Lògica en la Informàtica
Presentació

José Miguel Rivero

Dept. Ciències de la Computació
Universitat Politècnica de Catalunya (UPC)

Primavera 2025

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)

Presentació

Com estudiar teoria?

- classes de teoria: "intuïció"
- material online:
  - apunts per temes al Racó: definicions formals
  - pàgina web: ☞ www.cs.upc.edu/~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	dt 02/04/2025
Final:	[LProp] + LPO	dv 17/06/2025

Presentació

Com treballar al laboratori?

- 6 pràctiques. cadascuna en 2 sessions
- enunciats:

  `www.cs.upc.edu/~li/#practiques-de-laboratori`
- Lliurament de les pràctiques amb DATA LÍMIT per cadascuna
- FER també les pràctiques, no sols entendre-les
- laboratori: 40% de la nota LI
- dos exàmens:

  Part I: pràctiques 1,2,3 dt 02/04/2025
  Part II: pràctiques 4,5,6 dt 13/06/2025
- **No concentrar-se sols en laboratori o teoria**

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)

Crèdits

El material utilitzat en aquesta presentació ha estat extret del elaborat pel professor Robert Nieuwenhuis (Dept. CS, UPC) per l'assignatura *Lògica en la Informàtica* de la FIB.

En particular, del llibre *Lógica para informáticos* - Farré, R. [et al.], Marcombo, 2011. ISBN: 9788426716941.

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 conjunts**.

    Paradoxa de Russell: Sigui $S = \{ C \mid C \text{ no pertany a } C \}$.
    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*.

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

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"

Introducció i motivació

Perquè estudiar Logic in Computer Science?

- Què significa "formal"?
  - Que té una **sintaxi** i una **semàntica** (significat) definides de manera **inambigua**
- Què és una lògica?
  - sintaxi: què es una fórmula F?
  - semàntica:
    - què és una interpretació I?
    - quan una I SATISFÀ una F? $I \models F$?
- Intuitivament:
  - "Interpretació" $\equiv$ "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)

Perquè estudiar Logic in Computer Science?

- La "deducció intuitiva" que fem nosaltres ens enganya...
  Exemples:
- 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 $A \rightarrow B$ i tinc B, llavors A.
  NO és correcte.

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?
  - $A \to B$ és el mateix que $\neg A \vee B$
  - $\forall x\,(pen(x) \to \neg par(x)) \;\equiv\; \forall x\,(\neg pen(x) \vee \neg par(x))$
  - $\forall x\,(par(x) \to \neg pen(x)) \;\equiv\; \forall x\,(\neg par(x) \vee \neg pen(x)) \;\equiv\; \forall x\,(\neg pen(x) \vee \neg par(x))$
  - diu (dues vegades) que **no hi ha ningú que parli i pensi alhora**

Introducció i motivació

Perquè estudiar Logic in Computer Science?

- La “deducció intuitiva” que fem nosaltres ens enganya...
  Exemples:
- “1. El que no mata engreixa”. “2. l’enciam no engreixa”.
- Això implica que l’enciam mata?
  - $m =$ “l’enciam mata”
  - $e =$ “l’enciam engreixa”

  1. $\neg m \rightarrow e \;\equiv\; m \vee e$ (totes les coses, o bé maten, o bé engreixen. no hi ha res que ni mata ni engreixa)
     $\equiv\; \neg e \rightarrow m$
  2. $\neg e$
- Així sí implica m: que **l’enciam mata**...

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,. . .)

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.

Conceptes clau apresos en aquesta introducció

- Perquè estudiar *Logic in Computer Science*?
  - Estudi dels Fonaments
  - El llenguatge natural és imprecís, ambiguo
  - La "deducció intuitiva" que fem nosaltres ens enganya
- Què és una lògica?
  - sintaxis: què és una fórmula F?
  - semàntica:
    - què és una interpretació I?
    - quan una I SATISFÀ una F? $I \models F$?
- Aplicacions directes de la lògica en la informàtica
  - Verificació, aplicacions "crítiques", IA (raonament), BD deductives, prolog, ...
  - Ús de lògica per a resoldre problemes combinatoris, optimització, ...

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)

Crèdits

El material utilitzat en aquesta presentació ha estat extret del elaborat pel professor Robert Nieuwenhuis (Dept. CS, UPC) per l'assignatura *Lògica en la Informàtica* de la FIB.

En particular, del llibre *Lógica para informáticos* - Farré, R. [et al.], Marcombo, 2011. ISBN: 9788426716941.

Definició de Lògica Proposicional

- EN QUALSEVOL LÒGICA:
- Què és una lògica? Definició d'una lògica:
  - sintaxi:
    - què és una fórmula F?
  - semàntica:
    - què és una interpretació I?
    - quan una I SATISFÀ una F? notació: $I \models F$
- Fem servir I per a denotar interpretacions i F, G per a fórmules.

EN QUALSEVOL LÒGICA:

- I **és model** de F si I satisfà a F (es denota $I \models F$)
- F és **satisfactible** si F té algun model
- F és **insatisfactible** si F no té models
- 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 $F \models G$)
- F i G són **lògicament equivalents** si F i G tenen el mateixos models (es denota $F \equiv G$)

  Nota: Per definició tenim que $F \equiv G$ ssi $F \models G$ i $G \models F$.

- **Sintaxi**: les fórmules es construeixen amb un conjunto $\mathcal{P}$ de símbols de predicat: $p, q, r, \ldots$ (o "variables" $x_1, x_2, x_3, \ldots$) i les conectives:

$\wedge$	és	AND
$\vee$	és	OR
$\neg$	és	NOT

- Exemple de fòrmula F:

$$p \wedge ((q \vee \neg r) \wedge ((\neg p \vee r) \wedge \neg q))$$

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

- **Semàntica:**

  a) Una interpretació I és una funció $I : \mathcal{P} \to \{0,1\}$.
  Ens diu per cada símbol de $\mathcal{P}$ si és cert o fals.

  b) Quan una I SATISFÀ una F? $\quad I \models F$?
  Quan $eval_I(F) = 1$. Quan l'avaluació en I de F ens dona 1.

    - $eval_I(p) = I(p)$ si p pertany a $\mathcal{P}$, si p és un símbol de predicat (del conjunt $\mathcal{P}$)
    - $eval_I(\neg F) \quad = 1 - eval_I(F)$
    - $eval_I(F \wedge G) = min(eval_I(F), eval_I(G))$
    - $eval_I(F \vee G) = max(eval_I(F), eval_I(G))$

- Donada una I, per exemple $I(p) = 1$, $I(q) = 0$, $I(r) = 1$, i una F com
  $$p \wedge ((q \vee \neg r) \wedge ((\neg p \vee r) \wedge \neg q))$$
  quin és el cost de decidir si I és model de F? quant costa calcular $eval_I(F)$? És lineal!

Definició de Lògica Proposicional

- En qualsevol lògica:
  - I **és model** de F si I satisfà a F ($I \models F$)
  - F és **satisfactible** si F té algun model
  - F és **insatisfactible** si F no té models
  - 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 $F \models G$)
  - F i G són **lògicament equivalents** si F i G tenen el mateixos models (es denota $F \equiv G$)
- En Lògica Proposicional:
  - Quan I **satisfà** F? ($I \models F$) Quan $eval_I(F) = 1$.
  - $eval_I(p) = I(p)$, si $p \in \mathcal{P}$
  - $eval_I(\neg F) \quad = 1 - eval_I(F)$
  - $eval_I(F \land G) = min(eval_I(F), eval_I(G))$
  - $eval_I(F \lor G) = max(eval_I(F), eval_I(G))$

Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts: ☞ `p2.pdf`

1. Exercici 1 [interpretacions en LProp]
2. Exercici 2 [demostració en LProp]
3. Exercici 6 [demostració en LProp]

Exercici 1

1. (dificultat 1) Quantes interpretacions possibles hi ha en funció de $|\mathcal{P}|$?
(nota: si S és un conjunt, $|S|$ denota la seva cardinalitat, és a dir, el nombre d'elements de S).

Exercici 1

1. (dificultat 1) Quantes interpretacions possibles hi ha en funció de $|\mathcal{P}|$?

Podem fer la llista de totes els possibles I's (aquesta llista també és diu "taula de veritat"):
Per exemple, si $\mathcal{P} = \{p, q, r\}$ i $F = p \wedge ((q \vee \neg r) \wedge ((\neg p \vee r) \wedge \neg q))$

p	q	r	F
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**0**
1	1	0	**0**
1	1	1	**0**

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

En la pràctica, es fa SAT on la F donada és una *CNF* (*conjunctive normal form*): una fórmula que és un conjunt (ANDs) de clàusules (ORs de literals), on un literal és una variable (literal positiu) o una variable negada (literal negatiu).

Exercici 2

2. (dificultat 1) Demostra que $p \wedge \neg p$ és insatisfactible fent servir tan sols la definició de la LProp.

$p \wedge \neg p$ és insatisfactible	ssi	[per definició de insatisfactible]
$p \wedge \neg p$ no té cap model	ssi	[per definició de model]
$\forall I,\ I \not\models p \wedge \neg p$	ssi	[per definició de $\models$]
$\forall I,\ eval_I(p \wedge \neg p) = 0$	ssi	[per definició de $eval_I(\ldots \wedge \ldots)$]
$\forall I,\ min(eval_I(p), eval_I(\neg p)) = 0$	ssi	[per definició de $eval_I(\neg \ldots)$]
$\forall I,\ min(eval_I(p), 1 - eval_I(p)) = 0$	ssi	[per definició de $eval_I(p)$]
$\forall I,\ min(I(p), 1 - I(p)) = 0$	ssi	[donat que $I(p)$ sempre és 0 o 1, i per definició de min]
$0 = 0$	ssi	[perquè $0 = 0$ és cert]
cert		

Exercici 6

6. (dificultat 2) Demostra (fent servir tan sols la definició de la LProp) que, per a tota fórmula F,

F és tautologia ssi $\neg F$ és insatisfactible.

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

A) F és tautologia $\implies \ldots \implies \neg F$ és insatisfactible

B) $\neg F$ és insatisfactible $\implies \ldots \implies$ F és tautologia

En aquest cas farem una cadena de SSIs:

Exercici 6

6. (dificultat 2) Demostra (fent servir tan sols la definició de la LProp) que, per a tota fórmula F,

F és tautologia ssi $\neg F$ és insatisfactible.

F és tautologia	ssi	[per definició de tautologia]
$\forall I,\ I$ és model de F	ssi	[per definició de model]
$\forall I,\ I \models F$	ssi	[per definició de $\models$]
$\forall I,\ eval_I(F) = 1$	ssi	[per aritmètica]
$\forall I,\ 1 - eval_I(F) = 0$	ssi	[per avaluació d'un not]
$\forall I,\ eval_I(\neg F) = 0$	ssi	[per definició de $\models$]
$\forall I,\ I \not\models \neg F$	ssi	[per definició de model]
$\forall I,\ I$ no és model de $\neg F$	ssi	[per definició de insatisfactible]
$\neg F$ és insatisfactible		

Definició de Lògica Proposicional

Per al proper dia de classe:

☞ Capítol 2 dels apunts: exercicis del 7 en endavant:

5, 7, 8, 16, 21, 18, 23