# Lògica en la Informàtica

Tardor 2022. Teoria classe 4

# Lògica en la Informàtica

## Temari


1. Introducció i motivació
2. Definició de Lògica Proposicional (Lprop)
3. Deducció en Lògica Proposicional
4. Definició de Lògica de Primer Ordre (LPO)
5. Deducció en Lògica de Primer Ordre
6. Programació Lògica (Prolog)

# Lògica en la Informàtica. Deducció en Lògica Proposicional

## Col·lecció d'apunts bàsics de lògica.  Fitxer p3.pdf

1. Formes normals i clàusules
   - Fórmules com a conjunts
   - Literals
   - CNF i DNF
   - Clàusula
   - Conjunt de clàusules
   - Clàusula buida
   - Clàusula de Horn

# Lògica en la Informàtica. Deducció en Lògica Proposicional

Exercicis del tema 3

5. Demostra que una clàusula és una tautologia si, i només si, conté alhora p i ¬p per un cert símbol proposicional p.

Demostrarem que una clàusula de la forma p1 v … v pm v ¬q1 v … v ¬qn NO és tautologia ssi NO conté alhora p i ¬p.

p1 v...v pm v -q1 v...v -qn NO es tautologia ssi
E I, tal que I no és model de p1 v...v pm v -q1 v...v -qn ssi
E I, tal que NO I |= p1 v...v pm v -q1 v...v -qn ssi
E I, tal que max( eval_I(p1)... eval_I(pm), eval_I(-q1), ... eval_I(-qn)) = 0 ssi
E I, tal que I(pi)=0 per a totes les pi i I(qj)=1 per a totes les qj ssi
pi != qj per a tota i,j

# Lògica en la Informàtica. Deducció en Lògica Proposicional

6. Sigui S un conjunt de clàusules amb $\square \notin S$ . Demostra que S és satisfactible (donant un model per a S) en cadascuna de les següents situacions:

   a) Tota clàusula C de S té algún literal positiu.

   C és de la forma: $p1 \vee \ldots \vee pm \vee \neg q1 \vee \ldots \vee \neg qn$ $(m>1, n>=0)$

Sigui S = { C1, C2, … }. Cada clàusula Ci te almenys un literal positiu (un símbol de predicat sense negar).
Un model que satisfarà totes les clàusules de S és la I tal que I(p)=1 per tot símbol de predicat p.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

6. Sigui S un conjunt de clàusules amb $\square \notin S$ . Demostra que S és satisfactible (donant un model per a S) en cadascuna de les següents situacions:

    b) Tota clàusula de S té algún literal negatiu.

Sigui S = { C1, C2, … }. Cada clàusula Ci te almenys un literal negatiu (un símbol de predicat negat).
Un model que satisfarà totes les clàusules de S és la I tal que I(p)=0 per tot símbol de predicat p.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

6. Sigui S un conjunt de clàusules amb $\square \notin S$ . Demostra que S és satisfactible (donant un model per a S) en cadascuna de les següents situacions:

    c) Per tot símbol de predicat p es compleix que: o bé p apareix només en literals positius en S, o bé p apareix només en literals negatius en S.

És a dir, cada clàusula és de la forma $p_1 \vee \ldots \vee p_m \vee -q_1 \vee \ldots \vee -q_n$ on

- les pi's només apareixen en literals positius en la resta de les clàusulas
- las qi's només apareixen en literals negatius en la resta de les clàusulas

Un model que va a satisfer totes les clàusulas de S és la I tal que

$I(p)=0$ per tot símbol p tal que p només apareix negatiu

$I(p)=1$ per tot símbol p tal que p només apareix positiu

# Lògica en la Informàtica. Deducció en Lògica Proposicional

6. Sigui S un conjunt de clàusules amb $\square \notin S$ . Demostra que S és satisfactible (donant un model per a S) en cadascuna de les següents situacions:

    c) Per tot símbol de predicat p es compleix que: o bé p apareix només en literals positius en S, o bé p apareix només en literals negatius en S.

Un model que va a satisfer totes les clàusulas de S és la I tal que

    I(p)=0 per tot símbol p tal que p només apareix negatiu
    I(p)=1 per tot símbol p tal que p només apareix positiu

Nota: en realitat aquesta I va a satisfer TOTS els literals de TOTES les clàusules (quan en realitat en tenia prou amb complir UN literal de cada clàusula).

# Lògica en la Informàtica. Deducció en Lògica Proposicional

7. Donats n símbols proposicionals:

7a) Quantes clàusules diferents (com a conjunts de literals) hi ha?

Donat un conjunt S de k elements, quants subconjunts diferents té?

S = {e1 e2 ... ek}

0 0 ... 0 denota el subconjunt buit<br>
0 0 ... 1 denota el subconjunt {ek}<br>
....<br>
1 1 ... 1 denota el subconjunt S = {e1 e2 ... ek }

Això explica que hi ha 2^k subconjunts diferents.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

7. Donats n símbols proposicionals:

7a) Quantes clàusules diferents (com a conjunts de literals) hi ha?

Si tenim n símbols, quants literals hi ha?   2n

Per tant, hi ha 2^(2n) clàusules (subconjunts dels 2n literals) = 4^n.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

7. Donats n símbols proposicionals:

7a) Quantes clàusules diferents (com a conjunts de literals) hi ha?

Una altra manera de veure el mateix: per cadascun dels n símbols p, en una clàusula passarà una de les següents 4 situacions:

a) estan p i -p en la clàusula

b) està només p en la clàusula

c) està només -p en la clàusula

d) no està ni p ni -p en la clàusula

és a dir, hi ha 4^n possibilitats de clàusules.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

7. Donats n símbols proposicionals:

7b) Quantes d'aquestes clàusules són insatisfactibles?

Només 1, la clàusula buida.

En una clàusula  p1 v...v pk v -q1 v...v -qm   si k+m > 0, llavors sí és satisfactible.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

7. Donats n símbols proposicionals:

7c) Quantes clàusules diferents i que no són tautologiias hi ha?

3^n: per l'altra manera de resoldre el 7a): per cada p, desapareix el cas "a) estan p i -p en la clàusula."

# Lògica en la Informàtica. Deducció en Lògica Proposicional

7. Donats n símbols proposicionals:

7d) Quantes clàusules diferents que contenen exactament un literal per cada símbol proposicional hi ha?

2^n: per l'altra manera de resoldre el 7a): per cada p, desapareixen el cas a) i el cas d).

# Lògica en la Informàtica. Deducció en Lògica Proposicional

8. Proposa un algorisme eficient que, donat un conjunt de clàusules S, retorna un conjunt de clàusules S' (no necessàriament definit sobre els mateixos símbols de predicado que S) amb com a molt 3 literals per clàusula, que és equisatisfactible a S (és a dir, que és satisfactible si i només si S ho és).

Ajuda: és possible introduir algun símbol de predicat p nou, que signifiqui: “$l \vee l'$ és cert” per a algun parell de literals l i l'.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

8. Proposa un algorisme eficient que, donat un conjunt de clàusules S, retorna un conjunt de clàusules S' (no necessàriament definit sobre els mateixos símbols de predicado que S) amb com a molt 3 literals per clàusula, que és equisatisfactible a S (és a dir, que és satisfactible si i només si S ho és).

Si tinc una clàusula massa llarga (més de 3 lits), puc escriure-la com a l v l' v C on C és la resta de la clàusula.

Llavors S és de la forma { l v l' v C } ∪ S1.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

8. Proposa un algorisme eficient que, donat un conjunt de clàusules S, retorna un conjunt de clàusules S' (no necessàriament definit sobre els mateixos símbols de predicado que S) amb com a molt 3 literals per clàusula, que és equisatisfactible a S (és a dir, que és satisfactible si i només si S ho és).

Com podem expressar que  p <--> l v l'  mitjançant clàusules de màxim 3 literals? Recordem:  a --> b  ===  -a v b

p  -->  l v l'   ===  -p v l v l'

p  <--  l v l'   ===   { l --> p,  l' --> p }  ===  { -l v p,   -l' v p }

# Lògica en la Informàtica. Deducció en Lògica Proposicional

8. Proposa un algorisme eficient que, donat un conjunt de clàusules S, retorna un conjunt de clàusules S' (no necessàriament definit sobre els mateixos símbols de predicado que S) amb com a molt 3 literals per clàusula, que és equisatisfactible a S (és a dir, que és satisfactible si i només si S ho és).

Sigui S' el conjunt { p v C, -l v p, -l' v p, -p v l v l' } ∪ S1,

NOTA: aqui p és un símbol nou!!

Hem escurçat en 1 literal 1 clàusula.

Però puc repetir això tantes vegades com faci falta.

→ Falta veure que S és satisfactible ssi S' és satisfactible.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

8. Proposa un algorisme eficient que, donat un conjunt de clàusules S, retorna un conjunt de clàusules S' (no necessàriament definit sobre els mateixos símbols de predicado que S) amb com a molt 3 literals per clàusula, que és equisatisfactible a S (és a dir, que és satisfactible si i només si S ho és).

A) ==>: S és satisfactible ==> S' és satisfactible.
B) <==: S' és satisfactible ==> S és satisfactible.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

8. Proposa un algorisme eficient que, donat un conjunt de clàusules S, retorna un conjunt de clàusules S' (no necessàriament definit sobre els mateixos símbols de predicado que S) amb com a molt 3 literals per clàusula, que és equisatisfactible a S (és a dir, que és satisfactible si i només si S ho és).

A) ==>:    S és satisfactible  ==>  S' és satisfactible.

S és satisfactible                    ssi

E I  tq  I és modelo de S     ssi

E I  tq  I |= S

# Lògica en la Informàtica. Deducció en Lògica Proposicional

8. Proposa un algorisme eficient que, donat un conjunt de clàusules S, retorna un conjunt de clàusules S' (no necessàriament definit sobre els mateixos símbols de predicado que S) amb com a molt 3 literals per clàusula, que és equisatisfactible a S (és a dir, que és satisfactible si i només si S ho és).

…

E I  tq  I |= S

Si I |= l v l' llavors sigui I' la interpretació que ESTÉN la I amb I'(p)=1,
(I' es como I, excepto que además I'(p)=1)

si no, sigui I' la interpretació que ESTÉN la I amb I'(p)=0.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

8. Proposa un algorisme eficient que, donat un conjunt de clàusules S, retorna un conjunt de clàusules S' (no necessàriament definit sobre els mateixos símbols de predicado que S) amb com a molt 3 literals per clàusula, que és equisatisfactible a S (és a dir, que és satisfactible si i només si S ho és).

Si I |= l v l' llavors sigui I' la interpretació que ESTÉN la I amb I'(p)=1,
(I' es como I, excepto que además I'(p)=1)

si no, sigui I' la interpretació que ESTÉN la I amb I'(p)=0.

Tenim que I' |= S1, perquè I |= S1.

a més a més I' |= { p v C, -l v p, -l' v p, -p v l v l' } perquè (entre altres raons) I |= l v l’ v C.

Per això I' |= S' i per tant S' és sat.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

8. Proposa un algorisme eficient que, donat un conjunt de clàusules S, retorna un conjunt de clàusules S' (no necessàriament definit sobre els mateixos símbols de predicado que S) amb com a molt 3 literals per clàusula, que és equisatisfactible a S (és a dir, que és satisfactible si i només si S ho és).

B) <==: S' és satisfactible ==> S és satisfactible.

Sigui I' model de S'.
Sigui I la RESTRICCIÓ de I' "oblidant-nos" de la p.
I veiem que I |= S.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

9. Sigui S un conjunt de clàusules de Horn on la clàusula buida no està en S (□ ∉ S). Demostra que S és satisfactible (donant un model per a S) si no hi ha cap clàusula que només consti d'un únic literal positivo.

Totes les clàusules de S són de Horn i no-buides, és a dir, de la forma
p1 v...v pk v -q1 v...v -qm on k+m > 0, i k<=1. Com poden ser?
De la forma:

a) p
b) p v -q1 v...v -qn amb n>0
c) -q1 v...v -qn amb n>0

# Lògica en la Informàtica. Deducció en Lògica Proposicional

9. Sigui S un conjunt de clàusules de Horn on la clàusula buida no està en S (□ ∉ S). Demostra que S és satisfactible (donant un model per a S) si no hi ha cap clàusula que només consti d’un únic literal positivo.

~~a) p~~
b) p v -q1 v...v -qn   amb n>0
c)        -q1 v...v -qn   amb n>0

Ara diu que tampoc hi ha de tipus a): no hi ha cap clàusula que només consti d'un únic literal positiu.
Llavors és satisfactible: un model és la I on per a **tot símbol** p tenim I (p)=0.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

10. Demostra que l’enunciat de l’exercici previ és fals quan S no és de Horn.

Donem un contraexemple d'un conjunt de clàusules S on:

- la clàusula buida no està en S, i
- no hi ha cap clàusula que només consti d'un únic literal positiu,
- i en canvi, S és insatisfactible.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

10. Demostra que l’enunciat de l’exercici previ és fals quan S no és de Horn.

Donem un contraexemple d'un conjunt de clàusules S on:

- la clàusula buida no està en S, i
- no hi ha cap clàusula que només consti d'un únic literal positiu, i
- S és insatisfactible.

Perquè no sigui de Horn, se'ns ocorre posar la clàusula més senzilla que no és de Horn:

p v q

# Lògica en la Informàtica. Deducció en Lògica Proposicional

10. Demostra que l’enunciat de l’exercici previ és fals quan S no és de Horn.

Donem un contraexemple d'un conjunt de clàusules S on:

- la clàusula buida no està en S, i
- no hi ha cap clàusula que només consti d'un únic literal positiu, i
- S és insatisfactible.

Exemple: Sigui S el conjunt de clàusulas, que compleix les condicions però és insatisfactible:

p v q

-p

-q

# Lògica en la Informàtica. Deducció en Lògica Proposicional

10. Demostra que l'enunciat de l'exercici previ és fals quan S no és de Horn.

Un altre exemple:

p v  q
p v -q
-p v  q
-p v -q.

Això és insatisfactible, perquè cadascuna de les quatre interpretacions que hi ha és falsificada per una de les clàusules.
És a dir, per a tota I, hi ha una clàusula C en S tal que I no satisfà C.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

12. Per a una fórmula en DNF, quin és el millor algorisme possible per a decidir si és satisfactible? Quin cost té?

Una DNF (disjunctive normal form) és una disjunció (OR) de “cubs”, on cada cub és  p1 &...& pk & -q1 &...& -qm.
Una DNF = { C1 v...v Cn } és satisfactible ssi algun cub Ci és satisfactible.
Una cub p1 &...& pk & -q1 &...& -qm és satisfactible ssi no hi ha cap símbol que aparegui en un literal positiu del cub i també en un negatiu.
Per tant, el cost pot ser **lineal**.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

Col·lecció d'apunts bàsics de lògica.  Fitxer p3.pdf

Apartat 5. Resolució. Correcció i completitud

# Lògica en la Informàtica. Deducció en Lògica Proposicional

## Resolució. Correcció i completitud

```
p v C          -p v D
------------------------      per algun símbol  p
       C v D
```

La resolució és CORRECTA? És a dir, siguin com siguin como C, D i p, tenim que (p v C) & (-p v D) |= C v D ?

Sigui I un model de (p v C) & (-p v D). Hi han dos casos:

I(p)=1 Llavors I |= (p v C) & (-p v D) ==> I |= -p v D ==> I |= D ==> I |= C v D

I(p)=0 Llavors I |= (p v C) & (-p v D) ==> I |= p v C ==> I |= C ==> I |= C v D.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

## Resolució. Correcció i completitud

$$\frac{p \vee C \qquad -p \vee D}{C \vee D}$$

per algun símbol p

Exemple: Sigui S el conjunt de clàusules

{ p v q,<br>
p v -q,<br>
-p v q,<br>
-p v -q }.

Puc obtenir mitjançant resolució a partir de p v q i p v -q (sobre la q) i obtinc p v p que és el mateix que p.

Puc obtenir mitjançant resolució a partir de -p v q i -p v -q (sobre la q) i obtinc -p v -p que és el mateix que -p.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

## Resolució. Correcció i completitud

$$\frac{p \vee C \qquad -p \vee D}{C \vee D} \quad \text{per algun símbol } p$$

Exemple: Sigui S el conjunt de clàusules

{ p v q,
p v -q,
-p v q,
-p v -q }.

A partir de les dues clàusules noves p i -p, en un altre pas puc obtenir **la clàusula buida**

# Lògica en la Informàtica. Deducció en Lògica Proposicional

## Resolució. Correcció i completitud

$$\frac{p \vee C \qquad -p \vee D}{C \vee D} \quad \text{per algun símbol } p$$

Exemple: Sigui S el conjunt de clàusules

{ p v q,
p v -q,
-p v q,
-p v -q }.

S0 = S

S1 = S0 U { p, -p, q, -q , p v -p, q v -q, ... }

S2 = S1 U { [] ... }

S3 = S2 U ???

# Lògica en la Informàtica. Deducció en Lògica Proposicional

Resolució. Correcció i completitud

$$\frac{p \lor C \qquad -p \lor D}{C \lor D} \qquad \text{per algun símbol } p$$

Hi ha un teorema que diu:

S és insatisfactible SSI mitjançant resolució puc arribar a obtenir la clàusula buida ( formalment, SSI $\square \in Res(S)$ ).

# Lògica en la Informàtica. Deducció en Lògica Proposicional

16. Demostra que, per a tot conjunt finit de clàusules S, tenim que Res(S) és un conjunto finit de clàusules, si es consideren les clàusules com a conjunts de literals (per exemple, $C \vee p$ és la mateixa clàusula que $C \vee p \vee p$)

Si el conjunt inicial S té n símbols diferents, llavors EXISTEIXEN 2^(2n) clàusules diferents (que és un número gran, però **finit**).
Per tant, la resolució arribarà un moment, una S_i, tal que S_i = S_{i+1}, és a dir, que a partir d'aquesta S_i ja no afegim res nou, i totes les S_j a partir d'aquí seran iguals.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

17. Sigui S un conjunt de clàusules. Demostra que Res(S) es lògicamente equivalent a S.

Sigui I una interpretació **qualsevol**.
Hem de demostrar que $I \models S$ ssi $I \models Res(S)$.

a) $I \models Res(S) \implies I \models S$
b) $I \models S \implies I \models Res(S)$

# Lògica en la Informàtica. Deducció en Lògica Proposicional

17. Sigui S un conjunt de clàusules. Demostra que Res(S) es lògicamente equivalent a S.

Sigui I una interpretació qualsevol.

Hem de demostrar que $I \models S$ ssi $I \models Res(S)$.

a) $I \models Res(S) \implies I \models S$

trivialment, perquè S és un subconjunt de Res(S)

(per def. de Res(S) que és la unió de totes les Si's).

# Lògica en la Informàtica. Deducció en Lògica Proposicional

17. Sigui S un conjunt de clàusules. Demostra que Res(S) es lògicamente equivalente a S.

Sigui I una interpretació qualsevol.
Hem de demostrar que I |= S ssi I |= Res(S).
b) I |= S ==> I |= Res(S)
Hem obtingut Res(S) a partir de S, a força d'afegir, un nombre finit de vegades k, una conclusió per resolució a partir de clàusules que ja teníem.
Demostrarem que per a tota I, I |= S ==> I |= Res(S) per **inducció** sobre k.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

17. Sigui S un conjunt de clàusules. Demostra que Res(S) es lògicamente equivalente a S.

Sigui I una interpretació qualsevol.

Hem de demostrar que I |= S ssi I |= Res(S).

b) I |= S ==> I |= Res(S)

Si k=0, trivial perquè llavors S = Res(S).

# Lògica en la Informàtica. Deducció en Lògica Proposicional

17. Sigui S un conjunt de clàusules. Demostra que Res(S) es lògicamente equivalente a S.

Sigui I una interpretació qualsevol.
Hem de demostrar que $I \models S$ ssi $I \models Res(S)$.
b) $I \models S \implies I \models Res(S)$

Si k>0, suposa que el primer pas és de S a un S', afegint 1 clàusula per resolució a partir de S.

- ➤ Per correcció de la resolució $S \models S'$, per la qual cosa $I \models S \implies I \models S'$. A més a més, com $S \subseteq S'$, tenim també que $S' \models S$. Per tant, $S \equiv S'$.
- ➤ Per Hipòtesi d'Inducció, com el nombre de passos des de S' a Res(S) és k-1, tenim que $I \models Res(S)$. qed.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

18. La resolució és completa? Demostra-ho.

# Lògica en la Informàtica. Deducció en Lògica Proposicional

**Exercicis per la propera classe:**

- exercicis fins al 27, i també
- el sudoku.