin \mathcal{L}$, entonces como $\chi \sim Bin(k, \epsilon)$ tendremos que

$$\begin{aligned}
\mathbb{P}[\, M' \,\text{acepta}\,] &\leq \mathbb{P}[\chi \geq \frac{k}{2}] \\
&= \mathbb{P}[\chi - \mathbb{E}(\chi) \geq k\,(\frac{1}{2} - \epsilon)] \\
&\leq \mathbb{P}[|\chi - \mathbb{E}(\chi)| \geq k\,(\frac{1}{2} - \epsilon)] \\
&\leq \frac{\mathbb{V}ar\chi}{k^2(\frac{1}{2} - \epsilon)^2} \\
&= \frac{1}{k}\frac{\epsilon(1-\epsilon)}{(\frac{1}{2} - \epsilon)^2} \\
&= \delta
\end{aligned}$$

Luego se tiene que la máquina de Turing M' acepta cuando $x \in \mathcal{L}$con probabilidad al menos $1-\delta$ tal que $0<\delta<\epsilon<\frac{1}{2}$ y con probabilidad δ cuando $x \notin \mathcal{L}$. Luego $L \in$BPP$_\delta$.

Queda al lector propuesto demostrar que se tiene el resultado incluso para $\epsilon=\frac{1}{2}-\frac{1}{p(n)}$ donde p es un polinomio cualquiera y $n=|x|$. □

Definición 9 *Decimos que un lenguaje $\mathcal{L}$está en $\mathcal{RP}$si existe una máquina de Turing $M(\cdot,\cdot)$ y polinomios p y q tales que*

1. *Si $x \in \mathcal{L}$entonces $\mathbb{P}_{\rho\in\{0,1\}^{q(|x|)}}[\,M(x,\rho)=\text{acepta}]\ \geq \frac{2}{3}$*
2. *Si $x \notin L$ entonces $\mathbb{P}_{\rho\in\{0,1\}^{q(|x|)}}[\,M(x,\rho)=\text{acepta}]\ =\ 0$*
3. *Al operar M sobre la entrada x realiza a lo más $p(|\,x\,|)$ pasos.*

Ejercicio 2 *Demuestre que en la definición de RP es irrelevante el valor $\frac{2}{3}$ de la cota para la probabilidad de aceptar en el caso $x \in L$ y puede ser reemplazada por cualquier constante $\epsilon \in (\,0\,,\,1\,)$.*

Observación 3 *La definición de $\rfloor\wr\mathcal{RP}$es análoga sólo que exige que la probabilidad de aceptar cuando $x \in \mathcal{L}$sea igual a 1 y la probabilidad de aceptar cuando $x \notin \mathcal{L}$sea menor o igual a $\frac{1}{3}$.*

Definición 10 *Decimos que un lenguaje $\mathcal{L}$está en $\mathcal{ZPP}$si existe una máquina de Turing probabilística $M(\cdot)$ y un polinomio p tal que*

1. *$x \in L$ si y sólo si $M(\,x\,)=$ acepta.*
2. *La esperanza del número de pasos que M ejecuta en la entrada x es a lo más $p(\,|x|\,)$.*

Relaciones entre lenguajes

Se sabe que $\mathcal{P}\subseteq\ \mathcal{NP}$, que $\mathcal{P}\subseteq\ \mathcal{RP}\subseteq\ \mathcal{BPP}$. Además se cumple que $\mathcal{ZPP}=\ \mathcal{RP}\cap$ $\rfloor\wr\mathcal{RP}$, que $\rfloor\wr\mathcal{RP}\subseteq\ \mathcal{BPP}$y que $\mathcal{RP}\ \subseteq\ \ \mathcal{NP}$(basta adivinar el resultado del "lanzamiento de monedas" que realiza la máquina de Turing probabilísta).
Actualmente no se sabe si es cierto que $\mathcal{NP}\neq\ \mathcal{P}$aunque se conjetura que es cierto. Asimismo, aún no se clarifica cuál es la relación entre $\mathcal{BPP}$y $\mathcal{NP}$o entre $\mathcal{BPP}$y $\mathcal{P}$. [4]

[4] Un ejemplo de lenguaje que está es $\mathcal{BPP}$y no se sabe aún si está en $\mathcal{P}$es el lenguaje de todos los enteros primos (esto es, verificar primalidad).