

# INF155: Informática Teórica

## Clase 27: Problemas intratables II

Aldo Berrios Valenzuela

Martes 21 de Junio de 2016

### 1. Problemas Intratables P vs NP

---

**Teorema 1.1.** Si un problema NP-completo  $A$  está en P,  $P = NP$ .

**Demostración.** Ejercicio verlo. □

Notar que  $n^{\log_2(n)}$  no es polinomial ( $\log_2(n)$  crece al infinito, no está acotado por un polinomio)

$n^{\log_2(n)} = 2^{(\log_2(n))^2}$ , crece más lento que  $a^n$  para todo  $a > 1$  ya que  $a^n = 2^{n \log_2(a)}$ ,  $n \log_2(a)$  crece más rápido que  $(\log_a(n))^2$  (las demostraciones de  $n^{\log_2(n)} = 2^{(\log_2(n))^2}$  y  $a^n = 2^{n \log_2(a)}$  se encuentran en los anexos 2.1 y 2.2 respectivamente)

#### 1.1. SAT

Dada una fórmula lógica  $\phi(x_1, \dots, x_n)$  (escrita con variables  $\wedge, \vee, \neg$ ), determinar si hay una asignación de valores (verdadero o falso) a las variables  $x_i$  tal que  $\phi(x_1, \dots, x_n)$  sea verdadera (Satisfiability)

Plan:

- SAT es NP-completo (TM no determinista  $\leq_p$  SAT, teorema de Cook).
- 3SAT es NP-completo (SAT  $\leq_p$  3SAT)

⋮

**Definición 1.1.** Una variable o la negación de ella se llama literal, se anota  $x$  para la variable y  $\bar{x}$  para  $\neg x$ . Una cláusula es una disyunción<sup>1</sup> de literales, como  $x \vee \bar{y} \vee z$ .

**Definición 1.2.** Una fórmula lógica está en forma normal conjuntiva (CNF<sup>2</sup>) si es la conjunción<sup>3</sup> de cláusulas.

- CSAT: La fórmula lógica  $\phi(x_1, \dots, x_n)$  en CNF es satisfacible. CSAT es NP-completo (SAT  $\leq_p$  CSAT)
- $k$ -SAT: La fórmula lógica  $\phi(x_1, \dots, x_k)$  en CNF, donde cada cláusula tiene  $k$  literales.
- 3SAT es NP-completo.
- 1SAT, 2SAT tienen solución lineal (!).

---

<sup>1</sup>Una disyunción es un OR

<sup>2</sup>no confundir con Chomsky Normal Form

<sup>3</sup>Una conjunción es un AND

**Teorema 1.2** (Cook). SAT es NP-completo.

**Demostración.** SAT  $\in$  NP.

- SAT  $\in$  NP: Adivinar valores para las variables, calcular el valor de la expresión.
- TM no determinista  $M, \omega, p(n)$ ; determinista si  $\omega \in \mathcal{L}(M) \leq_p$  SAT con  $M$  aceptando en a lo más  $p(|\omega|)$  pasos.

Idea general: Armar una matriz de expresiones:

$$\begin{array}{cccc} x_{01} & x_{02} & \cdots & x_{0p(|\omega|)} \\ x_{11} & x_{12} & \cdots & x_{1p(|\omega|)} \\ \vdots & & & \\ x_{p(|\omega|)1} & \cdots & \cdots & x_{p(|\omega|)p(|\omega|)} \end{array} \quad (1.1)$$

Observaciones:

- Cada fila de la matriz representa una descripción instantánea de la máquina de turing.
- La primera fila es especial: ID  $M$  apronta a procesar  $\omega$
- Para la fila  $i$  a  $i + 1$ : movida legal de  $M$  a  $M$  acepté;  $ID_{i+1} = ID_i$ .
- Revisar si el estado es de aceptar.

Escribir esta expresión toma tiempo polinomial (en  $M, \omega$ )

□

**Teorema 1.3.** SAT  $\leq_p$  CSAT

**Demostración.** Aplicar de Morgan para llevar negaciones a variables, luego distribuir para lograr CNE,

Lo enredado es hacerlo en tiempo polinomial...

□

**Teorema 1.4.** CSAT  $\leq_p$  3SAT

**Demostración.** Problemas son cláusulas con más de 3 literales y cláusulas con menos de 3 literales.

- Menos de 3 literales, por ejemplo:  $x \vee y \rightsquigarrow (x \vee y \vee x_1) \wedge (x \vee y \vee \bar{x}_1)$  (con nueva variable  $x_1$ )
- Más de tres literales, por ejemplo:  $u \vee v \vee \omega \vee x \rightsquigarrow (u \vee v \vee x_1) \wedge (\bar{x}_1 \vee \omega \vee x)$

□

## 2. Anexos

**Anexo 2.1.** Demuestre que:

$$n^{\log_2(n)} = 2^{(\log_2(n))^2} \quad (2.1)$$

**Demostración.** Definimos la función  $f$  como sigue:

$$f(x) = 2^x \quad (2.2)$$

Como puede notar,  $f$  corresponde a la función inversa de  $\log_2$ . Entonces:

$$n^{\log_2(n)} = f\left(\log_2\left(n^{\log_2(n)}\right)\right) = f\left(\log_2(n) \cdot \log_2(n)\right) = f\left((\log_2(n))^2\right) = 2^{(\log_2(n))^2}$$

□

**Anexo 2.2.** Demuestre que:

$$a^n = 2^{n \log_2(a)} \quad (2.3)$$

*Demostración.* Sea  $f$  una función definida por:

$$f(x) = 2^x$$

Entonces:

$$a^n = f(\log_2(a^n)) = f(n \log_2(a)) = 2^{n \log_2(a)}$$

□