Lògica en la Informàtica
Definició de Lògica Proposicional (LProp)

José Miguel Rivero

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

Primavera 2025

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.

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)

Definició de Lògica Proposicional

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

1. Exercici 5 [demostració en LProp]
2. Exercici 7 [demostració en LProp]
3. Exercici 8 [demostració en LProp]
4. Exercici 16 [demostració en LProp]
5. Exercici 21 [equivalència lògica]
6. Exercici 18 [equivalències entre fòrmules]
7. Exercici 23 [lema de Substitució]

Exercici 5

5. (dificultat 1) Siguin F i G dues fórmules qualssevol. És cert que $F \vee G$ és tautologia si i només si alguna de les dues fórmules F o G ho és?. Demostra-ho fent servir només la definició de la LProp.

No, no és cert. Donarem un **contraexemple** de fórmules F i G per a les quals la propietat és falsa, és a dir, on $F \vee G$ és tautologia, però ni F ni G són tautologies.

Sigui F la fórmula p, on p és un símbol de predicat.
Sigui G la fórmula $\neg p$.

Exercici 5

5. (dificultat 1) Siguin F i G dues fórmules qualssevol. És cert que $F \vee G$ és tautologia si i només si alguna de les dues fórmules F o G ho és?. Demostra-ho fent servir només la definició de la LProp.

a) Tenim que $p \vee \neg p$ és tautologia perquè:

$p \vee \neg p$ és tautologia	ssi	[per definició de tautologia]
tota I és model de $p \vee \neg p$	ssi	[per definició de model]
$\forall I,\ I \models p \vee \neg p$	ssi	[per definició de $\models$]
$\forall I,\ eval_I(p \vee \neg p) = 1$	ssi	[per definició de $eval_I(\ldots \vee \ldots)$]
$\forall I,\ max(eval_I(p), eval_I(\neg p)) = 1$	ssi	[per definició de $eval_I(\neg \ldots)$]
$\forall I,\ max(eval_I(p), 1 - eval_I(p)) = 1$	ssi	[donat que $eval_I(p)$ sempre és 0 o 1, i per definició de max]

$$max(0, 1-0) = max(1, 1-1) = 1$$

Exercici 5

5. (dificultat 1) Siguin F i G dues fórmules qualssevol. És cert que $F \vee G$ és tautologia si i només si alguna de les dues fórmules F o G ho és?. Demostra-ho fent servir només la definició de la LProp.

a) Tenim que $p \vee \neg p$ és tautologia perquè:

$p \vee \neg p$ és tautologia	ssi	[per definició de tautologia]
tota I és model de $p \vee \neg p$	ssi	[per definició de model]
$\forall I,\ I \models p \vee \neg p$	ssi	[per definició de $\models$]
$\forall I,\ eval_I(p \vee \neg p) = 1$	ssi	[per definició de $eval_I(\ldots \vee \ldots)$]
$\forall I,\ max(eval_I(p), eval_I(\neg p)) = 1$	ssi	[per definició de $eval_I(\neg \ldots)$]
$\forall I,\ max(eval_I(p), 1 - eval_I(p)) = 1$	ssi	[donat que $eval_I(p)$ sempre és 0 o 1, i per definició de max]
$1 = 1$	ssi	[perquè $1 = 1$ és *cert*]
cert		

Exercici 5

5. (dificultat 1) Siguin F i G dues fórmules qualssevol. És cert que $F \vee G$ és tautologia si i només si alguna de les dues fórmules F o G ho és?. Demostra-ho fent servir només la definició de la LProp.

b) Però tenim que ni F ni G són tautologies:

F no és tautologia	ssi	[com F és p]
p no és tautologia	ssi	[per definició de tautologia]
$\exists I$, tq I no és model de p	ssi	[per definició de model]
$\exists I$, $I \not\models p$	ssi	[per definició de $\models$]
$\exists I$, $eval_I(p) = 0$	ssi	[per definició de $eval_I(p)$]
$\exists I$, $I(p) = 0$	ssi	cert

Exercici 5

5. (dificultat 1) Siguin F i G dues fórmules qualssevol. És cert que $F \vee G$ és tautologia si i només si alguna de les dues fórmules F o G ho és?. Demostra-ho fent servir només la definició de la LProp.

b) Però tenim que ni F ni G són tautologies:

G no és tautologia	ssi	[com G és $\neg p$]
$\neg p$ no és tautologia	ssi	[per definició de tautologia]
$\exists\, I$, tq I no és model de $\neg p$	ssi	[per definició de model]
$\exists\, I$, $I \not\models \neg p$	ssi	[per definició de $\models$]
$\exists\, I$, $eval_I(\neg p) = 0$	ssi	[per definició de $eval_I(\neg \ldots)$]
$\exists\, I$, $1 - eval_I(p) = 0$	ssi	[per definició de $eval_I(p)$]
$\exists\, I$, $1 - I(p) = 0$	ssi	[per aritmètica]
$\exists\, I$, $I(p) = 1$	ssi	cert

Exercici 7

7. (dificultat 2) Siguin F i G dues fórmules qualssevol. Demostra, fent servir tan sols la definició de la LProp, que F és conseqüència lògica de G, és a dir, $G \models F$ si i només si $G \wedge \neg F$ és insatisfactible.

F és conseqüència lògica de G	ssi	[per def. de conseqüència lògica]
tot model de G satisfà F	ssi	[per def. de model]
$\forall I$, si $I \models G$, llavors $I \models F$	ssi	[pel significat de si. . . llavors. . .]
$\forall I$, $I \not\models G$ o bé $I \models F$	ssi	[per def. de $\models$]
$\forall I$, $eval_I(G) = 0$ o bé $eval_I(F) = 1$		

$G \wedge \neg F$ és insatisfactible

Exercici 7

7. (dificultat 2) Siguin F i G dues fórmules qualssevol. Demostra, fent servir tan sols la definició de la LProp, que F és conseqüència lògica de G, és a dir, $G \models F$ si i només si $G \wedge \neg F$ és insatisfactible.

F és conseqüència lògica de G	ssi	[per def. de conseqüència lògica]
tot model de G satisfà F	ssi	[per def. de model]
$\forall I$, si $I \models G$, llavors $I \models F$	ssi	[pel significat de si...llavors...]
$\forall I$, $I \not\models G$ o bé $I \models F$	ssi	[per def. de $\models$]
$\forall I$, $eval_I(G) = 0$ o bé $eval_I(F) = 1$	ssi	[per aritmètica]
$\forall I$, $eval_I(G) = 0$ o bé $1 - eval_I(F) = 0$	ssi	[per def. de *min*]
$\forall I$, $min(eval_I(G), 1 - eval_I(F)) = 0$	ssi	[per def. de $eval_I(\neg \ldots)$]
$\forall I$, $min(eval_I(G), eval_I(\neg F)) = 0$	ssi	[per def. de $eval_I(\ldots \wedge \ldots)$]
$\forall I$, $eval_I(G \wedge \neg F) = 0$	ssi	[per def. de $\models$]
$\forall I$, $I \not\models (G \wedge \neg F)$	ssi	[per def. de model]
$G \wedge \neg F$ no té models	ssi	[per def. de insatisfactible]
$G \wedge \neg F$ és insatisfactible		

8. (dificultat 2) Siguin F i G dues fórmules qualssevol. Demostra, fent servir tan sols la definició de la LProp, que F és lògicament equivalent a G ssi $(G \wedge \neg F) \vee (F \wedge \neg G)$ és insatisfactible ssi $F \leftrightarrow G$ és tautologia.

Fem primer el primer ssi:

$(G \wedge \neg F) \vee (F \wedge \neg G)$ és insatisfactible

ssi [per definició de insatisfactible]

$\forall I,\ I \not\models (G \wedge \neg F) \vee (F \wedge \neg G)$

ssi [per definició de $\models$]

$\forall I,\ eval_I((G \wedge \neg F) \vee (F \wedge \neg G)) = 0$

ssi [per definició de *eval* $\vee$]

$\forall I,\ max(eval_I(G \wedge \neg F), eval_I(F \wedge \neg G)) = 0$

ssi [per definició de *eval* $\wedge$]

$\forall I,\ max(min(eval_I(G), eval_I(\neg F)), min(eval_I(F), eval_I(\neg G))) = 0$

8. (dificultat 2) Siguin F i G dues fórmules qualssevol. Demostra, fent servir tan sols la definició de la LProp, que F és lògicament equivalent a G ssi $(G \wedge \neg F) \vee (F \wedge \neg G)$ és insatisfactible ssi $F \leftrightarrow G$ és tautologia.

Fem primer el primer ssi (cont.):

$\forall I,\ max(min(eval_I(G), eval_I(\neg F)), min(eval_I(F), eval_I(\neg G))) = 0$

ssi [per definició de $eval$ $\neg$]

$\forall I,\ max(min(eval_I(G), 1 - eval_I(F)), min(eval_I(F), 1 - eval_I(G))) = 0$

ssi [per definició de min i max,
i perquè $eval$ sempre dona 0 o 1]

$\forall I,\ eval_I(F) = eval_I(G)$

ssi [perquè $eval$ sempre dona 0 o 1]

$\forall I,\ (eval_I(F) = 1 \text{ ssi } eval_I(G) = 1)$

ssi [per definició de $\models$]

$\forall I,\ (I \models F \text{ ssi } I \models G)$

Exercici 8

8. (dificultat 2) Siguin F i G dues fórmules qualssevol. Demostra, fent servir tan sols la definició de la LProp, que F és lògicament equivalent a G ssi $(G \wedge \neg F) \vee (F \wedge \neg G)$ és insatisfactible ssi $F \leftrightarrow G$ és tautologia.

Fem primer el primer ssi (cont.):

$\forall I$, ($I \models F$ ssi $I \models G$)

ssi [per definició de model]

$\forall I$, (I és model de F ssi I és model de G)

ssi [per definició de equivalència de models]

F i G tenen els mateixos models

ssi [per definició de equivalència lògica]

F és lógicamente equivalente a G.

Exercici 8

8. (dificultat 2) Siguin F i G dues fórmules qualssevol. Demostra, fent servir tan sols la definició de la LProp, que F és lògicament equivalent a G ssi $(G \land \neg F) \lor (F \land \neg G)$ és insatisfactible ssi $F \leftrightarrow G$ és tautologia.

Per al segon ssi:

$F \leftrightarrow G$ és tautologia

ssi [per def. $\leftrightarrow$]

$(F \to G) \land (G \to F)$ és tautologia

ssi [per def. $\to$]

$(\neg F \lor G) \land (\neg G \lor F)$ és tautologia

ssi [per def. de tautologia]

$\forall I$ és model de $(\neg F \lor G) \land (\neg G \lor F)$

ssi [per def. de model]

$\forall I,\ I \models (\neg F \lor G) \land (\neg G \lor F)$

ssi [per def. de $\models$]

$\forall I,\ eval_I(\ (\neg F \lor G) \land (\neg G \lor F)\ ) = 1$

Exercici 8

8. (dificultat 2) Siguin F i G dues fórmules qualssevol. Demostra, fent servir tan sols la definició de la LProp, que F és lògicament equivalent a G ssi $(G \wedge \neg F) \vee (F \wedge \neg G)$ és insatisfactible ssi $F \leftrightarrow G$ és tautologia.

Per al segon ssi (cont.):

$\forall I,\ eval_I(\ (\neg F \vee G) \wedge (\neg G \vee F)\ ) = 1$

ssi [per def. de $eval_I(\ \wedge\ )$]

$\forall I,\ min(eval_I(\neg F \vee G), eval_I(\neg G \vee F)) = 1$

ssi [per def. de $eval_I(\ \vee\ )$]

$\forall I,\ min(max(eval_I(\neg F), eval_I(G)), max(eval_I(\neg G), eval_I(F))) = 1$

ssi [per def. de $eval_I(\ \neg\ )$]

$\forall I,\ min(max(1 - eval_I(F), eval_I(G)), max(1 - eval_I(G), eval_I(F))) = 1$

ssi [per definició de *min* i *max*,
i perquè $eval_I$ sempre dona 0 o 1]

$\forall I,\ eval_I(F) = eval_I(G)$

i seguim igual que en la demostració anterior.

Definició de Lògica Proposicional

Conseqüència dels exercicis 6,7,8

En la pràctica ens interessa sempre, esbrinar aquest tipus de propietats:

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 $F \equiv G$)

Com ho podem fer, si l'única cosa que tenim és un SAT solver?

F és tautologia	ssi	$\neg F$ és insatisfactible.
G és conseqüència lògica de F	ssi	$F \land \neg G$ és insatisfactible.
F i G són lògicament equivalentes	ssi	$(G \land \neg F) \lor (F \land \neg G)$ és insatisfactible.

Exercici 16

16. (dificultat 2) Siguin F i G dues fórmules qualssevol. Si $F \to G$ és satisfactible i F és satisfactible, llavors G és satisfactible? Demostra-ho fent servir tan sols la definició de la LProp.

Farem un *intent de demostració*:

$F \to G$ és satisfactible	ssi	[per def. de $\to$]
$\neg F \vee G$ és satisfactible	ssi	[per def. de satisfactible]
$\neg F \vee G$ té algun model	ssi	[per def. de model]
$\exists\, I,\ I \models \neg F \vee G$	ssi	[per def. de $\models$]
$\exists\, I,\ eval_I(\neg F \vee G) = 1$	ssi	[per def. de $eval_I(\ \vee\ )$]
$\exists\, I,\ max(eval_I(\neg F), eval_I(G)) = 1$	ssi	[per def. de $eval_I(\ \neg\ )$]
$\exists\, I,\ max(1 - eval_I(F), eval_I(G)) = 1$	ssi	[per def. de max]
$\exists\, I,\ 1 - eval_I(F) = 1$ o bé $eval_I(G) = 1$	ssi	[per aritmètica]
$\exists\, I,\ eval_I(F) = 0$ o bé $eval_I(G) = 1$		

NO puc escriure $\exists\, I,\ eval_I(F) = 0 \vee eval_I(G) = 1$
NO té cap sentit, perquè $\vee$ és una connectiva que només té sentit dintre de fórmules, i $eval_I(F) = 0$ NO és una fórmula. Aquí estem raonant/explicant en català.

Exercici 16

16. (dificultat 2) Siguin F i G dues fórmules qualssevol. Si $F \rightarrow G$ és satisfactible i F és satisfactible, llavors G és satisfactible? Demostra-ho fent servir tan sols la definició de la LProp.

Farem un *intent de demostració (cont)*:

F és satisfactible	ssi	[per def. de satisfactible]
F té algun model	ssi	[per def. de model]
$\exists\, I',\ I' \models F$	ssi	[per def. de $\models$]
$\exists\, I',\ eval_{I'}(F) = 1$		

D'aquest intent de demostració, veiem que si existeix una I que no és model de F i un altre I' que sí és model de F, llavors ja és compleixen les dues condicions de que $F \rightarrow G$ és satisfactible i F és satisfactible, i això no implica res sobre la G!

Exercici 16

16. (dificultat 2) Siguin F i G dues fórmules qualssevol. Si $F \to G$ és satisfactible i F és satisfactible, llavors G és satisfactible? Demostra-ho fent servir tan sols la definició de la LProp.

> D'aquest intent de demostració, veiem que si existeix una I que no és model de F i un altre I' que sí és model de F, llavors ja és compleixen les dues condicions de que $F \to G$ és satisfactible i F és satisfactible, i això no implica res sobre la G!

Això ens inspira per a adonar-nos que la propietat és falsa, i per a donar aquest contraexemple:

> Sigui F la fórmula p
>
> Sigui G la fórmula $p \wedge \neg p$.

Llavors $F \to G$ és satisfactible i F és satisfactible, però G no ho és!

Exercici 16

16. (dificultat 2) Siguin F i G dues fórmules qualssevol. Si $F \rightarrow G$ és satisfactible i F és satisfactible, llavors G és satisfactible? Demostra-ho fent servir tan sols la definició de la LProp.

$F \rightarrow G$ és satisfactible	ssi	[per def. de F i G]
$p \rightarrow (p \wedge \neg p)$ és satisfactible	ssi	[per def. de $\rightarrow$]
$\neg p \vee (p \wedge \neg p)$ és satisfactible	ssi	[per def. de satisfactible]
$\neg p \vee (p \wedge \neg p)$ té algun model	ssi	[per def. de model]
$\exists\, I,\ I \models \neg p \vee (p \wedge \neg p)$	ssi	[per def. de $\models$]
$\exists\, I,\ eval_I(\neg p \vee (p \wedge \neg p)) = 1$	ssi	[per def. de $eval_I(\ \vee\ )$]
$\exists\, I,\ max(eval_I(\neg p), eval_I(p \wedge \neg p)) = 1$	ssi	[per def. de $eval_I(\ \neg\ )$]
$\exists\, I,\ max(1 - eval_I(p), eval_I(p \wedge \neg p)) = 1$	ssi	[per def. de max]
$\exists\, I,\ 1 - eval_I(p) = 1$ o $eval_I(p \wedge \neg p) = 1$	ssi	[per aritmètica]
$\exists\, I,\ eval_I(p) = 0$ o $eval_I(p \wedge \neg p) = 1$	ssi	[agafem la I tal que $I(p) = 0$]

cert

Exercici 16

16. (dificultat 2) Siguin F i G dues fórmules qualssevol. Si $F \to G$ és satisfactible i F és satisfactible, llavors G és satisfactible? Demostra-ho fent servir tan sols la definició de la LProp.

F és satisfactible	ssi	[per def. de F]
p és satisfactible	ssi	[per def. de satisfactible]
p té algun model	ssi	[per def. de model]
$\exists\, I',\ I' \models p$	ssi	[per def. de $\models$]
$\exists\, I',\ eval_{I'}(p) = 1$	ssi	[agafem la I' tal que $I'(p) = 1$]
cert		

G és insatisfactible (☞ veure exercici 2).

Exercici 21

21. (dificultat 2) Demostra que l'equivalència lògica és realment una relació d'**equivalència**.

Una relacion binària R sobre un conjunt S és un subconjunt del producte cartesià $S \times S$. És a dir R ens diu quines parelles estan relacionades, quines parelles (e, e') estan en R (on e i e' són elements de S).

R és **reflexiva** si (e, e) està en R per a tot e de S.
R és **simètrica** si (e, e') en R implica (e', e) en R per a tot e,e' de S.
R és **transitiva** si (e, e') en R i (e', e'') en R implica (e, e'') en R per a tot e,e',e'' de S.

I si R compleix les tres propietats llavors R és una relació d'**equivalència**.

Exercici 21

21. (dificultat 2) Demostra que l'equivalència lògica és realment una relació d'**equivalència**.

Una relacion binària R sobre un conjunt S és un subconjunt del producte cartesià $S \times S$. És a dir R ens diu quines parelles estan relacionades, quines parelles (e, e') estan en R (on e i e' són elements de S).

Altres notacions:

- com un predicat binari:

R és **reflexiva** si $R(e, e)$		per a tot e
R és **simètrica** si $R(e, e')$	implica $R(e', e)$	per a tot e,e'
R és **transitiva** si $R(e, e')$ i $R(e', e'')$	implica $R(e, e'')$	per a tot e,e',e''

Exercici 21

21. (dificultat 2) Demostra que l'equivalència lògica és realment una relació d'**equivalència**.

Una relacion binària R sobre un conjunt S és un subconjunt del producte cartesià $S \times S$. És a dir R ens diu quines parelles estan relacionades, quines parelles (e, e') estan en R (on e i e' són elements de S).

Altres notacions:

- com un predicat infix:

R és **reflexiva** si eRe		per a tot e	de S.
R és **simètrica** si eRe'	implica $e'Re$	per a tot e,e'	de S.
R és **transitiva** si eRe' i $e'Re''$	implica eRe''	per a tot e,e',e''	de S.

Per exemple si R és $>$, la notació infixa és molt més habitual: escribim $e > e'$, etc.

Exercici 18

18. (dificultat 2) Demostra les següents equivalències entre fórmules:

$F \wedge F$	$\equiv$	F	idempotència de $\wedge$
$F \vee F$	$\equiv$	F	idempotència de $\vee$
$F \wedge G$	$\equiv$	$G \wedge F$	conmutativitat de $\wedge$
$F \vee G$	$\equiv$	$G \vee F$	conmutativitat de $\vee$
$(F \wedge G) \wedge H$	$\equiv$	$F \wedge (G \wedge H)$	associativitat de $\wedge$
$(F \vee G) \vee H$	$\equiv$	$F \vee (G \vee H)$	associativitat de $\vee$

Aquestes tres propietats (idempotència, commutativitat, asociatividad de $\wedge$ i de $\vee$) ens indiquen que a vegades podem escriure les fórmules de manera més "relaxada", ometent alguns parèntesis. I també, que podem veure una CNF com un CONJUNT (and) de clàusules, i podem veure una clàusula com un CONJUNT (un or) de literals.

Exercici 18

18. (dificultat 2) Demostra les següents equivalències entre fórmules:

$$\begin{array}{lcll} \neg\neg F & \equiv & F & \text{doble negació} \\ \neg(F \wedge G) & \equiv & \neg F \vee \neg G & \text{llei de De Morgan 1} \\ \neg(F \vee G) & \equiv & \neg F \wedge \neg G & \text{llei de De Morgan 2} \end{array}$$

Aquestes tres propietats ens serveixen per a transformar fórmules "movent les negacions cap a dins", fins que només hi hagi negacions aplicades a símbols de predicat.

Exercici 18

18. (dificultat 2) Demostra les següents equivalèncias entre fórmules:

$$\begin{array}{rcll} (F \land G) \lor H & \equiv & (F \lor H) \land (G \lor H) & \text{distributivitat 1} \\ (F \lor G) \land H & \equiv & (F \land H) \lor (G \land H) & \text{distributivitat 2} \end{array}$$

Una vegada les negacions estan aplicades als símbols de predicat, aplicant distributivitat 1 $(F \land G) \lor H \Longrightarrow (F \lor H) \land (G \lor H)$ d'esquerra a dreta obtenim una CNF.

Hi ha un detall: Demostra que $p \land (q \lor q) \equiv p \land q$.
Podem "aplicar" alegrement la idempotència del $\lor$ sobre la subfórmula $q \lor q$?
No! Cal demostrar primer el Lema de Substitució de l'exercici 23.

Exercici 18

Demostrem, com a exemple, que $F \equiv \neg\neg F$:

$F \equiv \neg\neg F$

ssi [per def. d'equivalència lògica]

$\forall I$, (I és model de F ssi I és model de $\neg\neg F$)

ssi [per def. de model]

$\forall I, (I \models F \text{ ssi } I \models \neg\neg F)$

ssi [per def. de satisfacció]

$\forall I, (eval_I(F) = 1 \text{ ssi } eval_I(\neg\neg F) = 1)$

ssi [perquè $eval_I$ sempre dona 0 o 1]

$\forall I, eval_I(F) = eval_I(\neg\neg F)$

ssi [per def. de $eval_I(\neg)$]

$\forall I, eval_I(F) = 1 - eval_I(\neg F)$

ssi [per def. de $eval_I(\neg)$]

$\forall I, eval_I(F) = 1 - (1 - eval_I(F))$

ssi [per aritmètica]

$\forall I, eval_I(F) = eval_I(F)$

ssi

true

Exercici 23

23. (dificultat 3) Lema de Substitució.

Siguin F, G, G' fórmules qualssevol, amb $G \equiv G'$.
Si en F substituïm una aparició de una subfórmula G per G' obtenim una nova fórmula F' amb $F \equiv F'$.

En el exemple anterior:

$$\begin{array}{ll} F \text{ és} & p \wedge (q \vee q) \\ G \text{ és} & \phantom{p \wedge{}} (q \vee q) \\ G' \text{ és} & q \\ F' \text{ és} & p \wedge q\ . \end{array}$$

Definició de Lògica Proposicional

Per al proper dia de classe:

- ☞ Capítol 2 dels apunts:
  exercicis 26, 27, 28, 31, 32, 33, 34, 35, 36, 37.
- ☞ Capítol 3 dels apunts: p3.pdf

Lògica en la Informàtica
Definició de Lògica Proposicional (LProp)
Deducció en Lògica Proposicional

José Miguel Rivero

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

Primavera 2025

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.

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)

Definició de Lògica Proposicional

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

1. Exercici 26 [cost de SAT]
2. Exercici 27 [funcions booleanes]
3. Exercici 28 [funcions booleanes]
4. Exercici 31 [funcions booleanes]
5. Exercici 32 [funcions booleanes]
6. Exercici 37 [simplificació de codi]
7. Exercici 33 [formalització i raonament]
8. Exercici 34 [altres lògiques]
9. Exercici 35 [altres lògiques]

Deducció en Lògica Proposicional

Capítol 3 dels apunts: ☞ `p3.pdf`

Exercici 26

26. (dificultat 2) Suposem que $|\mathcal{P}| = 100$ i que ens interessa determinar si una fórmula F construïda sobre $\mathcal{P}$ és satisfactible o no. Si l'algorisme està basat en una anàlisi de la taula de veritat, i avaluar F en una interpretació I donada costa un microsegon (10^{-6} segons), quants anys trigarà?
Més endavant veurem tècniques que moltes vegades funcionen millor.

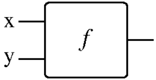
Quantes interpretacions hi ha? hi ha 2^{100}

$$\begin{array}{ll} 2^{10} = 1024 & \approx 10^3 \\ 2^{100} & \approx 10^{30} \end{array}$$

Avaluar-les totes trigarà $2^{100} \cdot 10^{-6}$ segons $\approx 10^{30} \cdot 10^{-6}$ segons $\approx 10^{24}$ segons $\approx 10^{24}/(365 \cdot 24 \cdot 3600)$ anys $\approx 4 \cdot 10^{16}$ anys aprox.

Exercici 27

27. (dificultat 2) Una funció booleana de n entrades és una funció $f : \{0,1\}^n \to \{0,1\}$, és a dir, una funció que prem com entrada una cadena de n bits i retorna un bit. Quantes funcions booleanes de n entrades hi ha?



Hi ha 2^{2^n}: 2 elevat a (2 elevat a n): hi ha tantes funcions com tires de 2^n bits

Exercici 27

27. (dificultat 2) Una funció booleana de n entrades és una funció $f : \{0,1\}^n \rightarrow \{0,1\}$, és a dir, una funció que prem com entrada una cadena de n bits i retorna un bit. Quantes funcions booleanes de n entrades hi ha?

Hi ha 2^{2^n}: 2 elevat a (2 elevat a n): hi ha tantes funcions com tires de 2^n bits

Exemple $n = 2$: tantes funcions com tires de 2^n bits = tires de 4 bits = 2^4

x	y	0	and	$\neg(x \to y)$	x	$\neg(y \to x)$	y	xor	or	$\cdots$
0	0	0	0	0	0	0	0	0	0	
0	1	0	0	0	0	1	1	1	1	
1	0	0	0	1	1	0	0	1	1	
1	1	0	1	0	1	0	1	0	1	

x	y	$\cdots$	nor	$=$	$\neg y$	$y \to x$	$\neg x$	$x \to y$	nand	1
0	0		1	1	1	1	1	1	1	1
0	1		0	0	0	0	1	1	1	1
1	0		0	0	1	1	0	0	1	1
1	1		0	1	0	1	0	1	0	1

Exercici 27

27. (dificultat 2) Una funció booleana de n entrades és una funció $f : \{0,1\}^n \to \{0,1\}$, és a dir, una funció que prem com entrada una cadena de n bits i retorna un bit. Quantes funcions booleanes de n entrades hi ha?

Hi ha 2^{2^n}: 2 elevat a (2 elevat a n): hi ha tantes funcions com tires de 2^n bits

Exemple $n = 3$: hi ha $2^8 = 256$

x	y	z
0	0	0
0	0	1
	$\dots$	
1	1	1

Si $n = 4$, hi ha $2^{64} \approx 65000$

Exercici 28

28. (dificultat 2) Cada fórmula F representa una única funció booleana: la que retorna 1 exactament per a aquellas cadenes de bits I tals que $eval_I(F) = 1$. Per això, dues fórmules són lògicament equivalents si i només si representen la mateixa funció booleana. Quantes funcions booleanes (o quantes fórmules lògicamente no-equivalents) hi ha en funció de $n = |\mathcal{P}|$?

Per exemple, la funció booleana "and" (de 2 entrades x, y) la podem representar mitjançant les fórmules

$x \wedge y$
$(x \wedge y) \wedge y$
$\neg(\neg x \vee \neg y)$
$\neg(\neg x \vee \neg y) \wedge y$
$\ldots$

Exercici 28

28. (dificultat 2) Cada fórmula F representa una única funció booleana: la que retorna 1 exactament per a aquellas cadenes de bits I tals que $eval_I(F) = 1$. Per això, dues fórmules són lògicament equivalents si i només si representen la mateixa funció booleana. Quantes funcions booleanes (o quantes fórmules lògicamente no-equivalents) hi ha en funció de $n = |\mathcal{P}|$?

Hi ha 2^{2^n} [2 elevat a (2 elevat a n)]: hi ha tantes funcions com tires de 2^n bits

Exercici 31

31. (dificultat 3) Escriu en una taula de veritat les 16 funcions booleanes de 2 entrades. Quantes de elles només depenen d'una de las dues entrades? Quantes depenen de zero entrades? Les altres, vistes com conectives lògicas, reben algun nom?

x	y	0	and	$\neg(x \to y)$	x	$\neg(y \to x)$	y	xor	or	$\cdots$
0	0	0	0	0	0	0	0	0	0	
0	1	0	0	0	0	1	1	1	1	
1	0	0	0	1	1	0	0	1	1	
1	1	0	1	0	1	0	1	0	1	

x	y	$\cdots$	nor	$=$	$\neg y$	$y \to x$	$\neg x$	$x \to y$	nand	1
0	0		1	1	1	1	1	1	1	1
0	1		0	0	0	0	1	1	1	1
1	0		0	0	1	1	0	0	1	1
1	1		0	1	0	1	0	1	0	1

Exercici 31

31. (dificultat 3) Escriu en una taula de veritat les 16 funcions booleanes de 2 entrades. Quantes de elles només depenen d'una de las dues entrades? Quantes depenen de zero entrades? Les altres, vistes com conectives lògicas, reben algun nom? Ja sabem que podem expressar qualsevol funció booleana amb el conjunt de tres connectives $\{\wedge, \vee, \neg\}$, és a dir, qualsevol funció booleana és equivalent a una fórmula construïda sobre aquestes tres connectives. És cert això també para algún conjunto de només dues de les 16 funcions? (Hi ha diverses maneres, però basta amb donar una sola.)

Sí, amb només *or* i *not*, per exemple: $x \wedge y \equiv \neg(\neg x \vee \neg y)$

Exercici 32

32. (dificultat 3) Demostra que qualsevol funció booleana de dues entrades es pot expressar amb només *nor* o bé amb només *nand*, on $nor(F, G)$ és $\neg(F \vee G)$, i $nand(F, G)$ és $\neg(F \wedge G)$.

Ho fem per *nand*:

$$\neg F \equiv F \; nand \; F$$

$$F \vee G \equiv \neg(\neg F \wedge \neg G) \equiv \neg F \; nand \; \neg G \equiv (F \; nand \; F) \; nand \; (G \; nand \; G)$$

$$F \wedge G \equiv \neg(F \; nand \; G) \equiv (F \; nand \; G) \; nand \; (F \; nand \; G)$$

Exercici 37

37. (dificultat 3) Considera el següent fragment de codi, que retorna un booleà:

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

Simplifíca'l sustituïnt els valors de retorn per un sol valor de retorn que sigui una expressió booleana en $i > 0$, a i b:

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

Exercici 37

37. (dificultat 3) Considera el següent fragment de codi, que retorna un booleà:

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

$$\text{if\_then\_else}(C, F, G) \equiv (C \rightarrow F) \wedge (\neg C \rightarrow G)$$

Exercici 37

37. (dificultat 3) Considera el següent fragment de codi, que retorna un booleà:

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

Tindrem tres símbols de predicat: a, b, $i > 0$.

a	b	$i > 0$	return	
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)

```
return ((not a) and (not b) and i>0) or
       (a and b and i>0);
```

```
return (a==b and i>0);
```

Exercici 33

33. (dificultat 3) Tres estudiants A, B i C són acusats d'introduir un virus en les sales d'ordinadors de la FIB. Durant l'interrogatori, les declaracions són les següents:

- A diu: "B ho va fer i C és innocent"
- B diu: "Si A és culpable llavors C també ho és"
- C diu: "Jo no ho vaig fer, ho va fer almenys un dels altres dos"

a) Són les tres declaracions contradictories?

b) Assumint que tots son innocents, qui o quins van mentir en la declaració?

c) Assumint que ningún va mentir, qui és innocent i qui és culpable?

Exercici 33

33. (dificultat 3) Tres estudiants A, B i C són acusats d'introduir un virus en les sales d'ordinadors de la FIB. Durant l'interrogatori, les declaracions són les següents:

- A diu: "B ho va fer i C és innocent"
- B diu: "Si A és culpable llavors C també ho és"
- C diu: "Jo no ho vaig fer, ho va fer almenys un dels altres dos"

Introduïm símbols de predicat: a,b,c que signifiquen: "A ho va fer", "B ho va fer", "C ho va fer".

Les tres declaracions són:

- A diu: $b \wedge \neg c$
- B diu: $a \rightarrow c$
- C diu: $\neg c \wedge (a \vee b)$

Exercici 33

33. (dificultat 3) Tres estudiants A, B i C són acusats d'introduir un virus en les sales d'ordinadors de la FIB. Durant l'interrogatori, les declaracions són les següents:

Les tres declaracions són:

- A diu: $b \land \neg c$
- B diu: $a \to c$
- C diu: $\neg c \land (a \lor b)$

a) Són les tres declaracions contradictories?

Per saber si poden ser veritat les tres declaracions, formalment, hem de veure si és satisfactible la conjunció (*and*) de les tres fórmules.

Només hi ha un model I: $I(a) = 0$, $I(b) = 1$, $I(c) = 0$.
Per tant, no són contradictòries.

Exercici 33

33. (dificultat 3) Tres estudiants A, B i C són acusats d'introduir un virus en les sales d'ordinadors de la FIB. Durant l'interrogatori, les declaracions són les següents:

Les tres declaracions són:

- A diu: $b \land \neg c$
- B diu: $a \rightarrow c$
- C diu: $\neg c \land (a \lor b)$

b) Assumint que tots son innocents, qui o quins van mentir en la declaració?

Si tots són innocents, A i C van mentir.

Exercici 33

33. (dificultat 3) Tres estudiants A, B i C són acusats d'introduir un virus en les sales d'ordinadors de la FIB. Durant l'interrogatori, les declaracions són les següents:

Les tres declaracions són:

- A diu: $b \wedge \neg c$
- B diu: $a \rightarrow c$
- C diu: $\neg c \wedge (a \vee b)$

c) Assumint que ningún va mentir, qui és innocent i qui és culpable?

Si ningú ha mentit, llavors B és culpable.

Exercici 34

34. (dificultat 2) Inventa i defineix formalment alguna altre lògica diferent a la lògica proposicional. Per exemple, si les interpretacions són funcions $I : \mathcal{P} \to \{0, 1, \perp\}$ que també poden donar "indefinit" $\perp$, es pot adaptar la noció de satisfacció de manera raonable, tot i que la resposta ja no serà binaria: l'"avaluació" d'una fórmula F en una interpretació I pot donar 1 (I satisfà F) o 0 (I no satisfà F) o $\perp$ (indefinit).

Podem fer-ho així, de manera raonable, suposant que $\perp$ en la nostra aplicació modela "no ho sé":

```
if ( x and y ) {
   ...
}
```

Exercici 34

34. (dificultat 2) Inventa i defineix formalment alguna altre lògica diferent a la lògica proposicional. Per exemple, si les interpretacions són funcions $I : \mathcal{P} \to \{0, 1, \perp\}$ que també poden donar "indefinit" $\perp$, es pot adaptar la noció de satisfacció de manera raonable, tot i que la resposta ja no serà binaria: l'"avaluació" d'una fórmula F en una interpretació I pot donar 1 (I satisfà F) o 0 (I no satisfà F) o $\perp$ (indefinit).

$eval_I(\neg F) =$

0	dona	1
1		0
$\perp$		$\perp$

$eval_I(F \wedge G) =$

0	0	dona	0
0	1		0
0	$\perp$		0
1	0		0
1	1		1
1	$\perp$		$\perp$
$\perp$	0		0
$\perp$	1		$\perp$
$\perp$	$\perp$		$\perp$

$eval_I(F \vee G) =$

0	0	dona	0
0	1		1
0	$\perp$		$\perp$
1	0		1
1	1		1
1	$\perp$		1
$\perp$	0		$\perp$
$\perp$	1		1
$\perp$	$\perp$		$\perp$

Exercici 34

34. (dificultat 2) Inventa i defineix formalment alguna altre lògica diferent a la lògica proposicional. Per exemple, si les interpretacions són funcions $I : \mathcal{P} \rightarrow \{0, 1, \perp\}$ que també poden donar "indefinit" $\perp$, es pot adaptar la noció de satisfacció de manera raonable, tot i que la resposta ja no serà binaria: l' "avaluació" d'una fórmula F en una interpretació I pot donar 1 (I satisfà F) o 0 (I no satisfà F) o $\perp$ (indefinit).

I si $\perp$ modela "no termina"? Per exemple, en un programa com:

```
if ( not f(...) ) {
}

if ( f(...) and g(...) ) {
}

if ( f(...) or g(...) ) {
}
```

Exercici 34

34. (dificultat 2) Inventa i defineix formalment alguna altre lògica diferent a la lògica proposicional. Per exemple, si les interpretacions són funcions $I : \mathcal{P} \rightarrow \{0, 1, \perp\}$ que també poden donar "indefinit" $\perp$, es pot adaptar la noció de satisfacció de manera raonable, tot i que la resposta ja no serà binaria: l'"avaluació" d'una fórmula F en una interpretació I pot donar 1 (I satisfà F) o 0 (I no satisfà F) o $\perp$ (indefinit).

$eval_I(\neg F) =$

0	dona	1
1		0
$\perp$		$\perp$

$eval_I(F \wedge G) =$

0	0	dona	0
0	1		0
0	$\perp$		0
1	0		0
1	1		1
1	$\perp$		$\perp$
$\perp$	0		$\perp$
$\perp$	1		$\perp$
$\perp$	$\perp$		$\perp$

$eval_I(F \vee G) =$

0	0	dona	0
0	1		1
0	$\perp$		$\perp$
1	0		1
1	1		1
1	$\perp$		1
$\perp$	0		$\perp$
$\perp$	1		$\perp$
$\perp$	$\perp$		$\perp$

Exercici 35

35. (dificultat 2) Com l'exercici anterior, però considerant $I : \mathcal{P} \rightarrow [0 \ldots 1]$, és a dir, l'interpretació d'un símbol p és una probabilitat (un número real entre 0 i 1). En aquest cas, l'avaluació d'una fórmula F en una interpretació I pot donar quelcom (remotament) semblant a la probabilitat de satisfacció de F en I. En la lògica que has definit, l'avaluació de F en una I determinada, i la de $F \wedge F$ en aquesta mateixa I donen el mateix resultat?

$$
\begin{array}{lcl}
eval_I(\neg F) & = & 1 - eval_I(F) \\
eval_I(F \wedge G) & = & eval_I(F) \cdot eval_I(G) \\
eval_I(F \vee G) & = & (eval_I(F) + eval_I(G)) - (eval_I(F) \cdot eval_I(G))
\end{array}
$$

Això ens ho hem inventat, però és incorrecte en general perquè les probabilitats de les subfórmules no són independents. Per exemple, l'avaluació de $F \wedge F$ hauria de donar el mateix que la de F i aquí no és així.

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)

Deducció en Lògica Proposicional

Capítol 3 dels apunts: ☞ `p3.pdf`

Formes normals i clàusules

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

- Fórmules com a conjunts
- Literals
- CNF i DNF
- Clàusula
- Conjunt de clàusules
- Clàusula buida
- Clàusula de Horn

Formes normals i clàusules

- **Fórmules com a conjunts**

$$
\begin{array}{lcll}
F \wedge F & \equiv & \mathrm{F} & \text{idempotència de } \wedge \\
F \wedge G & \equiv & G \wedge F & \text{conmutativitat de } \wedge \\
(F \wedge G) \wedge H & \equiv & F \wedge (G \wedge H) & \text{associativitat de } \wedge \\
 & + & & \\
\multicolumn{3}{c}{\text{Lema de Substitució}} &
\end{array}
$$

- Conjunció de fórmules. Exemple:

$$
\begin{aligned}
F &= (F_1 \wedge ((F_2 \wedge (F_3 \wedge F_4) \wedge F_2))) \\
&\equiv F_1 \wedge F_2 \wedge F_3 \wedge F_4 \\
F &= \{\, F_1,\ F_2,\ F_3,\ F_4 \,\}
\end{aligned}
$$

- Idem amb la connectiva $\vee$

Formes normals i clàusules

- **Literals**

  literals positius: $p, q, r \ldots$

  literals negatius: $\neg p, \neg q, \ldots$

- **CNF** (*conjuntive normal form*)

  conjunció de disjuncions de literals:

  $(l_{1,1} \vee \ldots \vee l_{1,k_1}) \wedge \ldots \wedge (l_{n,1} \vee \ldots \vee l_{n,k_n})$

- **DNF** (*disjuntive normal form*)

  disjunció de conjuncions de literals:

  $(l_{1,1} \wedge \ldots \wedge l_{1,k_1}) \vee \ldots \vee (l_{n,1} \wedge \ldots \wedge l_{n,k_n})$

- **Clàusula**

  disjunció de literals:

  $l_1 \vee \ldots \vee l_k \qquad \{l_1, \ldots, l_k\}$

Formes normals i clàusules

- **Literals**

  literals positius: $p, q, r \ldots$

  literals negatius: $\neg p, \neg q, \ldots$

- **CNF** (*conjuntive normal form*)

  conjunció de disjuncions de literals:

  $(l_{1,1} \vee \ldots \vee l_{1,k_1}) \wedge \ \ldots \ \wedge (l_{n,1} \vee \ldots \vee l_{n,k_n})$

- **DNF** (*disjuntive normal form*)

  disjunció de conjuncions de literals:

  $(l_{1,1} \wedge \ldots \wedge l_{1,k_1}) \vee \ \ldots \ \vee (l_{n,1} \wedge \ldots \wedge l_{n,k_n})$

- **Clàusula**

  disjunció de literals:

  $l_1 \vee \ldots \vee l_k$

  $p_1 \vee \ldots \vee p_m \vee \neg q_1 \vee \ldots \vee \neg q_n$

- CNF $=$ Conjunció de clàusules $=$ **Conjunt de clàusules**

Formes normals i clàusules

- **Clàusula buida**
  - Disjunció de 0 literals: $l_1 \vee \ldots \vee l_k$ amb $k = 0$.
    Es denota amb $\Box$ En fitxer de text: []
  - No és una fórmula segons la sintaxi de la LProp ...
- Extensió de la *sintaxi* de la LProp
  Si tenim que $n \geq 0$ i que $F_1, \ldots, F_n$ són fórmules,
  llavors també són fórmules:
  $\bigwedge_{i \in 1..n} F_i$ i $\bigvee_{i \in 1..n} F_i$
- Extensió de la *semàntica* de la LProp.
  Si I és una interpretació:

$$eval_I(\bigwedge_{i \in 1..n} F_i) = min(\ 1, eval_I(F_1), \ldots, eval_I(F_n)\ )$$
$$eval_I(\bigvee_{i \in 1..n} F_i) = max(0, eval_I(F_1), \ldots, eval_I(F_n)\ )$$

☞ Si $n = 0$ la conjunció de 0 fórmules és trivialment certa

☞ Si $n = 0$ la disjunció de 0 fórmules és trivialment falsa

- **Clàusula de Horn**

  $p_1 \vee \ldots \vee p_m \vee \neg q_1 \vee \ldots \vee \neg q_n$ amb $m \leq 1$

  ☞ màxim 1 literal positiu

Exercici 1

1. (dificultat 2) Demostra que, per a tota fórmula F, hi ha com a mínim una fórmula lògicamente equivalent que està en DNF. Ídem per a CNF. Ajuda: obtenir les fórmulas en CNF i DNF a partir de la taula de veritat per a F.

	p	q	r	F
v_0	0	0	0	0
v_1	0	0	1	1
v_2	0	1	0	0
v_3	0	1	1	0
v_4	1	0	0	1
v_5	1	0	1	0
v_6	1	1	0	1
v_7	1	1	1	0

$$\begin{aligned} \text{DNF: } F' &= v_1 \vee v_4 \vee v_6 \\ &= (\neg p \wedge \neg q \wedge r) \quad \vee \\ &\quad\ (p \wedge \neg q \wedge \neg r) \quad \vee \\ &\quad\ (p \wedge q \wedge \neg r) \end{aligned}$$

Exercici 1

1. (dificultat 2) Demostra que, per a tota fórmula F, hi ha com a mínim una fórmula lògicamente equivalent que està en DNF. Ídem per a CNF. Ajuda: obtenir les fórmulas en CNF i DNF a partir de la taula de veritat per a F.

	p	q	r	F
v_0	0	0	0	0
v_1	0	0	1	1
v_2	0	1	0	0
v_3	0	1	1	0
v_4	1	0	0	1
v_5	1	0	1	0
v_6	1	1	0	1
v_7	1	1	1	0

$$
\begin{aligned}
\text{CNF: } F'' &= \neg v_0 \wedge \neg v_2 \wedge \neg v_3 \wedge \neg v_5 \wedge \neg v_7 \\
&= \neg(\neg p \wedge \neg q \wedge \neg r) \quad \wedge \\
&\quad\ \neg(\neg p \wedge q \wedge \neg r) \quad \wedge \\
&\quad\ \neg(\neg p \wedge q \wedge r) \quad \wedge \\
&\quad\ \neg(p \wedge \neg q \wedge r) \quad \wedge \\
&\quad\ \neg(p \wedge q \wedge r)
\end{aligned}
$$

Exercici 1

1. (dificultat 2) Demostra que, per a tota fórmula F, hi ha com a mínim una fórmula lògicamente equivalent que està en DNF. Ídem per a CNF. Ajuda: obtenir les fórmulas en CNF i DNF a partir de la taula de veritat per a F.

	p	q	r	F
v_0	0	0	0	0
v_1	0	0	1	1
v_2	0	1	0	0
v_3	0	1	1	0
v_4	1	0	0	1
v_5	1	0	1	0
v_6	1	1	0	1
v_7	1	1	1	0

$$
\begin{aligned}
\text{CNF: } F'' &= \neg v_0 \wedge \neg v_2 \wedge \neg v_3 \wedge \neg v_5 \wedge \neg v_7 \\
&= (p \vee q \vee r) \qquad \wedge \\
&\quad\; (p \vee \neg q \vee r) \qquad \wedge \\
&\quad\; (p \vee \neg q \vee \neg r) \qquad \wedge \\
&\quad\; (\neg p \vee q \vee \neg r) \qquad \wedge \\
&\quad\; (\neg p \vee \neg q \vee \neg r)
\end{aligned}
$$

Exercici 2

2. (dificultat 3) Dona una manera de calcular una fórmula $\hat{F}$ en CNF para una fórmula F donada (amb $\hat{F} \equiv F$) sense necessitat de construir prèviament la taula de veritat. Ajuda: aplica equivalències lògiques com les lleis de De Morgan i la distributivitat i el Lema de Substitució.

1. Moure les negacions cap a dins:

$$\begin{array}{llll} \neg\neg F & \Longrightarrow & F & \text{doble negació} \\ \neg(F \wedge G) & \Longrightarrow & \neg F \vee \neg G & \text{llei de De Morgan 1} \\ \neg(F \vee G) & \Longrightarrow & \neg F \wedge \neg G & \text{llei de De Morgan 2} \end{array}$$

2. Aplicar distributivitat 1:

$$(F \wedge G) \vee H \quad \Longrightarrow \quad (F \vee H) \wedge (G \vee H)$$

3. Eliminar literals repetits en les clàusules, i clàusules trivialment certes

Exercici 3

3. (dificultat 3) Cada fórmula de lògica proposicional pot veure's com un circuit electrònic que té una porta lògica per cada connectiva $\wedge$, $\vee$, $\neg$ que aparegui en la fórmula (encara que les fórmules tenen estructura d'arbre, mentre els circuits en realitat permeten compartir subarbres repetits, és a dir, són grafs dirigits acíclics).

El problema del *disseny lògic* consisteix a trobar un circuit adequat que implementi una funció booleana donada. Per a aconseguir circuits *ràpids*, ens va bé representar la funció booleana com una fórmula en CNF (o DNF), perquè la *profunditat* del circuit serà com a màxim tres.

Però també és important utilitzar el mínim nombre de connectives (portes lògiques). Els mètodes d'obtenir CNFs vists en els exercicis anteriors, ens donen la CNF més curta en aquest sentit? Se t'acut alguna millora?

Exercici 4

4. (dificultat 1) La clàusula buida $\Box$ és el cas més senzill de fórmula insatisfactible. Una CNF que sigui un conjunt de zero clàusules, és satisfactible o insatisfactible?

Una CNF és una conjunció de clàusules.
Si es tracta d'una conjunció de zero clàusules, per definició és trivialment certa, i per tant, satisfactible.

Exercici 5

5. (dificultat 2) Demostra que una clàusula és una tautologia si, i només si, conté alhora p i $\neg p$ per un cert símbol proposicional p.

Demostrarem que una clàusula de la forma $p_1 \vee \cdots \vee p_m \vee \neg q_1 \vee \cdots \vee \neg q_n$ NO és tautologia ssi NO conté alhora p i $\neg p$.

$p_1 \vee \cdots \vee p_m \vee \neg q_1 \vee \cdots \vee \neg q_n$ NO és tautologia ssi

$\exists\, I$, tal que I no és model de $p_1 \vee \cdots \vee p_m \vee \neg q_1 \vee \cdots \vee \neg q_n$ ssi

$\exists\, I$, tal que $I \not\models p_1 \vee \cdots \vee p_m \vee \neg q_1 \vee \cdots \vee \neg q_n$ ssi

$\exists\, I$, tal que $eval_I(\, p_1 \vee \cdots \vee p_m \vee \neg q_1 \vee \cdots \vee \neg q_n\,) = 0$ ssi

$\exists\, I$, tal que $max(eval_I(p_1), \ldots, eval_I(p_m), eval_I(\neg q_1), \ldots, eval_I(\neg q_n)) = 0$ ssi

$\exists\, I$, tal que $I(p_i) = 0$ per a totes les p_i i $I(q_j) = 1$ per a totes les q_j ssi

$p_i \neq q_j$ per a tota i, j

Exercici 6

6. (dificultat 2) 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.

b) Tota clàusula C de S té algún literal negatiu.

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.

Exercici 6

6. (dificultat 2) Sigui S un conjunt de clàusules amb $\Box \notin S$.
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: $p_1 \vee \cdots \vee p_m \vee \neg q_1 \vee \cdots \vee \neg q_n \quad (m > 1, n \geq 0)$

Sigui $S = \{C_1, C_2, \ldots\}$. Cada clàusula C_i 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.

Exercici 6

6. (dificultat 2) 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 C de S té algún literal negatiu.

Sigui $S = \{C_1, C_2, \ldots\}$. Cada clàusula C_i 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.

Exercici 6

6. (dificultat 2) Sigui S un conjunt de clàusules amb $\Box \notin S$. 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 $p_1 \vee \cdots \vee p_m \vee \neg q_1 \vee \cdots \vee \neg q_n$ on

- les p_i's només apareixen en literals positius en la resta de les clàusulas
- les q_i's només apareixen en literals negatius en la resta de les clàusulas

Un model que satisfarà 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

Exercici 6

6. (dificultat 2) Sigui S un conjunt de clàusules amb $\Box \notin S$.
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 satisfarà 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 satisfarà TOTS els literals de TOTES les clàusules (quan en realitat en tenia prou amb complir UN literal de cada clàusula).

Deducció en Lògica Proposicional

Continguts del capítol 3 [`p3.pdf`] per al proper dia:

- Exercicis del 7 en endavant
- ☞ Resolució (pàg. 5)

Lògica en la Informàtica
Deducció en Lògica Proposicional

José Miguel Rivero

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

Primavera 2025

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.

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)

Sumari

Exercici 7

7. (dificultat 1) Donats n símbols proposicionals:

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

7b) Quantes d'aquestes clàusules són insatisfactibles?

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

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

Exercici 7

7. (dificultat 1) 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?

$S = \{e_1 \ e_2 \ \dots \ e_k\}$

$0 \ 0 \ \dots \ 0$	denota el subconjunt buit
$0 \ 0 \ \dots \ 1$	denota el subconjunt $\{e_k\}$
$\dots$	
$1 \ 1 \ \dots \ 1$	denota el subconjunt $S = \{e_1 \ e_2 \ \dots \ e_k\}$

Això explica que hi ha 2^k subconjunts diferents.

Exercici 7

7. (dificultat 1) 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) $= 4^n$

Exercici 7

7. (dificultat 1) 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 $\neg p$ en la clàusula
b) està només p en la clàusula
c) està només $\neg p$ en la clàusula
d) no està ni p ni $\neg p$ en la clàusula

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

Exercici 7

7. (dificultat 1) Donats n símbols proposicionals:

7b) Quantes d'aquestes clàusules són insatisfactibles?

Només 1, la clàusula buida.

En una clàusula $p_1 \vee \cdots \vee p_m \vee \neg q_1 \vee \cdots \vee \neg q_n$ si $m+n > 0$, llavors sí és satisfactible.

Exercici 7

7. (dificultat 1) Donats n símbols proposicionals:

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

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

Exercici 7

7. (dificultat 1) 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).

Exercici 8

8. (dificultat 4) 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 $l \vee l' \vee C$ on C és la resta de la clàusula.

Llavors S és de la forma $\{l \vee l' \vee C\} \cup S1$.

Com podem expressar que $p \leftrightarrow l \vee l'$ mitjançant clàusules de màxim 3 literals? Recordem: $a \rightarrow b \equiv \neg a \vee b$

$$p \rightarrow l \vee l' \equiv \neg p \vee l \vee l'$$
$$p \leftarrow l \vee l' \equiv \{ l \rightarrow p, l' \rightarrow p \} \equiv \{ \neg l \vee p, \neg l' \vee p \}$$

Exercici 8

8. (dificultat 4) 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 $\{\, p \vee C, \neg l \vee p, \neg l' \vee p, \neg p \vee l \vee l' \,\} \cup S1$,
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.

Exercici 8

8. (dificultat 4) 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 <u>equisatisfactible</u> a S (és a dir, que és satisfactible si i només si S ho és).

A) $\Longrightarrow$: S és satisfactible $\Longrightarrow$ S' és satisfactible.

B) $\Longleftarrow$: S' és satisfactible $\Longrightarrow$ S és satisfactible.

Exercici 8

8. (dificultat 4) 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 <u>equisatisfactible</u> a S (és a dir, que és satisfactible si i només si S ho és).

A) $\Longrightarrow$: S és satisfactible $\Longrightarrow$ S' és satisfactible.

S és satisfactible	ssi
$\exists I$ tq I és model de S	ssi
$\exists I$ tq $I \models S$	

Exercici 8

8. (dificultat 4) 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) $\Longrightarrow$: S és satisfactible $\Longrightarrow$ S' és satisfactible.

Si $I \models l \vee l'$ llavors sigui I' la interpretació que ESTÉN la I amb $I'(p) = 1$, (I' és com I, excepte que a més a més $I'(p) = 1$)
si no, sigui I' la interpretació que ESTÉN la I amb $I'(p) = 0$.

Tenim que $I' \models S1$, perquè $I \models S1$.
A més a més $I' \models \{\, p \vee C, \neg l \vee p, \neg l' \vee p, \neg p \vee l \vee l' \,\}$ perquè (entre altres raons) $I \models l \vee l' \vee C$.
Per això $I' \models S'$ i per tant S' és sat.

Exercici 8

8. (dificultat 4) 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) $\Longleftarrow$: S' és satisfactible $\Longrightarrow$ S és satisfactible.

Sigui I' model de S'.
Sigui I la RESTRICCIÓ de I' "oblidant-nos" de la p.
I veiem que $I \models S$.

Exercici 9

9. (dificultat 2) Sigui S un conjunt de clàusules de Horn on la clàusula buida no està en S ($\square \notin S$). 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 $p_1 \vee \cdots \vee p_k \vee \neg q_1 \vee \cdots \vee \neg q_m$ on $k+m > 0$, i $k \leq 1$. Com poden ser?

De la forma:

a) p

b) $p \vee \neg q_1 \vee \cdots \vee \neg q_n$ amb $n > 0$

c) $\neg q_1 \vee \cdots \vee \neg q_n$ amb $n > 0$

Exercici 9

9. (dificultat 2) Sigui S un conjunt de clàusules de Horn on la clàusula buida no està en S ($\square \notin S$). 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.

Ara diu que tampoc hi ha de tipus a): no hi ha cap clàusula que només consti d'un únic literal positiu.

a) $\cancel{p}$

b) $p \vee \neg q_1 \vee \cdots \vee \neg q_n \quad$ amb $n > 0$

c) $\neg q_1 \vee \cdots \vee \neg q_n \quad$ amb $n > 0$

Llavors és satisfactible: un model és la I on per a tot símbol p —que aparegui en les clàusules b) o c)— tenim $I(p) = 0$.

Exercici 10

10. (dificultat 2) 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.

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

$$p \vee q$$
$$\neg p$$
$$\neg q$$

Exercici 10

10. (dificultat 2) Demostra que l'enunciat de l'exercici previ és fals quan S no és de Horn.

Un altre exemple:

$$\begin{aligned} p &\vee q \\ p &\vee \neg q \\ \neg p &\vee q \\ \neg p &\vee \neg q. \end{aligned}$$

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.

Exercici 12

12. (dificultat 3) 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 $p_1 \wedge \cdots \wedge p_k \wedge \neg q_1 \wedge \cdots \wedge \neg q_m$.
Una DNF $= \{ C_1 \vee \cdots \vee C_n \}$ és satisfactible ssi algun cub C_i és satisfactible.

Una cub $p_1 \wedge \cdots \wedge p_k \wedge \neg q_1 \wedge \cdots \wedge \neg q_m$ 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**.

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

5. Resolució. Correcció i completitud

- Resolució
- Clausura sota resolució
- Clausura sota una regla deductiva qualsevol
- Correcció i completitud d'una regla deductiva
- Completitud refutacional de la resolució

Deducció en Lògica Proposicional

- **Resolució** (*regla deductiva*)

  Donadas dues clàusules: $p \vee C$ i $\neg p \vee D$

  Resolució per a lògica proposicional:

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

  Premisses: $p \vee C$, $\neg p \vee D$

  Conclusió: $C \vee D$

- La resolució és CORRECTA?

  És a dir, siguin com siguin C, D i p, tenim que
  $(p \vee C) \wedge (\neg p \vee D) \models C \vee D$?

Exercici 15

15. (dificultad 2) Demostra que la resolució és correcta.

Es CORRECTA si siguin com siguin C, D i p, tenim que
$(p \vee C) \wedge (\neg p \vee D) \models C \vee D$.

Sigui I un model de $(p \vee C) \wedge (\neg p \vee D)$.
Hi han dos casos:

$I(p) = 1$ Llavors
$I \models (p \vee C) \wedge (\neg p \vee D) \implies I \models \neg p \vee D \implies I \models D \implies I \models C \vee D$

$I(p) = 0$ Llavors
$I \models (p \vee C) \wedge (\neg p \vee D) \implies I \models p \vee C \implies I \models C \implies I \models C \vee D$

Exercici 15

15. (dificultat 2) Demostra que la resolució és correcta.

Es CORRECTA si siguin com siguin C, D i p, tenim que $(p \vee C) \wedge (\neg p \vee D) \models C \vee D$.

Sigui I un model de $(p \vee C) \wedge (\neg p \vee D)$.
Hi han dos casos (la demostració pel cas $I(p) = 0$ és similar):

- $I(p) = 1$:

  $I \models (p \vee C) \wedge (\neg p \vee D)$
  implica [per def. de $\models$]
  $eval_I((p \vee C) \wedge (\neg p \vee D)) = 1$
  implica [per def. de $eval_I(\wedge)$]
  $min(eval_I(p \vee C), eval_I(\neg p \vee D)) = 1$
  implica [per def. de $eval_I$, i de min]
  $eval_I(p \vee C) = 1$ **i** $eval_I(\neg p \vee D) = 1$
  implica [per def. de conjunció i]
  $eval_I(\neg p \vee D) = 1$

Exercici 15

15. (dificultad 2) Demostra que la resolució és correcta.

Es CORRECTA si siguin com siguin C, D i p, tenim que $(p \vee C) \wedge (\neg p \vee D) \models C \vee D$.

Sigui I un model de $(p \vee C) \wedge (\neg p \vee D)$.

Hi han dos casos (la demostració pel cas $I(p) = 0$ és similar):

- $I(p) = 1$ (cont.):

  $eval_I(\neg p \vee D) = 1$

  implica [per def. de $eval_I(\ \vee\ )$]

  $max(eval_I(\neg p), eval_I(D)) = 1$

  implica [per def. de $eval_I(\ \neg\ )$]

  $max(1 - eval_I(p), eval_I(D)) = 1$

  implica [per def. de $eval_I(p)$]

  $max(1 - I(p), eval_I(D)) = 1$

  implica [perquè $I(p) = 1$, i per aritmètica]

  $max(0, eval_I(D)) = 1$

Exercici 15

15. (dificultad 2) Demostra que la resolució és correcta.

Es CORRECTA si siguin com siguin C, D i p, tenim que
$(p \vee C) \wedge (\neg p \vee D) \models C \vee D$.

Sigui I un model de $(p \vee C) \wedge (\neg p \vee D)$.
Hi han dos casos (la demostració pel cas $I(p) = 0$ és similar):

- $I(p) = 1$ (cont.):

  $max(0, eval_I(D)) = 1$
  implica [per def. de max]
  $eval_I(D) = 1$
  implica [per def. de $eval_I$ i de max, i per qualsevol C]
  $max(eval_I(C), eval_I(D)) = 1$
  implica [per def. de $eval_I(\ \vee\ )$]
  $eval_I(C \vee D) = 1$
  implica [per def. de $\models$]
  $I \models C \vee D$

- **Clausura sota resolució**

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

Exemple: Sigui S el conjunt de clàusules:

$$\begin{array}{rl} \{ & p \vee q \\ & p \vee \neg q \\ & \neg p \vee q \\ & \neg p \vee \neg q \ \}. \end{array}$$

➤ Puc obtenir mitjaçant resolució a partir de $p \vee q$ i $p \vee \neg q$ (sobre la q) i obtinc $p \vee p$ que és el mateix que p.

➤ Puc obtenir mitjaçant resolució a partir de $\neg p \vee q$ i $\neg p \vee \neg q$ (sobre la q) i obtinc $\neg p \vee \neg p$ que és el mateix que $\neg p$.

- **Clausura sota resolució**

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

Exemple: Sigui S el conjunt de clàusules:

$$\begin{array}{l} \{\ \ p \vee q \\ \quad p \vee \neg q \\ \quad \neg p \vee q \\ \quad \neg p \vee \neg q\ \}. \end{array}$$

A partir de les dues clàusules noves p i $\neg p$, en un altre pas puc obtenir **la clàusula buida** $\Box$.

- **Clausura sota resolució**

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

Exemple: Sigui S el conjunt de clàusules:

$$\begin{array}{ll}
\{\; p \vee q & S_0 = S \\
\;\; p \vee \neg q & S_1 = S_0 \cup \{\, p, \neg p, q, \neg q, p \vee \neg p, q \vee \neg q, \ldots \} \\
\;\; \neg p \vee q & S_2 = S_1 \cup \{\, \Box, \ldots \} \\
\;\; \neg p \vee \neg q \;\}. & S_3 = S_2 \cup ???
\end{array}$$

$$Res(S) = \bigcup_{i=0}^{\infty} S_i$$

- **Correcció i completitud d'una regla deductiva**

  $S = \{F_1, \ldots, F_n\} \quad (S = F_1 \wedge \cdots \wedge F_n)$.

  $R(S)$ denota la clausura de S sota la regla deductiva R.

  - R és *correcta* si mitjançant R només podem obtenir conseqüències lògiques del que ja tenim.
    - si $F \in R(S)$ llavors $S \models F$.
  - R és *completa* si mitjançant R podem deduir totes les conseqüències lògiques.
    - si $S \models F$ llavors $F \in R(S)$.

Deducció en Lògica Proposicional

- **Completitud refutacional de la resolució**

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

La resolució és *refutacionalment completa*, és a dir, si S és insatisfactible llavors per resolució obtindré la clàusula buida.

Hi ha un teorema que diu:

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

Exercici 16

16. (dificultat 2) 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.

Exercici 17

17. (dificultat 3) 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) $I \models S \implies I \models Res(S)$

Exercici 17

17. (dificultat 3) 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 S_i's).

Exercici 17

17. (dificultat 3) 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 $I \models Res(S)$.

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

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, $I \models S \implies I \models Res(S)$ per **inducció** sobre k.

Exercici 17

17. (dificultat 3) 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 $I \models Res(S)$.

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

- Si $k = 0$, trivial perquè llavors $S = Res(S)$.
- 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 $I \models Res(S)$. $\blacksquare$

Exercici 18

18. (dificultat 2) La resolució és completa? Demostra-ho.

Completa: qualsevol conseqüència lògica es pot arribar a obtenir mitjançant resolució.

No. Contraexemple: Sigui S el conjunt buit de clàusules.
Llavors $S \models p \vee \neg p$. (perquè $p \vee \neg p$ és una tautologia).

Però NO podem obtenir $p \vee \neg p$ a partir de S mitjançant resolució.
Per tant NO podem obtenir qualsevol conseqüència lògica mitjançant resolució.

Per tant la resolució **NO** és completa.

Exercici 18

18. (dificultat 2) La resolució és completa? Demostra-ho.

Completa: qualsevol conseqüència lògica es pot arribar a obtenir mitjançant resolució.

Un altre contraexemple:
Sigui S qualsevol conjunt de clàusules que conté la clàusula buida.
Llavors S és insatisfactible.
I per tant tenim $S \models p$, on p és un símbol que no apareix en S.

Però NO podem obtenir p a partir de S mitjançant resolució.

Exercici 18

18. (dificultat 2) La resolució és completa? Demostra-ho.

Completa: qualsevol conseqüència lògica es pot arribar a obtenir mitjançant resolució.

Un altre contraexemple: $S = \{p, q\}$

$S \models p \vee q$ però NO podem obtenir $p \vee q$ per resolució a partir del conjunt de clàusules $\{p, q\}$.

Exercicis del capítol `p3.pdf` per al proper dia:

- exercicis fins al 27, i també
- ☞ **el sudoku** (pàg. 8)

Lògica en la Informàtica
Deducció en Lògica Proposicional

José Miguel Rivero

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

Primavera 2025

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.

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)

Sumari

Exercici 19

19. (dificultat 2) Sigui S un conjunt de clàusules insatisfactible. Per la completitud refutacional de la resolució, sabem que existeix una demostració per resolució de que $\Box \in Res(S)$. És aquesta demostració única?

Un petit incís:

Hi ha un teorema que diu: S és insatisfactible **SSI** $\Box$ està en $Res(S)$

$\Rightarrow$ "completitud refutacional" (és un cas particular de completitud)

$\Leftarrow$ pq $\Box$ està en $Res(S)$ $\Rightarrow$ $Res(S)$ insat $\Rightarrow$ S insat

(l'última $\Rightarrow$ ve de l'exercici 17: $Res(S)$ és lògicament equivalent a S)

Exercici 19

19. (dificultat 2) Sigui S un conjunt de clàusules insatisfactible. Per la completitud refutacional de la resolució, sabem que existeix una demostració per resolució de que $\square \in Res(S)$. És aquesta demostració única?

Tornant a l'exercici:

NO és única. Contraexemple. Sigui S el conjunt amb les 4 clàusules:

$$\begin{array}{r}
p \vee q \\
p \vee \neg q \\
\neg p \vee q \\
\neg p \vee \neg q.
\end{array}$$

Exercici 19

19. (dificultat 2) Sigui S un conjunt de clàusules insatisfactible. Per la completitud refutacional de la resolució, sabem que existeix una demostració per resolució de que $\square \in Res(S)$. És aquesta demostració única?

Podem obtenir $\square$ de més d'una manera:
Per exemple així:

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

Però també així:

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

Exercici 21

21. (dificultat 2) Demostra que el llenguatge de les clàusules de Horn és tancat sota resolució, és a dir, a partir de clàusules de Horn, per resolució només s'obtenen clàusules de Horn.

Una clàusula de Horn és una clàusula que té com a màxim 1 literal positiu.
Si tinc: $p \vee C$ i $\neg p \vee D$, i són de Horn, llavors hi ha 0 literals positius en C, i màxim 1 en D.

Per tant, en $C \vee D$ també hi ha com a màxim 1 literal positiu, és a dir, $C \vee D$ també és de Horn.

Exercici 25

25. (dificultat 2) Quina és la complexitat de Horn-SAT, és a dir, del problema de determinar si un conjunt de clàusules de Horn S és satisfactible?
Ajuda: pensa si la positive unit propagation decideix Horn-SAT, i analitza la complexitat d'això.

unit propagation: en el meu algorisme de SAT mitjançant backtracking (veure labo 1) si tinc una clàusula de la forma $l \vee C$ i una interpretació (parcial, que estic construint) on la part C és falsa, llavors el literal l ha de ser cert.

positive unit propagation: és el mateix, però només amb l positius

Exercici 25

25. (dificultat 2) Quina és la complexitat de Horn-SAT, és a dir, del problema de determinar si un conjunt de clàusules de Horn S és satisfactible?

    Ajuda: pensa si la positive unit propagation decideix Horn-SAT, i analitza la complexitat d'això.

Les clàusules d'un conjunt S de clàusules de Horn (sense la $\square$), poden ser de la forma:

a)	p	(positive unit clause)
b)	$p \vee \neg q_1 \vee \cdots \vee \neg q_n$	(1 positiu i n negatius, $n > 0$)
c)	$\neg q_1 \vee \cdots \vee \neg q_n$	(0 positius i n negatius, $n > 0$)

Un model de S HA DE satisfer totes les clàusules de tipus a) (les positive units). Si les propago, amb les clàusules de tipus b), obtinc noves positive units, que al seu torn poden propagar-se, etc, etc

Exercici 25

25. (dificultat 2) Quina és la complexitat de Horn-SAT, és a dir, del problema de determinar si un conjunt de clàusules de Horn S és satisfactible?
Ajuda: pensa si la positive unit propagation decideix Horn-SAT, i analitza la complexitat d'això.

Per exemple, S pot ser:

p	(tipus a)	
q	(tipus a)	
$r \lor \neg p \lor \neg q$	(tipus b)	$\rightarrow$ propago r
$r' \lor \neg p \lor \neg q \lor \neg r$	(tipus b)	$\rightarrow$ propago r'
$\neg r \lor \neg r'$	(tipus c)	$\rightarrow$ conflicte!!!

És clar que si em surt un conflicte, és insatisfactible. I si no em surt un conflicte? Llavors és satisfactible: hi ha un model! Quin és aquest model?

Exercici 25

25. (dificultat 2) Quina és la complexitat de Horn-SAT, és a dir, del problema de determinar si un conjunt de clàusules de Horn S és satisfactible?
    Ajuda: pensa si la positive unit propagation decideix Horn-SAT, i analitza la complexitat d'això.

El model és la I tal que $I(p) = 1$ per a totes les positive units p que he anat obtenint (les inicials i les altres) i $I(p) = 0$ per a tots els altres símbols p. Per què això és un model?

És model de les clàusules de tipus:

les a) ok
les b) que han propagat alguna cosa: ok
les b) que no han propagat res: ok, perquè tindrà un literal $\neg q$ on la q no està entre les positive units: $I(q) = 0$.
les c) (que no han donat conflicte): ok, perquè tindrà un literal $\neg q$ on la q no està entre les positive units: $I(q) = 0$.

Exercici 25

25. (dificultat 2) Quina és la complexitat de Horn-SAT, és a dir, del problema de determinar si un conjunt de clàusules de Horn S és satisfactible?
Ajuda: pensa si la positive unit propagation decideix Horn-SAT, i analitza la complexitat d'això.

Resumint:

sent S un conjunt de clàusules de Horn, S és insat SSI la **positive unit propagation** dona conflicte.

- Quin cost té?

Exercici 25

25. (dificultat 2) Quina és la complexitat de Horn-SAT, és a dir, del problema de determinar si un conjunt de clàusules de Horn S és satisfactible?
Ajuda: pensa si la positive unit propagation decideix Horn-SAT, i analitza la complexitat d'això.

- Quin cost té?

Això és lineal, si uso occur lists (veure labo 1) i un comptador per clàusula que compta el nombre de literals negatius que li queden. Quan el comptador es posa a zero $\rightarrow$ aquesta clàusula propaga.

NOTA: aquest algorisme només posa a cert aquells símbols p que HAN DE SER certs en QUALSEVOL model de S.
És a dir, el que es calcula és el model MINIMAL de S
(aquell model que té el mínim nombre de símbols a 1).

Exercici 25

25. (dificultat 2) Quina és la complexitat de Horn-SAT, és a dir, del problema de determinar si un conjunt de clàusules de Horn S és satisfactible?
    Ajuda: pensa si la positive unit propagation decideix Horn-SAT, i analitza la complexitat d'això.

Un altre exemple:

p
q
$r \vee \neg p \vee \neg q$
$r' \vee \neg p \vee \neg q \vee \neg r$
$r'' \vee \neg p \vee \neg q \vee \neg r'''$

Aquí, el model minimal quin és?

- $I(p) = I(q) = I(r) = I(r') = 1 \qquad I(r'') = I(r''') = 0.$

Però hi ha més models:

- $I(p) = I(q) = I(r) = I(r') = I(r'') = 1 \qquad I(r''') = 0.$
- el model on $I(p) = I(q) = I(r) = I(r') = I(r'') = I(r''') = 1$ (tots).

Exercici 25

25. (dificultat 2) Quina és la complexitat de Horn-SAT, és a dir, del problema de determinar si un conjunt de clàusules de Horn S és satisfactible?
    Ajuda: pensa si la positive unit propagation decideix Horn-SAT, i analitza la complexitat d'això.

Veiem que si S és de Horn i és satisfactible, llavors S té un model minimal únic.

És també cert això si S no és de Horn?

NO! Per exemple, $S = \{p \vee q\}$, llavors hi ha dos models minimals:

A) $I(p) = 1 \;\; I(q) = 0$.

B) $I(p) = 0 \;\; I(q) = 1$.

Exercici 26

26. (dificultat 2) Les clàusules de Krom són aquelles que tenen com a màxim dos literals. Quantes clàusules de Krom es poden construir amb n símbols de predicat? Demostra que basta un nombre quadràtic de passos de resolució per a decidir 2-SAT, és a dir, si un conjunt de clàusules de Krom és satisfactible o no.

Hi ha $2n$ literals. Cada clàusula de Krom és un subconjunt de com a màxim 2 literals d'aquests $2n$.

Hi ha $\binom{2n}{2}$	clàusules de Krom de dos literals	$+$
$\binom{2n}{1} = 2n$	clàusules de Krom de un literal	$+$
1	clàusula de Krom de 0 literals.	

$\binom{2n}{2} = 2n \cdot (2n-1)/2 = 4n^2 + \cdots = \mathcal{O}(n^2)$. Un número quadràtic.

Exercici 26

26. (dificultat 2) Les clàusules de Krom són aquelles que tenen com a màxim dos literals. Quantes clàusules de Krom es poden construir amb n símbols de predicat? Demostra que basta un nombre quadràtic de passos de resolució per a decidir 2-SAT, és a dir, si un conjunt de clàusules de Krom és satisfactible o no.

Cada pas de resolució em donarà una altra clàusula de Krom: "el llenguatge de les clàusules de Krom està tancat sota resolució".

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

la part C té com a màxim 1 literal i la part D també. Per tant $C \vee D$ té màxim dos.

Per això, la resolució acaba després d'un nombre quadràtic de passos. Si ho implementem bé, això ens dona un algorisme quadràtic per a 2-SAT.

Resumint

Ja tenim dos problemes concrets (subcasos del problema de SAT general) que són polinòmics:

- Horn-SAT
- 2-SAT

Exercici 27

27. (dificultad 3) Algorisme per a 2-SAT basat en detecció de cicles en un graf.

Sigui S un conjunt de clàusules de Krom.

Cada clàusula de Krom $l \vee l'$, en realitat representa dues implicacions:

$$\neg l \rightarrow l'$$
$$\neg l' \rightarrow l$$

Puc muntar un graf G a partir de S, amb totes aquestes arestes: cada clàusula de S em dona dues arestes en G.

Si en G tinc un camí $p \rightarrow \cdots \rightarrow \neg p$, llavors $S \models \neg p$.

Ho puc demostrar per correcció de la resolució.

Exercici 27

27. (dificultad 3) Algorisme per a 2-SAT basat en detecció de cicles en un graf.

Si tinc:

$$p \to l \to l' \to l'' \to \cdots \neg p, \text{ llavors } S \models \neg p$$

Per resolució:

$$\cfrac{\cfrac{\cfrac{\neg p \vee l \quad \neg l \vee l'}{\neg p \vee l'} \quad \neg l' \vee l''}{\neg p \vee l''}}{\begin{array}{c}\ddots \\ \hline \neg p \vee \neg p\end{array}}$$

que és $\neg p$

Exercici 27

27. (dificultad 3) Algorisme per a 2-SAT basat en detecció de cicles en un graf.

Si en G tinc un camí $p \to \cdots \to \neg p$, llavors $S \models \neg p$.

Similarment, si en G tinc un camí $\neg p \to \cdots \to p$, llavors $S \models p$.

I si tinc els dos camins, és que hi ha un cicle en G que conté un símbol p i el seu negat $\neg p$, ... i llavors S és insatisfactible.

I a l'inrevés, si S és insatisfactible, llavors hi ha un cicle en G que conté un símbol i el seu negat.

Resumint

S és insatisfactible SSI hi ha un cicle en G que conté un símbol i el seu negat.

Exercici 27

27. (dificultad 3) Algorisme per a 2-SAT basat en detecció de cicles en un graf.

➤ Quin és el cost d'aquest algorisme per a 2-SAT?

- Muntar el graf és lineal.
- Després detectar cicles, es fa amb l'algorisme de Strongly Connected Components (SCCs), (Components Fortament Connexes) que és lineal.

Resumint

Surt un algorisme lineal per a 2-SAT.

Pensem ara en SAT en general.

☞ Apunts de la web de LI: "Breu resum sobre NP i NP-completitud" (Robert Nieuwenhuis)

Remembering some intuitions about NP and NP-completeness
(for more formal definitions and details, see the slides of the EDA course on this same website)

Decision problems and complexity classes
Here we focus on decision problems, the ones with output "yes" or "no", and on classifying problems (not algorithms!) according to the time needed to solve them (with the best of the available algorithms), and we will call problem A harder than problem B if solving A needs more time than solving B.

Decision problems and complexity classes (cont.)

For example, given a sequence of integers, the problem of deciding whether it contains the integer 7 can be solved in linear time. We say that it belongs to the class of problems solvable in linear time. If moreover the input sequence is ordered, then we can say more: it belongs to a proper subclass of the problems solvable in linear time, namely the ones solvable in logarithmic time (in this case, by binary search). Here we see that in fact what matters is how fast the running time grows depending on the size of the input.

Other problems are not linear, but harder. The class of polynomial problems is called P. Note that all logarithmic, linear, quadratic, cubic, etc., problems are in P.

Decision problems and complexity classes (cont.)

Some other problems are even harder, and are not in P. The class of exponential problems is called EXP (solvable in time with the input size n in the exponent; note that for large enough n, the number 2^n is much larger than n^2, n^3, or n^k for whatever constant k). It is known that $P \subset EXP$ (there are problems in EXP that are not in P such as "generalized chess").

The class NP, membership in NP, NP-hardness and NP completeness

There is a special class, NP, for which it is known that P $\subseteq$ NP $\subseteq$ EXP. NP is the class of problems having a Nondeterministic Polynomial algorithm. Roughly, this means that a problem A is in NP if, whenever the answer to A for a given input is "yes", there is a "witness" (a "solution") that allows one to verify this "yes" in polynomial time.

The class NP, membership in NP, NP-hardness and NP completeness (cont.)

The most famous problem in NP is SAT, the problem of deciding whether a given propositional input formula F is satisfiable or not. This problem is clearly in NP: if the answer is "yes", the witness is the model, which can be checked in polynomial (even linear) time.

Another example of problem in NP is 3-colorability: can we color each node of a given graph G with one of three colors, such that adjacent nodes get different colors? Here the witness is the coloring, indicating each node's color.

The class NP, membership in NP, NP-hardness and NP completeness (cont.)

A problem P is called NP-hard if any other problem in NP can be polynomially reduced to P. SAT is NP-hard: any problem in NP can be polynomially reduced to (or solved by, or expressed as) a SAT problem.

This means that for any problem A in NP and input data D for A, we can build in polynomial time a SAT formula F that is satisfiable if, and only if, the answer to A on input D is "yes". Moreover, from a satisfiablity witness of F (i.e., a model), it is usually easy to reconstruct a witness (or a "solution") for A on input D.

The class NP, membership in NP, NP-hardness and NP completeness (cont.)

For example, we can reduce 3-colorability to SAT. Let G be a graph with n nodes. Introducing $3n$ propositional symbols x_{ic} meaning "node i gets color c", let F state,

- for each node i, that it gets at least one color (a clause $x_{i1} \vee x_{i2} \vee x_{i3}$) and,
- for each edge (i, j), that i and j do not get the same color (three clauses per edge: $\neg x_{i1} \vee \neg x_{j1}$, $\neg x_{i2} \vee \neg x_{j2}$, and $\neg x_{i3} \vee \neg x_{j3}$).

Then F is satisfiable iff G is 3-colorable, and from any model for F it is trivial to reconstruct a 3-coloring for G.

The class NP, membership in NP, NP-hardness and NP completeness (cont.)

Note that if SAT can be polynomially reduced to some problem P, then P is NP-hard too. Apart from SAT, many other problems in NP have been proved NP-hard too (doing such reductions, or chains of them).

Note that, by such reductions, if we had a polynomial algorithm for any single NP-hard problem, then we would have it for all problems in NP, that is, we would have P=NP. That would have dramatic consequences, because there are many very important real-world problems in NP.

It is unknown whether P=NP? In fact, there is a million-dollar prize (search "millenium problems") for whoever proves either P=NP or P $\neq$ NP.

The class NP, membership in NP, NP-hardness and NP completeness (cont.)

Since P $\subset$ EXP, at least one of the two inclusions in P $\subseteq$ NP $\subseteq$ EXP is strict, and it is believed that both are, i.e., P $\subset$ NP $\subset$ EXP.

A problem is called NP-complete if

A) it is in NP, and

B) it is NP-hard.

Exercici: Emplena la resta de la taula amb la complexitat algorítmica corresponent.

	SAT	TAUT
CNF	NP-complet	
DNF		

Exercici: Emplena la resta de la taula amb la complexitat algorítmica corresponent.

	SAT	TAUT
CNF	NP-complet	
DNF	lineal (1)	

(1) Per l'exercici 12, sabem que per una fórmula en DNF podem decidir si és satisfactible en temps lineal.

Exercici: Emplena la resta de la taula amb la complexitat algorítmica corresponent.

	SAT	TAUT
CNF	NP-complet	lineal (2)
DNF	lineal (1)	

(2) Una CNF $C_1 \wedge \cdots \wedge C_n$ és tautologia ssi totes les seves clàusules C_i són tautologias.
Sabem per l'exercici 5. que una clàusula és una tautologia ssi conté alhora p i $\neg p$.

Exercici: Emplena la resta de la taula amb la complexitat algorítmica corresponent.

	SAT	TAUT
CNF	NP-complet	lineal (2)
DNF	lineal (1)	NP-complet (3)

(3) Una DNF $Cub_1 \vee \cdots \vee Cub_N$ és tautologia ssi $\neg(Cub_1 \vee \cdots \vee Cub_N)$ és insat.
Movent les negacions cap a dins en $\neg(Cub_1 \vee \cdots \vee Cub_N)$ obtenim una CNF F.

Exercici: Emplena la resta de la taula amb la complexitat algorítmica corresponent.

	SAT	TAUT
CNF	NP-complet	lineal (2)
DNF	lineal (1)	NP-complet (3)

(3) [cont.] A l'inrevés també:
Una CNF $C_1 \wedge \cdots \wedge C_n$ és insatisfactible ssi
$\neg(C_1 \wedge \cdots \wedge C_n)$ és tautologia ssi
$Cub_1 \vee \cdots \vee Cub_N$ és tautologia, on $Cub_1 \vee \cdots \vee Cub_N$
és la DNF obtinguda movent les negacions cap a dins
en $\neg(C_1 \wedge \cdots \wedge C_n)$.

Exercici: Emplena la resta de la taula amb la complexitat algorítmica corresponent.

	SAT	TAUT
CNF	NP-complet	lineal (2)
DNF	lineal (1)	NP-complet (3)

(3) [cont.]

Una DNF $Cub_1 \vee \cdots \vee Cub_N$ és tautologia ssi $\neg(Cub_1 \vee \cdots \vee Cub_N)$ és insat.

Per tant, decidir si una DNF és tautologia ha de ser NP-complet! Si no, tindríem una manera de fer SAT en temps polinòmic comprovant si $Cub_1 \vee \cdots \vee Cub_N$ és tautologia!

Per al proper dia:

- exercicis següents, i
- apunts de la web de LI:
  ☞ "La transformació de Tseitin a CNF equisatisfactible."

Lògica en la Informàtica
Deducció en Lògica Proposicional

José Miguel Rivero

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

Primavera 2025

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.

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)

Racó: Col·lecció d'apunts bàsics de lògica. ☞ `p3.pf`

- Formes normals i clàusules
- Resolució. Correcció i completesa
- Nocions informals de decidibilitat i complexitat
- Resoldre problemes pràctics amb la lògica proposicional

Sumari

Necessitem decidir SAT per a fórmules qualssevol, però els SAT solvers només treballen amb CNFs (conjunts de clàusules).

Per tant, necessitem poder transformar fórmules qualssevol en CNFs.

Tseitin. Veure la presentació (en la web de Lògica en la Informàtica ☞ `https://www.cs.upc.edu/~li`) sobre la transformació de Tseitin d'una fórmula qualsevol a una CNF equisatisfactible.

Transformació a CNF via distributivitat

1. Aplica les tres regles de transformació mentre sigui possible:
   - $\neg\neg F \Rightarrow F$
   - $\neg(F \wedge G) \Rightarrow \neg F \vee \neg G$
   - $\neg(F \vee G) \Rightarrow \neg F \wedge \neg G$

   Després d'això, la fórmula està en Negation Normal Form (NNF)
2. Ara aplica la regla de distributivitat mentre sigui possible:
   - $F \vee (G \wedge H) \Rightarrow (F \vee G) \wedge (F \vee H)$

EXEMPLE: sigui F la fòrmula $(p \wedge q) \vee \neg(\neg p \wedge (q \vee \neg r))$

1. $(p \wedge q) \vee \neg(\neg p \wedge (q \vee \neg r)) \Rightarrow (p \wedge q) \vee (\neg\neg p \vee \neg(q \vee \neg r)) \Rightarrow$
   $(p \wedge q) \vee (p \vee (\neg q \wedge \neg\neg r)) \Rightarrow (p \wedge q) \vee (p \vee (\neg q \wedge r))$
2. $(p \wedge q) \vee (p \vee (\neg q \wedge r)) \Rightarrow (p \vee p \vee (\neg q \wedge r)) \wedge (q \vee p \vee (\neg q \wedge r)) \Rightarrow$
   $(p \vee p \vee \neg q) \wedge (p \vee p \vee r) \wedge (q \vee p \vee \neg q) \wedge (q \vee p \vee r) \Rightarrow$
   $(p \vee \neg q) \wedge (p \vee r) \wedge (q \vee p \vee r)$

Deducció en Lògica Proposicional

Per què la transformació via distributivitat pot fer créixer exponencialment la fórmula?

Perquè la regla de distributivitat

$F \vee (G \wedge H) \Longrightarrow (F \vee G) \wedge (F \vee H)$ DUPLICA la subfórmula F.

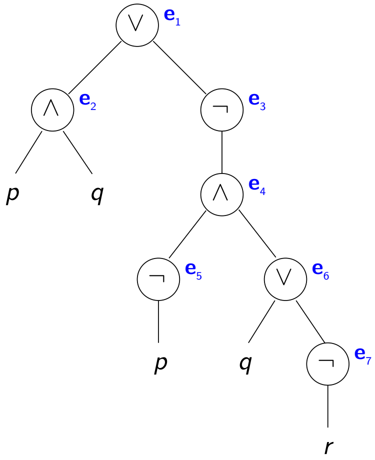
Exemple de cas pitjor: si F és una DNF $Cub_1 \vee \cdots \vee Cub_n$, on cada cub és un AND de k literals, la CNF tindrà TOTES les clàusules possibles amb un literal de cada cub, és a dir k^n clàusules (el literal del primer cub es pot triar de k maneres, el del segon també, etc.).

Exemple: $(p \wedge q \wedge r) \;\vee\; (p' \wedge q' \wedge r')$ donaria:

$$\begin{array}{lll} p \vee p', & p \vee q', & p \vee r', \\ q \vee p', & q \vee q', & q \vee r', \\ r \vee p', & r \vee q', & r \vee r' \end{array}$$

Deducció en Lògica Proposicional

Sigui F la fórmula $(p \wedge q) \vee \neg(\neg p \wedge (q \vee \neg r))$



- e_1
- $e_1 \leftrightarrow e_2 \vee e_3$
  - $\neg e_1 \vee e_2 \vee e_3$
  - $\neg e_2 \vee e_1$
  - $\neg e_3 \vee e_1$
- $e_2 \leftrightarrow p \wedge q$
  - $\neg p \vee \neg q \vee e_2$
  - $\neg e_2 \vee p$
  - $\neg e_2 \vee q$
- $e_3 \leftrightarrow \neg e_4$
  - $\neg e_3 \vee \neg e_4$
  - $e_3 \vee e_4$
- $e_4 \leftrightarrow e_5 \wedge e_6$
- $e_5 \leftrightarrow \neg p$
- $e_6 \leftrightarrow q \vee e_7$
- $e_7 \leftrightarrow \neg r$

Per això fem TSEITIN:

Introduïm un símbol nou per cada connectiva de la fórmula.
I generem les clàusules que "defineixen" el paper que juguen aquests símbols nous en la fórmula.

Per exemple, per a expressar que p és el símbol d'un node OR de dos fills amb símbols a, b, necessitem $p \leftrightarrow a \vee b$.

Per a això:

- expressem $p \rightarrow a \vee b$ mitjançant una clàusula de tres literals: $\neg p \vee a \vee b$
- expressem $p \leftarrow a \vee b$ que és $a \rightarrow p$ i $b \rightarrow p$, amb dues clàusules: $\neg a \vee p \quad \neg b \vee p$

Per això fem TSEITIN:

Introduïm un símbol nou per cada connectiva de la fórmula.
I generem les clàusules que "defineixen" el paper que juguen aquests símbols nous en la fórmula.

Per a expressar que p és el símbol d'un node AND de dos fills amb símbols a, b, necessitem $p \leftrightarrow a \wedge b$.

Per a això:

- expressem $p \rightarrow a \wedge b$ que és $p \rightarrow a$ i $p \rightarrow b$, amb dues clàusules: $\neg p \vee a \quad \neg p \vee b$
- expressem $p \leftarrow a \wedge b$ mitjançant una clàusula de tres literals: $\neg a \vee \neg b \vee p$.

En la presentació de la web de LI s'introdueixen també símbols i clàusules per als nodes NOT, però això no és necessari.

Per exemple, per a evitar el primer node NOT i el seu símbol $e3$, podem expressar directament que $e1 \Leftrightarrow e2 \vee \neg e4$, generant les clàusules:

$$
\begin{array}{l}
e1 \vee e2 \vee \neg e4, \\
\neg e2 \vee e1, \\
e4 \vee e1.
\end{array}
$$

Quins resultats obtenim?

Sigui F una fórmula.
Sigui Tseitin(F) la CNF de (el conjunt de les clàusules generades per) la transformació de Tseitin de F.

Llavors:

1. Tseitin(F) té clàusules de fins a 3 literals.
   Compte!: hi ha una clàusula unitària (d'1 només literal) que és el símbol auxiliar de l'arrel ($e1$ en l'exemple).
2. F i Tseitin(F) són EQUISATISFACTIBLES: F és satisfactible SSI Tseitin(F) és satisfactible
3. F i Tseitin(F) NO són logicamente equivalents
4. La mida de Tseitin(F) és lineal en la mida de F (3 clàusules per cada connectiva AND o OR de F) + l'arrell
5. Podem obtenir Tseitin(F) en temps lineal a partir de F
6. Podem reconstruir fàcilment un model de F a partir d'un model de Tseitin(F) ("oblidant-nos" dels símbols auxiliars)

Deducció en Lògica Proposicional

1. Tseitin(F) té clàusules de fins a 3 literals.
   Compte!: hi ha una clàusula unitària (d'1 només literal) que és el símbol auxiliar de l'arrel ($e1$ en l'exemple).
2. F i Tseitin(F) són EQUISATISFACTIBLES: F és satisfactible SSI Tseitin(F) és satisfactible
3. F i Tseitin(F) NO són logicamente equivalents
4. La mida de Tseitin(F) és lineal en la mida de F (3 clàusules per cada connectiva AND o OR de F) + l'arrell
5. Podem obtenir Tseitin(F) en temps lineal a partir de F
6. Podem reconstruir fàcilment un model de F a partir d'un model de Tseitin(F) ("oblidant-nos" dels símbols auxiliars)

Nota:

Sabent que SAT per a fórmules F qualssevol és NP-complet,
els punts 1,2,5 impliquen que 3-SAT també és NP-complet.

Deducció en Lògica Proposicional

Nota:

Si tenim una subfórmula amb ORs (o ANDs) niats, com a $p \vee (q \vee r)$ podem fer Tseitin com sempre, introduint per cada OR binari un símbol auxiliar i tres clàusules. Però també podem considerar que és una OR de tres entrades $p \vee q \vee r$, i generar un sol símbol auxiliar i quatre clàusules per a expressar $a \Leftrightarrow p \vee q \vee r$:

$\neg a \vee p \vee q \vee r$
$\neg p \vee a$
$\neg q \vee a$
$\neg r \vee a$

Això pot fer-se similarment per a ORs i ANDs de qualsevol nombre d'entrades.

Exercici: *The Transportation Company*

We need to plan the activities of a transportation company during a period of H hours. The company has T trucks, D drivers and there are N transportation tasks to be done, each one of which lasts one hour and needs one driver per truck.
Each task $i \in 1 \dots N$ needs K_i trucks, and has a list $L_i \subseteq \{1 \dots H\}$ of hours at which this task i can take place. For example, if $L_7 = \{3, 4, 8\}$ this means that task 7 can take place at hour 3, at hour 4 or at hour 8.
For each driver $d \in 1 \dots D$ there is a list of blockings $B_d \subseteq \{1 \dots H\}$ of hours at which driver D can *not* work.

Exercici: *The Transportation Company*

Explain how to use a SAT solver for planning this: for each task, when does it take place, and using which drivers. Clearly indicate which types of propositional variables you are using, and how many of each type, using the following format:

variables $t_{i,h}$ meaning "task i takes place at hour h"
for all tasks $i \in 1 \dots N$ and for all hours $h \in 1 \dots H$
Total: $N \cdot H$ variables.

Since H, D and N may be large, it is not allowed to use $O(H \cdot D \cdot N)$ variables (but using such a large number of clauses is fine).
Hint: you many use several types of variables, for example one type with $N \cdot H$ variables and another one with $N \cdot D$.

Also clearly indicate which clauses you need, and how many of each type, and how many literals each type of clause has. If you use any AMO, cardinality or pseudo-Boolean constraints, it is not necessary to convert these into CNF.

Codificació de restriccions numèriques en SAT

- AMO (*at most one*), ALO (*at least one*), *exactly one*:

$$l_1 + \cdots + l_n \leq 1$$
$$l_1 + \cdots + l_n \geq 1$$
$$l_1 + \cdots + l_n = 1$$

Codificació de restriccions numèriques en SAT

- AMO (*at most one*), ALO (*at least one*), *exactly one*:

$$l_1 + \cdots + l_n \leq 1$$
$$l_1 + \cdots + l_n \geq 1$$
$$l_1 + \cdots + l_n = 1$$

- *Cardinality constraints* en general:

$$l_1 + \cdots + l_n \leq K$$
$$l_1 + \cdots + l_n \geq K$$
$$l_1 + \cdots + l_n = K$$

Codificació de restriccions numèriques en SAT

- AMO (*at most one*), ALO (*at least one*), *exactly one*:

$$l_1 + \cdots + l_n \leq 1$$
$$l_1 + \cdots + l_n \geq 1$$
$$l_1 + \cdots + l_n = 1$$

- *Cardinality constraints* en general:

$$l_1 + \cdots + l_n \leq K$$
$$l_1 + \cdots + l_n \geq K$$
$$l_1 + \cdots + l_n = K$$

- *Pseudo-Boolean constraints*:

$$a_1 l_1 + \cdots + a_n l_n \leq K$$
$$a_1 l_1 + \cdots + a_n l_n \geq K$$
$$a_1 l_1 + \cdots + a_n l_n = K$$

Per al proper dia de classe:

- **Lògica de Primer Ordre (LPO)**
  Capítol 4 dels apunts: ☞ fitxer `p4.pdf`
- Exercicis del tema 4: $5, 6, \ldots$

Lògica en la Informàtica
Definició de Lògica de Primer Ordre (LPO)

José Miguel Rivero

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

Primavera 2025

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.

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)

Sumari

Recordatori: definició d'una Lògica

Recordem:

Què és una lògica?

- sintaxi:	-	què és una fórmula F ?
$+$		
- semàntica:	-a	què és una interpretació I ?
	-b	quan una I SATISFÀ una F ? $I \models F$?

Intuïtivament:

"Interpretació" $\equiv$ "situació de la vida real a modelar"

Una F "representa" aquelles I on se satisfà, es compleix.

Recordatori: definicions en qualsevol Lògica

Recordem:

Usem 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 $G \models F$.

Lògica de Primer Ordre vs. Lògica Proposicional

☞ Col·lecció d'apunts bàsics de lògica: `p4.pdf`

LPO: molt més poder expressiu que la LProp.
podem modelar moltes més coses de la vida real:
matemàtiques, verificació de programari, protocols, ...

LPO: deducció més costosa (en complexitat, decidibilitat)
que la LProp

Definició de la Lògica de Primer Ordre

☞ Col·lecció d'apunts bàsics de lògica: `p4.pdf`

LPO: Sintaxi:

símbols de variable: $\mathcal{X}$ notació: x,y,z (1)

(1) possiblement amb superíndexs o subíndexs

Definició de la Lògica de Primer Ordre

☞ Col·lecció d'apunts bàsics de lògica: `p4.pdf`

LPO: Sintaxi:

$$\left.\begin{array}{l}\left.\begin{array}{ll}\text{símbols de variable:} & \mathcal{X}\\ \text{símbols de funció:} & \mathcal{F}\end{array}\right\}\ \text{termes}\\ \begin{array}{ll}\text{símbols de predicat:} & \mathcal{P}\end{array}\quad \underline{\qquad\qquad}\end{array}\right\}\ \text{atoms}$$

Fórmules: àtoms combinats amb connectives $\wedge$ $\vee$ $\neg$ i amb quantificadors $\forall$ $\exists$
(compte amb la notació "text" que també es fa servir aquí: "per a tot" és A, "existeix" és E, etc.)

Exemple de Definició de LPO

Exemple:

$\mathcal{F}$ és:

f d'aritat 2	f^2
g d'aritat 1	g^1
h d'aritat 1	h^1
a d'aritat 0	a^0
b d'aritat 0	b^0

$\mathcal{P}$ és:

p d'aritat 2	p^2
q d'aritat 1	q^1
r d'aritat 0	r^0

Exemples de termes: $a \quad b \quad g(a) \quad f(x,a)$

$f(f(a,b),x) \qquad f(g(a),g(g(f(a,x)))) \quad \ldots$

de fet, si només tinc un símbol unari ja puc fabricar infinits termes:

$x \quad h(x) \quad h(h(x)) \quad h(h(h(x))) \quad \ldots$

Exemples d'àtoms: $r \quad q(a) \quad q(f(a,b)) \quad q(h(h(x))) \quad p(a,h(x)) \quad \ldots$

Exemples de fórmules:

$$F = \forall x\, \exists y\, (p(x,h(y)) \vee q(f(x,y)))$$

$$F' = \forall x\, p(g(x),a) \vee \exists y\, q(f(y,y))$$

Definició de la Lògica de Primer Ordre

LPO: Semàntica:

Una I consta de tres parts:

D_I: "el domini" de I (un conjunt no buit)

f_I: per cada símbol de funció f d'aritat n,

una funció $f_I : \overbrace{D_I \times \cdots \times D_I}^{n \texttt{ args}} \to D_I$ "la interpretació de f en I"

p_I: per cada símbol de predicat p d'aritat n,

una funció $p_I : \overbrace{D_I \times \cdots \times D_I}^{n \texttt{ args}} \to \{0, 1\}$ "la interpretació de p en I"

Intuïtivament, és com si hi hagués dos TIPUS: els Booleans i "els altres" (els elements de D_I).

F: prenen arguments de D_I i retornen D_I.

P: prenen arguments de D_I i retornen un Booleà.

PER AIXÒ NO TÉ SENTIT NIAR SÍMBOLS DE PREDICAT.

Exemple de Definició de LPO

Exemple (cont.):

$\mathcal{F}$ és:

f d'aritat 2	f^2
g d'aritat 1	g^1
h d'aritat 1	h^1
a d'aritat 0	a^0
b d'aritat 0	b^0

$\mathcal{P}$ és:

p d'aritat 2	p^2
q d'aritat 1	q^1
r d'aritat 0	r^0

Exemple de Definició en LPO

Exemple d'I:

$D_I = \{\circ, \$\}$

f_I: $D_I \times D_I \to D_I$

defineixo aquesta funció donant tots els casos:

$f_I(\$, \$) = \$$

$f_I(\$, \circ) = \circ$

$f_I(\circ, \$) = \$$

$f_I(\circ, \circ) = \$$

g_I: $D_I \to D_I$

$g_I(\$) = \circ$

$g_I(\circ) = \$$

h_I: $D_I \to D_I$

$h_I(\$) = \$$

$h_I(\circ) = \circ$

$a_I = \circ$

$b_I = \$$

Exemple de Definició en LPO

Exemple d'I (cont.):

$D_I = \{\circ, \$\}$

p_I: $D_I \times D_I \to \{0,1\}$
defineixo aquesta funció donant tots els casos:
$p_I(\$,\$) = 1$
$p_I(\$,\circ) = 0$
$p_I(\circ,\$) = 0$
$p_I(\circ,\circ) = 1$

q_I: $D_I \to \{0,1\}$
defineixo aquesta funció donant tots els casos:
$q_I(\$) = 1$
$q_I(\circ) = 0$

$r_I = 1$

Exemple de Definició en LPO

Tenim $I \models F$?

$$\forall x\, \exists y\, (p(x, h(y)) \vee q(f(x, y)))$$

$$\begin{aligned} p_I(\$, \$) &= 1 \\ p_I(\$, \circ) &= 0 \\ p_I(\circ, \$) &= 0 \\ p_I(\circ, \circ) &= 1 \end{aligned}$$

$$\begin{aligned} h_I(\$) &= \$ \\ h_I(\circ) &= \circ \end{aligned}$$

com p_I s'interpreta com a igualtat, i la h_I és la funció identitat (que "no fa res"), tenim que $\forall x\, \exists y\, p(x, h(y))$ es compleix: per a tota x del domini hi ha una y que és igual:

si $x = \$$ triem que la y sigui també \$

si $x = \circ$ triem que la y sigui també o

ni tan sols cal mirar la part $q(f(x, y))$.

Tenim que $I \models F$.

Exemple de Definició en LPO

Un altre exemple d'interpretació:

$D_I = \mathbb{N}$ (els nombres naturals)

f_I d'aritat 2 la suma de naturals: $f_I(n,m) = n + m$
g_I d'aritat 1 la funció "successor": $g_I(n) = n + 1$
h_I d'aritat 1 la funció "doble": $h_I(n) = 2n$
a_I d'aritat 0 7
b_I d'aritat 0 23

p_I d'aritat 2 l'ordre estricte de naturals: $p_I(n,m) = (n > m)$
q_I d'aritat 1 ens diu si és parell: $q_I(n) = (n \bmod 2 = 0)$
r_I d'aritat 0 0

Ara tenim $I \models F$?

$$\forall x\, \exists y\, (p(x, h(y)) \vee q(f(x,y)))$$

per a tota x existeix una y tal que $x > 2y$ o $x + y$ és parell ?
Això és cert, perquè per a tota x podem triar la y que sigui la mateixa x i llavors $x + y = x + x$ que és parell.
(no necessitem la primera meitat de l'or)

Noció d'avaluació d'una F en una I

☞ veure p4.pdf

- Sintàxi
- Interpretació
- Satisfacció
  - Assignació
  - Avaluació de termes
  - Avaluació de fórmules
  - Noció de satisfacció
- Fórmules tancades
  - Aparicions lliures i lligades de variables
  - Fórmules tancades
  - Avaluació de fórmules tancades
  - Satisfacció de fórmules tancades

Exercici 5

5. (dificultat 1) Sigui F la fórmula

$$\exists x\, \exists y\, \exists z\, (p(x,y) \wedge p(z,y) \wedge p(x,z) \wedge \neg p(z,x)).$$

Quines de les següents interpretacions són models de F ?

a) $D_I = \mathbb{N}$ i $p_I(m,n) = 1$ si i només si $m \leq n$.

b) $D_I = \mathbb{N}$ i $p_I(m,n) = 1$ si i només si $n = m + 1$.

c) $D_I = \mathcal{P}(\mathbb{N})$ (això denota parts de $\mathbb{N}$, és a dir, el conjunt de tots els subconjunts de $\mathbb{N}$), i $p_I(A,B) = 1$ si i només si $A \subseteq B$.

En format "text":

```
Ex Ey Ez ( p(x, y) & p(z, y) & p(x, z) & -p(z, x) )
```

Exercici 5

5. (dificultat 1) Sigui F la fórmula

$$\exists x\, \exists y\, \exists z\, (p(x,y) \wedge p(z,y) \wedge p(x,z) \wedge \neg p(z,x)).$$

a) $D_I = \mathbb{N}$ i $p_I(m,n) = 1$ si i només si $m \leq n$.

$\exists x\, \exists y\, \exists z\, (p(x,y) \wedge p(z,y) \wedge p(x,z) \wedge \neg p(z,x))$ s'avalua com

$$\begin{array}{cccc} x \leq y & z \leq y & x \leq z & z > x \\ 1 \quad 3 & 2 \quad 3 & 1 \quad 2 & 2 \quad 1 \end{array}$$

Sí, $I \models F$.

Exercici 5

5. (dificultat 1) Sigui F la fórmula

$$\exists x\, \exists y\, \exists z\, (p(x,y) \wedge p(z,y) \wedge p(x,z) \wedge \neg p(z,x)).$$

b) $D_I = \mathbb{N}$ i $p_I(m,n) = 1$ si i només si $n = m+1$.

$$\exists x\, \exists y\, \exists z\, (p(x,y) \wedge p(z,y) \wedge p(x,z) \wedge \neg p(z,x)) \quad \text{s'avalua com}$$

$$\underbrace{\underbrace{\underbrace{y = x{+}1 \quad y = z{+}1}_{x\,=\,z} \quad \underbrace{z = x{+}1}_{z\,=\,x+1}}_{\text{NO}}} \quad x \neq z{+}1$$

NO, I no és model de F.

Exercici 5

5. (dificultat 1) Sigui F la fórmula

$$\exists x\, \exists y\, \exists z\, (p(x,y) \wedge p(z,y) \wedge p(x,z) \wedge \neg p(z,x)).$$

c) $D_I = \mathcal{P}(\mathbb{N})$ i $p_I(A,B) = 1$ si i només si $A \subseteq B$.

$$\exists x\, \exists y\, \exists z\, (\underbrace{p(x,y)}_{\substack{x \subseteq y \\ \{1\}\ \{1,2,3\}}} \wedge \underbrace{p(z,y)}_{\substack{z \subseteq y \\ \{1,2\}\ \{1,2,3\}}} \wedge \underbrace{p(x,z)}_{\substack{x \subseteq z \\ \{1\}\ \{1,2\}}} \wedge \underbrace{\neg\, p(z,x)}_{\substack{z \not\subseteq x \\ \{1,2\}\ \{1\}}}) \quad \text{s'avalua}$$

Sí, $I \models F$.

Exercici 6

6. (dificultat 2) Expressa amb tres fórmules les propietats de reflexivitat, simetria i transitivitat d'un predicat binari p i demostra que cap de les tres fórmules és conseqüència lògica de (la conjunció de) les altres dues.

Una interpretació p_I d'un predicat binari p, és una funció $p_I : D_I \times D_I \to \{0, 1\}$. Ens adonem que en realitat p_I és el mateix que una relació binària sobre D_I:

p_I ens diu quines parelles d'elements de D_I donen 1 (estan en la relació).

Exercici 6

6. (dificultat 2) Expressa amb tres fórmules les propietats de reflexivitat, simetria i transitivitat d'un predicat binari p i demostra que cap de les tres fórmules és conseqüència lògica de (la conjunció de) les altres dues.

Recordem:

p és **reflexiu**		$p(e,e)$	per a tot e de S.
	FR: $\forall x\, p(x,x)$		
p és **simètric**	si $p(e,e')$	implica $p(e',e)$	per a tot e, e' de S.
	FS: $\forall x\, \forall y\, (p(x,y) \to p(y,x))$		
p és **transitiu**	si $p(e,e')$ i $p(e',e'')$	implica $p(e,e'')$	per a tot e, e', e'' de S.
	FT: $\forall x\, \forall y\, \forall z\, (p(x,y) \wedge p(y,z) \to p(x,z))$		

Exercici 6

1r cas: FR no és conseqüència lògica de $FS \wedge FT$.

Sigui I la interpretació on $D_I = \{*\}$ i $p_I(*,*) = 0$.
Llavors tenim que I no és model de FR.
Però I sí que és model de FS:

$$\forall x \, \forall y \, (p(x,y) \rightarrow p(y,x)) \equiv \forall x \, \forall y \, (\neg p(x,y) \vee p(y,x))$$

i I també és model de FT:

$$\forall x \, \forall y \, \forall z \, (p(x,y) \wedge p(y,z) \rightarrow p(x,z)) \equiv \forall x \, \forall y \, \forall z \, (\neg p(x,y) \vee \neg p(y,z) \vee p(x,z))$$

Per tant, tenim que FR no és conseqüència lògica de $FS \wedge FT$.

Exercici 6

2n cas: FS no és conseqüència lògica de $FR \wedge FT$.

Sigui I la interpretació on $D_I = \{a, b\}$

$p_I(a, a) = 1$ (per reflexivitat)
$p_I(a, b) = 1$ per a incomplir la simetria, juntament amb la línia següent
$p_I(b, a) = 0$ per a incomplir la simetria, juntament amb la línia anterior
$p_I(b, b) = 1$ (per reflexivitat).

Tenim que I no és model de FS, però sí de FR i de FT.

Per tant, tenim que FS no és conseqüència lògica de $FR \wedge FT$.

Exercici 6

3r cas: FT no és conseqüència lògica de $FR \wedge FS$.

Sigui I la interpretació on $D_I = \{a, b, c\}$

Imposem, per aquest ordre:

FR : per reflexivitat

$\neg FT$: per a incomplir la transitivitat

FS : per simetria

Exercici 6

3r cas: FT no és conseqüència lògica de $FR \wedge FS$.

Sigui I la interpretació on $D_I = \{a, b, c\}$

		FR	$\neg FT$	FS
$p_I(a,a)$	$=$	1		
$p_I(a,b)$	$=$		1	
$p_I(a,c)$	$=$		0	
$p_I(b,a)$	$=$			1
$p_I(b,b)$	$=$	1		
$p_I(b,c)$	$=$		1	
$p_I(c,a)$	$=$			0
$p_I(c,b)$	$=$			1
$p_I(c,c)$	$=$	1		

Tenim que I no és model de FT, però sí de FR i de FS.

Per tant, tenim que FT no és conseqüència lògica de $FR \wedge FS$.

Definició de la Lògica de Primer Ordre

Exercicis del capítol `p4.pdf` per al proper dia:

☞ 7, 8, 9, 10, 11, 12, 16, 21 en endavant.

Lògica en la Informàtica
Definició de Lògica de Primer Ordre (LPO)

José Miguel Rivero

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

Primavera 2025

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.

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)

Sumari

Exercici 7

7. (dificultat 3) Sigui $\mathcal{F}$ el conjunt $\{f^2, c^0\}$ i sigui $\mathcal{P}$ el conjunt $\{p^2\}$. Considerem una interpretació I tal que $D_I = \mathbb{N}$ i $p_I(n, m) = 1$ si i només si $n \leq m$.
Troba tres interpretacions diferents per a f i c que satisfacin la fórmula

$$\forall x\, (p(f(x, c), x) \wedge p(x, f(x, c)))$$

i que només dues d'elles satisfacin la fórmula

$$\forall x\, \forall y\, (p(f(x, y), f(y, x)) \wedge p(f(y, x), f(x, y)))$$

$\forall x\, (p(f(x, c), x) \wedge p(x, f(x, c))$

$\forall x\ (f(x, c) \leq x\ \wedge\ x \leq f(x, c))$ això implica que: $\forall x\ f(x, c) = x$

Exercici 7

7. (dificultat 3) Sigui $\mathcal{F}$ el conjunt $\{f^2, c^0\}$ i sigui $\mathcal{P}$ el conjunt $\{p^2\}$. Considerem una interpretació I tal que $D_I = \mathbb{N}$ i $p_I(n, m) = 1$ si i només si $n \leq m$.
Troba tres interpretacions diferents per a f i c que satisfacin la fórmula

$$\forall x\, (p(f(x, c), x) \wedge p(x, f(x, c)))$$

i que només dues d'elles satisfacin la fórmula

$$\forall x\, \forall y\, (p(f(x, y), f(y, x)) \wedge p(f(y, x), f(x, y)))$$

1a I: f_I és la suma, i c_I és 0

2a I: f_I és el producte, i c_I és 1

3a I: $f_I(n, m) = n$, i c_I és qualsevol natural, per exemple el 7

Exercici 7

7. (dificultat 3) Sigui $\mathcal{F}$ el conjunt $\{f^2, c^0\}$ i sigui $\mathcal{P}$ el conjunt $\{p^2\}$. Considerem una interpretació I tal que $D_I = \mathbb{N}$ i $p_I(n, m) = 1$ si i només si $n \leq m$.
Troba tres interpretacions diferents per a f i c que satisfacin la fórmula

$$\forall x\,(p(f(x,c),x) \wedge p(x,f(x,c)))$$

i que només dues d'elles satisfacin la fórmula

$$\forall x\,\forall y\,(p(f(x,y),f(y,x)) \wedge p(f(y,x),f(x,y)))$$

$$\forall x\,\forall y\;\;(f(x,y) \leq f(y,x) \wedge f(y,x) \leq f(x,y))$$

Exercici 7

7. (dificultat 3) Sigui $\mathcal{F}$ el conjunt $\{f^2, c^0\}$ i sigui $\mathcal{P}$ el conjunt $\{p^2\}$. Considerem una interpretació I tal que $D_I = \mathbb{N}$ i $p_I(n,m) = 1$ si i només si $n \leq m$.
Troba tres interpretacions diferents per a f i c que satisfacin la fórmula

$$\forall x\, (p(f(x,c),x) \wedge p(x, f(x,c)))$$

i que només dues d'elles satisfacin la fórmula

$$\forall x\, \forall y\, (p(f(x,y), f(y,x)) \wedge p(f(y,x), f(x,y)))$$

Nota: això implica que $\forall x\, \forall y\; f(x,y) = f(y,x)$, ja que $n \leq m$ i $m \leq n \;\rightarrow\; n = m$. És a dir, f_I ha de ser commutativa.

Exercici 7

7. (dificultat 3) Sigui $\mathcal{F}$ el conjunt $\{f^2, c^0\}$ i sigui $\mathcal{P}$ el conjunt $\{p^2\}$. Considerem una interpretació I tal que $D_I = \mathbb{N}$ i $p_I(n, m) = 1$ si i només si $n \leq m$.
Troba tres interpretacions diferents per a f i c que satisfacin la fórmula

$$\forall x \, (p(f(x, c), x) \wedge p(x, f(x, c)))$$

i que només dues d'elles satisfacin la fórmula

$$\forall x \, \forall y \, (p(f(x, y), f(y, x)) \wedge p(f(y, x), f(x, y)))$$

En la 1a I, $f_I(n, m) = n + m$ (la suma) és commutativa
En la 2a I, $f_I(n, m) = n \cdot m$ (el producte) és commutativa
En la 3a I, $f_I(n, m) = n$ NO és commutativa. ok.

Una altra opció amb f_I no commutativa: $f_I(n, m) = n^m$ i $c_I = 1$.

Exercici 8

8. (dificultat 2) Sigui F la fórmula
$\forall x \, \exists y \; p(x,y) \; \wedge \; \forall x \, \exists y \; \neg p(x,y)$. F és satisfactible? Demostra-ho.

Sigui I la interpretació tal que:

$D_I = \mathbb{Z}$ (els enters)

$p_I(n,m) = n < m$.

Aquesta I és un model.

Exercici 8

8. (dificultat 2) Sigui F la fórmula
$\forall x\, \exists y\, p(x,y) \ \wedge\ \forall x\, \exists y\, \neg p(x,y)$. F és satisfactible? Demostra-ho.

Un altre model:

$D_I = \{a, b\}$
$p_I(a,a) = 1$
$p_I(a,b) = 0$
$p_I(b,a) = 0$
$p_I(b,b) = 1$

Exercici 8

8. (dificultat 2) Sigui F la fórmula
$\forall x\, \exists y\, p(x,y) \,\wedge\, \forall x\, \exists y\, \neg p(x,y)$. F és satisfactible? Demostra-ho.

Si I és una interpretació, diem que el nombre d'elements d'I és $|D_I|$, el nombre d'elements de D_I. Així mateix, diem que I és un model finit quan D_I és finit, i parlem de la cardinalitat d'I per a referir-nos a la cardinalitat de D_I.

Quin és el mínim nombre d'elements que ha de tenir un model de F?

$D_I = \{a\}$
tant si definim $p_I(a,a) = 1$ com si definim $p_I(a,a) = 0$, la fórmula no es compleix! Per tant, no és possible amb 1 sol element en el domini, però sí amb 2.

Exercici 9

9. (dificultat 3) Considera els conjunts de símbols i parells d'interpretacions I_1 i I_2 següents. Per a cada cas, dona una fórmula F que és una certa en una d'elles i falsa en l'altra, i raona informalment per què és així.

a) $\mathcal{P} = \{r^2\}$,
I_1 té com a domini els naturals i el predicat s'interpreta com l'ordre (és a dir $r_I(n,m) = 1$ si i només si $n \leq m$);
I_2 té com a domini els enters i el predicat s'interpreta també com l'ordre.

$$D_{I_1} = \mathbb{N} \qquad r_{I_1}(n,m) = (n \leq m)$$
$$D_{I_2} = \mathbb{Z} \qquad r_{I_2}(n,m) = (n \leq m)$$

per a tot enter existeix un altre menor estricte
però: per a tot natural NO existeix un altre menor estricte (perquè és fals per al zero)

9. (dificultat 3) Considera els conjunts de símbols i parells d'interpretacions I_1 i I_2 següents. Per a cada cas, dona una fórmula F que és una certa en una d'elles i falsa en l'altra, i raona informalment per què és així.

a) $\mathcal{P} = \{r^2\}$,
I_1 té com a domini els naturals i el predicat s'interpreta com l'ordre (és a dir $r_I(n, m) = 1$ si i només si $n \leq m$);
I_2 té com a domini els enters i el predicat s'interpreta també com l'ordre.

Sigui F la fórmula $\forall x \, \exists y \, \neg r(x, y)$. $[\neg r(x, y) = \neg(x \leq y) = x > y]$
Tenim que I_1 no és model de F i I_2 sí que és model de F.
I_1 no és model perquè si x és 0, llavors no existeix cap y tal que $\neg(0 \leq y)$, és a dir, tal que $0 > y$.
En canvi, en I_2, els enters, per a tota x sí que existeix una y tal que $x > y$.

Exercici 9

9. (dificultat 3) Considera els conjunts de símbols i parells d'interpretacions I_1 i I_2 següents. Per a cada cas, dona una fórmula F que és una certa en una d'elles i falsa en l'altra, i raona informalment per què és així.

b) $\mathcal{P} = \{r^2\}$,
I_1 té com a domini els enters i el predicat s'interpreta com l'ordre;
I_2 té com a domini els racionals i el predicat s'interpreta també com l'ordre.

$D_{I_1} = \mathbb{Z} \qquad r_{I_1}(n, m) = (n \leq m)$
$D_{I_2} = \mathbb{Q} \qquad r_{I_2}(n, m) = (n \leq m)$
per a tot parell d'elements x i y tals que $x > y$, existeix un z tal que $x > z \wedge z > y$ (es diu que els racionals són "densos")

9. (dificultat 3) Considera els conjunts de símbols i parells d'interpretacions I_1 i I_2 següents. Per a cada cas, dona una fórmula F que és una certa en una d'elles i falsa en l'altra, i raona informalment per què és així.

b) $\mathcal{P} = \{r^2\}$,
I_1 té com a domini els enters i el predicat s'interpreta com l'ordre;
I_2 té com a domini els racionals i el predicat s'interpreta també com l'ordre.

Si F és la fórmula:

$$\forall x\, \forall y\, (\underbrace{\neg r(x,y)}_{x>y} \to \exists z\, (\underbrace{\neg r(x,z)}_{x>z} \land \underbrace{\neg r(z,y)}_{z>y}))$$

llavors $I_2 \models F$, però $I_1 \not\models F$. I_2 és model de F però I_1 no ho és.

Exercici 9

9. (dificultat 3) Considera els conjunts de símbols i parells d'interpretacions I_1 i I_2 següents. Per a cada cas, dona una fórmula F que és una certa en una d'elles i falsa en l'altra, i raona informalment per què és així.

c) $\mathcal{P} = r^2$. El domini tant de I_1 com de I_2 són els números enters, per a
I_1 el predicat r s'interpreta com "tenir el mateix resto módul 2", i per
I_2 el predicat r s'interpreta com "tenir el mateix resto módul 3".

$$D_{I_1} = \mathbb{Z} \qquad r_{I_1}(n, m) = (n \bmod 2 = m \bmod 2)$$

"tenir la mateixa paritat"

$$D_{I_2} = \mathbb{Z} \qquad r_{I_2}(n, m) = (n \bmod 3 = m \bmod 3)$$

"tenir la mateixa "triaritat"???"

Exercici 9

9. (dificultat 3) Considera els conjunts de símbols i parells d'interpretacions I_1 i I_2 següents. Per a cada cas, dona una fórmula F que és una certa en una d'elles i falsa en l'altra, i raona informalment per què és així.

c) $\mathcal{P} = r^2$. El domini tant de I_1 com de I_2 són els números enters, per a
I_1 el predicat r s'interpreta com "tenir el mateix resto módul 2", i per
I_2 el predicat r s'interpreta com "tenir el mateix resto módul 3".

F expressa que hi ha tres elements amb diferent paritat:
Si F és la fórmula: $\exists x\, \exists y\, \exists z\; (\neg r(x, y) \wedge \neg r(x, z) \wedge \neg r(y, z))$
llavors $I_2 \models F$, però $I_1 \not\models F$. I_2 és model de F
però I_1 no ho és.

Exercici 10

10. (dificultat 2) Suposa que en $\mathcal{P}$ només hi ha símbols de predicat de aritat zero. Llavors, la sintaxi de les fórmules, en què es diferencia de la de la lògica proposicional? I la semàntica?

Si només hi ha símbols de predicat de aritat zero $(p,q,r,\ldots)$ quines fórmules hi ha?
Sintaxi:
Els àtoms seran $p,q,r,\ldots$ sense termes, i, per tant, sense variables, i, per tant, les fórmules seran sense quantificadors.
Les fórmules seran combinacions de $p,q,r,\ldots$ amb connectives $\wedge$, $\vee$, $\neg$.
SÓN les fórmules de la lògica proposicional.

Exercici 10

10. (dificultat 2) Suposa que en $\mathcal{P}$ només hi ha símbols de predicat de aritat zero. Llavors, la sintaxi de les fórmules, en què es diferencia de la de la lògica proposicional? I la semàntica?

Semàntica:
Encara que hi hagués símbols de funció, aquests no sortiran en les fórmules, per la qual cosa la seva interpretació és irrellevant.
De la mateixa manera, D_I també és irrellevant, perquè no hi ha variables ni quantificadors en les fórmules.
Queda definir en I com s'interpreten els símbols de predicat (que són tots de aritat zero):

$p_I: \quad \rightarrow \{0,1\}$
$q_I: \quad \rightarrow \{0,1\}$
$r_I: \quad \rightarrow \{0,1\}$
etc.

En què es diferencia això d'una I en lògica proposicional, que era: $I : \mathcal{P} \rightarrow \{0,1\}$? En Res.

Exercici 10

10. (dificultat 2) Suposa que en $\mathcal{P}$ només hi ha símbols de predicat de aritat zero. Llavors, la sintaxi de les fórmules, en què es diferencia de la de la lògica proposicional? I la semàntica?

Conclusió: la LProp és un cas (molt, molt) particular de la LPO.

16. (dificultat 2) Demostra alguna de les següents equivalències:

$$
\begin{array}{lll}
\neg\, \forall x\, F & \equiv & \exists x\, \neg F \\
\neg\, \exists x\, F & \equiv & \forall x\, \neg F \\
\forall x \forall y\, F & \equiv & \forall y \forall x\, F \\
\exists x \exists y\, F & \equiv & \exists y \exists x\, F \\
 & & \\
\forall x\, F \ \wedge\ \forall x\, G & \equiv & \forall x\, (F \wedge G) \\
\exists x\, F \ \vee\ \exists x\, G & \equiv & \exists x\, (F \vee G) \\
 & & \\
\forall x\, F \ \rightarrow\ \exists x\, G & \equiv & \exists x\, (F \rightarrow G) \\
 & & \\
\forall x\, F \ \vee\ G & \equiv & \forall x\, (F \vee G)\,, \ \text{si } x \text{ no es lliure en } G \\
\forall x\, F \ \wedge\ G & \equiv & \forall x\, (F \wedge G)\,, \ \text{si } x \text{ no es lliure en } G \\
\exists x\, F \ \vee\ G & \equiv & \exists x\, (F \vee G)\,, \ \text{si } x \text{ no es lliure en } G \\
\exists x\, F \ \wedge\ G & \equiv & \exists x\, (F \wedge G)\,, \ \text{si } x \text{ no es lliure en } G
\end{array}
$$

Exercici 17

17. (dificultat 2) Demostra que les equivalències següents **no** són certes en general (és a dir, per a qualsevol parell de fórmules F, G):

$$\begin{array}{lcl} \forall x\, F \;\vee\; \forall x\, G & \equiv & \forall x\,(F \vee G) \\ \exists x\, F \;\wedge\; \exists x\, G & \equiv & \exists x\,(F \wedge G) \end{array}$$

En tots dos casos, hi ha alguna de la dues fórmules que sigui conseqüència lògica de l'altra? (no cal que demostris això últim, ja ho farem quan tinguem un càlcul deductiu).

Exercici 18

18. (dificultat 2) Demostra que les fórmules $\forall x \, \exists y \, F$ i $\exists y \, \forall x \, F$ no són lògicament equivalents en general. Hi ha alguna de les dues fórmules que sigui conseqüència lògica de l'altra? (no cal que demostris això últim, ja ho farem quan tinguem un càlcul deductiu).

Definició de la Lògica de Primer Ordre

Per al proper dia de classe:

- ☞ Exercicis del capítol `p4.pdf`: 21, 22, 23.
- ☞ Lògica de Primer Ordre amb Igualtat (LPOI)

Lògica en la Informàtica
Definició de Lògica de Primer Ordre (LPO)

José Miguel Rivero

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

Primavera 2025

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.

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)

Sumari

Definició de la Lògica de Primer Ordre

Una fórmula F "EXPRESSA" coses:
les propietats dels seus models.

Continguts: ☞ `p4.pdf`

- Exercicis: 21, 22, 23
  - 21 Expressar límit inferior de la mida dels models
  - 22 Expressar mida infinita del models
  - 23 Existència de models de mida superior
- Lògica de Primer Ordre amb Igualtat (LPOI)
- Exercicis de LPOI: 24, 26, 27, 28, 32

Exercici 21

21. (dificultat 3) Dona una fórmula F_3 tal que tot model de F_3 tingui almenys 3 elements. Generalitza-ho a n qualsevol.
Ajuda: defineix la propietat reflexiva d'un símbol de predicat binari p, i a més expressa que hi ha parells d'elements e_i i e_j en el domini tals que $p_I(e_i, e_j) = 0$.

Comencem així. Sigui F la fórmula:

$$\begin{array}{ll} \forall x\, p(x,x) & \quad\text{(reflexivitat)} \\ \quad\wedge & \\ \exists x\, \exists y\, \neg p(x,y) & \end{array}$$

Qualsevol model I de F tindrà almenys DOS elements:

$$\begin{array}{ll} D_I = \{e_1, e_2\} & \\ p_I(e_1, e_1) = 1 & \quad\text{(PER REFLEXIVITAT)} \\ p_I(e_1, e_2) = 0 & \\ p_I(e_2, e_1) = 0 & \\ p_I(e_2, e_2) = 1 & \quad\text{(PER REFLEXIVITAT)} \end{array}$$

Exercici 21

21. (dificultat 3) Dona una fórmula F_3 tal que tot model de F_3 tingui almenys 3 elements. Generalitza-ho a n qualsevol.
Ajuda: defineix la propietat reflexiva d'un símbol de predicat binari p, i a més expressa que hi ha parells d'elements e_i i e_j en el domini tals que $p_I(e_i, e_j) = 0$.

Comencem així. Sigui F la fórmula:

$$\begin{array}{ll} \forall x\, p(x,x) & \quad\text{(reflexivitat)} \\ \quad\wedge & \\ \exists x\, \exists y\, \neg p(x,y) & \end{array}$$

Perquè si hi hagués només un:

$$D_I = \{e1\}$$

tindríem

$$p_I(e_1, e_1) = 1 \qquad \text{(PER REFLEXIVITAT)}$$

i no es compliria la part $\exists x\, \exists y\, \neg p(x,y)$.

Exercici 21

21. (dificultat 3) Dona una fórmula F_3 tal que tot model de F_3 tingui almenys 3 elements. Generalitza-ho a n qualsevol.
Ajuda: defineix la propietat reflexiva d'un símbol de predicat binari p, i a més expressa que hi ha parells d'elements e_i i e_j en el domini tals que $p_I(e_i, e_j) = 0$.

Ho podem generalitzar a tres o més elements, així:

Sigui F la formula:

$$\begin{array}{ll} \forall x\, p(x,x) & \text{(reflexivitat)} \\ \quad\wedge & \\ \exists x\, \exists y\, \exists z\, (\neg p(x,y) \wedge \neg p(x,z) \wedge \neg p(y,z)) & \end{array}$$

Exercici 21

21. (dificultat 3) Dona una fórmula F_3 tal que tot model de F_3 tingui almenys 3 elements. Generalitza-ho a n qualsevol.
Ajuda: defineix la propietat reflexiva d'un símbol de predicat binari p, i a més expressa que hi ha parells d'elements e_i i e_j en el domini tals que $p_I(e_i, e_j) = 0$.

I en general per a mínim n elements en el domini:

$$\forall x\, p(x,x) \qquad \text{(reflexivitat)}$$
$$\wedge$$
$$\exists x_1 \cdots \exists x_n\, (\neg p(x_1, x_2) \wedge \neg p(x_1, x_3) \wedge \cdots \wedge \neg p(x_{n-1}, x_n))$$

(una fórmula de mida quadràtica)

Exercici 22

22. (dificultat 5) Escriu una fórmula F tal que si $I \models F$ llavors D_I té infinits elements. Ajuda: pensa en la relació "ser estrictament menor que" i expressa (entre altres coses) que "no hi ha màxim" tal com ocorre en els naturals.

Definició: un **ordre estricte** és una relacion binària irreflexiva i transitiva.

Usem un símbol binari p que té aquestes dues propietats.
Sigui F la fórmula:

$$\forall x\, \neg p(x,x) \qquad \text{(irreflexivitat)}$$
$$\wedge$$
$$\forall x\, \forall y\, \forall z\, (p(x,y) \wedge p(y,z) \rightarrow p(x,z)). \qquad \text{(transitivitat)}$$

equivalentment: $\forall x\, \forall y\, \forall z\, (\neg p(x,y) \vee \neg p(y,z) \vee p(x,z))$

Exercici 22

22. (dificultat 5) Escriu una fórmula F tal que si $I \models F$ llavors D_I té infinits elements. Ajuda: pensa en la relació "ser estrictament menor que" i expressa (entre altres coses) que "no hi ha màxim" tal com ocorre en els naturals.

Sigui F la fórmula:

$$\begin{array}{ll} \forall x\, \neg p(x,x) & \text{(irreflexivitat)} \\ \quad \wedge & \\ \forall x\, \forall y\, \forall z\, (p(x,y) \wedge p(y,z) \rightarrow p(x,z)). & \text{(transitivitat)} \end{array}$$

En qualsevol model I de F, tenim que p_I és una relacion d'ordre estricte sobre D_I.

Això fa que necessitem que D_I sigui infinit en qualsevol model I de F? No, perquè tindríem el model de F:

$D_I = \{a\}$

$p_I(a,a) = 0$

Exercici 22

22. (dificultat 5) Escriu una fórmula F tal que si $I \models F$ llavors D_I té infinits elements. Ajuda: pensa en la relació "ser estrictament menor que" i expressa (entre altres coses) que "no hi ha màxim" tal com ocorre en els naturals.

Per això afegim: $\forall x \, \exists y \; p(x,y)$ a la nostra F:

$$\forall x \, \neg p(x,x) \qquad \text{(irreflexivitat)}$$
$$\wedge$$
$$\forall x \, \forall y \, \forall z \; (p(x,y) \wedge p(y,z) \rightarrow p(x,z)) \qquad \text{(transitivitat)}$$
$$\wedge$$
$$\forall x \, \exists y \; p(x,y) \qquad \text{("existència de successors")}$$

Per què aquesta F només té models infinits?

- Reducció a l'absurd.

Exercici 22

22. (dificultat 5) Escriu una fórmula F tal que si $I \models F$ llavors D_I té infinits elements. Ajuda: pensa en la relació "ser estrictament menor que" i expressa (entre altres coses) que "no hi ha màxim" tal com ocorre en els naturals.

Suposem que existís un model finit I, amb:

$D_I = \{e_1 \ldots e_k\}$

Per la part $\forall x\, \exists y\; p(x, y)$, necessito que $p_I(e_1, e)=1$ per a algun element "e" de D_I. Diguem-li e_2 a aquest element e.

També necessito

$p_I(e_2, e)=1$ per a algun element "e" de D_I. No pot ser e_2, ni tampoc e_1: tindriem $p_I(e_1, e_2)$ i $p_I(e_2, e_1)$ i per transitivitat tindríem $p_I(e_1, e_1)$ que contradiu la irreflexivitat. Per tant, el successor de e_2 ha de ser un element al qual podem anomenar e_3.

Exercici 22

22. (dificultat 5) Escriu una fórmula F tal que si $I \models F$ llavors D_I té infinits elements. Ajuda: pensa en la relació "ser estrictament menor que" i expressa (entre altres coses) que "no hi ha màxim" tal com ocorre en els naturals.

Suposem que existís un model finit I, amb:

$D_I = \{e_1 \ldots e_k\}$

També necessito

$p_I(e_3, e) = 1$ per a algun element "e" de D_I. Per les mateixes raons, no pot ser e_3 ni e_2, ni tampoc e_1.

Exercici 22

22. (dificultat 5) Escriu una fórmula F tal que si $I \models F$ llavors D_I té infinits elements. Ajuda: pensa en la relació "ser estrictament menor que" i expressa (entre altres coses) que "no hi ha màxim" tal com ocorre en els naturals.

Suposem que existís un model finit I, amb:
$D_I = \{e_1 \dots e_k\}$

Una vegada hem entès això, (per inducció) podem demostrar (no ho farem aquí) que no podem introduir "cicles" en la relacion p_I, del tipus:

$$p_I(e_1, e_2) \wedge p_I(e_2, e_3) \wedge \cdots \wedge p_I(e_n, e_1)$$

La qual cosa ens porta a una contradicció, perquè... qui serà el successor de e_k?
Ningú!

Exercici 23

23. (dificultat 5) Demostra que si una fórmula té models amb n elements, també té models amb m elements per a qualsevol $m \geq n$ i fins i tot models infinits.

Això és una altra manera de dir que NO podem expressar amb una fórmula F, que els models de F tindran com a màxim 2 elements, o com a màxim k elements, per a alguna k.

Exercici 23

23. (dificultat 5) Demostra que si una fórmula té models amb n elements, també té models amb m elements per a qualsevol $m \geq n$ i fins i tot models infinits.

Exemple de com "clonar" un element "a" de D_I: tinc p de aritat 2, i tinc la interpretació I amb:

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

$$p_I(a, a) = 1$$
$$p_I(a, b) = 0$$
$$p_I(b, a) = 1$$
$$p_I(b, b) = 0$$

Exercici 23

23. (dificultat 5) Demostra que si una fórmula té models amb n elements, també té models amb m elements per a qualsevol $m \geq n$ i fins i tot models infinits.

Sigui F qualsevol formula tal que $I \models F$.

Clonar l'element a, afegint el seu clon a' obtenint una I' de manera que $I' \models F$:

$$D_{I'} = \{a, b, a'\}$$

$$\begin{array}{lcl}
p_{I'}(a,a) & = & 1 \\
p_{I'}(a,b) & = & 0 \\
p_{I'}(a,a') & = & \quad 1 \quad \longleftarrow \text{ amb } a',\ p_{I'} \text{ es comporta igual que amb } a \\
p_{I'}(b,a) & = & 1 \\
p_{I'}(b,b) & = & 0 \\
p_{I'}(b,a') & = & \\
p_{I'}(a',a) & = & \\
p_{I'}(a',b) & = & \\
p_{I'}(a',a') & = &
\end{array}$$

Exercici 23

23. (dificultat 5) Demostra que si una fórmula té models amb n elements, també té models amb m elements per a qualsevol $m \geq n$ i fins i tot models infinits.

Sigui F qualsevol formula tal que $I \models F$.

Clonar l'element a, afegint el seu clon a' obtenint una I' de manera que $I' \models F$:

$$D_{I'} = \{a, b, a'\}$$

$$\begin{array}{lcll}
p_{I'}(a,a) & = & 1 & \\
p_{I'}(a,b) & = & 0 & \\
p_{I'}(a,a') & = & & 1 \\
p_{I'}(b,a) & = & 1 & \\
p_{I'}(b,b) & = & 0 & \\
p_{I'}(b,a') & = & & 1 \\
p_{I'}(a',a) & = & & 1 \\
p_{I'}(a',b) & = & & 0 \\
p_{I'}(a',a') & = & & 1
\end{array}$$

F pot ser per exemple:

$$\forall x\, \exists y\, (p(x,y) \vee p(y,x))$$

Lògica de Primer Ordre amb Igualtat (LPOI)

- En els exercicis 21 i 22 vam veure que en LPO podem expressar que hi ha ALMENYS k elementos en el domini.
- Però en l'exercici 23, veiem que en LPO NO podem expressar que hi ha COM A MOLT k elements en el domini.
- Això és el que ens motiva a introduir una lògica més expressiva, la LPOI.

Expressivitat d'una lògica: quines situacions de la vida real podem descriure o distingir?

Per exemple, en LProp $\mathcal{P} = \{plou, fa\_sol, esta\_ennuvolat\}$ cada I "modela" una sitacion de la vida real: per exemple, NO *plou*, NO *fa_sol* i SÍ *esta_ennuvolat*. Una F el que fa és distingir un subconjunt de les I's: els MODELS de F.

Expressivitat d'una lògica: quines situacions de la vida real podem descriure o distingir?

En LPO el mateix, però les interpretacions són molt més complexes: quin domini hi ha, com s'interpreten els símbols.

> Amb una F podem distingir les I's que tenen almenys 2 elements en el seu D_I. O infinits elements.
>
> Però NO podem expressar que hi ha com a màxim 2 (o k) elements (exercici 23).

Això ens motiva a introduir una altra lògica que estén la LPO, que és la LPOI, que sí que permet expressar aquest tipus de coses.

Lògica de Primer Ordre amb Igualtat (LPOI)

Què és la LPOI?

Sintaxi: F: és com LPO, però hi ha un simbolo de predicat "predefinit" binari eq^2

Semantica: I: és com LPO, però eq_I sempre serà "ser el mateix element del domini"

$eq_I(e_1, e_1) = 1$ per a tot element e_1 de D_I

$eq_I(e_1, e_2) = 0$ si e_1 i e_2 són elements diferents de D_I

$I \models F \quad (eval_I(F))$ com LPO.

Exercicis de LPOI

- Exercici 24 Expressar límit superior de la mida dels models
- Exercici 26 Expressar mida exacta dels models
- Exercici 27 Monoide. Exemples de monoides
- Exercici 28 Grup. Exemples de grups
- Exercici 32 Fórmula que discrimina dues interpretacions

Exercici 24

24. (dificultat 2) Escriu una fórmula F de LPOI que expressi que per a tot model I de F:

a) hi ha com a màxim 1 element en el domini d'I

b) hi ha com a màxim 2 elements en el domini d'I

c) hi ha com a màxim n elements en el domini d'I, per a una n donada

d) hi ha exactament n elements en el domini d'I, per a una n donada

Exercici 24

24. (dificultat 2) Escriu una fórmula F de LPOI que expressi que per a tot model I de F:

a) hi ha com a màxim 1 element en el domini d'I

Tres maneres alternatives de fer-ho:

$\forall x\,\forall y\ eq(x,y)$	amb l'altra notació:	$\forall x\,\forall y\ x=y$
$\forall x\ eq(x,a)$		$\forall x\ x=a$
$\exists x\,\forall y\ eq(x,y)$		$\exists x\,\forall y\ x=y$

Exercici 24

24. (dificultat 2) Escriu una fórmula F de LPOI que expressi que per a tot model I de F:

b) hi ha com a màxim 2 elements en el domini d'I

Tres maneres alternatives de fer-ho:

$$\forall x\, \forall y\, \forall z\ (eq(x,y) \vee eq(x,z) \vee eq(y,z))$$

$$\forall x\ (eq(x,a) \vee eq(x,b))$$

$$\exists x\, \exists y\, \forall z\ (eq(x,z) \vee eq(y,z))$$

24. (dificultat 2) Escriu una fórmula F de LPOI que expressi que per a tot model I de F:

c) hi ha com a màxim n elements en el domini d'I, per a una n donada

Tres maneres alternatives de fer-ho:

$$\forall x_1 \cdots \forall x_{n+1} \left(\bigvee_{1\leq i<j\leq n+1} eq(x_i, x_j)\right)$$
(una fórmula de mida quadràtica)

$$\forall x \, (eq(x, a_1) \vee \cdots \vee eq(x, a_n))$$
(una fórmula de mida lineal)

$$\exists x_1 \cdots \exists x_n \forall y \, (eq(y, x_1) \vee \cdots \vee eq(y, x_n))$$
(una fórmula de mida lineal)

Exercici 24

24. (dificultat 2) Escriu una fórmula F de LPOI que expressi que per a tot model I de F:

d) hi ha exactament n elements en el domini d'I, per a una n donada

$$(\forall x\ (eq(x, a_1) \vee \cdots \vee eq(x, a_n)))\qquad\qquad \text{(màxim } n\text{)}$$

$$\wedge$$

$$(\textstyle\bigwedge_{1 \leq i < j \leq n} \neg eq(a_i, a_j))$$

(com a mínim n: una formula de mida quadràtica)

$$\neg eq(a_1, a_2) \wedge \neg eq(a_1, a_3) \wedge \cdots \wedge \neg eq(a_{n-1}, a_n)$$

Exercici 26

26. (dificultat 2)

   a) Sigui p un símbol de predicat unari. Escriu una fórmula F de LPOI que expressi que hi ha un únic element que compleix p. (en mates a vegades s'escriu $\exists! x\, p(x)$). Això vol dir: que expressi que per a tot model I de F hi ha un únic element a en D_I amb $p_I(a) = 1$.

   $\exists x\ (p(x) \wedge \forall y\, (\neg eq(x, y) \rightarrow \neg p(y)))$

   Una altra manera, amb una constant a:

   $p(a) \wedge \forall x\, (\neg eq(a, x) \rightarrow \neg p(x))$

Exercici 26

26. (dificultat 2)

    b) Escriu una altra F expressant que hi ha exactament 2.

    $$p(a) \wedge p(b) \wedge \neg eq(a, b) \wedge \forall x\, (\neg eq(a, x) \wedge \neg eq(b, x) \rightarrow \neg p(x)))$$

Exercici 27

27. (dificultat 2) Un *monoide* és un model de la següent fórmula:

$$\forall x\, \forall y\, \forall z\ \ (x \cdot y) \cdot z = x \cdot (y \cdot z) \qquad [\cdot \text{ es associatiu}]$$
$$\wedge$$
$$\forall x\ \ x \cdot e = x \qquad [e \text{ és l'element neutre per la dreta}]$$
$$\wedge$$
$$\forall x\ \ e \cdot x = x \qquad [e \text{ és l'element neutre per l'esquerra}]$$

on $\cdot$ és un símbol de funció binària i e és un símbol de constant. Observa que hem usat notació infix (com fem amb el símbol $=$ per a la igualtat). Amb la notació habitual (i amb f en comptes de $\cdot$) la fórmula $\forall x\, \forall y\, \forall z\ \ (x \cdot y) \cdot z = x \cdot (y \cdot z)$ s'escriuria $\forall x\, \forall y\, \forall z\ \ f(f(x, y), z) = f(x, f(y, z))$.

Exercici 27

27. (dificultat 2) Un *monoide* és un model de la següent fórmula:

$$\forall x\, \forall y\, \forall z\ \ (x \cdot y) \cdot z = x \cdot (y \cdot z) \qquad [\,\cdot \text{ es associatiu}\,]$$

$$\wedge$$

$$\forall x\ \ x \cdot e = x \qquad [\, e \text{ és l'element neutre per la dreta}\,]$$

$$\wedge$$

$$\forall x\ \ e \cdot x = x \qquad [\, e \text{ és l'element neutre per l'esquerra}\,]$$

Notació: $eq(x, y)$ $\quad x = y$

Notació: $\cdot(x, y)$ $\quad x \cdot y$ $\quad$ símbol de funció binari.

En notació prefix la associativitat seria:

$$\forall x\, \forall y\, \forall z\ \ \cdot(\cdot(x, y), z) = \cdot(x, \cdot(y, z))$$

Exercici 27

27. (dificultat 2) Un *monoide* és un model de la següent fórmula:

$$\forall x\, \forall y\, \forall z\ \ (x \cdot y) \cdot z = x \cdot (y \cdot z) \qquad [\,\cdot \text{ es associatiu}\,]$$
$$\wedge$$
$$\forall x\ \ x \cdot e = x \qquad [\,e \text{ és l'element neutre per la dreta}\,]$$
$$\wedge$$
$$\forall x\ \ e \cdot x = x \qquad [\,e \text{ és l'element neutre per l'esquerra}\,]$$

Exemples de monoides:

$D_I = \mathbb{N}$ els naturals
$\cdot_I = +$
$e_I = 0$

$D_I = \mathbb{Q}$ els racionals
$\cdot_I = +$
$e_I = 0$

$D_I = \mathbb{Z}$ els enters
$\cdot_I = +$
$e_I = 0$

$D_I = \mathbb{R}$ els reals
$\cdot_I = +$
$e_I = 0$

i tots aquests amb $\times, 1$

Exercici 27

27. (dificultat 2) Un *monoide* és un model de la següent fórmula:

$$\forall x\, \forall y\, \forall z\ \ (x \cdot y) \cdot z = x \cdot (y \cdot z) \qquad [\cdot \text{ es associatiu}]$$

$$\wedge$$

$$\forall x\ \ x \cdot e = x \qquad [\,e \text{ és l'element neutre per la dreta}\,]$$

$$\wedge$$

$$\forall x\ \ e \cdot x = x \qquad [\,e \text{ és l'element neutre per l'esquerra}\,]$$

c)

$D_I = \mathcal{P}(\mathbb{N})$ els conjunts de naturals

$\cdot_I = \cap$ intersecció

$e_I = \mathbb{N}$

Exercici 27

27. (dificultat 2) Un *monoide* és un model de la següent fórmula:

$$\forall x \, \forall y \, \forall z \;\; (x \cdot y) \cdot z = x \cdot (y \cdot z) \qquad [\cdot \text{ es associatiu}]$$
$$\wedge$$
$$\forall x \;\; x \cdot e = x \qquad [\, e \text{ és l'element neutre per la dreta}\,]$$
$$\wedge$$
$$\forall x \;\; e \cdot x = x \qquad [\, e \text{ és l'element neutre per l'esquerra}\,]$$

d)

Els strings amb concatenació i l'string buit ($\lambda =$ "lambda")

D_I = cadenes de 0s i 1s

$\cdot_I$ = concatenació $\quad (S1 @ S2) @ S3 = S1 @ (S2 @ S3)$

on @ és la concatenació

$e_I = \lambda$

Exercici 27

27. (dificultat 2) Un *monoide* és un model de la següent fórmula:

$$\forall x\,\forall y\,\forall z\;\;(x\cdot y)\cdot z = x\cdot(y\cdot z) \qquad [\,\cdot\text{ es associatiu}\,]$$
$$\wedge$$
$$\forall x\;\; x\cdot e = x \qquad [\,e\text{ és l'element neutre per la dreta}\,]$$
$$\wedge$$
$$\forall x\;\; e\cdot x = x \qquad [\,e\text{ és l'element neutre per l'esquerra}\,]$$

f)

$$D_I = \{\alpha, \beta\}$$
$$\cdot_I(\alpha,\alpha) = \alpha$$
$$\cdot_I(\alpha,\beta) = \beta$$
$$\cdot_I(\beta,\alpha) = \beta$$
$$\cdot_I(\beta,\beta) = \alpha$$

Exercici 27

27. (dificultat 2) Un *monoide* és un model de la següent fórmula:

$\forall x \, \forall y \, \forall z \;\; (x \cdot y) \cdot z = x \cdot (y \cdot z)$ [$\cdot$ es associatiu]

f)

$D_I = \{\alpha, \beta\}$
$\cdot_I(\alpha, \alpha) = \alpha$
$\cdot_I(\alpha, \beta) = \beta$
$\cdot_I(\beta, \alpha) = \beta$
$\cdot_I(\beta, \beta) = \alpha$

s'haurien de fer els 8 casos

α	α	α	
α	α	β	
α	β	α	$\leftarrow$ (1)
α	β	β	
β	α	α	
β	α	β	
β	β	α	
β	β	β	

(1) com a exemple, comprobarem aquest cas

Exercici 27

27. (dificultat 2) Un *monoide* és un model de la següent fórmula:

$\forall x\, \forall y\, \forall z\ \ (x \cdot y) \cdot z = x \cdot (y \cdot z)$ [$\cdot$ es associatiu]

f)

$$D_I = \{\alpha, \beta\}$$
$$\cdot_I(\alpha, \alpha) = \alpha$$
$$\cdot_I(\alpha, \beta) = \beta$$
$$\cdot_I(\beta, \alpha) = \beta$$
$$\cdot_I(\beta, \beta) = \alpha$$

$$\underbrace{\underbrace{\alpha \cdot \beta}_{\beta} \cdot \alpha}_{\beta} = \underbrace{\alpha \cdot \underbrace{\beta \cdot \alpha}_{\beta}}_{\beta}$$

Exercici 27

27. (dificultat 2) Un *monoide* és un model de la següent fórmula:

$\forall x \;\; x \cdot e = x$ [*e* és l'element neutre per la dreta]

$\forall x \;\; e \cdot x = x$ [*e* és l'element neutre per l'esquerra]

f)

$$D_I = \{\alpha, \beta\}$$
$$\cdot_I(\alpha,\alpha) = \alpha$$
$$\cdot_I(\alpha,\beta) = \beta$$
$$\cdot_I(\beta,\alpha) = \beta$$
$$\cdot_I(\beta,\beta) = \alpha$$

$$e_I = \alpha$$

s'han de comprovar els casos:

$$\cdot_I(\alpha, e_I) = \alpha$$
$$\cdot_I(\beta, e_I) = \beta$$

$$\cdot_I(e_I, \alpha) = \alpha$$
$$\cdot_I(e_I, \beta) = \beta$$

Exercici 28

28. (dificultat 2) un *grup* és un monoide que a més satisfà:

$$\forall x \, \exists y \, (x \cdot y = e \wedge y \cdot x = e)$$

i diu que "y és l'invers de x".

Quines de les interpretacions anteriors eren grups?

Exercici 28

28. (dificultat 2) un *grup* és un monoide que a més satisfà:

$$\forall x\, \exists y\ (x \cdot y = e \wedge y \cdot x = e)$$

i diu que "y és l'invers de x".

Una altra manera de definir els grups és fent explícita l'operació unària invers i:

$$\forall x\, (x \cdot i(x) = e \wedge i(x) \cdot x = e)$$

Exercici 28

28. (dificultat 2) un *grup* és un monoide que a més satisfà:

$$\forall x\, \exists y\, (x \cdot y = e \land y \cdot x = e)$$

i diu que "y és l'invers de x".

Exemples de grups:

$D_I = \mathbb{N}$ els naturals
$\cdot_I = +$
$e_I = 0$ NO és grup, perquè no hi ha invers

$D_I = \mathbb{Z}$ els enters
$\cdot_I = +$
$e_I = 0$
$i_I = -n$ SÍ és grup

$D_I = \mathbb{Q}$ els racionals
$\cdot_I = +$
$e_I = 0$
$i_I = -n$ SÍ és grup

$D_I = \mathbb{R}$ els reals
$\cdot_I = +$
$e_I = 0$
$i_I = -n$ SÍ és grup

I tots aquests amb $\times$, 1 ?

Exercici 28

28. (dificultat 2) un *grup* és un monoide que a més satisfà:

$$\forall x \, \exists y \, (x \cdot y = e \wedge y \cdot x = e)$$

i diu que "y és l'invers de x".

$D_I = \mathbb{Z}$ els enters

$\cdot_I = \times$

$e_I = 1$

NO és grup, perquè no hi ha invers

Exercici 28

28. (dificultat 2) un *grup* és un monoide que a més satisfà:

$$\forall x\, \exists y\, (x \cdot y = e \wedge y \cdot x = e)$$

i diu que "y és l'invers de x".

$D_I = \mathbb{Q}$ els racionals

$\cdot_I = \times$

$e_I = 1$

$i_I(n) = 1/n$

SÍ és grup si traiem el zero del domini: $D_I = \mathbb{Q} \setminus \{0\}$

Exercici 28

28. (dificultat 2) un *grup* és un monoide que a més satisfà:

$$\forall x\, \exists y\, (x \cdot y = e \wedge y \cdot x = e)$$

i diu que "y és l'invers de x".

$D_I = \mathbb{R}$ els reals
$\cdot_I = \times$
$e_I = 1$
$i_I(n) = 1/n$

SÍ és grup si traiem el zero del domini: $D_I = \mathbb{R} \setminus \{0\}$

Exercici 28

28. (dificultat 2) un *grup* és un monoide que a més satisfà:

$$\forall x\, \exists y\, (x \cdot y = e \wedge y \cdot x = e)$$

i diu que "y és l'invers de x".

c)

$D_I = \mathcal{P}(\mathbb{N})$ els conjunts de naturals
$\cdot_I = \cap$ la intersecció
$e_I = \mathbb{N}$

NO és grup. NO hi ha invers.
Perquè fos grup, necessitaríem que per a tot conjunt de naturals x hagués un altre, $i(x)$, tal que $x \cap i(x) = \mathbb{N}$.
I això no existeix.

Exercici 28

28. (dificultat 2) un *grup* és un monoide que a més satisfà:

$$\forall x \, \exists y \, (x \cdot y = e \wedge y \cdot x = e)$$

i diu que "y és l'invers de x".

d)

D_I = cadenes de 0s i 1s

$\cdot_I$ = concatenació $\qquad (S1 @ S2) @ S3 = S1 @ (S2 @ S3)$

on @ és la concatenació

$e_I = \lambda$ $\qquad$ (lambda, la cadena buida)

NO és grup perquè no hi ha invers

NO és grup commutatiu

Exercici 28

28. (dificultat 2) un *grup* és un monoide que a més satisfà:

$$\forall x \, \exists y \, (x \cdot y = e \wedge y \cdot x = e)$$

i diu que "y és l'invers de x".

f)

$$D_I = \{\alpha, \beta\}$$
$$\cdot_I(\alpha, \alpha) = \alpha$$
$$\cdot_I(\alpha, \beta) = \beta$$
$$\cdot_I(\beta, \alpha) = \beta$$
$$\cdot_I(\beta, \beta) = \alpha$$

$$e_I = \alpha$$

$$i_I(\alpha) = \alpha \qquad \alpha \cdot i_I(\alpha) = i_I(\alpha) \cdot \alpha = e_I = \alpha$$
$$i_I(\beta) = \beta \qquad \beta \cdot i_I(\beta) = i_I(\beta) \cdot \beta = e_I = \alpha$$

SÍ és grup amb aquesta interpretació de l'invers

SI és grup commutatiu

Exercici 28

28. (dificultat 2) un *grup* és un monoide que a més satisfà:

$$\forall x \, \exists y \; (x \cdot y = e \; \wedge \; y \cdot x = e)$$

i diu que "y és l'invers de x".

Un altre possible exemple:

$$D_I = \mathbb{N}$$
$$\cdot_I(n, m) = mcd(n, m)$$

Això és associatiu, perquè
$mcd(x, mcd(y, z)) = mcd(mcd(x, y), z)$.
Però no hi ha element neutre, per tant no és monoide.

Exercici 32

32. Per als conjunts de símbols i parells d'interpretacions següents, escriu una fórmula F de LPOI sobre el vocabulari (els símbols) donat, tal que F és certa en una d'elles i falsa en l'altra.

a) (dificultat 2) Conjunt de símbols de funció: $\{f^2\}$,
I_1 té com a domini els naturals $\mathbb{N}$ i f s'interpreta com el producte
I_2 té com a domini $\mathcal{P}(\mathbb{N})$ i f s'interpreta com la intersecció

Si F és la fórmula $\forall x\, f(x,x)=x$ llavors
$I_1 \not\models F$ però $I_2 \models F$

Si F és la fórmula $\neg \forall x\, f(x,x)=x$ llavors
$I_1 \models F$ però $I_2 \not\models F$

Exercici 32

32. Per als conjunts de símbols i parells d'interpretacions següents, escriu una fórmula F de LPOI sobre el vocabulari (els símbols) donat, tal que F és certa en una d'elles i falsa en l'altra.

b) (dificultat 2) Conjunt de símbols de funció: $\{f^1\}$,
I_1 té domini els naturals $\mathbb{N}$
I_2 té domini els enters $\mathbb{Z}$
En tots dos casos el símbol f s'interpreta com la funció "següent", és a dir, $f_I(n) = n + 1$.

Si F és la fórmula $\forall x\, \exists y\; f(y) = x$ llavors
$I_1 \not\models F$ però $I_2 \models F$

Exercici 32

32. Per als conjunts de símbols i parells d'interpretacions següents, escriu una fórmula F de LPOI sobre el vocabulari (els símbols) donat, tal que F és certa en una d'elles i falsa en l'altra.

c) (dificultat 4) $\{f^2, g^2\}$,
I_1 té domini els reals $\mathbb{R}$
I_2 té domini els racionals $\mathbb{Q}$
En tots dos casos f i g s'interpreten com la suma i el producte respectivament.
Ajuda: fabrica el dos i expressa que arrel de dos existeix.

$\exists x \forall y\ g(x,y)=y$ (això expressa que x és el 1, i per això $f(x,x)$ serà 2)

Afegim alguna cosa i tenim:

Si F és la fórmula $\exists x \exists z (\forall y\, g(x,y)=y \wedge g(z,z)=f(x,x))$
llavors $I_1 \models F$ però $I_2 \not\models F$

Definició de la Lògica de Primer Ordre

Per al proper dia de classe:

- Comença a estudiar el capítol 5: Deducció en LPO.

  ☞ `p5.pdf`

Lògica en la Informàtica
Deducció en Lògica de Primer Ordre (LPO)

José Miguel Rivero

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

Primavera 2025

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.

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)

Sumari

1. Decidibilitat en Lògica de Primer Ordre
   - Quins problemes són decidibles en LPO (o LPOI) i quins no?

2. Tema 5: Deducció en LPO
   - Transformació a forma clausal EQUISATISFACTIBLE
   - Resolució en LPO

Definició de la Lògica de Primer Ordre

Quins problemes són decidibles en LPO (o LPOI) i quins no?

- Considerem problemes Booleans = problemes de decisió = problemes amb resposta si/no.

Definició

Un problema és DECIDIBLE si:
existeix algun procediment que sempre contesta correctament, en temps finit (és a dir, acaba).

Dins dels problemes decidibles, distingim classes de complexitat (en temps): logarítmic, lineal, quadràtic, polinòmic, exponencial, NP-complet, ...

Dins dels problemes INdecidibles, distingim altres classes: semi-decidibles, co-semi-decidibles, ...

Definició de la Lògica de Primer Ordre

Quins problemes són decidibles en LPO (o LPOI) i quins no?

En el context de la lògica, dos problemes importants són:

1: evaluació d'una formula: donades I i F, tenim $I \models F$?

2: SAT: donada una F, existeix alguna I tal que $I \models F$?

	avaluació	SAT
LProp:	lineal	NP-complet
LPO:	indecidible	indecidible

SAT en LPO és *co-semi-decidible*: existeix algun procediment que,

- si la resposta és NO (és a dir, F és insat), llavors contesta correctament "NO" en temps finit
- si la resposta és SI (és a dir, F és sat), o contesta correctament "SI" o no acaba

Quins problemes són decidibles en LPO (o LPOI) i quins no?

En general:

Un problema és *semi-decidible* si existeix algun procediment que,

- si la resposta és SI, llavors contesta correctament en temps finit
- si la resposta és NO, o contesta correctament "NO" o no acaba

Un problema és *CO-semi-decidible* si existeix algun procediment que,

- si la resposta és NO, llavors contesta correctament en temps finit
- si la resposta és SI, o contesta correctament "SI" o no acaba

Definició de la Lògica de Primer Ordre

Avaluació en LPO amb domini **finit** (és a dir, la I donada té domini finit) sí que és *decidible*. Per què?

Exemple d'avaluació en LPO amb domini finit:

Sigui la I següent:

$$D_I = \{a, b\}$$
$$p_I(a, a) = 1$$
$$p_I(a, b) = 1$$
$$p_I(b, a) = 1$$
$$p_I(b, b) = 0$$

Sigui la F següent:

$$\forall x \, \exists y \; p(x, y)$$

$$\forall x_1 \, \exists y_1 \, \forall x_2 \, \exists y_2 \cdots \forall x_n \, \exists y_n \; F$$

decidible però pot ser exponencial.

Definició de la Lògica de Primer Ordre

Avaluació en LPO amb domini **INfinit** és *INdecidible* en general.
Per què?

El halting problem (el problema de la parada):
- donat un programa P, (o el que és el mateix, una màquina de Turing), P acaba?

Aquest problema es va demostrar que era indecidible.

A partir de aqui, es van demostrar indecidibles altres problemes, mitjançant reduccions entre problemes.
Per exemple, si pots reduir el "halting problem" a "SAT en LPO" (fer-ho mitjançant SAT en LPO),... llavors SAT en LPO també ha de ser indecidible!

Avaluació en LPO amb domini **INfinit** és *INdecidible* en general. Per què?

El problema "Arrell": donat un polinomi com a $x^3y^2 + 3x^4 + \cdots = 0$, té solucions ("arrells") enteres?

(trobar arrells de polinomis sobre diverses variables de grau arbitrari i amb productes entre variables).

Aquest problema es va demostrar que era indecidible. Es diu Hilbert's tenth problem.

Definició de la Lògica de Primer Ordre

Avaluació en LPO amb domini **INfinit** és *INdecidible* en general.
Per què?

Podem reduir "Arrell" a l'avaluació en LPO amb domini infinit (fer "Arrell" mitjançant avaluació en LPO amb domini infinit):
Sigui I la interpretació amb $D_I = \mathbb{Z}$ (els enters) on $\{f^2, g^2\}$ s'interpreten com a suma i producte.

$$
\begin{aligned}
F \;=\; & \exists x\, \exists y \\
& \quad \exists z\, ((\forall y\, f(z, y) = y) \\
& \qquad \wedge \\
& f(g(\underbrace{g(x, g(x, x))}_{x^3}, \underbrace{g(y, y)}_{y^2}), \underbrace{f(f(g(g(x, x), g(x, x)), g(g(x, x), g(x, x))), g(g(x, x), g(x, x)))}_{+ \qquad x^4 + x^4 + x^4} \underbrace{, \ldots) = z)}_{+ \cdots = 0}
\end{aligned}
$$

Tenim $I \models F$ ssi $x^3y^2 + 3x^4 + \cdots = 0$ té arrells senceres.

Definició de la Lògica de Primer Ordre

Avaluació en LPO amb domini **INfinit** és *INdecidible* en general. Per què?

Reduir "Arrell" al problema d'avaluació en LPO amb domini infinit: Si em donen un polinomi P, puc construir una fórmula F_P, tal que si I és la interpretació: $D_I = \mathbb{Z}$ (els enters) on $\{f^2, g^2\}$ s'interpreten com a suma i producte tenim $I \models F_P$ ssi P té arrells senceres.

Tema 5: Deducció en LPO

En L.Prop. teníem diversos mètodes per a SAT: $p \vee q \vee \neg r$

- El millor mètode per a SAT estava basat en un algorisme de backtracking amb propagació, etc., que explora el conjunt de possibles models (totes les interpretacions).
  En LPO aquest mètode no existeix.
  No hi ha manera de "enumerar" totes les I's.
- Però en L.Prop. vam veure un altre, basat en resolució, amb el teorema:

  un cjto de clausulas S és insat ssi $\Box \in Res(S)$.

  En LPO, l'únic mètode per a SAT que estudiarem és el basat en resolució.

Deducció en Lògica de Primer Ordre

Una clàusula en LPO és una disjunció de literals, com en L.Prop, però en LPO els literals ja no són símbols de predicat o símbols de predicat negats, sinó que són ÀTOMS, o ÀTOMS NEGATS.

Poden contenir variables, que TOTES s'entenen que estan universalment quantificades:

$\forall x_1 \cdots \forall x_m \; L_1 \vee \cdots \vee L_n$ però normalment els $\forall x_1 \cdots \forall x_m$ no els escrivim.

Un exemple de Prolog (Neboda, Tia, Mare):

`tia(N,T) :- mare(N,M), germana(M,T).`

$\texttt{tia(N,T)} \leftarrow \texttt{mare(N,M)} \wedge \texttt{germana(M,T)}$

$\texttt{tia(N,T)} \vee \neg(\texttt{mare(N,M)} \wedge \texttt{germana(M,T)})$

$\texttt{tia(N,T)} \vee \neg\texttt{mare(N,M)} \vee \neg\texttt{germana(M,T)}$ és una clàusula de Horn de LPO (i no escrivim els "per a tot" $\forall\texttt{N}\,\forall\texttt{T}\,\forall\texttt{M}$)

Un literal és un àtom $p(t_1, \ldots, t_n)$ o un àtom negat $\neg p(t_1, \ldots, t_n)$.

Per a què volem SAT en LPO?

Per al mateix que en L.Prop, per a les aplicacions pràctiques, i tenim les propietats:

F SAT ?
F insat ?
F Taut ? ssi $\neg F$ insat $\rightarrow$ (1)
$F \models G$? ssi $F \wedge \neg G$ insat
$F \equiv G$? ssi $F \wedge \neg G \;\vee\; G \wedge \neg F$ insat

(1) Taut en LPO és *semi-decidible* (pq és equivalent a un problema de INsat)

Per a poder fer SAT en LPO mitjançant resolució:

Transformació a forma clausal EQUISATISFACTIBLE

(com passava amb la transformació de Tseitin en L.Prop)

"Forma clausal" = conjunt (conjunció, un AND) de clàusules.

Equisatisfactible: Si una formula F té el cjto de clàusules S com a forma clausal, tenim que: F sat ssi S sat.

Deducció en Lògica de Primer Ordre

Per a poder fer SAT en LPO mitjançant resolució:

Transformació a forma clausal EQUISATISFACTIBLE

(com passava amb la transformació de Tseitin en L.Prop)

1. Moviment de les negacions cap a dins
2. Eliminació de conflictes de nom de variable
3. [Opcional] Moviment de quantificadors cap a dins mentre sigui possible
4. Eliminació de quantificadors existencials o Skolemización
5. Moviment de quantificadors universals cap a fora
6. Aplicar distributivitat

Deducció en Lògica de Primer Ordre

Transformació a forma clausal EQUISATISFACTIBLE:

1. Moviment de les negacions cap a dins:

   $\neg(F \wedge G) \;\Rightarrow\; \neg F \vee \neg G$

   $\neg(F \vee G) \;\Rightarrow\; \neg F \wedge \neg G$

   $\neg\neg F \;\Rightarrow\; F$

   $\neg\exists x\, F \;\Rightarrow\; \forall x\, \neg F$

   $\neg\forall x\, F \;\Rightarrow\; \exists x\, \neg F$

2. Eliminació de conflictes de nom de variable:

   per exemple: $\forall x\, p(x) \wedge \exists x\, q(x) \;\Longrightarrow\; \forall x\, p(x) \wedge \exists x'\, q(x')$

3. [Opcional; és només per eficiència] Moviment de quantificadors cap a dins mentre sigui possible:

   per exemple: $\forall x\, (p(a) \wedge q(x)) \;\Longrightarrow\; p(a) \wedge \forall x\, q(x)$

Transformació a forma clausal EQUISATISFACTIBLE:

4. Eliminació de quantificadors existencials o Skolemització: (és l'únic pas dels 6 que NO preserva l'equivalència logica; però sí la equisatisfactibilitat)

   2 exemples:

   1. $\forall x\, \exists y\ p(\, x, y\,) \ \xrightarrow{sk}\ \forall x\ p(\, x, f_y(x)\,)$ on f_y és un símbol de funció nou "fresc"
   2. $\exists y\, \forall x\ p(\, x, y\,) \ \xrightarrow{sk}\ \forall x\ p(\, x,\ \ c_y\ \ )$ on c_y és un símbol de funció nou "fresc" (en aquest cas, una cte)

Transformació a forma clausal EQUISATISFACTIBLE:

4. Eliminació de quantificadors existencials o Skolemització: (és l'únic pas dels 6 que NO preserva l'equivalència logica; però sí la equisatisfactibilitat)

   Recordem $\forall x\, \exists y\ p(x,y) \not\equiv \exists y\, \forall x\ p(x,y)$:

   1. $\forall x\, \exists y\ p(x,y)$
   2. $\exists y\, \forall x\ p(x,y)$

   Si tenim la interpretació I tal que:

   $D_I = \{a, b\}$
   $p_I(a,a) = 0$
   $p_I(a,b) = 1$
   $p_I(b,a) = 1$
   $p_I(b,b) = 0$

   tenim que $I \models \forall x\, \exists y\ p(x,y)$,
   però $I \not\models \exists y\, \forall x\ p(x,y)$.

   Intuïtivament, tenim que $\exists y\, \forall x\ p(x,y) \models \forall x\, \exists y\ p(x,y)$.

Transformació a forma clausal EQUISATISFACTIBLE:

4. Eliminació de quantificadors existencials o Skolemització:

La Skolemización NO dona una fórmula lògicament equivalent:

tenim $\forall x\, \exists y\ p(x,y) \ \xrightarrow{sk}\ \forall x\ p(x, f_y(x))$

donem una I tal que:

- $I \models \forall x\, \exists y\ p(x,y)$
- $I \not\models \forall x\ p(x, f_y(x))$

$D_I = \{a, b\}$

$p_I(a,a) = 0 \qquad f_{y_I}(a) = a$
$p_I(a,b) = 1 \qquad f_{y_I}(b) = a$
$p_I(b,a) = 1$
$p_I(b,b) = 0$

això és model de $\forall x\, \exists y\ p(x,y)$ però NO és model de $\forall x\ p(x, f_y(x))$.

Transformació a forma clausal EQUISATISFACTIBLE:

4. Eliminació de quantificadors existencials o Skolemització:

La Skolemización NO dona una fórmula lògicament equivalent:

tenim $\forall x\, \exists y\ p(x,y) \xrightarrow{sk} \forall x\ p(x, f_y(x))$

donem una I tal que:

- $I \models \forall x\, \exists y\ p(x,y)$
- $I \not\models \forall x\ p(x, f_y(x))$

$D_I = \{a, b\}$

$p_I(a,a) = 0 \qquad f_{y_I}(a) = b$
$p_I(a,b) = 1 \qquad f_{y_I}(b) = a$
$p_I(b,a) = 1$
$p_I(b,b) = 0$

En canvi, si interpreto f_y així (és a dir, "bé", com ho feia l'existeix en $\forall x\, \exists y\ p(x,y)$), llavors f_{y_I} "tria" el valor adequat perquè I SÍ QUE sigui model de $\forall x\, p(x, f_y(x))$.

Transformació a forma clausal EQUISATISFACTIBLE:

4. Eliminació de quantificadors existencials o Skolemització:

La Skolemización NO dona una fórmula lògicament equivalent

En general:

Si $F \xrightarrow{sk} F'$ llavors donat un model de F puc construir un model de F', i viceversa: F i F' són **equisatisfactibles**.

Transformació a forma clausal EQUISATISFACTIBLE:

5. Moviment de quantificadors universals cap a fora

   Per exemple:

   $$F \wedge \forall x\, G \implies \forall x\, (F \wedge G)$$

6. Distributivitat amb: $(F \wedge G) \vee H \Rightarrow (F \vee H) \wedge (G \vee H)$

   (això pot fer créixer la fórmula exponencialment, perquè la part H es duplica; hi ha mètodes similars a Tseitin per a evitar aquest problema).

Resolució en LPO

- Resolució en L.Proposicional
- Regla de resolució en LPO
- No terminació de la resolució en LPO

Deducció en Lògica de Primer Ordre

Recordem: Resolució en L.Proposicional:

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

Teorema

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

Aquest teorema també és cert en LPO (bé, "gaire bé cert"; veure més endavant)

Recordem: Resolució en L.Proposicional:

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

Teorema

S insat ssi $\Box \in Res(S)$

$S_0 = S$

$S_1 = S_0 \cup Res_1(S_0)$ $\quad$ ($Res_1(S_0)$ = el que puc obtenir en 1 pas de resolució a partir de S_0)

$S_2 = S_1 \cup Res_1(S_1)$ $\quad$ ($Res_1(S_1)$ = el que puc obtenir en 1 pas de resolució a partir de S_1)

... ...

$Res(S) = \bigcup_{i=0}^{\infty} S_i$

Resolució en LPO

$$\frac{A \vee C \qquad\qquad \neg B \vee D}{(C \vee D)\sigma}$$

A, B són àtoms
si $\sigma = mgu(A, B)$
(*most general unifier*)

Exemple: x, y són vars a, b són ctes:

$$\frac{p(a, x) \vee q(x) \qquad \neg p(y, b) \vee r(y)}{(\ q(x) \vee r(y)\ )\sigma}$$

si $\sigma = \{x = b, y = a\}$
$q(b) \vee r(a)$

$$\frac{p(a, b) \vee q(b) \qquad \neg p(a, b) \vee r(a)}{q(b) \vee r(a)}$$

En LPO, la resolució pot no acabar:

$$\frac{p(a) \qquad \neg p(x) \vee p(f(x))}{p(f(a))} \qquad mgu(\, p(a), p(x)\,) = \{x\!=\!a\}$$

$$\frac{p(f(a)) \qquad \neg p(x) \vee p(f(x))}{p(f(f(a)))} \qquad mgu(\, p(f(a)), p(x)\,) = \{x\!=\!f(a)\}$$

1. $p(a)$
2. $\neg p(x) \vee p(f(x))$

---

3. $p(f(a))$
4. $p(f(f(a)))$
5. $p(f(f(f(a))))$
6. $\ldots$

puc obtenir, amb $mgu(p(a), p(x)) = \{x\!=\!a\}$:
3. amb la 2. amb $mgu(\, p(f(a)), p(x)\,) = \{x\!=\!f(a)\}$:
4. amb la 2. amb $mgu(\, p(f(f(a))), p(x)\,) = \{x\!=\!f(f(a))\}$:

En LPO, la resolució pot no acabar:

$$\frac{p(a) \qquad \neg p(x) \vee p(f(x))}{p(f(a))} \qquad mgu(\, p(a), p(x)\,) = \{x = a\}$$

$$\frac{p(f(a)) \qquad \neg p(x) \vee p(f(x))}{p(f(f(a)))} \qquad mgu(\, p(f(a)), p(x)\,) = \{x = f(a)\}$$

El mateix exemple, de forma més “natural”:

1. $nat(0)$
2. $\neg nat(x) \vee nat(succ(x))$

Deducció en Lògica de Primer Ordre

En LPO, la resolució pot no acabar:

Un altre exemple de no-terminació, sense símbol de funció:

1. $\neg p(x,y) \vee \neg p(y,z) \vee p(x,z)$

$$
\begin{array}{ll}
1. & \neg p(x,y) \;\vee\; \neg p(y,z) \;\vee\; \underline{p(x,z)} \\
 & \underline{\neg p(x',y')} \vee \neg p(y',z') \vee p(x',z')
\end{array}
$$

el *mgu* és $\{x' = x, y' = z\}$ i obtenim:

2. $\neg p(x,y) \vee \neg p(y,z) \vee \neg p(z,z') \vee p(x,z')$
(una mena de "transitivitat de 4")

... etc

Per al proper dia de classe:

- Unificació
- Veure el capítol 5 dels apunts, i els exercicis.

☞ `p5.pdf`

Lògica en la Informàtica
Deducció en Lògica de Primer Ordre (LPO)

José Miguel Rivero

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

Primavera 2025

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.

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)

Sumari

Unificació. Algorisme d'unificació

Unificació

- una "substitució" σ és un conjunt de parells variable-terme: $\{x_1 = t_1, \ldots, x_n = t_n\}$
- "aplicar una substitució": si σ és $\{x = f(a), y = b\}$ i t és el terme (o àtom) $g(f(x), y))$ llavors $t\sigma$ és $g(f(f(a)), b)$
- dos termes s i t són "unificables" si existeix una σ tal que $s\sigma = t\sigma$

- σ és l'"unificador més general" (*most general unifier, mgu*) de dos termes s i t si:
  - $s\sigma = t\sigma$ (σ és unificador)
  - i a més és l'unificador més general: per a tot σ', si tenim que $s\sigma' = t\sigma'$ llavors hi ha un σ'' tal que σ' és $\sigma\sigma''$

Unificació

Per exemple: unificar $f(x, y)$ amb $f(a, z)$
el *mgu* $\sigma = \{x = a, y = z\}$

Un altre unificador σ' pot ser $\sigma' = \{x = a, y = a, z = a\}$
però no és el més general. És un cas particular del *mgu* σ.

Existeix $\sigma'' = \{y = a, z = a\}$
i tinc que $\sigma' = \sigma\sigma''$.

Unificació. Algorisme d'unificació

Algorisme d'unificació:

Jo vull unificar dos termes s i t (o dos àtoms s i t, a l'efecte d'unificació és el mateix).

Escriurem el problema d'unificació com a conjunts d'igualtats $\{s = t\}$:

0.	$E \cup \{t = t\}$	$\Longrightarrow$	E
1.	$E \cup \{f(\ldots) = g(\ldots)\}$	$\Longrightarrow$	fallo
			si $f \neq g$ (no són unificables!)
2.	$E \cup \{f(s_1 \ldots s_n) = f(t_1 \ldots t_n)\}$	$\Longrightarrow$	$E \cup \{s_1 = t_1 \ldots s_n = t_n\}$
3.	$E \cup \{x = t\}$	$\Longrightarrow$	fallo
			si x apareix en t, i
			x no ÉS t (per ex. $x = f(x)$)
4.	$E \cup \{x = t\}$	$\Longrightarrow$	$E\{x = t\} \cup \{x = t\}$
			si x NO apareix en t, i
			a més x SI apareix en E

Unificació. Algorisme d'unificació

Algorisme d'unificació:

Exemple:

$$\begin{array}{lcl} \{f(x,g(x,a)) = f(h(b),z)\} & \stackrel{2}{\Longrightarrow} & \\ \{\underbrace{x = h(b)}_{x=t},\ \underbrace{z = g(x,a)}_{E}\} & \stackrel{4}{\Longrightarrow} & \{x = h(b),\ z = g(h(b),a)\} \end{array}$$

això és el *mgu*!

podria tornar a aplicar la regla 4, però no faria res (per això exigim que x SÍ QUE aparegui en E).

El que hem de pensar:

- aquestes regles acaben? (o necessitem alguna cosa més perquè acabin?)
- donen lloc al *mgu* del problema inicial?

Exemple de recapitulació

Petit exemple de recapitulació

Exemple:

Vull saber si $F \models G$ en LPO. Aqui F i G són fórmules qualssevol. Què faig?

$F \models G$	ssi
la fórmula $F \wedge \neg G$ és insat	ssi
$S = \mathit{forma\_clausal}(F \wedge \neg G)$ és insat	ssi (*gairebé*)
la clàusula buida $\Box$ està en $\mathit{Res}(S)$	(puc obtenir $\Box$ mitjançant resolució a partir del conjunt de clàusules S)

Exemple de recapitulació

Petit exemple de recapitulació

Exemple:

Vull saber si $F \models G$ en LPO. Aqui F i G són fórmules qualssevol. Què faig?

$$S_0 = S$$
$$S_1 = S_0 \cup Res_1(S_0)$$
$$\ldots$$
$$Res(S) = \bigcup_{i=0}^{\infty} S_i$$

Un detall important!! $Res(S)$ NO és exactament la clausura sota **només resolució**!!
Cal una regla deductiva addicional: la **factorització**.
Veurem per què:

Exemple de recapitulació

Un exemple. Un conjunt S de dues clàusules:

$$\{\, p(x) \vee p(y), \;\; \neg p(z) \vee \neg p(z') \,\}$$

S és SAT o INSAT? Suposem que S és SAT.

Llavors hi hauria un model I amb almenys un element en el seu domini D_I (els dominis sempre són no-buits).

Diguem-li "e" a aquest element: $D_I = \{e, \ldots\}$

Com pot ser $p_I(e)$? cert o fals?

- per la primera clàusula $\forall x \forall y\ (p(x) \vee p(y))$ en el cas on $x = y = e$, necessito que $p_I(e) = 1$
- per la segona clàusula (cas on $z = z' = e$) necessito que $p_I(e) = 0$

Contradicció! No existeix cap model!! Per tant, S és INSAT.

Exemple de recapitulació

Com que S és INSAT, mitjançant resolució hauríem de poder obtenir $\square$ a partir de S.

$$\{\, p(x) \vee p(y),\ \ \neg p(z) \vee \neg p(z') \,\}$$

Però no és possible obtenir $\square$!!

Puc fer, per exemple, aquesta resolució:

$$\frac{p(x) \vee p(y) \qquad \neg p(z) \vee \neg p(z')}{p(y) \vee \neg p(z')} \qquad mgu(p(x), p(z)) = \{x = z\}$$

En les clàusules de S, els literals no comparteixen variables.

L'única cosa que puc obtenir mitjançant resolució són altres clàusules on els literals tampoc comparteixen variables.

I per això, en aquest exemple, sempre continuaré obtenint clàusules de dos literals!

I mai sortirà la clàusula buida $\square$!!

Exemple de recapitulació

Aquest exemple demostra que la resolució per si sola NO és refutacionalment completa!!

Si només considerem resolució, NO és veritat que

S insat SSI $\square \in Res(S)$.

Què és el que falta?

Fem el mateix "tipus" d'exemple en L.proposicional:

$$\boxed{\begin{array}{r} p \vee q \\ p \vee \neg q \\ \neg p \vee q \\ \neg p \vee \neg q \end{array}}$$

INSAT. Per resolució:

$$\dfrac{\dfrac{\dfrac{p \vee q \quad p \vee \neg q}{p \vee p}}{p} \qquad \dfrac{\dfrac{\neg p \vee q \quad \neg p \vee \neg q}{\neg p \vee \neg p}}{\neg p} \quad \longleftarrow \ (*)}{\square}$$

(*) aquest pas d'eliminar literals repetits l'hem de simular (estendre) en LPO!!

Regla deductiva de Factorització

Factorització

La Regla deductiva de Factorització en LPO és la que fa això. És la següent:

$$\frac{A \vee B \vee C}{(A \vee C)\,\sigma}$$

on A i B átoms (literals POSITIUS),
i C és la resta de la clàusula
$\sigma = mgu(A, B)$

Per exemple:

$$\frac{\overbrace{p(a,x)}^{A} \vee \overbrace{p(y,b)}^{B} \vee \overbrace{q(x,y) \vee \cdots}^{C}}{\begin{array}{c}(\ p(a,x) \vee q(x,y) \vee \cdots)\,\sigma \\ p(a,b) \vee q(b,a) \vee \cdots\end{array}}$$

$$\begin{aligned}\sigma &= mgu(p(a,x), p(y,b)) \\ &= \{x = b, y = a\}\end{aligned}$$

Regla deductiva de Factorització

En què es basa això?

Si tinc:	$\forall x \, \forall y$	$p(a,x) \vee p(y,b) \vee q(x,y)$
en particular, tinc:		$p(a,b) \vee p(a,b) \vee q(b,a)$
		(és a dir, el mateix on $x = b$, $y = a$)

i sobre això puc eliminar literals repetits com en L.Prop:

$$p(a,b) \vee q(b,a)$$

Regla deductiva de Factorització

Tornem a l'exemple del conjunt S de dues clàusules:

1. $\{\, p(x) \vee p(y),$
2. $\neg p(z) \vee \neg p(z') \,\}$

Puc aplicar factorització a la clàusula 1.! :

3. $\dfrac{p(x) \vee p(y)}{p(x)}$ $\qquad \sigma = mgu(p(x), p(y)) = \{y = x\}$ (aqui la part C és buida)
4. $\neg p(z')$ $\qquad$ per resolució entre 2. i 3.: $\sigma = mgu(p(z), p(x)) = \{x = z\}$
5. $\square$ $\qquad$ per resolució entre 3. i 4.: $\sigma = mgu(p(z'), p(x)) = \{x = z'\}$

Completitud refutacional en LPO

El teorema que SÍ QUE és veritat en LPO:

$$S \ \text{insat} \ \text{SSI} \ \Box \in \mathit{ResFact}(S)$$

Calculem $\mathit{ResFact}(S)$ per nivells:

$$S_0 = S$$
$$S_{i+1} = S_i \cup \mathit{Res}_1(S_i) \cup \mathit{Fact}_1(S_i)$$

$$\mathit{ResFact}(S) = \bigcup_{i=0}^{\infty} S_i$$

Completitud refutacional en LPO

Última observació: Què passa si S és (un conjunt de clàusules de) Horn?

1. La regla de factorització no s'aplica a clàusules de Horn.
2. Si S és de Horn, fent resolució només obtinc clàusules de Horn.

Si S és Horn, llavors:

S insat SSI $\square \in Res(S)$

(Si S és Horn no necessito factorització!!!,
perquè mai tindré ocasió d'aplicar-la!)

Deducció en Lògica de Primer Ordre

Comentaris sobre el tema 6

- Programació lògica

  Llegeix els apunts del tema 6. ☞ `p6.pdf`

  - un programa Prolog és un conjunt de clausulas de Horn de LPO
  - executar un programa Prolog és fer resolució (amb una estratègia determinada, no és exactament per nivells $S_0, S_1, S_2, \ldots$)

- Mira els exercicis d'examen on es fa això. Per exemple, l'exercici 6 de l'examen final de 2017 tardor.

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

   (a) "Tot drac està feliç si tots els seus fills poden volar"
   (b) "Els dracs verds poden volar"
   (c) "Un drac és verd si és fill d'almenys un drac verd"

Demostra per resolució que la conjunció de (a), (b) i (c) implica que:

   (d) "Tots els dracs verds són feliços"

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:
   (a) "Tot drac està feliç si tots els seus fills poden volar"
   (b) "Els dracs verds poden volar"
   (c) "Un drac és verd si és fill d'almenys un drac verd"

Demostra per resolució que la conjunció de (a), (b) i (c) implica que:
   (d) "Tots els dracs verds són feliços"

$$
\begin{array}{lcl}
\textit{esfeliç}(x) & \equiv & \text{"}x \text{ és feliç"} \\
\textit{fillde}(x,y) & \equiv & \text{"un fill de } x \text{ és } y\text{"} \\
\textit{esverd}(x) & \equiv & \text{"}x \text{ és verd"} \\
\textit{vola}(x) & \equiv & \text{"}x \text{ pot volar"}
\end{array}
$$

- Necessitem un predicat unari *esdrac*(x) ???
  Funcionaria, però NO cal, podem assumir que tots l'elements del domini són dracs.

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

(a) "Tot drac està feliç si tots els seus fills poden volar"

$$\begin{array}{lcl} \mathit{esfeliç}(x) & \equiv & \text{"}x \text{ és feliç"} \\ \mathit{fillde}(x,y) & \equiv & \text{"un fill de } x \text{ és } y\text{"} \\ \mathit{esverd}(x) & \equiv & \text{"}x \text{ és verd"} \\ \mathit{vola}(x) & \equiv & \text{"}x \text{ pot volar"} \end{array}$$

(a) "Tot drac està feliç si tots els seus fills poden volar"

(a) $\forall x\, (\quad \ldots \quad \rightarrow \mathit{esfeliç}(x)\,)$

on $\ldots$ ha de dir que tots els fills de x poden volar:

$\forall y\, (\, \mathit{fillde}(x,y) \rightarrow \mathit{vola}(y)\,)$

I ens queda:

(a) $\forall x\, (\, \forall y\, (\, \mathit{fillde}(x,y) \rightarrow \mathit{vola}(y)\,) \rightarrow \mathit{esfeliç}(x)\,)$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

   (b) "Els dracs verds poden volar"

$$
\begin{array}{lcl}
\mathit{esfeliç}(x) & \equiv & \text{“}x \text{ és feliç”} \\
\mathit{fillde}(x, y) & \equiv & \text{“un fill de } x \text{ és } y\text{”} \\
\mathit{esverd}(x) & \equiv & \text{“}x \text{ és verd”} \\
\mathit{vola}(x) & \equiv & \text{“}x \text{ pot volar”}
\end{array}
$$

(b) "Els dracs verds poden volar"

(b) $\forall x \; (\mathit{esverd}(x) \rightarrow \mathit{vola}(x) \;)$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

(c) "Un drac és verd si és fill d'almenys un drac verd"

$$
\begin{array}{lcl}
\mathit{esfeliç}(x) & \equiv & \text{“}x\text{ és feliç”} \\
\mathit{fillde}(x, y) & \equiv & \text{“un fill de }x\text{ és }y\text{”} \\
\mathit{esverd}(x) & \equiv & \text{“}x\text{ és verd”} \\
\mathit{vola}(x) & \equiv & \text{“}x\text{ pot volar”}
\end{array}
$$

(c) "Un drac és verd si és fill d'almenys un drac verd"

(c) $\forall x \; (\quad \ldots \quad \rightarrow \mathit{esverd}(x) \;)$

on $\ldots$ ha de dir que x és fill de almenys un drac verd:

$\exists\, y \; (\; \mathit{fillde}(y, x) \wedge \mathit{esverd}(y) \;)$

I ens queda:

(c) $\forall x \; (\; \exists y \; (\; \mathit{fillde}(y, x) \wedge \mathit{esverd}(y) \;) \; \rightarrow \; \mathit{esverd}(x) \;)$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

La conjunció de (a), (b) i (c) implica (d) SSI

$(a) \wedge (b) \wedge (c) \quad \models \quad (d)$ SSI

$(a) \wedge (b) \wedge (c) \wedge \neg(d)$ INSAT SSI

$S = \texttt{formaclausal}(\,(a) \wedge (b) \wedge (c) \wedge \neg(d)\,)$ INSAT SSI

$\Box \in \mathit{ResFact}(S)$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

Passem tot a forma clausal:

(d) "Tots els dracs verds són feliços"

$(\neg d)$ "No tots els dracs verds són feliços"

$$
\begin{array}{lll}
(\neg d) & \neg \forall x\, (& verd(x) \rightarrow esfeliç(x)\,) \\
 & \neg \forall x\, (& \neg verd(x) \vee esfeliç(x)\,) \\
 & \exists x\, (& verd(x) \wedge \neg esfeliç(x)\,)
\end{array}
$$

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

Passem tot a forma clausal:

(*a*) "Tot drac està feliç si tots els seus fills poden volar"

(*a*) $\forall x\, (\, \forall y\, (\, \mathit{fillde}(x, y) \rightarrow \mathit{vola}(y)\,)\, \rightarrow\, \mathit{esfeliç}(x)\,)$

$$\forall x \; (\; \forall y \; (\; \mathit{fillde}(x,y) \; \rightarrow \; \mathit{vola}(y) \;) \; \rightarrow \; \mathit{esfeliç}(x) \;)$$

- eliminem les $\rightarrow$

$$\forall x \; (\; \neg \forall y \; (\; \mathit{fillde}(x,y) \; \rightarrow \; \mathit{vola}(y) \;) \; \vee \; \mathit{esfeliç}(x) \;)$$
$$\forall x \; (\; \neg \forall y \; (\; \neg \mathit{fillde}(x,y) \; \vee \; \mathit{vola}(y) \;) \; \vee \; \mathit{esfeliç}(x) \;)$$

- moure les $\neg$

$$\forall x \; (\; \exists y \; \neg (\; \neg \mathit{fillde}(x,y) \; \vee \; \mathit{vola}(y) \;) \; \vee \; \mathit{esfeliç}(x) \;)$$

- moure les $\neg$ (de Morgan)

$$\forall x \; (\; \exists y \; (\; \mathit{fillde}(x,y) \; \wedge \; \neg \mathit{vola}(y) \;) \; \vee \; \mathit{esfeliç}(x) \;)$$

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

Passem tot a forma clausal:

(*a*) "Tot drac està feliç si tots els seus fills poden volar"

(*a*) [*cont.*] $\forall x\ (\ \exists y\ (\ fillde(x,y) \wedge \neg\, vola(y)\ )\ \vee\ esfeliç(x)\ )$

$\forall x\ (\ \exists y\ (\ fillde(x,y) \wedge \neg\, vola(y)\ )\ \vee\ esfeliç(x)\ )$

➤ Skolemizar (eliminar el $\exists$)

$\forall x\ (\ (\ fillde(x, f_y(x)) \wedge \neg\, vola(f_y(x))\ )\ \vee\ esfeliç(x)\ )$

➤ distributivitat $(F \wedge G) \vee H) \Rightarrow (F \vee H) \wedge (G \vee H)$

$$\forall x\ (\ (\ fillde(x, f_y(x)) \vee esfeliç(x)\ ) \wedge$$
$$(\ \neg\, vola(f_y(x)) \vee esfeliç(x)\ )$$

Això ens done dues clàusules:

(*a*1) $fillde(x, f_y(x)) \vee esfeliç(x)$

(*a*2) $\neg\, vola(f_y(x)) \vee esfeliç(x)$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

Passem tot a forma clausal:

$(\neg d)$ "No tots els dracs verds són feliços"

$(\neg d)$ $\exists x\,(\,esverd(x) \land \neg\, esfeli\c{c}(x)\,)$

$\exists x \quad (\quad esverd(x) \quad \land \quad \neg\, esfeli\c{c}(x) \quad)$

Per al proper dia de classe:

- Recorda la lliçó de l'examen parcial.
  Per a estudiar teoria de LI:
  - repassa els materials del que hem estudiat, i
  - FÉS ELS EXÀMENS PENJATS, començant pels últims, cap als anteriors, treballant sempre primer l'enunciat SENSE resoldre, i després l'examen resolt.
- Continueu fent els exercicis del tema 5. Pròxima classe els farem, i també els d'examen que em proposeu!!!

Lògica en la Informàtica
Deducció en Lògica de Primer Ordre (LPO)

José Miguel Rivero

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

Primavera 2025

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.

Temari

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)

Sumari

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:
   (a) “Tot drac està feliç si tots els seus fills poden volar”
   (b) “Els dracs verds poden volar”
   (c) “Un drac és verd si és fill d'almenys un drac verd”

Demostra per resolució que la conjunció de (a), (b) i (c) implica que:

   (d) “Tots els dracs verds són feliços”

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:
   - (a) "Tot drac està feliç si tots els seus fills poden volar"
   - (b) "Els dracs verds poden volar"
   - (c) "Un drac és verd si és fill d'almenys un drac verd"

Demostra per resolució que la conjunció de (a), (b) i (c) implica que:

- (d) "Tots els dracs verds són feliços"

$$\begin{aligned} \mathit{esfeliç}(x) &\equiv \text{“}x \text{ és feliç”} \\ \mathit{fillde}(x,y) &\equiv \text{“un fill de } x \text{ és } y\text{”} \\ \mathit{esverd}(x) &\equiv \text{“}x \text{ és verd”} \\ \mathit{vola}(x) &\equiv \text{“}x \text{ pot volar”} \end{aligned}$$

- Necessitem un predicat unari *esdrac*(x) ???
  Funcionaria, però NO cal, podem assumir
  que tots l'elements del domini són dracs.

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

(a) "Tot drac està feliç si tots els seus fills poden volar"

$$
\begin{aligned}
\mathit{esfeliç}(x) &\equiv \text{“}x \text{ és feliç”} \\
\mathit{fillde}(x, y) &\equiv \text{“un fill de } x \text{ és } y\text{”} \\
\mathit{esverd}(x) &\equiv \text{“}x \text{ és verd”} \\
\mathit{vola}(x) &\equiv \text{“}x \text{ pot volar”}
\end{aligned}
$$

(a) "Tot drac està feliç si tots els seus fills poden volar"

(a) $\forall x\ (\quad \ldots \quad \rightarrow \mathit{esfeliç}(x)\ )$

on $\ldots$ ha de dir que tots els fills de x poden volar:

$\forall y\ (\ \mathit{fillde}(x, y) \rightarrow \mathit{vola}(y)\ )$

I ens queda:

(a) $\forall x\ (\ \forall y\ (\ \mathit{fillde}(x, y) \rightarrow \mathit{vola}(y)\ ) \rightarrow \mathit{esfeliç}(x)\ )$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

   (b) "Els dracs verds poden volar"

$$
\begin{array}{lcl}
\mathit{esfeliç}(x) & \equiv & \text{"}x\text{ és feliç"} \\
\mathit{fillde}(x, y) & \equiv & \text{"un fill de }x\text{ és }y\text{"} \\
\mathit{esverd}(x) & \equiv & \text{"}x\text{ és verd"} \\
\mathit{vola}(x) & \equiv & \text{"}x\text{ pot volar"}
\end{array}
$$

   (b) "Els dracs verds poden volar"

   (b) $\forall x\ (\mathit{esverd}(x) \rightarrow \mathit{vola}(x)\ )$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

   (c) "Un drac és verd si és fill d'almenys un drac verd"

$$
\begin{array}{lcl}
\mathit{esfeliç}(x) & \equiv & \text{"}x\text{ és feliç"} \\
\mathit{fillde}(x,y) & \equiv & \text{"un fill de }x\text{ és }y\text{"} \\
\mathit{esverd}(x) & \equiv & \text{"}x\text{ és verd"} \\
\mathit{vola}(x) & \equiv & \text{"}x\text{ pot volar"}
\end{array}
$$

(c) "Un drac és verd si és fill d'almenys un drac verd"

(c) $\forall x\ (\quad \ldots \quad \rightarrow \mathit{esverd}(x)\ )$

on $\ldots$ ha de dir que x és fill de almenys un drac verd:

$\exists\, y\ (\ \mathit{fillde}(y,x) \wedge \mathit{esverd}(y)\ )$

I ens queda:

(c) $\forall x\ (\ \exists y\ (\ \mathit{fillde}(y,x) \wedge \mathit{esverd}(y)\ )\ \rightarrow\ \mathit{esverd}(x)\ )$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

La conjunció de (*a*), (*b*) i (*c*) implica (*d*) SSI

$(a) \land (b) \land (c) \quad \models \quad (d)$ SSI

$(a) \land (b) \land (c) \land \neg(d)$ INSAT SSI

$S = \texttt{formaclausal}(\,(a) \land (b) \land (c) \land \neg(d)\,)$ INSAT SSI

$\Box \in ResFact(S)$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

Passem tot a forma clausal:

(d) "Tots els dracs verds són feliços"

$(\neg d)$ "No tots els dracs verds són feliços"

$$
\begin{array}{lllll}
(\neg d) & \neg \forall x\,(& verd(x) & \rightarrow & esfeliç(x)\,) \\
 & \neg \forall x\,(& \neg\, verd(x) & \vee & esfeliç(x)\,) \\
 & \exists x\,(& verd(x) & \wedge & \neg\, esfeliç(x)\,)
\end{array}
$$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

Passem tot a forma clausal:

(*a*) "Tot drac està feliç si tots els seus fills poden volar"

(*a*) $\forall x\, (\, \forall y\, (\, fillde(x,y) \rightarrow vola(y)\,)\, \rightarrow\, esfeliç(x)\,)$

$$\forall x \;(\; \forall y \;(\; fillde(x,y) \rightarrow vola(y) \;) \rightarrow esfeliç(x) \;)$$

- eliminem les $\rightarrow$

$$\forall x \;(\; \neg\forall y \;(\; fillde(x,y) \rightarrow vola(y) \;) \vee esfeliç(x) \;)$$
$$\forall x \;(\; \neg\forall y \;(\; \neg fillde(x,y) \vee vola(y) \;) \vee esfeliç(x) \;)$$

- moure les $\neg$

$$\forall x \;(\; \exists y \;\neg(\; \neg fillde(x,y) \vee vola(y) \;) \vee esfeliç(x) \;)$$

- moure les $\neg$ (de Morgan)

$$\forall x \;(\; \exists y \;(\; fillde(x,y) \wedge \neg vola(y) \;) \vee esfeliç(x) \;)$$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

Passem tot a forma clausal:

(a) "Tot drac està feliç si tots els seus fills poden volar"

$$(a)\ [cont.]\ \forall x\ (\ \exists y\ (\ fillde(x,y) \land \neg\, vola(y)\ )\ \lor\ esfeliç(x)\ )$$

$$\forall x\ (\ \exists y\ (\ fillde(x,y) \land \neg\, vola(y)\ )\ \lor\ esfeliç(x)\ )$$

➤ Skolemizar (eliminar el $\exists$)

$$\forall x\ (\ (\ fillde(x, f_y(x)) \land \neg\, vola(f_y(x))\ )\ \lor\ esfeliç(x)\ )$$

➤ distributivitat $(F \land G) \lor H) \Rightarrow (F \lor H) \land (G \lor H)$

$$\forall x\ (\ (\ fillde(x, f_y(x)) \lor esfeliç(x)\ ) \land$$
$$(\ \neg\, vola(f_y(x)) \lor esfeliç(x)\ )$$

Això ens done dues clàusules:

($a1$) $fillde(x, f_y(x)) \lor esfeliç(x)$

($a2$) $\neg\, vola(f_y(x)) \lor esfeliç(x)$

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

Passem tot a forma clausal:

(b) "Els dracs verds poden volar"

(b) $\forall x \; (esverd(x) \rightarrow vola(x)\;)$

$\forall x \quad (\quad esverd(x) \quad \rightarrow \quad vola(x) \quad)$

- eliminem qles $\rightarrow$

$\forall x \quad (\quad \neg\, esverd(x) \quad \vee \quad vola(x) \quad)$

Això ens done una clàusula:

(b) $\neg\, esverd(x) \vee vola(x)$

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

Passem tot a forma clausal:

(c) "Un drac és verd si és fill d'almenys un drac verd"

(c) $\forall x \,(\, \exists y \,(\, \mathit{fillde}(y,x) \wedge \mathit{esverd}(y)\,) \;\rightarrow\; \mathit{esverd}(x)\,)$

$\forall x \;(\; \exists y \;(\; \mathit{fillde}(y,x) \;\wedge\; \mathit{esverd}(y) \;) \;\rightarrow\; \mathit{esverd}(x) \;)$

- eliminem les $\rightarrow$

$\forall x \;(\; \neg\exists y \;(\; \mathit{fillde}(y,x) \;\wedge\; \mathit{esverd}(y) \;) \;\vee\; \mathit{esverd}(x) \;)$

- moure les $\neg$

$\forall x \;(\; \forall y \;\neg(\; \mathit{fillde}(y,x) \;\wedge\; \mathit{esverd}(y) \;) \;\vee\; \mathit{esverd}(x) \;)$

- moure les $\neg$ amb de Morgan

$\forall x \;(\; \forall y \;(\; \neg\mathit{fillde}(y,x) \;\vee\; \neg\,\mathit{esverd}(y) \;) \;\vee\; \mathit{esverd}(x) \;)$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

Passem tot a forma clausal:

(c) "Un drac és verd si és fill d'almenys un drac verd"

(c) [cont.] $\forall x \; (\; \forall y \; (\; \neg fillde(y,x) \vee \neg \; esverd(y) \;) \;) \vee esverd(x) \;)$

$$\forall x \quad (\quad \forall y \quad (\quad \neg fillde(y,x) \quad \vee \quad \neg \; esverd(y) \quad) \quad \vee \quad esverd(x) \quad)$$

$$\forall x \quad \quad \forall y \quad (\quad \neg fillde(y,x) \quad \vee \quad \neg \; esverd(y) \quad \quad \vee \quad esverd(x) \quad)$$

Això ens done una clàusula:

(c) $\neg \, fillde(y,x) \vee \neg \, esverd(y) \vee esverd(x)$

és com:

```
esverd(x) :- fillde(y,x), esverd(y).
```

$\texttt{esverd(x)} \leftarrow \texttt{fillde(y,x)} \wedge \texttt{esverd(y)}$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

Passem tot a forma clausal:

$(\neg d)$ "No tots els dracs verds són feliços"

$(\neg d)$ $\exists x\,(\,esverd(x) \wedge \neg\, esfeliç(x)\,)$

$$\exists x \quad (\quad esverd(x) \quad \wedge \quad \neg\, esfeliç(x) \quad)$$

➤ Skolem

$$esverd(c_x) \quad \wedge \quad \neg\, esfeliç(c_x)$$

Això ens done dues clàusules:

$(\neg d1)$ $esverd(c_x)$

$(\neg d2)$ $\neg\, esfeliç(c_x)$

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

Juntem totes les clàusules:

$$
\begin{array}{rl}
(a1) & \mathit{fillde}(x, f_y(x)) \lor \mathit{esfeliç}(x) \\
(a2) & \neg\, \mathit{vola}(f_y(x)) \lor \mathit{esfeliç}(x) \\
(b) & \neg\, \mathit{esverd}(x) \lor \mathit{vola}(x) \\
(c) & \neg\, \mathit{fillde}(y, x) \lor \neg\, \mathit{esverd}(y) \lor \mathit{esverd}(x) \\
(\neg d1) & \mathit{esverd}(c_x) \\
(\neg d2) & \neg\, \mathit{esfeliç}(c_x)
\end{array}
$$

Hem de fer resolució (i factorització, degut a que no és de Horn), i intentar obtenir $\square$:

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

$a1.$	$\mathit{fillde}(x, f_y(x)) \vee \mathit{esfeliç}(x)$
$a2.$	$\neg \mathit{vola}(f_y(x)) \vee \mathit{esfeliç}(x)$
$b.$	$\neg \mathit{esverd}(x) \vee \mathit{vola}(x)$
$c.$	$\neg \mathit{fillde}(y, x) \vee \neg \mathit{esverd}(y) \vee \mathit{esverd}(x)$
$\neg d1.$	$\mathit{esverd}(c_x)$
$\neg d2.$	$\neg \mathit{esfeliç}(c_x)$

		amb:	*mgu*:	
1.	$\mathit{vola}(c_x)$	res $d1$. amb b.	$\{x = c_x\}$	$\leftarrow (*)$
2.	$\mathit{fillde}(c_x, f_y(c_x))$	res $d2$. amb $a1$.	$\{x = c_x\}$	
3.	$\neg \mathit{esverd}(c_x) \vee \mathit{esverd}(f_y(c_x))$	res 2. amb c.	$\{y = c_x, x = f_y(c_x)\}$	
4.	$\mathit{esverd}(f_y(c_x))$	res 3. amb $\neg d1$.	$\{\}$	
5.	$\neg \mathit{vola}(f_y(c_x))$	res $a2$. amb $\neg d2$.	$\{x = c_x\}$	
6.	$\neg \mathit{esverd}(f_y(c_x))$	res 5. amb b.	$\{x = f_y(c_x)\}$	
7.	$\square$	res 4. amb 6.	$\{\}$	

$(*)$: Al final aquesta clàusula 1. no la fem servir per res!

Exercici 7

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

$$
\begin{array}{rl}
a1. & \mathit{fillde}(x, f_y(x)) \vee \mathit{esfeliç}(x) \\
a2. & \neg \mathit{vola}(f_y(x)) \vee \mathit{esfeliç}(x) \\
b. & \neg \mathit{esverd}(x) \vee \mathit{vola}(x) \\
c. & \neg \mathit{fillde}(y, x) \vee \neg \mathit{esverd}(y) \vee \mathit{esverd}(x) \\
\neg d1. & \mathit{esverd}(c_x) \\
\neg d2. & \neg \mathit{esfeliç}(c_x)
\end{array}
$$

Nota:

Hi ha altres formes d'obtenir $\square$.

6) Consider the following Prolog program and its well-known behaviour:

```
animals([dog,lion,elephant]).
bigger(lion,cat).
faster(lion,cat).
better(X,Y):- animals(L), member(X,L), bigger(X,Y), faster(X,Y).

    ?- better(U,V).
    U = lion
    V = cat
```

In Prolog, a list like `[dog,lion,elephant]` is in fact represented as a term `f(dog,f(lion,f(elephant,emptylist)))`.

Therefore, we assume that the program also contains the standard clauses for `member` like this:

```
member( E, f(E,_) ).
member( E, f(_,L) ):- member(E,L).
```

6) Consider the following Prolog program and its well-known behaviour:

```
animals([dog,lion,elephant]).
bigger(lion,cat).
faster(lion,cat).
better(X,Y):- animals(L), member(X,L), bigger(X,Y), faster(X,Y).

member( E, f(E,_) ).
member( E, f(_,L) ):- member(E,L).
```

Express the program as a set of first-order clauses P and prove that $\exists u\, \exists v\ \mathit{better}(u, v)$ is a logical consequence of P. Which values did the variables u and v get (by unification) in your proof? **Only write the steps and values. No explanations.**

Examen final de 2017 tardor. Exercici 6

```
better(X,Y):- animals(L), member(X,L), bigger(X,Y), faster(X,Y).
```

$$better(X,Y) \leftarrow animals(L) \wedge member(X,L) \wedge bigger(X,Y) \wedge faster(X,Y)$$

$$a \rightarrow b \equiv \neg a \vee b$$

$$better(X,Y) \vee \neg(\, animals(L) \wedge member(X,L) \wedge bigger(X,Y) \wedge faster(X,Y)\,)$$

$$better(X,Y) \vee \neg animals(L) \vee \neg member(X,L) \vee \neg bigger(X,Y) \vee \neg faster(X,Y)$$

Examen final de 2017 tardor. Exercici 6

Les clàusulas de P són les següents:

1. $animals(f(dog, f(lion, f(elephant, emptylist))))$
2. $bigger(lion, cat)$
3. $faster(lion, cat)$
4. $better(X, Y) \vee \neg animals(L) \vee \neg member(X, L) \vee \neg bigger(X, Y) \vee \neg faster(X, Y)$
5. $member(E, f(E, \_))$
6. $member(E, f(\_, L)) \vee \neg member(E, L)$

La negació de $\exists u \exists v \; better(u, v)$ és:

$\neg \exists u \exists v \; better(u, v)$

$\forall u \neg \exists v \; better(u, v)$

$\forall u \forall v \neg better(u, v)$

que en forma clausal (ometent els $\forall u \forall v$) és:

7. $\neg better(u, v)$

Resolució en LPO:

$$\frac{A \lor C \qquad\qquad \neg B \lor D}{(C \lor D)\sigma}$$

A, B són àtoms
si $\sigma = mgu(A, B)$
(*most general unifier*)

He d'obtenir la clàusula buida $\square$ mitjançant resolució a partir d'aquestes 7 clàusules.

1. $animals(f(dog, f(lion, f(elephant, emptylist))))$
2. $bigger(lion, cat)$
3. $faster(lion, cat)$
4. $better(X, Y) \lor \neg animals(L) \lor \neg member(X, L) \lor \neg bigger(X, Y) \lor \neg faster(X, Y)$
5. $member(E, f(E, \_))$
6. $member(E, f(\_, L)) \lor \neg member(E, L)$
7. $\neg better(u, v)$

Examen final de 2017 tardor. Exercici 6

1. $animals(f(dog, f(lion, f(elephant, emptylist))))$
2. $bigger(lion, cat)$
3. $faster(lion, cat)$
4. $better(X, Y) \vee \neg animals(L) \vee \neg member(X, L) \vee \neg bigger(X, Y) \vee \neg faster(X, Y)$
5. $member(E, f(E, \_))$
6. $member(E, f(\_, L)) \vee \neg member(E, L)$
7. $\neg better(u, v)$

	res entre	**mgu**	
	4+7	$\{u = X, v = Y\}$	obtenim:
8.	$\neg animals(L) \vee \neg member(X, L) \vee \neg bigger(X, Y) \vee \neg faster(X, Y)$		
	1+8	$\{L = f(dog, f(lion, f(elephant, emptylist)))\}$	obtenim:
9.	$\neg member(X, f(dog, f(lion, \ldots) \vee \neg bigger(X, Y) \vee \neg faster(X, Y)$		
	6+9	$\{E = X, L = f(lion, f(elephant, emptylist))\}$	obtenim:
10.	$\neg member(X, f(lion, \ldots)) \vee \neg bigger(X, Y) \vee \neg faster(X, Y)$		
	5+10	$\{X = lion, E = lion, \_ = f(elephant, emptylist)\}$	obtenim:
11.	$\neg bigger(lion, Y) \vee \neg faster(lion, Y)$		
	2+11	$\{Y = cat\}$	obtenim:
12.	$\neg faster(lion, cat)$		
	3+12	$\{\ \}$	obtenim:
13.	$\square$		

Examen final de 2017 tardor. Exercici 6

...

7. $\neg better(u, v)$

	res entre	**mgu**	
	4+7	$\{u = X, v = Y\}$	obtenim:
8.	$\neg animals(L) \vee \neg member(X, L) \vee \neg bigger(X, Y) \vee \neg faster(X, Y)$		
	1+8	$\{L = f(dog, f(lion, f(elephant, emptylist)))\}$	obtenim:
9.	$\neg member(X, f(dog, f(lion, \ldots) \vee \neg bigger(X, Y) \vee \neg faster(X, Y)$		
	6+9	$\{E = X, L = f(lion, f(elephant, emptylist))\}$	obtenim:
10.	$\neg member(X, f(lion, \ldots)) \vee \neg bigger(X, Y) \vee \neg faster(X, Y)$		
	5+10	$\{X = lion, E = lion, \_ = f(elephant, emptylist)\}$	obtenim:
11.	$\neg bigger(lion, Y) \vee \neg faster(lion, Y)$		
	2+11	$\{Y = cat\}$	obtenim:
12.	$\neg faster(lion, cat)$		
	3+12	$\{\ \}$	obtenim:
13.	$\square$		

$u = X = lion$

$v = Y = cat$

Hem vist que no sols hem demostrat que $P \models \exists u \exists v\ better(u, v)$, sinó que fins i tot hem calculat dos valors concrets d'u i v.

4) (3 points) For 4a and 4b, just write the simplest and cleanest possible formula F. Use no more predicate or function symbols than just p. Give no explanations.

4a) Write a satisfiable first-order formula F, using only a *binary* predicate p, such that all models I of F have an infinite domain D_I.

4b) Write a satisfiable formula F of first-order logic with equality, using only a *unary* predicate p, such that F expresses that there is a single element satisfying p, that is, all models I of F have a single (unique) element e in its domain D_I such that $p_I(e) = 1$.

4a) Write a satisfiable first-order formula F, using only a *binary* predicate p, such that all models I of F have an infinite domain D_I.

Resposta:

$\forall x\, \neg p(x,x)$	(irreflexivitat)
$\wedge$	
$\forall x\, \forall y\, \forall z\, (\, p(x,y) \wedge p(y,z) \rightarrow p(x,z)\,)$	(transitivitat)
$\wedge$	
$\forall x\, \exists\, y\; p(x,y)$	("existència de successors")

4b) Write a satisfiable formula F of first-order logic with equality, using only a *unary* predicate p, such that F expresses that there is a single element satisfying p, that is, all models I of F have a single (unique) element e in its domain D_I such that $p_I(e) = 1$.

Resposta:

$$\exists x\,(\quad p(x) \quad\land\quad \forall y\,(\neg eq(x,y) \rightarrow \neg p(y))\quad)$$

5) (3 points) Let F be the first-order formula
$\exists x \forall y \exists z\ (\ p(z, y) \wedge \neg p(x, y)\ )$.

5a) Give a model I of F with $D_I = \{a, b, c\}$.

5b) Is it true that $F \models \forall x\ p(x, x)$?

5c) Is there any model of F with a single-element domain?

5) (3 points) Let F be the first-order formula
$\exists x \forall y \exists z \; (\; p(z, y) \wedge \neg p(x, y) \;)$.

5a) Give a model I of F with $D_I = \{a, b, c\}$.

Solució:

$$
\begin{aligned}
&D_I = \{a, b, c\} \\
&p_I(a, a) = 0 \\
&p_I(a, b) = 0 \\
&p_I(a, c) = 0 \\
&p_I(b, a) = 1 \\
&p_I(b, b) = 1 \\
&p_I(b, c) = 1 \\
&p_I(c, a) = \text{no importa} \\
&p_I(c, b) = \text{no importa} \\
&p_I(c, c) = \text{no importa}
\end{aligned}
$$

5) (3 points) Let F be the first-order formula
$\exists x \forall y \exists z\ (\ p(z,y) \wedge \neg p(x,y)\ )$.

5b) Is it true that $F \models \forall x\ p(x,x)$?

Solució:

No. perquè existeix una I tal que I és model de F, és a dir $I \models F$, però I no és model de $\forall x\, p(x,x)$.
I aquesta I és la de l'apartat **5a)**.

5) (3 points) Let F be the first-order formula
$\exists x \forall y \exists z\ (\ p(z,y) \land \neg p(x,y)\ )$.

5c) Is there any model of F with a single-element domain?

Solució:

No. Si tinguéssim $D_I = \{e\}$, hauríem de definir

$$p_I(e,e) = 1$$

o bé

$$p_I(e,e) = 0$$

En tots dos casos, si $x{=}e$, $y{=}e$, i $z{=}e$, no es compleix
$p_I(e,e) \land \neg p_I(e,e)$

6) (4 points) Formalize and prove by resolution that sentence F is a logical consequence of the first five:

A: All people that have electric cars are ecologists.

B: If someone has a grandmother, then that someone has a mother whose mother is that grandmother.

C: A person is an ecologist if his/her mother is an ecologist.

D: Mary is John's grandmother.

E: Mary has an electric car.

F: John is an ecologist.

6) (4 points) Formalize and prove by resolution that sentence F is a logical consequence of the first five:

A: All people that have electric cars are ecologists.
B: If someone has a grandmother, then that someone has a mother whose mother is that grandmother.
C: A person is an ecologist if his/her mother is an ecologist.
D: Mary is John's grandmother.
E: Mary has an electric car.
F: John is an ecologist.

Resposta (predicats):

$$\begin{array}{lcl} hasEcar(x) & \equiv & \text{“}x \text{ has an electric car”} \\ isEcologist(x) & \equiv & \text{“}x \text{ is an ecologist”} \\ mother(x,y) & \equiv & \text{“}y \text{ is the mother of x”} \\ grandma(x,y) & \equiv & \text{“}y \text{ is the grandmother of x”} \end{array}$$

Per al proper dia de classe:

- Per a estudiar teoria de LI:
  - repassa els materials del que hem estudiat, i
  - FÉS ELS EXÀMENS PENJATS, començant pels últims, cap als anteriors, treballant sempre primer l'enunciat SENSE resoldre, i després l'examen resolt.
- Continueu fent els exercicis del tema 5. Pròxima classe els farem, i també els d'examen que em proposeu!!!

Lògica en la Informàtica
Deducció en Lògica de Primer Ordre (LPO)

José Miguel Rivero

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

Primavera 2025

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.

6) (4 points) Formalize and prove by resolution that sentence F is a logical consequence of the first five:

A: All people that have electric cars are ecologists.

B: If someone has a grandmother, then that someone has a mother whose mother is that grandmother.

C: A person is an ecologist if his/her mother is an ecologist.

D: Mary is John's grandmother.

E: Mary has an electric car.

F: John is an ecologist.

6) (4 points) Formalize and prove by resolution that sentence F is a logical consequence of the first five:

A: All people that have electric cars are ecologists.

B: If someone has a grandmother, then that someone has a mother whose mother is that grandmother.

C: A person is an ecologist if his/her mother is an ecologist.

D: Mary is John's grandmother.

E: Mary has an electric car.

F: John is an ecologist.

Resposta (predicats):

$$
\begin{array}{lcl}
hasEcar(x) & \equiv & \text{"}x \text{ has an electric car"} \\
isEcologist(x) & \equiv & \text{"}x \text{ is an ecologist"} \\
mother(x, y) & \equiv & \text{"}y \text{ is the mother of x"} \\
grandma(x, y) & \equiv & \text{"}y \text{ is the grandmother of x"}
\end{array}
$$

6) (4 points) Formalize and prove by resolution that sentence F is a logical consequence of the first five:

A: All people that have electric cars are ecologists.

$$\begin{array}{lcl} hasEcar(x) & \equiv & \text{“}x \text{ has an electric car”} \\ isEcologist(x) & \equiv & \text{“}x \text{ is an ecologist”} \\ mother(x,y) & \equiv & \text{“}y \text{ is the mother of x”} \\ grandma(x,y) & \equiv & \text{“}y \text{ is the grandmother of x”} \end{array}$$

A: $\forall x\,(hasEcar(x) \rightarrow isEcologist(x))$

A $\neg hasEcar(x) \vee isEcologist(x)$

6) (4 points) Formalize and prove by resolution that sentence F is a logical consequence of the first five:

B: If someone has a grandmother, then that someone has a mother whose mother is that grandmother.

$hasEcar(x)$	$\equiv$	"x has an electric car"
$isEcologist(x)$	$\equiv$	"x is an ecologist"
$mother(x,y)$	$\equiv$	"y is the mother of x"
$grandma(x,y)$	$\equiv$	"y is the grandmother of x"

NO és correcte:

$$\forall x \ \ (\exists y \, (\, grandma(x,y) \ \rightarrow \ \exists z \, (mother(x,z) \wedge mother(z,y) \,) \ \ ) \ \ )$$
$$\forall x \ \ (\exists y \, (\, \neg grandma(x,y) \ \vee \ \exists z \, (mother(x,z) \wedge mother(z,y) \,) \ \ ) \ \ )$$

6) (4 points) Formalize and prove by resolution that sentence F is a logical consequence of the first five:

B: If someone has a grandmother, then that someone has a mother whose mother is that grandmother.

NO és correcte:

$$\forall x \ (\exists y (\neg grandma(x,y) \vee \exists z (mother(x,z) \wedge mother(z,y))))$$

Aquesta formalització del llenguatge natural no és adecuada:
si tenim una situació I amb persones $D_I = \{p1, p2, avia\}$ i on l'àvia de $p1$ es $avia$: $\quad grandma_I(p1, avia) = 1$
i on tota la resta és fals (ningú és mare de ningú, etc.) llavors I satisfà la fórmula, perquè $\forall x \exists y \neg grandma(x,y)$. De fet, amb aquesta formalització no és possible obtenir la clàusula buida.

6) (4 points) Formalize and prove by resolution that sentence F is a logical consequence of the first five:

B: If someone has a grandmother, then that someone has a mother whose mother is that grandmother.

El que sí és correcte és:

$\forall x\, \forall y\ (\ grandma(x,y)\ \rightarrow\ \exists z\ (\ mother(x,z) \wedge mother(z,y)\ )\ )$

- Elim. $\rightarrow$

$\forall x\, \forall y\ (\ \neg grandma(x,y)\ \vee\ \exists z\ (mother(x,z) \wedge mother(z,y)\ )\ )$

- Skolem:

$$\forall x\, \forall y\ (\ \neg grandma(x,y)\ \vee\ (\ mother(x, f_z(x,y)) \wedge mother(f_z(x,y), y)\ )\ \ )$$

- Distrib:

B1 $\neg grandma(x,y) \vee mother(x, f_z(x,y))$
B2 $\neg grandma(x,y) \vee mother(f_z(x,y), y))$

6) (4 points) Formalize and prove by resolution that sentence F is a logical consequence of the first five:

C: A person is an ecologist if his/her mother is an ecologist.

$$\begin{array}{lcl} hasEcar(x) & \equiv & \text{“}x \text{ has an electric car”} \\ isEcologist(x) & \equiv & \text{“}x \text{ is an ecologist”} \\ mother(x,y) & \equiv & \text{“}y \text{ is the mother of x”} \\ grandma(x,y) & \equiv & \text{“}y \text{ is the grandmother of x”} \end{array}$$

C: $\forall x\, (\ \exists y\, (\, mother(x,y) \wedge isEcologist(y)\,) \ \rightarrow\ isEcologist(x)\ )$

$\forall x\, (\neg \exists y\, (\, mother(x,y) \wedge isEcologist(y)\,) \ \vee\ isEcologist(x)\,)$

$\forall x\, (\ \forall y\, \neg (\, mother(x,y) \wedge isEcologist(y)\,) \ \vee\ isEcologist(x)\,)$

$\forall x\, (\ \forall y\, (\neg\, mother(x,y) \vee \neg\, isEcologist(y)\,) \vee isEcologist(x)\,)$

C $\neg\, mother(x,y) \vee \neg\, isEcologist(y) \vee isEcologist(x)$

6) (4 points) Formalize and prove by resolution that sentence F is a logical consequence of the first five:

$$
\begin{array}{lcl}
hasEcar(x) & \equiv & \text{“}x \text{ has an electric car”} \\
isEcologist(x) & \equiv & \text{“}x \text{ is an ecologist”} \\
mother(x, y) & \equiv & \text{“}y \text{ is the mother of x”} \\
grandma(x, y) & \equiv & \text{“}y \text{ is the grandmother of x”}
\end{array}
$$

D: Mary is John's grandmother.

D $grandma(john, mary)$

E: Mary has an electric car.

E $hasEcar(mary)$

F: John is an ecologist.

¬ F: John is not an ecologist.

¬ F $\neg isEcologist(john)$

Examen final de 2020 tardor. Exercici 6

Volem demostrar que $A \wedge B \wedge C \wedge D \wedge E \models F$.

I això passa ssi $A \wedge B \wedge C \wedge D \wedge E \wedge \neg F$ és insatisfactible.

A	$\neg hasEcar(x) \vee isEcologist(x)$
B1	$\neg grandma(x, y) \vee mother(x, f_z(x, y))$
B2	$\neg grandma(x, y) \vee mother(f_z(x, y), y))$
C	$\neg mother(x, y) \vee \neg isEcologist(y) \vee isEcologist(x)$
D	$grandma(john, mary)$
E	$hasEcar(mary)$
¬F	$\neg isEcologist(john)$

He d'obtenir la $\square$ mitjançant resolució a partir d'aquestes 7 clàusules.

Examen final de 2020 tardor. Exercici 6

A	$\neg hasEcar(x) \vee isEcologist(x)$
B1	$\neg grandma(x, y) \vee mother(x, f_z(x, y))$
B2	$\neg grandma(x, y) \vee mother(f_z(x, y), y))$
C	$\neg mother(x, y) \vee \neg isEcologist(y) \vee isEcologist(x)$
D	$grandma(john, mary)$
E	$hasEcar(mary)$
¬F	$\neg isEcologist(john)$

	res entre	mgu	
	E+A	$\{x = mary\}$	obtenim:
1.	$isEcologist(mary)$		
	D+B1	$\{x = john, y = mary\}$	obtenim:
2.	$mother(john, f_z(john, mary))$		
	D+B2	$\{x = john, y = mary\}$	obtenim:
3.	$mother(f_z(john, mary), mary)$		
	2+C	$\{x = john, y = f_z(john, mary)\}$	obtenim:
4.	$\neg isEcologist(f_z(john, mary)) \vee isEcologist(john)$		
	4+¬F	$\{\ \}$	obtenim:
5.	$\neg isEcologist(f_z(john, mary))$		

Examen final de 2020 tardor. Exercici 6

A	$\neg hasEcar(x) \vee isEcologist(x)$
B1	$\neg grandma(x,y) \vee mother(x, f_z(x,y))$
B2	$\neg grandma(x,y) \vee mother(f_z(x,y), y)$
C	$\neg mother(x,y) \vee \neg isEcologist(y) \vee isEcologist(x)$
D	$grandma(john, mary)$
E	$hasEcar(mary)$
¬F	$\neg isEcologist(john)$

res entre **mgu**

1. $isEcologist(mary)$
2. $mother(john, f_z(john, mary))$
3. $mother(f_z(john, mary), mary)$
4. $\neg isEcologist(f_z(john, mary)) \vee isEcologist(john)$
5. $\neg isEcologist(f_z(john, mary))$

3+C $\{x = f_z(john, mary), y = mary\}$ obtenim:

6. $\neg isEcologist(mary) \vee isEcologist(f_z(john, mary))$

6+5 $\{\ \}$ obtenim:

7. $\neg isEcologist(mary)$

1+7 $\{\ \}$ obtenim:

8. $\square$

Per a estudiar teoria de LI:

- repassa els materials que hem estudiat.
- FÉS ELS EXÀMENS PENJATS, començant pels últims, cap als anteriors, treballant sempre primer l'enunciat SENSE resoldre, i després l'examen resolt.
- continua fent els exercicis del tema 5.

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.

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 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)

Introducció i motivació

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.

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?
  - $A \rightarrow B$ és el mateix que $\neg A \vee B$
  - $\forall x\,(pen(x) \rightarrow \neg par(x)) \;\equiv\; \forall x\,(\neg pen(x) \vee \neg par(x))$
  - $\forall x\,(par(x) \rightarrow \neg pen(x)) \;\equiv\; \forall x\,(\neg par(x) \vee \neg pen(x)) \;\equiv\; \forall x\,(\neg pen(x) \vee \neg par(x))$
  - 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. $\neg m \to e \;\equiv\; m \vee e$ (totes les coses, o bé maten, o bé engreixen. no hi ha res que ni mata ni engreixa)
     $\equiv\; \neg e \to m$
  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.

Introducció i motivació

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? $\quad I \models F$?
- 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

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.

Definició de Lògica Proposicional

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 $G \models F$.

Definició de Lògica Proposicional

- **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 $I : \mathcal{P} \rightarrow \{0,1\}$.
  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 $\mathcal{P}$)
    - $eval_I(\neg F) = 1 - eval_I(F)$
    - $eval_I(F \wedge G) = min(eval_I(F), eval_I(G))$
    - $eval_I(F \vee G) = max(eval_I(F), eval_I(G))$
- 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 $p \in \mathcal{P}$
  - $eval_I(\neg F) \quad = 1 - eval_I(F)$
  - $eval_I(F \wedge G) = min(eval_I(F), eval_I(G))$
  - $eval_I(F \vee G) = max(eval_I(F), eval_I(G))$

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 $F = p \land ((q \lor \neg r) \land ((\neg p \lor r) \land \neg q))$

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 $\implies \dots \implies$ $\neg F$ és insatisfactible

B) $\neg F$ és insatisfactible $\implies \dots \implies$ 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