NP - completitud

- Clase P : problemas de decisión resolubles en tiempo polinomial (respecto del tamaño de los datos)
- Algoritmos no - deterministas.
  - Admiten instrucciones de ramificación limitada (o eq. dobles).
  - Si una rama tiene éxito, el algoritmo termina con éxito.
- Visualizaciones alternativas
  - Paralelismo eventualmente no - acotado.
  - Oráculo mágico que escoge el camino exitoso, cuando éste existe.
- Clase NP : problemas resolubles por un algoritmo polinomial no - determinista
- Definición alternativa vía verificadores polinómicos de pruebas (polinómicas)

$$A = \{ w \mid V \text{ acepta algún par } \langle w, c \rangle \}$$

↳ en tiempo polinomico para w

- Equivalencia de las dos pruebas
  - Cada rama exitosa es un certificado verificado polinómicamente
  - Todos los posibles certificados polinomiales se pueden generar (no-deterministicamente) en tiempo polinomial, y después cada uno es chequeado.
- Problemas NP - completos : Si están en P, entonces $P = NP$.

Reducciones (polinómicas) entre problemas

- $P_1 \ll_p P_2 \ ::= \ \exists R$ polinomial $\forall n \ P_1(n) \Leftrightarrow P_2(R(n))$
- Q es NP - duro sii $\forall P_1 \in NP \ \exists R$ que reduce P_1 a Q
- Q es NP - completo sii $Q \in NP$ y es NP - duro.

El problema SAT

- Problema de decisión de la satisfactibilidad de fórmulas proposicionales.
- Trivialmente está en NP
- En principio hay que generar y probar cada valoración de las variables de la fórmula, por lo que el correspondiente algoritmo de backtracking sería exponencial.

El problema del CIRCUIT VALUE

- Se trata de calcular la salida de un circuito booleano secuencial (con puertas AND, OR, NOT) dadas sus entradas.
- CV $\in$ P, basta ir calculando ordenadamente las salidas de cada puerta
- CV es P-completo

  Dem: Se genera un circuito "cuadriculado homogéneo" que codifica el funcionamiento de cada MT que trabaja en tiempo polinomial

---

Teorema de Cook: SAT es NP-completo

- El problema CIRCUIT SAT: averiguar si un circuito booleano secuencial puede producir como salida 1.
- CIRCUIT SAT $\leq_p$ SAT
  - La igualdad se codifica con $(\neg g \vee x) \wedge (g \vee \neg x)$
  - La negación se codifica como $(\neg g \vee \neg h) \wedge (g \vee h)$
  - AND an OR se codifican pasando a FNC $g \Leftrightarrow (h \overset{\wedge}{\vee} h')$
- CIRCUIT SAT es NP-completo.

  Idea genial en la demostración: podemos suponer que cada transición de las máquinas a codificar es una elección entre 2 alternativas: ¡añadimos una variable booleana para codificar cada elección y la máquina "se convierte" en determinista!
  El resto de la prueba ¡es análoga a la de CIRCUIT_VALUE!

Th. 8.1. CIRCUIT VALUE es P- completo

Def: Tabla de computación de una máquina que trabaja en $O(|x|^k)$

- Filas: estado de la cinta con símbolos en Σ
- Representación del estado de la máquina mediante nuevos pares σ_q, $\sigma \in \Sigma$, $q \in K$.
- Estados finales representados simplemente por "yes" o "no".
- Tablas aceptadoras contienen algún "yes".

Def: Circuito que evalúa una tabla

- Representación binaria de los símbolos de la tabla
- Cada componente de una nueva fila sólo depende del símbolo en la correspondiente posición en la fila anterior, junto con los dos símbolos vecinos.
- La función (booleana) que calcula los bits de una posición a partir de los de las tres posiciones que influyen en ella puede ser computada por el circuito asociado a una expresión booleana $(\wedge, \vee, \neg)$ de tamaño constante (dependiente de Σ y K).
- Insertando dicho circuito sobre cada componente de la tabla se genera el circuito de tamaño $O(|x|^{2k})$ que establece la reducción perseguida
- De hecho podemos ver que el circuito puede ser construido en espacio logarítmico pues al efecto basta con generar las puertas de entrada a partir de la entrada, y contar hasta $|x|^k$ para generar el resto de blancos. Después hemos de anidar dos bucles de dicha longitud para ir construyendo las correspondientes copias del circuito básico que computa cada casilla de la tabla.

Corolario: La evaluación de circuitos monótonos (sin negación) también es un problema P- completo.

Dem: En todo circuito podemos llevar las negaciones a las entradas aplicando las Leyes de Morgan, lo que puede hacerse en espacio logarítmico pues basta ir contando el número de negaciones anidadas que afectan a cada conectivo.

Teorema de Cook (Th. 8.2) : SAT es NP-completo

- SAT está en NP : se adivinan los valores que satisfacen la fórmula y se comprueba que ello en verdad es así en tiempo polinomial.
- Veamos que todo problema en NP puede ser reducido a CIRCUIT SAT

  Sea $M = (K, \Sigma, \Delta, s)$ que decide L en tiempo n^k.

  Supongamos que en cada par estado-símbolo tenemos exactamente 2 elecciones (en caso contrario se introducen estados auxiliares intermedios)

  Codificamos las dos elecciones posibles con los valores 0 y 1, de manera que una cadena de elecciones no-deterministas durante $|x|^k - 1$ pasos se corresponde con un valor en $\{0,1\}^{|x|^k - 1}$

  Incorporamos una nueva variable c_i que se utiliza en el i-ésimo paso de la computación, apareciendo como entrada adicional de los circuitos que computan la i-ésima fila de la tabla de computación de la máquina
- Claramente la construcción puede hacerse en espacio logarítmico y el circuito obtenido es satisfactible sii la entrada que se le facilita está en L.

Variantes del problema SAT

- 3 SAT : Satisfactibilidad de fórmulas en forma normal conjuntiva con cláusulas con 3 literales a lo sumo
  - La reducción de CIRCUIT SAT a SAT produce fórmulas tales
- (3 SAT, 3, 2) : Satisfactibilidad de fórmulas en forma normal conjuntiva con cláusulas con 3 literales a lo sumo y cada variable repetida a lo sumo 3 veces, no todas ellas afirmada o negada.
  - x se desdobla en $x_1, \dots, x_k$ cuya igualdad se impone vía $(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge \dots \wedge (\neg x_{k-1} \vee x_k) \wedge (\neg x_k \vee x_1)$
- 2 SAT está en P
  - G definido con vértices x y $\neg x$ conectados por arcos que se corresponden con su presencia conjunta en una cláusula de la fórmula con el primero de ellos negado
  - ϕ insatisfactible si en $G(\phi)$ hay alguna variable x con $\neg x$ alcanzable desde x y x alcanzable desde $\neg x$.

Tma 9.1 : ϕ es insatisfactible sii $\exists x$ | en $G(\phi)$ hay un ciclo conteniendo a x y $\neg x$

Demost : Los arcos de $G(\phi)$ reflejan implicaciones lógicas, por lo que la existencia de un ciclo tal en $G(\phi)$ implica que $\phi \Rightarrow (x \Leftrightarrow \neg x)$, con lo que ha de ser ϕ = false.

Para el recíproco, si no hay ningún ciclo tal ejecutamos el siguiente algoritmo para encontrar una valoración que haga cierta ϕ.

MIENTRAS quede alguna variable x sin valorar

HACEMOS tomamos el literal ℓ de x para el que no hay ningún camino de ℓ a $\neg\ell$ en $G(\phi)$

asignamos a todos los literales ℓ' con $\ell \Rightarrow \ell'$ en $G(\phi)$ el valor true, y a sus negaciones false

{ con ello se establece el invariante de que todo nodo accesible desde otro marcado true lo está así también }

{ la simetría dual de $G(\phi)$ hace que todo antecesor de un nodo marcado con false ha de estar marcado igual }

{ nunca podemos tener $\ell \Rightarrow \ell'$ y $\ell \Rightarrow \neg\ell'$ pues la asimetría dual querería $\ell' \Rightarrow \neg\ell$ y así un ciclo conteniendo ℓ y $\neg\ell$ }

Al terminar se satisfacen todas las implicaciones en ϕ en virtud del primer invariante reseñado antes.

MAX 2 SAT : Máxima satisfactibilidad de un conjunto de cláusulas $\ell \vee \ell'$

Tma 9.2 : MAX 2 SAT es NP-completo

Lema : De entre x, y, z, w, $\neg x \vee \neg y$, $\neg y \vee \neg z$, $\neg z \vee \neg x$, $x \vee \neg w$, $y \vee \neg w$, $z \vee \neg w$ podemos satisfacer simultáneamente a lo sumo 7, pero ello es así sólo siempre que se cumple $x \vee y \vee z$. En caso contrario sólo podemos llegar a 6.

Reducción de 3 SAT a MAX2 SAT

Cada cláusula $\alpha \vee \beta \vee \gamma$ de ϕ se codifica por medio de las 10 cláusulas anteriores con α, β y γ como x, y y z, respectivamente, introduciendo una w nueva en cada caso

Podemos satisfacer $7m$ cláusulas de ϕ, con m cláusulas, sii ϕ es satisfactible

El problema NAESAT ("no todos los literales de cada cláusula valen lo mismo")

. Buscamos una valoración que haga ciertas las cláusulas, pero nunca a base de hacer ciertos al mismo tiempo sus tres literales.

— NAESAT es NP-completo

. La reducción de CIRCUIT SAT a 3SAT lo es también a NAESAT.

. Completamos las cláusulas con menos de tres literales con la misma variable z.

Dem: • Si C bajo σ es cierto $\Rightarrow$ $\phi(C)$ bajo σ lo es también

Podemos tomar $\sigma(z) = \text{false}$ y sigue siendo cierta. Entonces en ninguna cláusula de $\phi(C)$ todos los literales son ciertos. Ello es trivial cuando z forma parte de ellas. Nos quedan sólo las cláusulas correspondientes a las puertas AND y OR. Para las primeras tenemos $(\neg g \vee h \vee z)$, $(\neg g \vee h' \vee z)$ y $(\neg h \vee \neg h' \vee g)$. No podemos tener $\neg h$, $\neg h'$ y g simultáneamente, pues en tal caso las dos primeras cláusulas serían falsas. El caso de OR es análogo.

• Si $\phi(C)$ es NAESAT-satisfactible bajo σ, también lo será bajo su complementario $\overline{\sigma}$. En uno de los casos tendremos $\overline{\sigma}(z) = \text{false}$, con lo que la traducción a 3-SAT es cierta bajo σ, con lo que el circuito C lo sería también.

ALGUNOS PROBLEMAS DE TEORIA DE GRAFOS

. Grafos no dirigidos: los arcos son conjuntos con dos vértices $[i,j]$.

$G = (V, E)$

Conjunto de vértices independientes

. $I \subseteq V$ tal que $\neg \exists\ i,j \in V$, $[i,j] \in E$.

. Conjuntos maximales de vértices independientes: INDEPENDENT SET.

Dado $k \in \mathbb{N}$ ¿ $\exists\ I_k$ conj. vértices independientes con $|I_k| = k$?

. INDEPENDENT SET es NP-completo

. Razonamiento basado en los triángulos contenidos en el grafo: para cada triángulo sólo uno de sus vértices podría estar en cada conjunto de vértices independientes.

. Podemos limitarnos a considerar grafos cuyos vértices se pueden clasificar a partir de la existencia de m triángulos separados que los contienen a todos.

— Reducimos 3-SAT a INDEPENDET SET restringido a dichos grafos

. Dada ϕ constituida por m cláusulas $C_1, \ldots, C_m$, construimos