Circuit SAT

Jardines Mendoza César Eduardo
Mendoza Castillo María Fernanda

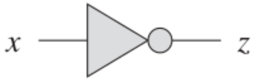
Universidad Nacional Autónoma de México

21 de Marzo del 2020

Circuit-SAT

Compuertas Lógicas

- Elementos de un circuito combinacional
- Entradas y salidas booleanas 0, 1
- El número de entradas es constante
- Compuertas básicas:
  NOT

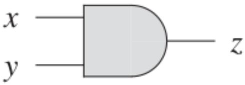


x	$\neg x$
0	1
1	0

Figura: Compuerta NOT

Compuertas Lógicas

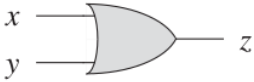
AND



x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Figura: Compuerta AND

OR



x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Circuito Combinacional

Un **circuito combinacional** es un conjunto de compuertas lógicas conectadas por cables.
No tiene ciclos.

Definiciones Auxiliares

- La **entrada de un circuito** es una entrada que no esta conectada a una salida.
- La **salida de un circuito** es la salida que no esta conectada a una entrada.
- La **asignación de verdad** es el conjunto de valores de entrada.

Circuito Cominacional

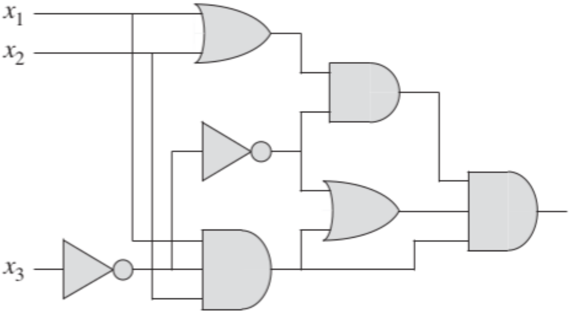


Figura: Circuito combinacional.

Definiciones Auxiliares

- Un **cable** conecta la salida de una compuerta a la entrada de otra(s).
- **fan-out** es el número de entradas a las que se conecto con un único cable.
- El **tamaño** de un circuito es el número de compuertas lógicas mas el número de cables que contiene.

Satisfacibilidad

Si una asignación de verdad causa que la salida sea 1, decimos que es **satisfacible**.

Circuit-SAT

Problema Circuit-SAT

"Dado un circuito combinacional con compuertas NOT, AND, OR, ¿es satisfacible?"

$$\text{Circuit-SAT} = \{< C > | C \text{ es un circuito combinacional satisfacible}\}$$

Problema Circuit-SAT

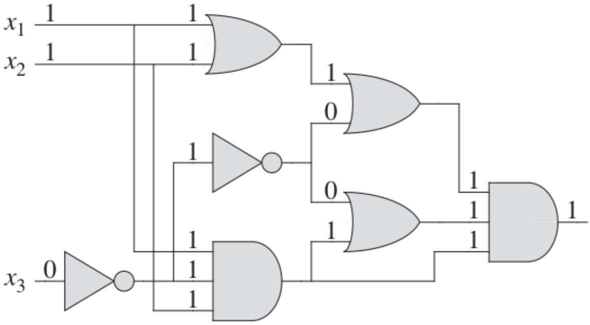


Figura: Circuito satisfacible.

Circuit-SAT

¿NP-Completo?

k entradas $\Rightarrow 2^k$ asignaciones posibles $\Rightarrow \Omega(2^k)$ en tiempo $\Rightarrow$ NP-completo

Definiciones Auxiliares

- Un **certificado** es el conjunto de asignaciones de los cables de un circuito.

¿Circuit-SAT $\in$ NP?

Lema. *El problema circuit-SAT pertenece a la clase NP.*

¿Circuit-SAT $\in$ NP?

Dem: Construimos un algoritmo polinomial A

$\rightarrow$ C es un circuito booleano combinacional.
$\rightarrow$ S un certificado de C.
$\leftarrow$ 1 si S satisface a C, 0 en otro caso.

A: Sea d una compuerta lógica de C con $x_1 ... x_n$ entradas y $z = d(x_1 ... x_n)\ \forall d \in C$

1. Verificamos que la salida del cable de d según S sea igual a z.
2. **if** la salida de C == 1
3. **then** return 1
4. **else** return 0

¿Circuit-SAT $\in$ NP?

Siendo S de longitud polinomial respecto al tamaño de C y A polinomial en tiempo(lineal).

$\Rightarrow$ Circuit-SAT $\in$ NP

$\square$

¿Es NP-Duro?

Lema. *El problema circuit-SAT es NP-duro.*

Definiciones Auxiliares

- El **program counter** indica la instrucción a ejecutar. Se le da el orden.
- Una **configuración** del programa A es algún estado particular de la memoria(programa, PC, máquina auxiliar de estados, cadena de entrada x, certificado y, memoria de trabajo).

¿Es NP-Duro?

Dem: Sea L cualquier lenguaje en NP.
Mostrar un algoritmo polinomial F que dada una cadena $x \in L$,
$C = f(x) \Leftrightarrow$ C $\in$ circuit-SAT
Sea $T(n)$ el peor caso de A verificando L, $T(n) = O(n^k)$ siendo polinomial por A y el tamaño de las entradas.

M es el circuito que muestra la secuencia de configuraciones $(c_0 \ldots c_{T(n)})$ de A.
Finalmente A devueve 0 o 1 en alguno de los bits de $c_{T(n)}$.

¿Es NP-Duro?

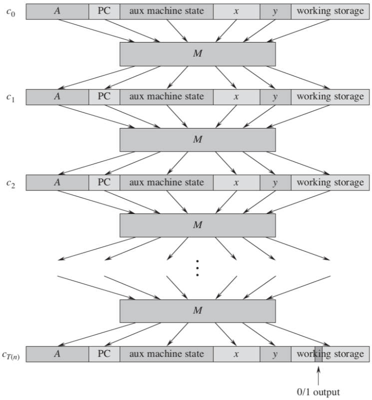


Figura: Configuraciones de A.

Circuit-SAT

El algoritmo de reducción F de un circuito se centra en construir una sola combinación de circuitos que compute todas las configuraciones producidas por una configuración inicial.

El algoritmo de reducción F de un circuito se centra en construir una sola combinación de circuitos que compute todas las configuraciones producidas por una configuración inicial.

- $T(n)$ lo denotaremos como el peor tiempo de ejecución del algoritmo en longitud n

El algoritmo de reducción F de un circuito se centra en construir una sola combinación de circuitos que compute todas las configuraciones producidas por una configuración inicial.

- $T(n)$ lo denotaremos como el peor tiempo de ejecución del algoritmo en longitud n
- M el circuito de combinaciones

El algoritmo de reducción F de un circuito se centra en construir una sola combinación de circuitos que compute todas las configuraciones producidas por una configuración inicial.

- $T(n)$ lo denotaremos como el peor tiempo de ejecución del algoritmo en longitud n
- M el circuito de combinaciones
- c_i i configuraciones

La idea es basicamente pegar las $T(n)$ copias del circuito M (las $C_{T(n)}$ generadas) y con la salida del i-esimo circuito que se produce en la configuración C_i se estará alimentando directamente de la entrada del circuito (i + 1) por lo que las configuraciones en lugar de almacenarse en la memoria de la computadora estas se consideraran como valores en los cables que conectan las copias de M.

¿Es Satisfacible?

Dada una entrada x se debe calcular un circuito $C = f(x)$ que sea satisfactoria si y solo si existe un certificado y tal que $A(x, y) = 1$.

Las entradas correspondientes al programa A, el contador inicial del programa, la entrada x y el estado inicial de la memoria directamente son valores conocidos.

Por lo que el circuito calcula $C(y) = A(x, y)$ para cualquier entrada y de longitud $O(n^k)$

dem: La reducción de nuestro algoritmo polinomial sea satisfacible.

$\rightarrow$ Supongamos que existe un certificado y de longitud $O(n^k)$ tal que $A(x, y) = 1$. Entonces, si aplicamos los bits de y a las entradas de C, la salida de C es $C(y) = A(x, y) = 1$. Por lo tanto, si existe un certificado, entonces C es satisfactoria.
$\leftarrow$ Supongamos que C es satisfactoria. Por lo tanto, existe una entrada y a C tal que $C(y) = 1$, de lo cual concluimos que $A(x, y) = 1$. Por lo tanto el algoritmo calcula correctamente una función de reducción y es satisfacible.

¿Es Polinomial?

Dem: Que nuestro algoritmo de reducción se ejecuta en tiempo polinomial $n = |x|$.

Puntos a considerar

Puntos a considerar:

- Sabemos que el número de bits requeridos para representar una configuración es polinomial en n

Puntos a considerar:

- Sabemos que el número de bits requeridos para representar una configuración es polinomial en n
- El programa para la secuencia de configuraciones tiene un tamaño constante e independiente de la longitud de su entrada x

Puntos a considerar:

- Sabemos que el número de bits requeridos para representar una configuración es polinomial en n
- El programa para la secuencia de configuraciones tiene un tamaño constante e independiente de la longitud de su entrada x
- La longitud de la entrada x es n, y la longitud del certificado y es $O(n^k)$

Suponemos que la memoria de circuit-SAT es contigua.
$\rightarrow$ El algoritmo se ejecutará en la mayoría de los pasos $O(n^k)$, la cantidad de almacenamiento de trabajo requerida por la secuencias de configuraciones también es polinómica en n. Entonces si el circuito combinacional M que implementa el hardware de la computadora tiene un tamaño polinómico en la longitud de una configuración, que es $O(n^k)$ entonces el tamaño de M es polinomial en n.

El circuito C consta de como máximo $t = O(n^k)$ copias de M, y por lo tanto tiene un tamaño polinomial en n. El algoritmo de reducción puede construir C a partir de x en tiempo polimial, ya que cada paso de la construcción lleva tiempo polinomial.

$\square$

Gracias por su atención :)