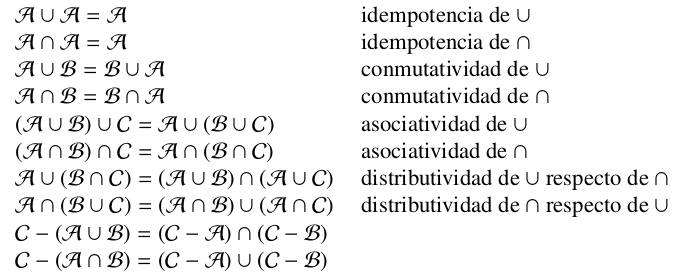
**Propiedades de las operaciones sobre conjuntos**



**Símbolos necesarios**

∀ - para todo

∃ - existe

**≡** - igual

**Satisfacción:** Sean I una interpretación y F una fórmula, ambas sobre el vocabulario

P. La evaluación en I de F, denotada eval I (F), es una función que por cada fórmula da

un valor de {0, 1}.

si F es un sı́mbolo p de P entonces eval I (F) = I(p)

**eval I ( (F ∧ G) )** = min( eval I (F), eval I (G) )

**eval I ( (F ∨ G) )** = max( eval I (F), eval I (G) )

**eval I (¬F)** = 1 − eval I (F)

**Satisfactibilidad, tautologı́a, consecuencia, y equivalencia lógica**

* Una fórmula **F es satisfactible** si tiene algún modelo, es decir, si existe alguna **interpretación I tal que I |= F.**
* Una fórmula **F es una tautologı́a** (o es válida), si toda interpretación es modelo de F, es decir, si para **toda interpretación I tenemos I |= F.**
* F es **consecuencia lógica de G** si todo modelo de G también lo es de F, es decir, si **para toda interpretación I sobre P tal que I |= G tenemos I |= F.** Sobrecargando el operador “|= ”, esto se denotará por G |= F.
* F y G son **lógicamente equivalentes** si tienen los mismos modelos, es decir, si para toda interpretación I sobre P tenemos I |= G si y sólo si I |= F. Esto se denotará por **G ≡ F**.

El sı́mbolo “|= ” denota:

* satisfacción cuando a su izquierda hay una interpretación I;

**I |= F** denota que **I satisface F**.

* cuando en la izquierda hay una fórmula G, el sı́mbolo “|= ” denota

consecuencia lógica; ası́, **G |= F**, denota que **todo modelo de G satisface F.**

**Demostraciones**

* **Consecuencia lógica:** F |= G **≡**  F -> G **≡** ¬F v G

**Per a demostrar:** F ^ ¬G és insat

* **Lógicamente equivalente:**  G **≡** F **≡** ( I |= G) <-> ( I |= F) **≡**  ((¬F ∨ G) ∧ (¬G ∨ F))

**Per a demostrar:**  ((F ∨ ¬G) ∧ (G ∨ ¬F)) es insat

**F -> G** abreviatura de **(¬F ∨ G)**

**F <-> G** abreviatura de **((¬F ∨ G) ∧ (¬G ∨ F))**

|  | SAT | TAUTOLOGÍA |
| --- | --- | --- |
| CNF | NP-Completo | lineal |
| DNF | lineal | NP-Completo |

**Tautología en CNF:** S is a tautology iff all clauses C in S are tautologies. A clause is a tautology iff it contains some predicate symbol p and its negation¬p. So the best method is to check this: linear time.

AMO’s

<https://www.cs.upc.edu/~roberto/li/codif.pdf>

* **Ladder encoding: (menores cláusulas)**

n auxiliary variables a1. . . an.

3n clauses:¬ai∨ai+1,¬xi∨ai,¬ai∨¬xi+1

for 1≤i≤n

* **The Heule-3 encoding: (menores clausulas)**

n/2 auxiliary variables

3n clauses, by expressing x1+···+xn≤1 as x1+x2+x3+a≤1 ∧ ¬a+x4+···+xn≤1

using(4/2)= 6 clauses forx1+x2+x3+a≤1

* **the quadratic encoding (menor nombre de auxiliars)**

no auxiliary vars

the(n2)clauses ¬xi ∨ ¬xj

for 1≤i < j≤n.

| DECIDIBLE | That there exists a procedure that, given the input data, **answers correctly whether the answer of Qis yes or no, and always terminates** |
| --- | --- |
| SEMI-DECIDIBLE | That there exists a procedure such that, given the input data, if it terminates it **answers correctly whether the answer of Qis yes or no**, and it **always terminates if the answer is yes** |
| CO-SEMI-DECIDIBLE | That there exists a procedure such that, given the input data, if it terminates **it answers correctly whether the answer of Qis yes or no**, and it **always terminates if the answer is no.** |

**2-SAT resolution**

1. Res(S) only contains 2-literal clauses (cannot get larger clauses by resolution from 2-literal clauses).
2. Only a quadratic number of 2-literal clauses exist, so |Res(S)| is quadratic and can be computed in polynomial time.
3. S insat iff empty clause in Res(S).