Lògica en la Informàtica

Tardor 2022. Teoria classe 2

Lògica en la Informàtica

Temari

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

Lògica en la Informàtica. Introducció i motivació

- Perquè estudiar *Logic in Computer Science*?
  - Els grecs, els matemàtics, els informàtics
  - Estudi dels Fonaments
  - El llenguatge natural és imprecís, ambiguo
  - Què és una lògica?
    - sintaxis: què és una fórmula F?
    - semàntica:
      - què és una interpretació I?
      - quan una I SATISFÀ una F? I |= F?
  - La “deducció intuitiva” que fem nosaltres ens enganya
  - Aplicacions directes de la lògica en la informàtica

Lògica en la Informàtica. Definició de Lògica Proposicional

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

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

Lògica en la Informàtica. Definició de Lògica Proposicional

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

  & és AND

  v és OR

  \- és NOT

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

Lògica en la Informàtica. Definició de Lògica Proposicional

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

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

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

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

5. Siguin F i G dues fórmules qualsevol. És cert que $F \lor G$ és tautologia si i només si alguna de les dues fórmules F o G ho és?. Demostra fent servir només la definició de la LProp.

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

5. Siguin F i G dues fórmules qualsevol. És cert que F $\vee$ G és tautologia si i només si alguna de les dues fórmules F o G ho és?. Demostra 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 v 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 -p.

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

5. Siguin F i G dues fórmules qualsevol. És cert que F ∨ G és tautologia si i només si alguna de les dues fórmules F o G ho és?. Demostra fent servir només la definició de la LProp.

   a) Tenim que p v -p és tautologia perquè:

p v -p és tautologia	ssi	[per definició de tautologia]
tota I és model de p v -p	ssi	[per definició de model]
A I, I |= p v -p	ssi	[per definició de |=]
A I, eval_I(p v -p) = 1	ssi	[per definició de eval_I(… v …)]
A I, max(eval_I(p), eval_I(-p)) = 1	ssi	[per definició de eval_I(- …)]
A I, max(eval_I(p), 1 - eval_I(p)) = 1	ssi	[donat que eval_(p) sempre és 0 o 1, i per definició de max: max(0, 1-0) = max (1, 1-1) = 1]
A I, 1 = 1	ssi	[perquè 1 = 1 és cert]
cert		

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

5. Siguin F i G dues fórmules qualsevol. És cert que F ∨ G és tautologia si i només si alguna de les dues fórmules F o G ho és?. Demostra 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]
E I, tq I no és model de p	ssi	[per definició de model]
E I, no I |= p	ssi	[per definició de |=]
E I, eval_I(p) = 0	ssi	[per definició de eval_I(p)]
E I, I(p) = 0	ssi	cert

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

5. Siguin F i G dues fórmules qualsevol. És cert que F ∨ G és tautologia si i només si alguna de les dues fórmules F o G ho és?. Demostra 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 -p]
-p no és tautologia	ssi	[per definició de tautologia]
E I, tq I no és model de -p	ssi	[per definició de model]
E I, no I |= -p	ssi	[per definició de |=]
E I, eval_I(-p) = 0	ssi	[per definició de eval_I(-...)]
E I, 1 - eval_I(p) = 0	ssi	[per definició de eval_I(p)]
E I, 1 - I(p) = 0	ssi	[per aritmètica]
E I, I(p) = 1	ssi	cert

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

7. 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 $G \land \neg F$ és insatisfactible.

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

7. 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 |= F   si i només si G ∧ ¬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]
A I, si I |= G, llavors I |= F	ssi	[pel significat de si... llavors …]
A I, no I |= G o bé I |= F	ssi	[per def. de |=]
A I, eval_I(G) = 0 o bé eval_I(F) = 1	ssi	[per aritmètica]
A I, eval_I(G) = 0 o bé 1-eval_I(F) = 0	ssi	[per def. de min]
A I, min(eval_I(G), 1-eval_I(F)) = 0	ssi	[per def. de eval_I(-…)]
A I, min(eval_I(G), eval_I(¬F)) = 0	ssi	[per def. de eval_I(… & ...)]
A I, eval_I(G ∧ ¬F)=0	ssi	[per def. de model]
G ∧ ¬F no té models	ssi	[per def. de insatisfactible]
G ∧ ¬F és insatisfactible		

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

8. 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

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

8. 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]
A I, no $I \models (G \wedge \neg F) \vee (F \wedge \neg G)$	ssi	[per definició de $\models$]
A I, $eval\_I((G \wedge \neg F) \vee (F \wedge \neg G)) = 0$	ssi	[per definició de eval v]
A I, $max(eval\_I(G \wedge \neg F), eval\_I(F \wedge \neg G)) = 0$	ssi	[per definició de eval &]
A I, $max(min(eval\_I(G), eval\_I(\neg F)), min(eval\_I(F), eval\_I(\neg G))) = 0$	ssi	[per definició de eval -]
A 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]
A I, $eval\_I(F) = eval\_I(G)$	ssi	[perquè eval sempre dona 0 o 1]
A I, ($eval\_I(F) = 1$ ssi $eval\_I(G) = 1$)	ssi	[per definició de $\models$]
A I, ($I \models F$ ssi $I \models G$)	ssi	[per definició de model]
A 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.		

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

8. 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 ∧ ¬F) v (F ∧ ¬G) és insatisfactible ssi F <--> G és tautologia
   Per al segon ssi:

F <--> G és tautologia ssi [per def <-->]
(F --> G) & (G --> F) és tautologia ssi [per def -->]
(-F v G) & (-G v F) és tautologia ssi [per def. de tautologia]
tota I és model de (-F v G) & (-G v F) ssi [per def. de model]
A I, I |= (-F v G) & (-G v F) ssi [per def. de |=]
A I, eval_I((-F v G) & (-G v F)) = 1 ssi [per def. de eval_I(&)]
A I, min(eval_I(-F v G), eval_I(-G v F)) = 1 ssi [per def. de eval_I(v)]
A I, min(max(eval_I(-F), eval_I(G)), max(eval_I(-G), eval_I(F))) = 1 ssi [per def. de eval_I(-)]
A 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]

A I, eval_I(F) = eval_I(G)
y seguim igual que en la demostració anterior

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

Conseqüència dels exercicis 6,7,8:

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

F és **satisfactible** si F té algun model
F és **insatisfactible** si F no té models
F és **tautología** si tota I és model de F
G és **conseqüència lògica** de F si tot model de F satisfà G (es denota F |= G)
F i G són **lògicamente equivalents** si F i G tenen els mateixos models (es denota F ≡ G)

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

F és tautologia ssi -F és insatisfactible.
G és conseqüència lògica de F ssi $F \land \neg G$ és insatisfactible.
F y G són lògicamente equivalentes ssi $(G \land \neg F) \lor (F \land \neg G)$ és insatisfactible.

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

16. Siguin F i G dues fórmules qualsevols.
    Si F -> 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 -> G és satisfactible ssi [per def. de ->]
-F v G és satisfactible ssi [per def. de satisfactible]
-F v G te algun model ssi [per def. de model]
E I, I |= -F v G ssi [per def. de |=]
E I, eval_I(-F v G) = 1 ssi [per def. de eva_I(I v)]
E I, max(eval_I(-F), eval_I(G)) = 1 ssi [per def. de eval_I(-)]
E I, max(1-eval_I(F), eval_I(G)) = 1 ssi [per def. de max]
E I, 1-eval_I(F) = 1 **o** eval_I(G) = 1 ssi [per aritmètica]
E I, eval_I(F) = 0 **o** eval_I(G) = 1

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

16. Siguin F i G dues fórmules qualsevols.
    Si F -> 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 -> G és satisfactible	ssi	[per def. de ->]
…		
E I, max(eval_I(-F), eval_I(G)) = 1	ssi	[per def. de eval_I(-)]
E I, max(1-eval_I(F), eval_I(G)) = 1	ssi	[per def. de max]
E I, 1-eval_I(F) = 1 **o** eval_I(G) = 1	ssi	[per aritmètica]
E I, eval_I(F) = 0 **o** eval_I(G) = 1		

NO puc escriure E I, eval_I(F)=0 v eval_I(G)=1

NO té cap sentit, perquè v és una conectiva que només té sentit dintre de fórmules, y eval_I(F) = 0 NO és una fórmula. Aquí estem raonant/explicant en català/castellà.

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

16. Siguin F i G dues fórmules qualsevols.
    Si F -> 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]
E I', I' |= F	ssi	[per def. de |=]
E 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 -> G és satisfactible i F és satisfactible, i això no implica res sobre la G!

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

16. Siguin F i G dues fórmules qualsevols.
    Si F -> 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 -> 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 & -p.
Llavors F -> G és satisfactible i F és satisfactible, però G no ho és!

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

16. Siguin F i G dues fórmules qualsevols.
    Si F -> G és satisfactible i F és satisfactible, llavors G és satisfactible?
    Demostra-ho fent servir tan sols la definició de la LProp.

F -> G és satisfactible ssi [per def. de F i G]
p -> (p & -p) és satisfactible ssi [per def. de ->]
-p v (p & -p) és satisfactible ssi [per def. de satisfactible]
-p v (p & -p) té algun model ssi [per def. de modelo]
E I, I |= -p v (p & -p) ssi [per def. de |=]
E I, eval_I(-p v (p & -p)) = 1 ssi [per def. de eval_I(v)]
E I, max(eval_I(-p), eval_I(p & -p)) = 1 ssi [per def. de eval_I(-)]
E I, max(1-eval_I(p), eval_I(p & -p)) = 1 ssi [per def. de max]
E I, 1-eval_I(p) = 1 o eval_I(p & -p) = 1 ssi [per aritmètica]
E I, eval_I(p) = 0 o eval_I(p & -p) = 1 ssi [agafem la I tal que I(p) = 0]
cert

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

16. Siguin F i G dues fórmules qualsevols.
    Si F -> 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]
E I', I' |= p	ssi	[per def. de |=]
E I', eval_I'(p) = 1	ssi	[agafem la I' tal que I'(p) = 1]
cert		

G és insatisfactible (veure exercici 2).

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

21. 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 x 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**.

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

21. 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 x 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 tot e de S.
  R és **simètrica** si R(e,e') implica R(e',e) per tot e,e' de S.
  R és **transitiva** si R(e,e') i R(e',e'') implica R(e,e'') per tot e,e',e'' de S.

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

21. 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 x 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 una relació infixa:
  R és **reflexiva** si eRe per tot e de S.
  R és **simètrica** si eRe' implica e'Re per tot e,e' de S.
  R és **transitiva** si eRe' i e'Re'' implica eRe'' per tot e,e',e'' de S.

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

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

18. Demostra les següents equivalències entre fórmules:

$F \land F$	$\equiv$	F	idempotència de $\land$
$F \lor F$	$\equiv$	F	idempotència de $\lor$
$F \land G$	$\equiv$	$G \land F$	conmutativitat de $\land$
$F \lor G$	$\equiv$	$G \lor F$	conmutativitat de $\lor$
$(F \land G) \land H$	$\equiv$	$F \land (G \land H)$	associativitat de $\land$
$(F \lor G) \lor H$	$\equiv$	$F \lor (G \lor H)$	associativitat de $\lor$

Aquestes tres propietats (idempotència, commutativitat, asociatividad de & i de v) 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.

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

18. Demostra les següents equivalències entre fórmules:

$\neg\neg F$	$\equiv$	F	doble negació
$\neg(F \land G)$	$\equiv$	$\neg F \lor \neg G$	llei de De Morgan 1
$\neg(F \lor G)$	$\equiv$	$\neg F \land \neg G$	llei de De Morgan 2

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.

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

18. Demostra les següents equivalèncias entre fórmules:

$(F \land G) \lor H \;\; \equiv \;\; (F \lor H) \land (G \lor H)$ distributivitat 1
$(F \lor G) \land H \;\; \equiv \;\; (F \land H) \lor (G \land H)$ distributivitat 2

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 v sobre la subfórmula q v q?
No! Cal demostrar primer el següent *Lema de Substitució*:

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

23. Lema de Substitució.

    Siguin F, G, G' fórmules qualsevols, 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:

    F és $p \wedge (q \vee q)$

    G és $(q \vee q)$

    G' es q

    F' és $p \wedge q$.

Lògica en la Informàtica. Definició de Lògica Proposicional

Exercicis del capítol 2 dels apunts

Per al pròxim dia: Exercicis 26, 27, 28, 31, 32, 33, 34, 35, 36, 37.