Logica

domingo, 13 de febrero de 2022 21:34

- Logica
  - sintaxis — F fórmula
  - semántica
    - I interpretación
    - $I \models F$ satisface una interpretación

	m	e	$(\neg m \rightarrow e) \wedge \neg e$
interpret. →	0	0	0
	0	1	0
	1	0	1 → satisface interpretación
	1	1	0

F sat ?

F insat ? ↗ preguntas al sat solver

F tautologia ? ssi $\neg F$ insat

* $F \models G$? ssi $F \wedge \neg G$ insat

→ $\forall I \models F \Rightarrow I \models F$

→ $F \models G$ significa que F es consequencia logica de G

* ⚠ La gent s'equivoca

$F \equiv G$ ssi $(F \wedge \neg G) \vee (G \wedge \neg F)$ insat

→ $F \models G \wedge G \models F$

Para el sat solver, queremos representar F como un conj de cláusulas

Demostración: Convencer al lector de lo que quiere demostrar es cierto. El nivel de detalle depende del lector. En nuestro caso, sera un PC ⇒ dummy

2-Lógica Proposicional

martes, 22 de febrero de 2022 8:22

Sintaxis

$P = \{p, q, r, \dots\}$

Operandos : $\wedge$, $\vee$, $\neg$

Semántica

$I : P \to \{0, 1\}$

Evaluaciones

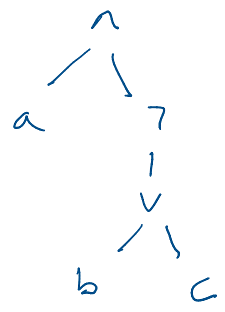
- $eval_I(p) = I(p)$ si $p \in P$
- $eval_I(F \wedge G) = \min(eval_I(F), eval_I(G))$
- $eval_I(F \vee G) = \max(eval_I(F), eval_I(G))$
- $eval_I(\neg F) = 1 - eval_I(F)$

Tipo de problema

$I \models F$? Tiene coste lineal. Se puede ver F como un árbol

ejemplo :

$a \wedge \neg(b \vee c)$



Cap 2 ej 2

martes, 22 de febrero de 2022 8:40

2| Demuestra que $P \wedge \neg P$ es insat usando SOLO la definición de la lógica

↳ ⚠ NO se pueden usar las leyes de De Morgan!

* Por el momento, estamos SOLO aplicando definiciones, sin saber mucho a donde ir

$P \wedge \neg P$ insat $\iff$ por def de insat

$P \wedge \neg P$ no tiene modelos $\iff$ por def de modelo

$\forall I,\ I \not\models P \wedge \neg P$ $\iff$ por def de satisfacción ($\models$)

$\forall I,\ eval_I(P \wedge \neg P) = 0$ $\iff$ por def de $eval_I(\ldots \wedge \ldots)$

$\forall I,\ \min(eval_I(P), eval_I(\neg P)) = 0$ $\iff$ por def de $eval_I(\neg \ldots)$

$\forall I,\ \min(eval(P), 1 - eval(P)) = 0$ $\iff$ por def de $eval_I(P)$ si $p \in \mathcal{P}$

$\forall I,\ \min(I(P), 1 - I(P)) = 0$ $\iff$ porque $I(P)$ siempre es 0 o 1

↳ análisis por casos (se podría hacer explícito)

true

$\square$

Cap 2 ej 6

martes, 22 de febrero de 2022 8:57

Demuestra F Tautología $\Leftrightarrow$ $\neg F$ insat usando SOLO las definiciones de la lógica proposicional

Dos formas de proceder:

- cadena $\Leftrightarrow$
- cadena $\Rightarrow$ n cadena $\Leftarrow$

F taut $\Leftrightarrow$ por def de taut

$\forall I,\ I \models F$ $\Leftrightarrow$ por def de $I \models F$

$\forall I,\ eval_I(F) = 1$ $\Leftrightarrow$ por aritmética

eval. en I de F

* Si no sabes como continuar, ir desde el final

$\forall I,\ 1 - eval_I(F) = 0$ $\Leftrightarrow$ por def de $eval_I(\neg \ldots)$

$\forall I,\ eval_I(\neg F) = 0$ $\Leftrightarrow$ por def de $I \not\models F$

$\forall I,\ I \not\models \neg F$ $\Leftrightarrow$ por def de insat

$\neg F$ insat

- orden en que se ha escrito

Pregunta extra:

martes, 22 de febrero de 2022 9:09

¿Es cierto que, si F y G son formulas, que $F \wedge \neg G$ es una tautología?

Usa solo las def de la log. proposicional

Para solucionarlo, se tiene que dar un contraejemplo.

NO. Contraejemplo:

Sea $F = p$
$G = p$

Entonces $p \wedge \neg p$ no es taut:

Dato $F \wedge G$, debo dar I tq $I \not\models F$

sea I tq $I(p) = 0$

Entonces $I \not\models p \wedge \neg p$

⋮

Cap 2 ej 5

martes, 22 de febrero de 2022 9:15

Sean F y G fórmulas cualquiera, es cierto que $F \vee G$ es taut $\Leftrightarrow$

$\Leftrightarrow$ alguna de las dos lo es ?

$F \vee G$ Taut ssi F es Taut o G es Taut ?

↳ NO poner conectivos lógicos en frases !

NO es cierto. Contraejemplo:

F es p : no es tautología, pq si $I(p) = 0$, Tenemos $I \not\models p$

G es $\neg p$: no es taut, pq si $I(p) = 1$, Tenemos $I \not\models \neg p$

pero $p \vee \neg p$ es taut, ... demostrar !

| estamos demostrando que $\not\Leftarrow$

Ejercicios extra:

martes, 22 de febrero de 2022 9:22

7, 8, 16

TO DO: 18

Cap 2 ej 21

martes, 22 de febrero de 2022 9:24

Comprueba que $\equiv$ es realmente una relación de equivalencia

Dadas 3 fórmulas F, G, H

- $F \equiv G \Rightarrow F$ y G comparten los mismos modelos $\Rightarrow G$ y F comparten los mismos modelos $\Rightarrow G \equiv F$ (simétrica)
- $F \equiv F \Rightarrow F$ comparte los mismos modelos consigo misma. (reflexiva)
-

Relaciones

martes, 22 de febrero de 2022 9:25

- Una relación binaria R sobre un conjunto S es un subconjunto del producto cartesiano

$$R \subseteq S \times S$$

Ejemplo:

$S = \{a, b, c\}$

R puede ser $\{(a,a), (a,c), (c,b)\}$

- Una relación es de equivalencia si R es:
  - Reflexiva: Si $(e,e) \in R, \forall e \in S$
  - Simétrica: Si $\forall e, e' \in S$ $(e,e') \in R \rightarrow (e',e) \in R$
  - Transitiva: Si $\forall e, e', e'' \in S$

    $(e,e') \in R$ y $(e',e'') \in R \rightarrow (e,e'') \in R$

- Cuando una relación R es de equivalencia, S se "divide" en clases de equivalencia.

Cap 2 ej 18 – TO DO

martes, 22 de febrero de 2022 11:21

Lema de substitución

martes, 1 de marzo de 2022 8:12

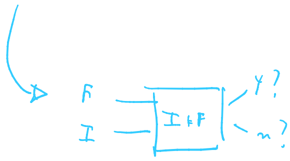
Cap 2 ej 23

Dem por inducción.

Cap 2 ej 26

martes, 1 de marzo de 2022 8:13

Supongamos que $|P| = 100$, y que evaluar una F en I cuesta $1 \mu s$ (10^{-6} s)



Supon que haces el alg. SAT tonto. cuanto tarda?

$2^{100} \simeq 10^{30}$ opciones

$$10^{30} \text{ op} \cdot 10^{-6} \text{ s/op} = 10^{24} \text{ s} \cdot \frac{1 \text{ año}}{(365 \cdot 24 \cdot 3600) \text{ s}} = 3,\ldots \cdot 10^{16} \text{ años}$$

Cap 2 ej 27

martes, 1 de marzo de 2022 8:18

Función booleana de n entradas, dados n bits, devuelve 1

$$f : \{0,1\}^n \longmapsto \{0,1\}$$

Cuantas funciones de n entradas existen?

$$2^{2^n} = 2^{(2^n)}$$

27. (dificultad 2) Una *función booleana* de n entradas es una función $f : \{0, 1\}^n \to \{0, 1\}$, es decir, una función que toma como entrada una cadena de n bits y devuelve un bit. ¿Cuántas funciones booleanas de n entradas hay?

ejemplo:

x	y	0	$\wedge$	$x \not\Rightarrow y$	x	$y \not\Rightarrow x$	y	XOR	$\vee$	nor	$=$	$\bar{y}$	$y \Rightarrow x$	$\bar{x}$	$x \Rightarrow y$	nand	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Extra:

Dadas las ∞ funciones posibles para n entradas, $\exists\ 2^{2^n}$ clases de eq.

Se pueden representar en forma de fórmula, circuitos, ...

Uso de NAND

martes, 1 de marzo de 2022 8:32

En la logica prop, usamos $\vee, \wedge, \neg$. Pero cualquier fórmula lógica se puede representar con una NAND. ¿Puedes demostrarlo?

Para demostrarlo, se tiene que dem. que $\wedge, \vee, \neg$ se pueden expresar con nand.

- $\neg F \equiv F \text{ nand } F$
- $F \wedge G \equiv \neg(F \text{ nand } G) \equiv (F \text{ nand } G) \text{ nand } (F \text{ nand } G)$
- $F \vee G \equiv \neg(\neg F \wedge \neg G) \equiv \neg F \text{ nand } \neg G \equiv$

  $\equiv (F \text{ nand } F) \text{ nand } (G \text{ nand } G)$

Se ha escogido la representación con 3 símbolos por ser más claros, y para poder usarse.

Cap 2 ej 37 - Simplificar programa

martes, 1 de marzo de 2022 8:41

37. (dificultad 3) Considera el siguiente fragmento de código, que devuelve un booleano:

```
int i;
bool a, b;
...
(1) if (a and i>0)
      return b;
(2) else if (a and i<=0)
      return false;
(3) else if (a or b)
      return a;
(4) else
      return (i>0);
```

Simplifícalo sustituyendo los valores de retorno por un solo valor de retorno que sea una expresión booleana en $i > 0$, a y b:

```
int i;
bool a, b;
return ...;
```

El codigo no es una expr. booleana pura, pq se evalua con un orden.

Por ejemplo, la AND no es commutativa

$\exists$ un int, pero al estar en comparaciones, se usa como bool

Hacemos tabla y vamos rellenando

a	b	i>0		
0	0	0	0	(4)
0	0	1	1	(4)
0	1	0	0	(3)
0	1	1	0	(3)
1	0	0	0	(2)
1	0	1	0	(1)
1	1	0	0	(2)
1	1	1	1	(1)

El programa devuelve ↑

return (a == b and i>0)

Ej 34. Inventa otra lógica

martes, 1 de marzo de 2022 8:57

34. (dificultad 2) Inventa y define formalmente alguna otra lógica distinta a la lógica proposicional. Por ejemplo, si las interpretaciones son funciones $I: \mathcal{P} \to \{0, 1, \perp\}$ que también pueden dar "indefinido" $\perp$, se puede adaptar la noción de satisfacción de manera razonable, aunque la respuesta ya no será binaria: la "evaluación" de una fórmula F en una interpretación I puede dar 1 (I satisface F) o 0 (I no satisface F) o $\perp$ (indefinido).

$$I : P \longmapsto \{0, 1, \perp\}$$

$$eval_I(p) = I(p) \quad \text{si } p \in P$$

$$eval_I(F \wedge G) = ?$$

x	y	¬x	x ∧ y	x ∨ y	
0	0		0		
0	1		0		
0	⊥		0		
1	0		0		
1	1		1		
1	⊥		⊥		
⊥	0		0		
⊥	1		⊥		
⊥	⊥		⊥		

← manera arbitraria, definida por nosotros.

manera arbitraria, definida por nosotros.
Depende de la aplicación que tengamos en mente, se puede definir de una forma u otra.

Ej 35 Inventa otra lógica

martes, 1 de marzo de 2022 9:05

35. (dificultad 2) Como el ejercicio anterior, pero considerando $I:\mathcal{P} \rightarrow [0 \ldots 1]$, es decir, la interpretación de un símbolo p es una probabilidad (un número real entre 0 y 1). En este caso, la evaluación de una fórmula F en una interpretación I puede dar algo (remotamente) parecido a la probabilidad de satisfacción de F en I. En la lógica que has definido, ¿la evaluación de F en una I determinada, y la de $F \wedge F$ en esa misma I dan el mismo resultado?

$$I : \mathcal{P} \longmapsto [0 \ldots 1] \in \mathbb{R}$$

↳ probabilidad ≡ lógica difusa

$$eval_I(p) = I(p) \quad \text{si } p \in \mathcal{P}$$

$$eval_I(\neg F) = 1 - eval_I(F)$$

$$eval_I(F \wedge G) = eval_I(F) \cdot eval_I(G)$$

↳ ⚠ es nuestra lógica, así que supondremos que los eventos son independientes

$$eval_I(F \vee G) = eval_I(F) + eval_I(G) - eval_I(F \wedge G)$$

Sudokus con SAT

martes, 8 de marzo de 2022 8:10

				7				
	8							
				5				
					4			
		3						

Variables

$X_{ijk} \equiv$ "la casilla (i, j) tiene el valor k"

para $i,j,k \in [1,9]$

$\Downarrow$

9^3 vars $= 729$

Cláusulas

Qué es un sudoku:

a) En cada casilla (i,j) hay exactamente 1 valor k:

- al menos 1 valor At Least One (ALO):

  C: $\forall i,j$, 1 cláusula $X_{ij1} \vee X_{ij2} \vee \dots \vee X_{ij9}$

- como mucho 1 valor At Most One (AMO:

  C: $\forall i,j$ las cláusulas $\overline{X_{ijk}} \vee \overline{X_{ijk'}}$ para $1 \le k < k' \le 9$

  $\overline{X_{ij1}} \vee \overline{X_{ij2}}$

$$\overline{X_{ij1}} \vee \overline{X_{ij2}}$$
$$\overline{X_{ij1}} \vee \overline{X_{ij3}}$$
$$\vdots$$
$$\overline{X_{ij8}} \vee \overline{X_{ij9}}$$

nb clausulas $\rightarrow \binom{9}{2} = \frac{9 \cdot 8}{2} = 36$ por casilla

b) En cada fila i, cada valor k 1 vez (en 1 j)

por cada (i,k) exactamente 1 j

ALO $X_{i1k} \vee \dots \vee X_{i9k}$

AMO $\overline{X_{i1k}} \vee \overline{X_{i2k}}$
$$\vdots$$
$$\overline{X_{i8k}} \vee \overline{X_{i9k}}$$

c) Ídem por cada j,k exactly 1

d) ídem para bloque

Total clausulas para definir sudoku general:

$4 \cdot 81 \cdot 36$

e) Rellenar el problema, añadiendo cláusulas unarias de las casillas ocupadas.

En ejemplo :

Un ejemplo:

$$X_{228}$$

$$X_{157}$$

$$X_{355}$$

Cláusulas (repaso)

martes, 8 de marzo de 2022 8:30

$$\underbrace{p_1 \vee \dots \vee p_n}_{\text{lit } +} \vee \underbrace{\neg q_1 \vee \dots \vee \neg q_m}_{\text{lit } -}$$

- $m = 0$ cláusula positiva
- $n = 0$ " negativa
- $n \leq 1$ cláusulas de Horn $\rightarrow \theta$ polinom
- $n = m = 0$ cláusula vacía [ ]
- $n + m \leq 2$ 2-SAT $\rightarrow \theta$ pol.
- $n + m \leq 3$ 3-SAT $\rightarrow \theta$ exp

3 – Deducción en lógica proposicional

martes, 8 de marzo de 2022 8:45

Cláusula vacía:

- de $\vee$ → siempre FALSO
- de $\wedge$ → siempre CIERTO

$$eval_I(l_1 \vee \ldots \vee l_n) = max(0, eval_I(l_1), \ldots, eval_I(l_n))$$

$$eval_I(c_1 \vee \ldots \vee c_n) = min(1, \ldots\ldots)$$

Ex 5

martes, 8 de marzo de 2022 8:47

5. (dificultad 2) Demuestra que una cláusula C es una tautología si, y sólo si, contiene a la vez p y $\neg p$ para un cierto símbolo proposicional p.

$\Leftarrow$ sea $C = p \vee \neg p \vee C'$

sea I cualquier interpretación

Tenemos que ver que $I \models C$

$I \models C$ ssi $eval_I(C) = 1$ ssi

ssi $max(0, eval_I(p), eval_I(\neg p), \ldots) = 1$

$\Rightarrow$ Por reducción al absurdo.

Supón que no es tautología, Entonces

$\exists$ una I que pone a falso p y $\neg p$ $\Rightarrow$ absurd

Demostración alternativa usando $\Leftrightarrow$

Demostraremos $C = p_1 \vee \ldots \vee p_m \vee \neg q_1 \vee \ldots \vee \neg q_n$ no taut. $\Leftrightarrow$

$\Leftrightarrow$ $p_i \neq q_j$ $\forall$ $i \in 1 \ldots m$, $j \in 1 \ldots n$ $\Leftrightarrow$

$\Leftrightarrow$ $\exists I$ tq $I \not\models p_1 \vee \ldots \vee p_m \vee \neg q_1 \vee \ldots \vee \neg q_n$ $\Leftrightarrow$

$\Leftrightarrow$ $\exists I$ tq $max(0, eval_I(p_1), \ldots, eval_I(p_m), eval_I(\neg q_1), \ldots, eval_I(\neg q_n)) = 0$ $\Leftrightarrow$

$1 - eval_I(q_1) \ldots 1 - eval_I(q_1)$

$\Leftrightarrow \ldots$

Idem en archivo

6

martes, 8 de marzo de 2022 9:05

6. (dificultad 2) Sea S un conjunto de cláusulas con $\square \notin S$. Demuestra que S es satisfactible (dando un modelo para S) en cada una de las siguientes situaciones:

   a) Toda cláusula de S tiene algún literal positivo.

   Sí, es satisfactible. La I tq $I(p)=1$ $\forall p \in P$ es modelo!

   b) Toda cláusula de S tiene algún literal negativo.

   Sí, es satisfactible. La I tq $I(p)=0$ $\forall p \in P$ es modelo!

   c) Para todo símbolo de predicado p se cumple que: o bien p aparece sólo en literales positivos en S, o bien p aparece sólo en literales negativos en S.

   Sí, es sat. La I que pone a 1 todos los lit + y a 0 todos los lit − es modelo.

Ex 7

martes, 8 de marzo de 2022 9:12

7. (dificultad 2) Dados n símbolos proposicionales: $|P| = n$

a) ¿Cuántas cláusulas distintas (como conjuntos de literales) hay?

Hay $2n$ literales (cada símbolo en + o −)

Este cjt de $2n$ lits tiene 2^{2n} subconjuntos distintos.

$$2^{2n} = 2^n 2^n = 4^n$$

4^n pq a cada símbolo p le pueden pasar 4 cosas en una cláusula:

a) p solo sale + en C
b) " " " − en C
c) p no sale en C
d) p sale en + y − en C

b) ¿Cuántas de estas cláusulas son insatisfactibles?

1 La única cláusula que es insat es $[\,]$

c) ¿Cuántas cláusulas distintas y que no son tautologías hay?

3^n

Por cada símbolo ahora pueden pasar 3 cosas (la d) no)

d) ¿Cuántas cláusulas distintas que contienen exactamente un literal por cada símbolo proposicional hay?

≡ con exactamente 1 lit por cada $p \in P$?

2^n

Ahora solo pueden ser los casos a) y b)

Horn SAT es polinómico (lineal!)

martes, 8 de marzo de 2022 9:25

Decidir si un cjt S de cláusulas de Horn es satisfactible.

Cada cláusula de Horn C puede ser de 4 tipos:

a) $\square$ (cláusula vacía)

b) p (1 lit +)

c) $p \vee \bar{q}_1 \vee \dots \vee \bar{q}_m$ (1 + y $m > 1$ neg)

d) $\bar{q}_1 \vee \dots \vee \bar{q}_m$ ($m \geqslant 1$ neg)

Si $\square \in S \Rightarrow S$ es insat.

Suponemos que $\square \notin S$.

Si no hay de tipo a) ni de tipo b) $\Rightarrow$ S es sat (modelo todo 0)

" " " " " a) " " " d) $\Rightarrow$ S es sat (modelo todo 1)

Para tratar el caso que tenga b), c), d):

- Hago positive unit propagation.
- Si conflicto: insat!
- Si acaba sin conflicto, hay modelo I:

$I(p) = 1 \quad \forall p$ unit propagation

$I(p) = 0$ para los demás

Las de tipo b) y d) se cumplen trivialmente. Las de tipo

Las de tipo b) y d) se cumplen trivialmente. Las de tipo c), dependerá de la propagación: o se cumple pq ha sido propagado o pq quedan en neg.

Algoritmo de resolución:

Hago un contador de negs por cada cláusula. Al propagar, decremento el contador. Si llega a 0:

- Propagar si es de tipo c)
- Insat si tipo d)

Tiempo: $\Theta(n)$

Propiedad: Las cláusulas de Horn $\exists!$ modelo minimal (min nb de cláusulas a 1)

El algoritmo construye el modelo minimal.

Miras

martes, 8 de marzo de 2022 9:52

er 12, ...

Ex 12: DNF

martes, 15 de marzo de 2022 8:10

Decidir SAT para una DNF, ¿Que costo tiene?

DNF: Disjuntive Normal Form

$F = \text{Cubo } 1 \vee \dots \vee \text{Cubo } n$

1 cubo : $P_1 \wedge \dots \wedge P_n \wedge \neg q_1 \wedge \dots \wedge \neg q_m$

Observaciones:

a) F es satisfactible ssi algun cubo i lo es

b) un cubo $P_1 \wedge \dots \wedge \neg q_1 \wedge \dots \neg q_m$ es sat ssi $\nexists i, j$ tq $P_i = q_j$

	CNF	DNF
SAT	NP-Completo	lineal
TAUT	lineal	NP-comp

CNF.SAT ≡ DNF.TAUT

Recordatorio:

NP-Completitud: Problemas de decisión (sol = si/no)

↳ Un problema p es NP-completo si

a) $p \in NP$ → tienen testigos que se pueden verificar en tiempo pol.

b) p es NP-hard → cualquier problema $p' \in NP$ lo podemos resolver mediante $p \equiv$ reducir a p

Reducir coloreado de grafos a SAT

Reducir coloreado de grafos a SAT

3 colores

n nodos

m aristas

$X_{i,j}$ = "el nodo i tiene el color j"

Para reducirlo: buscar ALO, AMO, repeticiones.

Resolución

martes, 15 de marzo de 2022 8:49

Es una regla deductiva para Lógica Proposicional

regla de inferencia

$$(p \vee C) \wedge (\neg p \vee D) \models C \vee D \ ? \qquad \text{SI}$$

Sea I tq $I \models (*)$

a) Si $I(p) = 0 \Rightarrow I \models C$ y $I \models C \vee D$

b) Si $I(p) = 1 \Rightarrow I \models D$ y "

Th

S es cjt de cláusulas.

S insat ssi $\square \in \text{Res}(S)$

la clausura bajo resolución de S

Ejemplo

$$S = \{ p \vee q, \ \neg p \vee q, \ p \vee \neg q, \ \neg p \vee \neg q \}$$

$$\dfrac{\dfrac{p \vee q \qquad \neg p \vee q}{q} \qquad \dfrac{p \vee \neg q \qquad \neg p \vee \neg q}{\neg q}}{\square}$$

Definición de Resolución

$Res(S)$

$$\begin{cases} S_0 = S \\ S_{i+1} = S_i \cup \left\{ C \vee D \;\middle|\; \begin{array}{l} p \vee C \in S_i \\ \neg p \vee D \in S_i \end{array} \text{ y } \right\} \\ Res(S) = \bigcup_{i=0}^{\infty} S_i \end{cases}$$

Como $\exists$ nº finito de cláusulas (4^n) distintos, $Res(S)$ acaba.

2-SAT $\in$ Polinomico bajo Res.

$Res(S) \equiv S \rightarrow$ dem por inducción.

Transformación de Tseitin

martes, 15 de marzo de 2022 9:27

Para transformar a forma clausal (CNF) sin explotar (tamaño exp)

$F \Rightarrow T(F)$

Propiedades

- F sat ssi $T(F)$ es sat
- $T(F)$ es CNF (de hecho, es 3-SAT)
- $|T(F)|$ es lineal en $|F|$
- Puedo hacer la transformación en tiempo lineal.

Ejemplo:

Dada una formula F, se hace el árbol y se añaden auxiliares a los $\vee$ y $\wedge$.

En total, por cada $\wedge$ y $\vee$ hay 3 clausulas + 1 clausula por la raíz (en positivo)

Cardinality constrains

martes, 22 de marzo de 2022 8:34

$l_1 + \dots + l_n \geq k$ at least

$l_1 + \dots + l_n \leq k$ at most $\leftarrow$ all subsets of size $k+1$, at least 1 false: $\overline{l_1} \vee \dots \vee \overline{l_{k+1}}$ $\vdots$ $\rightarrow \binom{n}{k+1}$

$= k$

En la web de la asig hay 2 videos:

- Transportation company → Prob con cardinality consts y Pseudo-Boolean consts.
- Codif. de constrains numéricos — sol → ej: $3x + 7\overline{y} + 4z + 6t \geq 5$

Explican cómo generar menos cláusulas.

At Most 1

encoder	# clausulas	# vars aux
cuadrática	$\frac{n \cdot (n-1)}{2}$	0
ladder	$3n$	n
Heule-3	$3n$	$n/2$
Heule-4	$10n/3$	$n/3$
logarítmica	$n \log n$	$\log n$

Ladder encoding

$$x_1 \quad \overline{x}_2 \ \cdots \ x_{i-1} \quad x_i \quad x_{i+1} \ \cdots \ x_{n-1} \quad x_n$$

$$a_1 \to a_2 \ \cdots \ a_{i-1} \to a_i \to a_{i+1} \ \cdots \ a_{n-1} \quad a_n$$

$$a_i = \text{"}\sum_{k=1}^{i} x_i \leq 1\text{"} = \text{"}x_1 + \ldots + x_i \leq 1\text{"}$$

$\overline{x_i} \vee a_i$ $\qquad$ $x_i \to a_i$

$\overline{a_i} \vee a_{i+1}$ $\qquad$ $a_1 \to a_2$

$\overline{a_i} \vee \overline{x_{i+1}}$ $\qquad$ $a_i \to \overline{x_{i+1}}$ → para asegurar que sólo 1 es cierto.

Heule encoding

$x_1 + \ldots + x_n \leq 1$

iff

$x_1 + x_2 + x_3 + a \leq 1$ → hacer con codif cuadrática

$\begin{cases} \overline{x}_1 \vee \overline{x}_2 \\ \vdots \\ \overline{x}_3 \vee \overline{a} \end{cases}$ 6 clausulas

AND

$\overline{a} + x_4 + \ldots + x_n \leq 1$ → heule recursivamente

Para quitar 2 necesito 1 clausula ant

Logarítmica

$x_0 + x_1 + \ldots + x_7 \leq 1$

	$\underline{5}$
a_1	1
a_2	0
a_4	1

$\overline{x_5} \vee a_1$ $\qquad$ $\overline{x_6} \vee \overline{a_1}$

$\overline{x_5} \vee \overline{a_2}$ $\qquad$ $\overline{x_6} \vee a_2$

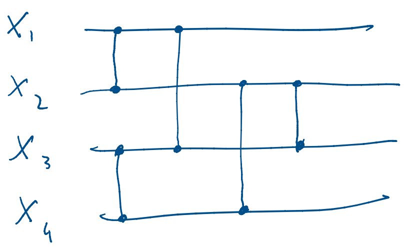
$\overline{x_5} \vee a_4$ $\qquad$ $\overline{x_6} \vee a_4$

$a_4 \quad 1 \qquad x_5 \vee a_4 \qquad x_6 \vee a_4$

$5_{10)} \to 101_{2)}$

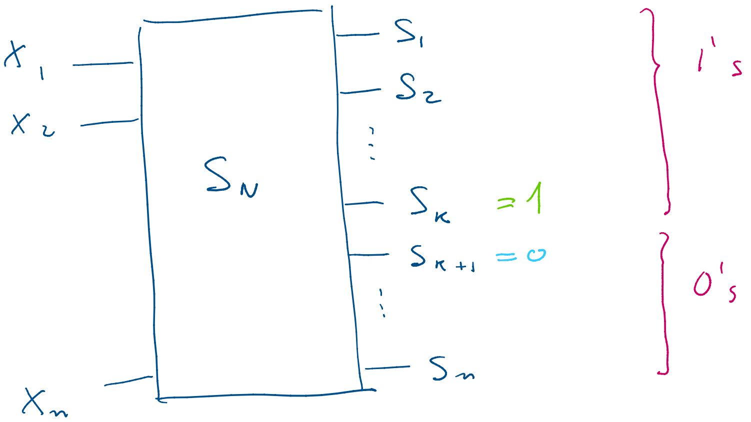
At Most / Least k

Se usan Sorting Networks

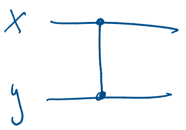


$x_1 + \ldots + x_n \geq k$

$x_1 + \ldots + x_n \leq k$



Falta expresar clausulas



$a = \max(x, y) \qquad a \Leftrightarrow x \vee y \longrightarrow$ 3 clausulas de T_s.

$b = \min(x, y) \qquad b \Leftrightarrow x \wedge y \longrightarrow$ " " " "

Cada comparador: 6 clausulas, 2 vars nuevas.

Los sorting Nets. necesitan $O(n \log^2 n)$ clausulas y vars

$n = \#$ comparadores

Cardinality Networks $\to O(n \log^2 k)$

va bién si $n \gg k$

Pseudo-Boolean Consts.

martes, 22 de marzo de 2022 9:33

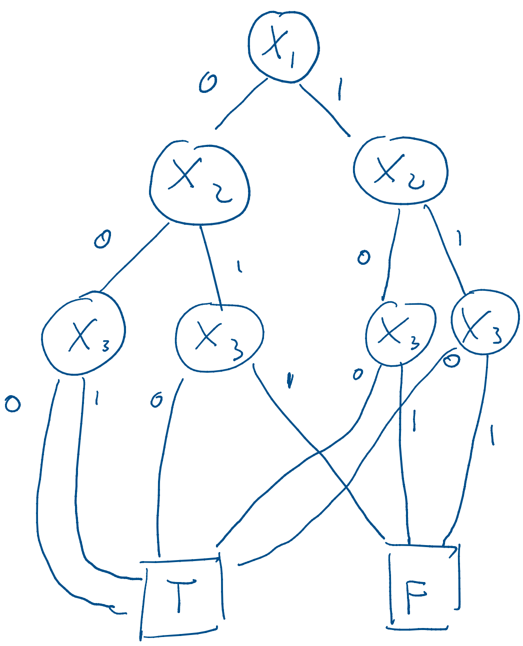
$$a_1 x_1 + \dots + a_n x_n \geq k$$
$$\leq k$$
$$= k$$

↳ 0-1 Int Linear Constrains

Lo codificamos con un **ROBDD** (es como un autómata / árbol de decisión)

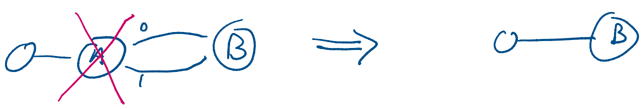
- R → Reduced
- O → Order
- B → Binary
- D → Decision
- D → Diagram

ej: $2x_1 + 3x_2 + 5x_3 \leq 6$

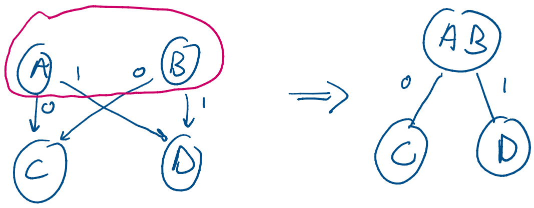


→ BDD NO reducido.
∃ 2 reduc posibl.

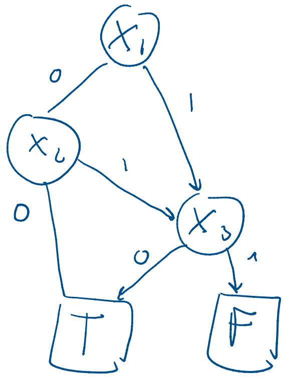
Reducción 1: eliminas nodos si las 2 opciones van al mismo sitio



Reducción 2: Si 2 nodos hacen lo mismo, se fusionan



ROBDD



2-SAT lineal

martes, 29 de marzo de 2022 8:10

Se puede hacer un grafo de $\Rightarrow$.

Si en S hay $p \to \dots \to \neg p$ then $S \models \bar{p}$
$\neg p \to \dots \to p$ then $S \models p$

$S \models \square$

$\Downarrow$

S Insat

cjto de cláusulas

Th

S insat ssi $\exists$ ciclo que contiene p y $\bar{p}$ para algún $p \in \mathcal{P}$.

Algoritmo:

dado S (cjto de 2-SAT)

monta el grafo $G(S)$ y comprueba si $\exists$ ciclo en $G(S)$ con p y $\bar{p}$

$$\text{si } l \vee l' \in S \Rightarrow \begin{cases} \neg l \to l' \\ \neg l' \to l \end{cases} \in G(S)$$

4 - Lógica de primer orden

martes, 29 de marzo de 2022 8:23

Sintaxis y Semantica

martes, 29 de marzo de 2022 8:23

Sintaxis

$\mathcal{X} = \{x, y, z, \ldots\}$ → dereche Tipo

$\mathcal{F} = \{f^2, g^1, h^1, \ldots a^0, \ldots\}$

términos: $x, y, f(x,y)$

átomos

$\mathcal{P} = \{r^2, \ldots\}$ → dereche bool

átomos → NO anidados. Solo preds que contienen términos.

Formulas: quantif + atomos + & y V

⚠ Priodidad de los cuantificadores!

$\forall$ y $\exists$ > $\wedge, \vee$

$$\forall x \; p(x) \wedge q(x) \;\begin{cases} = \; (\forall x \; p(x)) \wedge q(x) \\ \neq \; \forall x \; (p(x) \wedge q(x)) \end{cases}$$

Semántica

Una I tiene 3 cosas:

- D_I: el dominio de I (un cjt no-vacío)

- $\forall f^n \in \mathcal{F}$, una función $f_I : D_I^n \to D_I$

— $\forall f \in \mathcal{F}$, una función $f_I : D_I \longrightarrow D_I$

función $\nearrow$ ↑ interpretación en I de f ← del símbolo de función f

— $\forall p^n \in \mathcal{P}$, un pred $p_I : D_I^n \rightarrow \{0,1\}$

↳ es en realidad una relación binaria en D_I

Ejemplo

• $F : \forall x \exists y \left(p(x, g(y)) \wedge \neg q(x) \right)$

← definimos I para practicar

- $D_I = \mathbb{R}$
- $f_I(x,y) = x+y$ ⚠ Ponerlos
- $g_I(x) = x^2$
- $h_I(x) = 1-x$
- $a_I = 1$ → $1 \in \mathbb{R} \equiv 1_{\mathbb{R}}$
- $p_I(x,y) = x+y > 0$
- $q_I(x) = x^2 > 1$
- $r_I = 1$ → $1 \in \{0,1\} = 1 \in$ bool

$I \vDash F$? NO

por q, $x^2 \not> 1 \Rightarrow x \in [-1,1] \Rightarrow$

$\Rightarrow \exists x$ tq $I \nvDash F \Rightarrow \cancel{\forall} x \; I \vDash F$

Otro ejemplo de I.

- $D_I = \{\square, \triangle\}$
- $f_I(\triangle, \triangle) = \triangle$
  - $f_I(\triangle, \square) = \triangle$
  - $f_I(\square, \triangle) = \square$
  - $f_I(\square, \square) = \square$
- $g_I(\square) = \triangle$
  - $g_I(\triangle) = \square$
- $h_I(\square) = \square$
  - $h_I(\triangle) = \triangle$

⋮

$f_I \quad (\square, \square) \to \square$

$f_I \quad (\square, \square) \to \square$

$\vdots$

Ex 5

martes, 29 de marzo de 2022 9:21

5. (dificultad 1) Sea F la fórmula $\exists x \exists y \exists z \ (p(x, y) \wedge p(z, y) \wedge p(x, z) \wedge \neg p(z, x))$.
Cuáles de las siguientes interpretaciones son modelos de F?

a) $D_I = \mathbb{N}$ y $p_I(m, n) = 1$ si y sólo si $m \leq n$. SI

$x \leq y$, $z \leq y$, $x \leq z$, $z \nleq x$

Para $x = 2$, $y = 2$, $z = 1$ se cumple, pq es $\exists_x \exists_y \exists_z$

b) $D_I = \mathbb{N}$ y $p_I(m, n) = 1$ si y sólo si $n = m + 1$. NO

$\left.\begin{array}{l} y = x+1 \\ y = z+1 \end{array}\right\} x = z$ $\qquad z = x+1$

incompatibles

c) $D_I = \mathcal{P}(\mathbb{N})$ (esto denota *partes de* $\mathbb{N}$, es decir, el conjunto de todos los subconjuntos de $\mathbb{N}$), y $p_I(A, B) = 1$ si y sólo si $A \subseteq B$. SI

$P(\mathbb{N})$ a veces se escribe como $2^{\mathbb{N}}$

$x \subseteq y$ $\quad x \subseteq z$ $\qquad y = \{1, 2, 3\}$

$z \subseteq y$ $\quad z \nsubseteq x$ $\qquad x = \{1\}$

$z = \{1, 2\}$

Ex 6

martes, 29 de marzo de 2022 9:33

6. (dificultad 2) Expresa con tres fórmulas las propiedades de reflexividad, simetría y transitividad de un predicado binario p y demuestra que ninguna de las tres fórmulas es consecuencia lógica de (la conjunción de) las otras dos.

$p^2 \in P$

$F_R : \forall x \quad p(x,x)$

$F_S : \forall x \, \forall y \, \big(p(x,y) \rightarrow p(y,x) \big)$

$F_T : \forall x \, \forall y \, \forall z \, \big((p(x,y) \wedge p(y,z)) \rightarrow p(x,z) \big)$

<u>Demostraciones:</u>

a) $F_S \wedge F_T \not\models F_R$

$D_I = \{a\}$

$P_I(a,a) = 0$

$I \models F_S \qquad \neg p(a,a) \vee p(a,a)$

$I \models F_T \qquad$ similar

$I \not\models F_R$

b) $F_R \wedge F_T \not\models F_S$

$D_I = \{a, b\}$

$D_I = \{a, b\}$

$P_I(a, a) = 1$ → cumple F_R

$a, b = 1$ } no cumple F_S

$b, a = 0$

$b, b = 1$ → cumple F_R

Si se hacen los 8 casos, se ve que $I \models F_T$

c) $F_R \wedge F_S \not\models F_T$

$D_I = \{a, b, c\}$

P_I	$I \models F_R$	$I \not\models F_T$	$I \models F_S$
$(a, a) =$	1		
$a\ b =$		1	
$a\ c =$		0	
$b\ a =$			1
$b\ b =$	1		
$b\ c =$		1	
$c\ a =$			0
$c\ b =$			1
$c\ c =$	1		

Lab

miércoles, 23 de febrero de 2022 8:38

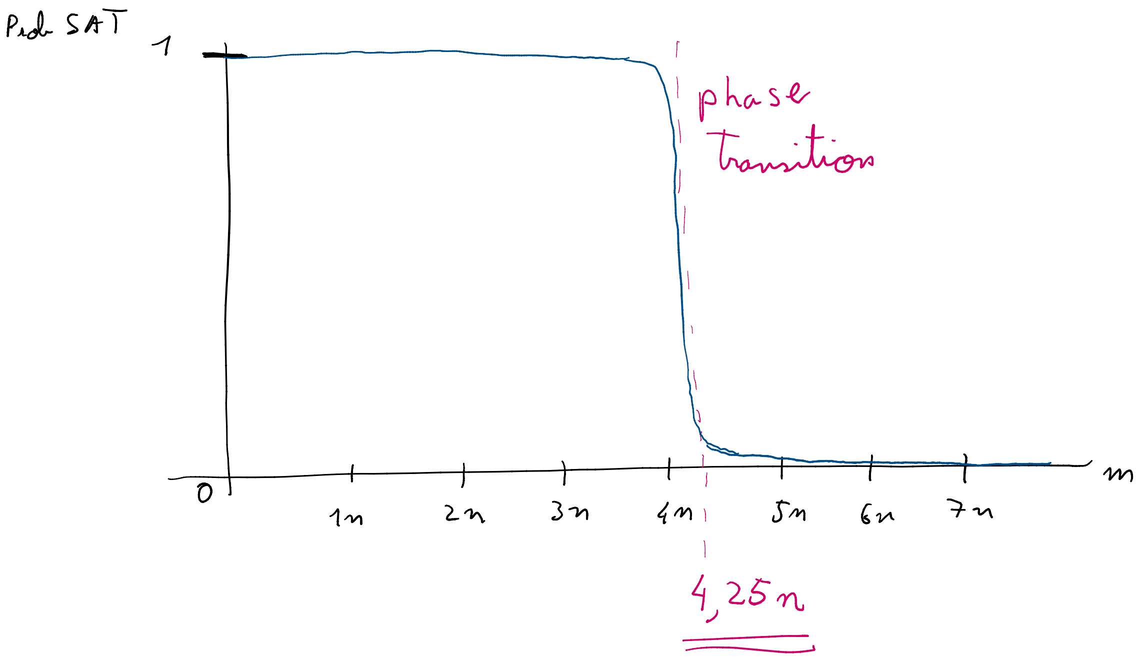
SAT

miércoles, 23 de febrero de 2022 8:38

Problemas con SAT

Dependiendo del numero de clausulas y vars, el problema es + o - dificil.

Con 3-SAT generado random, cada clausula de 3 literales se carga 1/8 de posibilidades.



n = # vars

m = # clausulas

Prolog

martes, 1 de marzo de 2022 9:11

Cláusulas de Horn: más $\underline{\underline{1}}$ lit +

En prolog, todo son 5

SAT → NP completo

SAT con C. de Horn → lineal

cabeza cola

F_1 :— F_2, F_3

Programa prolog ≡ BD deductiva

Cosas:

- todo acabado en .
- Consultar archivo: [nombre]. where archivo = nombre.pl
  ↳ + bien, cargar en BD interna
- Va a la primera clausula de la BD cuya cabeza sea unificable con el objetivo.

dar valores a las variables para que sean =

- El ámbito de las variables es la cláusula
- Tiene una pila de backtracking, para mirar las alternativas
- Si le añades ; después de la respuesta, pides +. Si pulsas ↵, se acaba y no te da más.