# Lògica en la Informàtica

Tardor 2022. Teoria classe 1

# Lògica en la Informàtica. Presentació

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. Presentació

## Com estudiar teoria?

- classes de teoria: “intuïció”
- material online:
    - apunts per temes al Racó: definicions formals
    - pàgina web: [www.cs.upc.edu/~rivero/Teaching/LI](www.cs.upc.edu/~rivero/Teaching/LI)
        - apunts més específics
        - molts exàmens de teoria (també resolts)
- mètode per superar l’assignatura:
    - ENTENDRE bé les classes i els apunts. Llavors FER els exercicis i els exàmens
- teoria: 60% de la nota
- dos exàmens:
    - Parcial: LProp 02/11/2022
    - Final: [LProp] + LPO 16/01/2023

# Lògica en la Informàtica. Presentació

Com treballar al laboratori?

- 6 pràctiques. cadascuna en 2 sessions
- enunciats:
  - www.cs.upc.edu/~rivero/Teaching/LI/#practiques-de-laboratori
- FER també les pràctiques, no sols entrede-les
- laboratori: 40% de la nota LI
- dos exàmens:
  - Part I:  pràctiques 1,2,3    2/11/2022
  - Part II: pràctiques 4,5,6    9/01/2023
- **No concentrar-se sols en laboratori o teoria**

# Lògica en la Informàtica. Introducció i motivació

Perquè estudiar *Logic in Computer Science*?

- La història: els grecs, els matemàtics, els informàtics
  - Els grecs. La lògica permet deduir conclusions vertaderes de premisses vertaderes, i fa possible la deducció
  - Els matemàtics [començaments S. XX]. Necessitat de formalitzar les matemàtiques: **teoria de conjuntos**.
    - Paradoxa de Russell: Sigui S = { C | C no pertany a C }. Llavors, ¿S pertany a S?
- Per què Lògica Matemàtica NO és LI:
  - necessitats completament diferents: la deducció es eficient, computabilitat, expressivitat
  - avui en dia el 99% de les publicacions en lògica són de Computer Science.

# Lògica en la Informàtica. Introducció i motivació

Perquè estudiar *Logic in Computer Science*?

- Estudi dels Fonaments
  - estudiar les "eines del moment" (que no existien fa 20 anys, ni existiran d’aquí a 20 anys)?
  - o estudiar els Fonaments? (que romanen, i que permeten aprendre qualsevol “eina del moment”)?
- Fonaments en Informàtica
  - matemàtiques (sobre tot discretes)
  - algoritmia
  - limitacions inherents de la computació: complexitat, calculabilitat, …
  - teoria d’autòmatas i llenguatges
  - lògica

# Lògica en la Informàtica. Introducció i motivació

Perquè estudiar *Logic in Computer Science*?

- El llenguatge natural és imprecís, ambiguo
  - “They are hunting dogs”
  - "Aquí vendemos zapatos de piel de señora"
  - "El perro está listo para comer"
- Fins i tot en àmbits com:
  - Control aeri
  - Marc legal
  - Especificació de software
- Necessitem “formalitzar”

# Lògica en la Informàtica. Introducció i motivació

Perquè estudiar *Logic in Computer Science*?

- Què significa "formal"?
  - Que té una **sintaxis** i una **semàntica** (significat) definides de manera **inambigua**
- Què és una lògica?
  - sintaxis: què es una fórmula F?
  - semàntica:
    - què és una interpretació I?
    - quan una I SATISFÀ una F? I |= F?
- Intuitivament:
    - "Interpretació" == "situació de la vida real a modelar"
    - Una F "representa" aquelles I on se satisfà F, on es compleix.
- Aquí veurem dues lògiques: LProp i LPO (amb algunes variantes)

# Lògica en la Informàtica. Introducció i motivació

Perquè estudiar *Logic in Computer Science*?

- La “deducció intuitiva” que fem nosaltres ens enganya… Exemple:
- Rajoy: "La gente honrada paga sus impuestos. Yo pago mis impuestos."

  Tot i que els pagui, això no implica que el sigui honrat)

  "Implicació invertida": Si A --> B i tinc B, llavors A. **NO és correcte**.

# Lògica en la Informàtica. Introducció i motivació

Perquè estudiar *Logic in Computer Science*?

- La “deducció intuitiva” que fem nosaltres ens enganya… Exemples:
- Lao Tse: "Els que pensen no parlen. Els que parlen no pensen."
    - diu dues vegades el mateix?
    - A ---> B     és el mateix que   -A v B
    - Ax p(x) --> -h(x)     ===     Ax -p(x) v -h(x)
    - Ax h(x) --> -p(x)     ===     Ax -h(x) v -p(x)     ===     Ax -p(x) v -h(x)
    - diu (dues vegades) que **no hi ha ningú que parli i pensi alhora**

# Lògica en la Informàtica. Introducció i motivació

Perquè estudiar *Logic in Computer Science*?

- La “deducció intuitiva” que fem nosaltres ens enganya… Exemples:
- "1. Lo que no mata engorda". "2. La lechuga no engorda".
- Això implica que la lechuga mata?
  - m = "la lechuga mata"
  - e = "la lechuga engorda"

  1. -m --> e === m v e (totes les coses, o bé maten, o bé engorden. No hi ha res que ni mate ni engorde) === -e --> m

  2. -e

- Així sí implica m: que **la lechuga mata**…

# Lògica en la Informàtica. Introducció i motivació

Perquè estudiar *Logic in Computer Science*?

Aplicacions directes de la lògica en la informàtica:

- Verificació de hardware i de software
  - demostració de correcció (terminació, etc.)
  - testing
- Aplicacions "crítiques" en:
  - vides humanes: centrals nuclears, químicas, avions, trànsit, cotxes, trens,... "safety"
  - confidencialitat: diners electrònics, signatura electrònic, dades bancaris... "security"
  - economia: la borsa, la telefonia, el sistema eléctric...
- Intel·ligència artificial, web semàntica (representaciò del coneixement: ontologías, description logics, sistemes experts, ...)

# Lògica en la Informàtica. Introducció i motivació

Perquè estudiar *Logic in Computer Science*?

Aplicacions directes de la lògica en la informàtica:

- Bases de dades
- Programació lògica (prolog)
- Ús de lògica per a resoldre problemes d'optimització, planificació...: per exemple, https://barcelogic.com/
  - especificació/formalització fent servir lògica
  - "solvers" lògics, per exemple, SAT solvers.

# Lògica en la Informàtica. Definició de Lògica Proposicional

- EN QUALSEVOL LÒGICA:
- Què és una lògica?  Definició d'una lògica:
  - sintaxis:
    - què és una fòrmula F?
  - semàntica:
    - què és una interpretació I?
    - quan una I SATISFÀ una F?        notació:  I |= F
- Fem servir I per a denotar interpretacions i F,G per a fórmules.

# Lògica en la Informàtica. Definició de Lògica Proposicional

- EN QUALSEVOL LÒGICA:
    - I **és model** de F si I satisfà a F (se denota I |= F )
    - F és **satisfactible** si F té algún model
    - F és **insatisfactible** si F no té model
    - F és **tautologia** si tota I és model de F
    - G és **conseqüència lògica** de F si tot model de F satisfà G (es denota F |= G)
    - F i G són **lògicament equivalents** si F i G tenen els mateixos models (es denota F ≡ G)

Nota: Per definició tenim que F ≡ G ssi F |= G i G |= F.

# Lògica en la Informàtica. Definició de Lògica Proposicional

- **Sintaxis**: les fórmules es construeixen amb un conjunto P de símbols de predicat: p,q,r,... (o "variables" x1,x2,x3…) i les conectives:

  & és AND

  v és OR

  - és NOT

- Exemple de fòrmula F: p & ((q v -r) & ((-p v r) & -q))

# Lògica en la Informàtica. Definició de Lògica Proposicional

- **Semàntica**:
  - a) Una interpretació I és una funció I: P --> {0,1}. Ens diu per cada símbol de P si és cert o fals.
  - b) Quan una I SATISFÀ una F?    I |= F?   Quan eval_I(F) = 1.  Quan l'avaluació en I de F ens dona 1.
    - eval_I(   p   )    =  I(p)          si p pertany a P, si p és un símbol de predicat (del conjunt P)
    - eval_I(  - F  )    =  1 -  eval_I(F)
    - eval_I( F & G ) =  min( eval_I(F), eval_I(G) )
    - eval_I( F v G )  =  max( eval_I(F), eval_I(G) )
- Donada una I, per exemple I(p)=1,  I(q)=0,  I(r)=1,  i una F com

  p & ((q v -r) & ((-p v r) & -q))

  quin és el cost de decidir si I és model de F? quant costa calcular eval_I(F)?

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

1. Quantes interpretacions posibles hi ha en funció de |P|?

   (nota: si S es un conjunt, |S| denota la seva cardinalitat, és a dir, el número d'elements de S).

# Lògica en la Informàtica. Definició de Lògica Proposicional

Podem fer la llista de totes les possibles I's (aquesta llista també es diu “taula de veritat”):

Per exemple, si P = {p,q,r} i F = p & ((q v -r) & ((-p v r) & -q)) tenim

| pqr | F |
|---|---|
| 000 | 0 |
| 001 | 0 |
| 010 | 0 |
| 011 | 0 |
| 100 | 0 |
| 101 | 0 |
| 110 | 0 |
| 111 | 0 |

i veiem que el nostre exemple de F és **INsatisfactible**: no té cap I que la satisfaga, no té cap model.

En la práctica, es fa SAT on la F donada és una CNF (conjunctive normal form): una fórmula que és un conjunt (ANDs) de clàusules (ORs de literales), on un literal és una variable (literal positivo) o una variable negada (literal negativo).

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

2. Demostra que p & -p és insatisfactible fent servir tan sols la definició de la LProp.

| | | |
|---|---|---|
| p & -p és insatisfactible | ssi | [ per definició de insatisfactible ] |
| p & -p no té cap model | ssi | [ per definició de model ] |
| A I, no I \|= p & -p | ssi | [ per definició de \|= ] |
| A I, eval_I( p & -p ) = 0 | ssi | [ per definició de eval_I( … & … ) ] |
| A I, min( eval_I(p), eval_I(-p) ) = 0 | ssi | [ per definició de eval_I( - … ) ] |
| A I, min( eval_I(p), 1-eval_I( p) ) = 0 | ssi | [ donat que eval_I(p) sempre és 0 o 1, i per definició de min ] |
| A I, 0 = 0 | ssi | [ perquè 0=0 es cert ] |
| cert | | |

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

2. Demostra que p & -p  és insatisfactible fent servir tan sols la definició de la LProp.

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

2. Demostra que p & -p  és insatisfactible fent servir tan sols la definició de la LProp.

p & -p  és insatisfactible          ssi    [ per definició de insatisfactible    ]

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

2. Demostra que p & -p és insatisfactible fent servir tan sols la definició de la LProp.

| | | |
|---|---|---|
| p & -p és insatisfactible | ssi | [ per definició de insatisfactible ] |
| p & -p no té cap model | ssi | [ per definició de model ] |

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

2. Demostra que p & -p  és insatisfactible fent servir tan sols la definició de la LProp.

| | | |
|---|---|---|
| p & -p  és insatisfactible | ssi | [ per definició de insatisfactible ] |
| p & -p  no té cap model | ssi | [ per definició de model ] |
| A  I,  no   I  \|=  p & -p | ssi | [ per definició de \|= ] |

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

2. Demostra que p & -p és insatisfactible fent servir tan sols la definició de la LProp.

| | | |
|---|---|---|
| p & -p és insatisfactible | ssi | [ per definició de insatisfactible ] |
| p & -p no té cap model | ssi | [ per definició de model ] |
| A I, no I \|= p & -p | ssi | [ per definició de \|= ] |
| A I, eval_I( p & -p ) = 0 | ssi | [ per definició de eval_I( … & … ) ] |

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

2. Demostra que p & -p és insatisfactible fent servir tan sols la definició de la LProp.

p & -p és insatisfactible ssi [ per definició de insatisfactible ]

p & -p no té cap model ssi [ per definició de model ]

A I, no I |= p & -p ssi [ per definició de |= ]

A I, eval_I( p & -p ) = 0 ssi [ per definició de eval_I( … & … ) ]

A I, min( eval_I(p), eval_I(-p) ) = 0 ssi [ per definició de eval_I( - … ) ]

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

2. Demostra que p & -p és insatisfactible fent servir tan sols la definició de la LProp.

| | | |
|---|---|---|
| p & -p és insatisfactible | ssi | [ per definició de insatisfactible ] |
| p & -p no té cap model | ssi | [ per definició de model ] |
| A I, no I \|= p & -p | ssi | [ per definició de \|= ] |
| A I, eval_I( p & -p ) = 0 | ssi | [ per definició de eval_I( … & … ) ] |
| A I, min( eval_I(p), eval_I(-p) ) = 0 | ssi | [ per definició de eval_I( - … ) ] |
| A I, min( eval_I(p), 1-eval_I( p) ) = 0 | ssi | [ donat que eval_I(p) sempre és 0 o 1, i per definició de min ] |

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

2. Demostra que p & -p  és insatisfactible fent servir tan sols la definició de la LProp.

| | | |
|---|---|---|
| p & -p  és insatisfactible | ssi | [ per definició de insatisfactible ] |
| p & -p  no té cap model | ssi | [ per definició de model ] |
| A  I,  no   I  \|=  p & -p | ssi | [ per definició de \|= ] |
| A  I,  eval_I( p & -p ) = 0 | ssi | [ per definició de eval_I( … & … ) ] |
| A  I,  min( eval_I(p),  eval_I(-p) ) = 0 | ssi | [ per definició de eval_I( - … ) ] |
| A  I,  min( eval_I(p), 1-eval_I( p) ) = 0 | ssi | [ donat que eval_I(p) sempre és 0 o 1, i per definició de min ] |
| A  I,  0 = 0 | ssi | [ perquè 0=0 es cert ] |

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

2. Demostra que p & -p és insatisfactible fent servir tan sols la definició de la LProp.

| | | |
|---|---|---|
| p & -p és insatisfactible | ssi | [ per definició de insatisfactible ] |
| p & -p no té cap model | ssi | [ per definició de model ] |
| A I, no I \|= p & -p | ssi | [ per definició de \|= ] |
| A I, eval_I( p & -p ) = 0 | ssi | [ per definició de eval_I( … & … ) ] |
| A I, min( eval_I(p), eval_I(-p) ) = 0 | ssi | [ per definició de eval_I( - … ) ] |
| A I, min( eval_I(p), 1-eval_I( p) ) = 0 | ssi | [ donat que eval_I(p) sempre és 0 o 1, i per definició de min ] |
| A I, 0 = 0 | ssi | [ perquè 0=0 és cert ] |
| cert | | |

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

6. Demostra (fent servir tan sols la definició de la LProp) que, per a tota fórmula F,

F és tautologia ssi -F és insatisfactible

Podem fer una cadena de SSIs o demostrar les dues implicacions per separat:

A) F és tautologia ==> .... ==> -F és insatisfactible

B) -F és insatisfactible ==> .... ==> F és tautologia

En aquest cas farem una cadena de SSIs:

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

6. Demostra (fent servir tan sols la definició de la LProp) que, per a tota fórmula F,

F és tautologia ssi -F és insatisfactible

F és tautologia ssi [per definició de tautologia ]

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

6. Demostra (fent servir tan sols la definició de la LProp) que, per a tota fórmula F,

F és tautologia ssi -F és insatisfactible

| | | |
|---|---|---|
| F és tautologia | ssi | [per definició de tautologia ] |
| A I, I és model de F | ssi | [per definició de model ] |

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

6. Demostra (fent servir tan sols la definició de la LProp) que, per a tota fórmula F,

F és tautologia ssi -F és insatisfactible

| | | |
|---|---|---|
| F és tautologia | ssi | [per definició de tautologia ] |
| A I, I és model de F | ssi | [per definició de model ] |
| A I, I \|= F | ssi | [per definició de \|= ] |

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

6. Demostra (fent servir tan sols la definició de la LProp) que, per a tota fórmula F,

F és tautologia ssi -F és insatisfactible

| | | |
|---|---|---|
| F és tautologia | ssi | [per definició de tautologia ] |
| A I, I és model de F | ssi | [per definició de model ] |
| A I, I \|= F | ssi | [per definició de \|= ] |
| A I, eval_I( F ) = 1 | | |

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

6. Demostra (fent servir tan sols la definició de la LProp) que, per a tota fórmula F,

F és tautologia ssi -F és insatisfactible

F és tautologia ssi [per definició de tautologia ]
A I, I és model de F ssi [per definició de model ]
A I, I |= F ssi [per definició de |= ]
A I, eval_I( F ) = 1

-F és insatisfactible ssi [per definició de insatisfactible ]

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

6. Demostra (fent servir tan sols la definició de la LProp) que, per a tota fórmula F,

F és tautologia  ssi  -F és insatisfactible

| | | |
|---|---|---|
| F és tautologia | ssi | [per definició de tautologia ] |
| A I, I és model de F | ssi | [per definició de model ] |
| A I, I \|= F | ssi | [per definició de \|= ] |
| A I, eval_I( F ) = 1 | | |
| | | |
| A I, I no és model de -F | ssi | [per definició de model ] |
| -F és insatisfactible | ssi | [per definició de insatisfactible ] |

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

6. Demostra (fent servir tan sols la definició de la LProp) que, per a tota fórmula F,

F és tautologia ssi -F és insatisfactible

F és tautologia ssi [per definició de tautologia ]
A I, I és model de F ssi [per definició de model ]
A I, I |= F ssi [per definició de |= ]
A I, eval_I( F ) = 1

A I, no I |= -F ssi [per definició de |= ]
A I, I no és model de -F ssi [per definició de model ]
-F és insatisfactible ssi [per definició de insatisfactible ]

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

6. Demostra (fent servir tan sols la definició de la LProp) que, per a tota fórmula F,

F és tautologia ssi -F és insatisfactible

| | | |
|---|---|---|
| F és tautologia | ssi | [per definició de tautologia ] |
| A I, I és model de F | ssi | [per definició de model ] |
| A I, I \|= F | ssi | [per definició de \|= ] |
| A I, eval_I( F ) = 1 | | |
| | | |
| A I, eval_I( -F ) = 0 | ssi | [per eval d’un not ] |
| A I, no I \|= -F | ssi | [per definició de \|= ] |
| A I, I no és model de -F | ssi | [per definició de model ] |
| -F és insatisfactible | ssi | [per definició de insatisfactible ] |

# Lògica en la Informàtica. Definició de Lògica Proposicional

## Exercicis del capítol 2 dels apunts

6. Demostra (fent servir tan sols la definició de la LProp) que, per a tota fórmula F,

F és tautologia ssi -F és insatisfactible

| | | |
|---|---|---|
| F és tautologia | ssi | [per definició de tautologia ] |
| A I, I és model de F | ssi | [per definició de model ] |
| A I, I \|= F | ssi | [per definició de \|= ] |
| A I, eval_I( F ) = 1 | ssi | [per aritmètica ] |
| A I, 1 - eval_I( F ) = 0 | | |
| A I, eval_I( -F ) = 0 | ssi | [per eval d'un not ] |
| A I, no I \|= -F | ssi | [per definició de \|= ] |
| A I, I no és model de -F | ssi | [per definició de model ] |
| -F és insatisfactible | ssi | [per definició de insatisfactible ] |