El algoritmo de compresión de datos de Lempel Ziv y la catástrofe del bit

Elvira Mayordomo

Workshop MOISES

Febrero 2004

Contenido

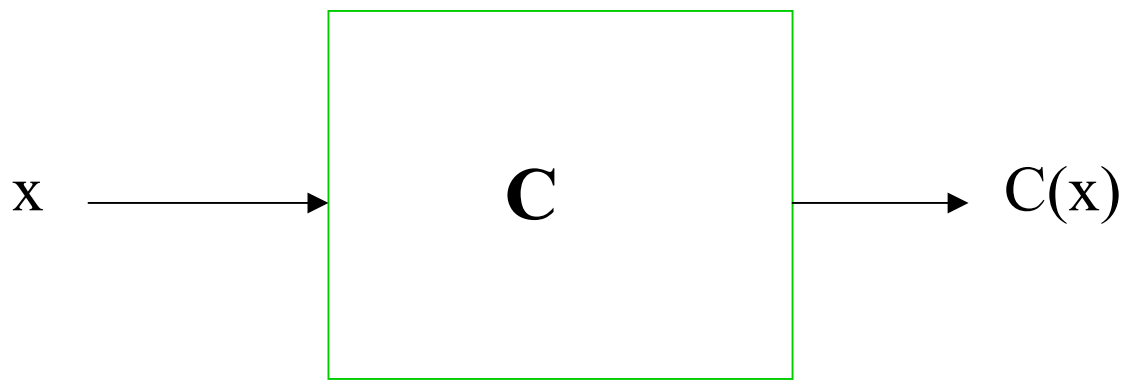
- Algoritmos de compresión sin pérdidas
- LZ78: idea principal
- ¿Cuánto comprime?
- La catástrofe del bit

Hoy

- Ziv, Lempel: “Compression of individual sequences via variable rate coding” IEEE Trans. Inf. Th., 24 (1978), 530-536
- Sheinwald: “On the Ziv-Lempel proof and related topics”. Procs. IEEE 82 (1994), 866-871

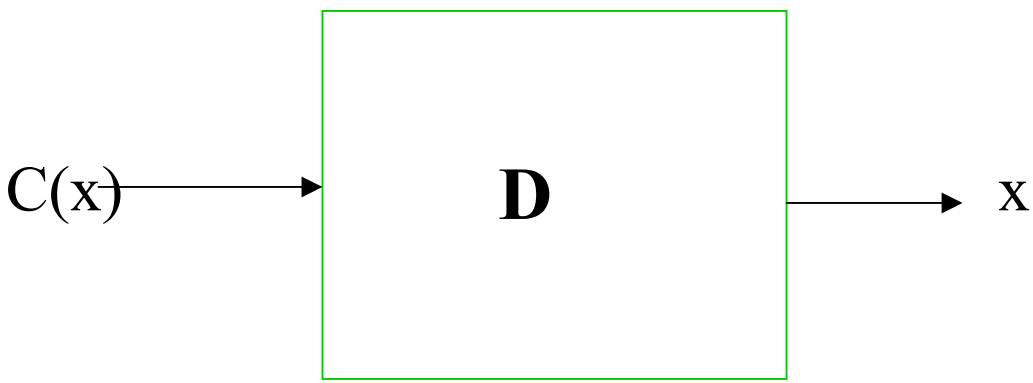
Algoritmos de compresión

- **La entrada y la salida son strings (cadenas, secuencias finitas)**



Algoritmos de compresión

- **A partir de la salida se puede reconstruir la entrada** **siempre: lossless compressor**



Objetivo

- **Comprimir lo máximo posible todos los strings ???**
- **Eso es imposible, ¿por qué?**
- **Nos conformamos con comprimir “lo fácil de comprimir”**

El LZ78

- Es el algoritmo de compresión más utilizado (y estudiado)
- De él se derivan el compress de Unix, el formato GIF y los algoritmos LZW, LZMW

LZ78: idea principal

- **Partimos el string en trozos (frases) de forma que cada trozo es uno de los anteriores más un símbolo**

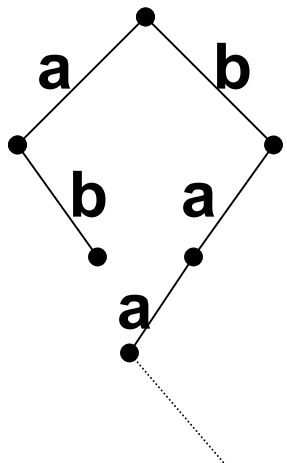
ababbabaaabaaabba

LZ78: idea principal

- **Partimos el string en trozos (frases) de forma que cada trozo es uno de los anteriores más un símbolo**

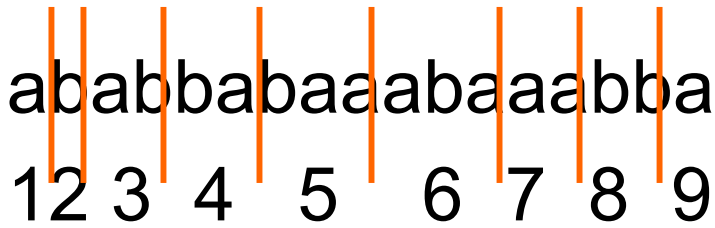
LZ78: idea principal

ababbabaabaababbabaa



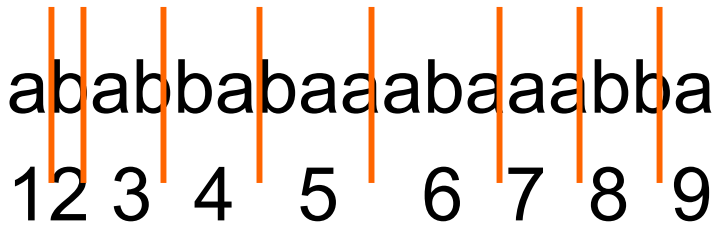
LZ78: idea principal

- **Numeramos las frases**



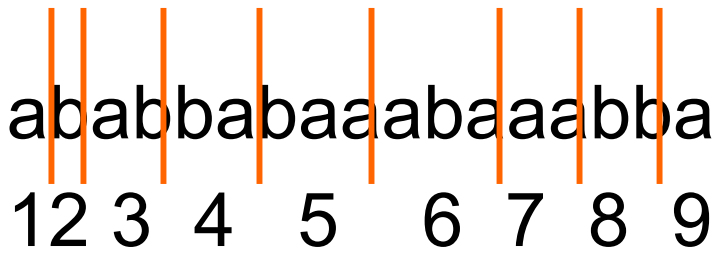
LZ78: idea principal

- **Numeramos las frases**



- **Cada frase es una de las anteriores más un símbolo**

LZ78: idea principal



(0,a) (0,b) (1,b) (2,a) (4,a) (3,a) (1,a) (2,b) (1,)

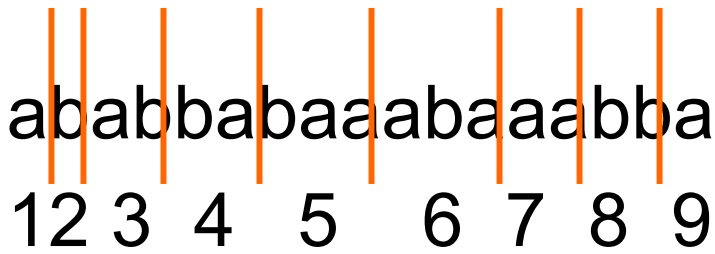
1 2 3 4 5 6 7 8 9

0 es la frase vacía

LZ78: idea principal

- **Se trata de “parsear” la entrada en frases diferentes**
- **¿Cuál es la salida exactamente?**

LZ78



(0,a) (0,b) (1,b) (2,a) (4,a) (3,a) (1,a) (2,b) (1,)

1 2 3 4 5 6 7 8 9

- **Codificamos** (0,a)(0,b)(1,b)(2,a)...

LZ78: la salida

- **Codificamos la parte correspondiente a la frase número** n **que es**

  (i,s)

  utilizando $\lceil \log_2(n) \rceil$ **bits para** i

  y para el símbolo s**, depende de cuantos símbolos diferentes haya**

LZ78: ¿cuánto comprime?

- **El tamaño de LZ(x) depende sólo del número de frases en que dividimos x**

a ab aba abaa abaab 15 símbolos

b a ba bb aa 8 símbolos

LZ78: ¿cuánto comprime?

- **Si t(x) es el número de frases en que LZ78 divide a x:**

$$|LZ(x)| = \sum_1^{t(x)} \left(\lceil \log_2(n+1) \rceil + \log_2 \alpha \right)$$

$$\approx \quad t(x) \left(\log_2(t(x)) + \log_2 \alpha \right)$$

Compresores de estados finitos

- **Dado A FSC y una partición cualquiera en frases distintas de**

  $x = w(1) \ldots w(f(x))$:

  $$|A(x)| \geq f(x) \log_2(f(x)) - o(|x|)$$

Compresores de estados finitos

- **Dado A FSC**

$$\lim_n \frac{|A(z[1..n])|}{n} \geq \lim_n \frac{|LZ(z[1..n])|}{n}$$

- **Luego el algoritmo de LZ es asintóticamente mejor que los FSC.**

El algoritmo LZ

Al ser universal para los compresores de estados finitos, se puede creer que sólo tendrá propiedades "óptimas"

Pero no es así ...

La catástrofe del bit

¿Qué pasa si añades un bit delante de una secuencia muy compresible?

1001010010101010 ...

1100101001010101 0 ...

Preguntas abiertas

- **¿Existe la catástrofe del bit?**

Experimentalmente parece que sí ??

No hay demostración

Preguntas abiertas

- **Si existe, ¿cómo la solucionamos?**

Aunque sea una solución parcial ...