

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



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)

- 1 Unificació. Algorisme d'unificació
- 2 Exemple de recapitulació
- 3 Regla deductiva de Factorització
- 4 Completitud refutacional en LPO
- 5 Exercici 7 [“Tot drac està feliç...” (Schöning, Exercise 85)]

Unificació

- una “substitució” σ és un conjunt de parells variable-terme:
 $\{x_1 = t_1, \dots, x_n = t_n\}$
- “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. $E \cup \{t = t\} \implies E$
- 1. $E \cup \{f(\dots) = g(\dots)\} \implies$ fallo
si $f \neq g$ (no són unificables!)
- 2. $E \cup \{f(s_1 \dots s_n) = f(t_1 \dots t_n)\} \implies E \cup \{s_1 = t_1 \dots s_n = t_n\}$
- 3. $E \cup \{x = t\} \implies$ fallo
si x apareix en t , i
 x no ÉS t (per ex. $x = f(x)$)
- 4. $E \cup \{x = t\} \implies E \cup \{x = t\} \cup \{x = t\}$
si x NO apareix en t , i
a més x SI apareix en E

Unificació. Algorisme d'unificació

Algorisme d'unificació:

Exemple:

$$\begin{array}{ccc} \{f(x, g(x, a)) = f(h(b), z)\} & \xRightarrow{2} & \\ \underbrace{\{x = h(b)\}}_{x=t}, \underbrace{\{z = g(x, a)\}}_E & \xRightarrow{4} & \{x = h(b), z = g(h(b), a)\} \end{array}$$

això és el *mgui*!

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 *mgui* del problema inicial?

Exemple de recapitulació

Petit exemple de recapitulació

Exemple:

Vull saber si $F \models G$ en LPO. Aquí F i G són fórmules qualssevol.
Què faig?

$F \models G$

la fórmula $F \wedge \neg G$ és insat

$S = \text{forma_clausal}(F \wedge \neg G)$ és insat

la clàusula buida \square està en $\text{Res}(S)$

ssi

ssi

ssi (*gairebé*)

(puc obtenir \square 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. Aquí F i G són fórmules qualssevol.
Què faig?

$$S_0 = S$$

$$S_1 = S_0 \cup \text{Res}_1(S_0)$$

...

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

Un detall important!! $\text{Res}(S)$ 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.

Lavors 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: $D_I = \{e, \dots\}$

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:

$$\frac{p(x) \vee p(y) \quad \neg p(z) \vee \neg p(z')}{p(y) \vee \neg p(z')} \quad mgu(p(x), p(z)) = \{x = z\}$$

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

$$S \text{ insat SSI } \Box \in \text{Res}(S).$$

Què és el que falta?

Fem el mateix “tipus” d'exemple en L.proposicional:

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

INSAT. Per resolució:

$$\begin{array}{ccc} \frac{p \vee q \quad p \vee \neg q}{p \vee p} & \frac{\neg p \vee q \quad \neg p \vee \neg q}{\neg p \vee \neg p} & \leftarrow (*) \\ \frac{p}{\Box} & \frac{\neg p}{\Box} & \end{array}$$

(*) 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:

$$\frac{A \vee B \vee C}{(A \vee C) \sigma} \quad \begin{array}{l} \text{on } A \text{ i } B \text{ átomos (literals POSITIUS),} \\ \text{i } C \text{ és la resta de la clàusula} \\ \sigma = mgu(A, B) \end{array}$$

Per exemple:

$$\frac{\overbrace{p(a, x)}^A \vee \overbrace{p(y, b)}^B \vee \overbrace{q(x, y) \vee \dots}^C}{(p(a, x) \vee q(x, y) \vee \dots) \sigma}$$
$$\begin{array}{l} p(a, b) \vee q(b, a) \vee \dots \\ \sigma = mgu(p(a, x), p(y, b)) \\ \quad = \{x=b, y=a\} \end{array}$$

Regla deductiva de Factorització

En què es basa això?

Si tinc:

$$\forall x \forall y \quad p(a, x) \vee p(y, b) \vee q(x, y)$$

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.!: :

$$\frac{p(x) \vee p(y)}{p(x)}$$

$$4. \neg p(z')$$

$$5. \square$$

$$\sigma = mgu(p(x), p(y)) = \{y=x\}$$

(aqui la part C és buida)

per resolució entre 2. i 3.:

$$\sigma = mgu(p(z), p(x)) = \{x=z\}$$

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:

$$S \text{ insat } SSI \quad \square \in ResFact(S)$$

Calculem $ResFact(S)$ per nivells:

$$S_0 = S$$

$$S_{i+1} = S_i \cup Res_1(S_i) \cup Fact_1(S_i)$$

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



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 $\iff \square \in Res(S)$

(Si S és Horn no necessito factorització!!!,
perquè mai tindrè ocasió d'aplicar-la!)

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, \dots)

- Mira els exercicis d'examen on es fa això. Per exemple, l'exercici 6 de l'examen final de 2017 tardor.

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”

$esfeliç(x)$	\equiv	“x és feliç”
$fillde(x, y)$	\equiv	“un fill de x és y”
$esverd(x)$	\equiv	“x és verd”
$vola(x)$	\equiv	“x pot volar”

► Necessitem un predicat unari $esdrac(x)$???

Funcionaria, però NO cal, podem assumir que tots l'elements del domini són dracs.

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

(a) “Tot drac està feliç si tots els seus fills poden volar”

$esfeliç(x) \equiv$ “x és feliç”

$fillde(x, y) \equiv$ “un fill de x és y”

$esverd(x) \equiv$ “x és verd”

$vola(x) \equiv$ “x pot volar”

(a) “Tot drac està feliç si tots els seus fills poden volar”

(a) $\forall x (\dots \rightarrow esfeliç(x))$

on ... ha de dir que tots els fills de x poden volar:

$\forall y (fillde(x, y) \rightarrow vola(y))$

I ens queda:

(a) $\forall x (\forall y (fillde(x, y) \rightarrow vola(y)) \rightarrow esfeliç(x))$

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

(b) “Els dracs verds poden volar”

$esfeliç(x) \equiv$ “x és feliç”

$fillde(x, y) \equiv$ “un fill de x és y”

$esverd(x) \equiv$ “x és verd”

$vola(x) \equiv$ “x pot volar”

(b) “Els dracs verds poden volar”

(b) $\forall x (esverd(x) \rightarrow vola(x))$

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

(c) “Un drac és verd si és fill d'almenys un drac verd”

$esfeliç(x) \equiv$ “x és feliç”

$fillde(x, y) \equiv$ “un fill de x és y”

$esverd(x) \equiv$ “x és verd”

$vola(x) \equiv$ “x pot volar”

(c) “Un drac és verd si és fill d'almenys un drac verd”

(c) $\forall x (\dots \rightarrow esverd(x))$

on ... ha de dir que x és fill de almenys un drac verd:

$\exists y (fillde(y, x) \wedge esverd(y))$

I ens queda:

(c) $\forall x (\exists y (fillde(y, x) \wedge esverd(y)) \rightarrow esverd(x))$

7. (Dificultat 3) (Schöning, Exercise 85) Formalitza els següents fets:

La conjunció de (a) , (b) i (c) implica (d) SSI

$(a) \wedge (b) \wedge (c) \models (d)$ SSI

$(a) \wedge (b) \wedge (c) \wedge \neg(d)$ INSAT SSI

$S = \text{formaclausals}((a) \wedge (b) \wedge (c) \wedge \neg(d))$ INSAT SSI

$\square \in \text{ResFact}(S)$

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”

($\neg d$) $\neg \forall x (\text{verd}(x) \rightarrow \text{esfeliç}(x))$

$\neg \forall x (\neg \text{verd}(x) \vee \text{esfeliç}(x))$

$\exists x (\text{verd}(x) \wedge \neg \text{esfeliç}(x))$

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”

$$(a) \forall x (\forall y (\text{fillde}(x, y) \rightarrow \text{vola}(y)) \rightarrow \text{esfeliç}(x))$$

$$\forall x (\quad \forall y (\quad \text{fillde}(x, y) \rightarrow \text{vola}(y)) \rightarrow \text{esfeliç}(x))$$

► eliminem les \rightarrow

$$\forall x (\quad \neg \forall y (\quad \text{fillde}(x, y) \rightarrow \text{vola}(y)) \vee \text{esfeliç}(x))$$

$$\forall x (\quad \neg \forall y (\quad \neg \text{fillde}(x, y) \vee \text{vola}(y)) \vee \text{esfeliç}(x))$$

► moure les \neg

$$\forall x (\quad \exists y \neg (\quad \neg \text{fillde}(x, y) \vee \text{vola}(y)) \vee \text{esfeliç}(x))$$

► moure les \neg (de Morgan)

$$\forall x (\quad \exists y (\quad \text{fillde}(x, y) \wedge \neg \text{vola}(y)) \vee \text{esfeliç}(x))$$

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.] $\forall x (\exists y (\text{fillde}(x, y) \wedge \neg \text{vola}(y)) \vee \text{esfeliç}(x))$

$$\forall x ((\exists y (\text{fillde}(x, y) \wedge \neg \text{vola}(y)) \vee \text{esfeliç}(x)))$$

► Skolemitzar (eliminar el \exists)

$$\forall x ((\text{fillde}(x, f_y(x)) \wedge \neg \text{vola}(f_y(x))) \vee \text{esfeliç}(x))$$

► distributivitat $(F \wedge G) \vee H \Rightarrow (F \vee H) \wedge (G \vee H)$

$$\forall x ((\text{fillde}(x, f_y(x)) \vee \text{esfeliç}(x)) \wedge (\neg \text{vola}(f_y(x)) \vee \text{esfeliç}(x)))$$

Això ens done dues clàusules:

(a1) $\text{fillde}(x, f_y(x)) \vee \text{esfeliç}(x)$

(a2) $\neg \text{vola}(f_y(x)) \vee \text{esfeliç}(x)$



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”

$(\neg d) \exists x (esverd(x) \wedge \neg esfeliç(x))$

$\exists x (esverd(x) \wedge \neg esfeliç(x))$

Per al proper dia de classe:

- Recorda la lliçó de l'examen parcial.

Per a estudiar teoria de Ll:

- 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!!!