(L) # Logique : Pr : Sylvain Salvati.

→ 6 CM puis TD/TP

$0,2\ C_{TD} + 0,4\ DS_1 + 0,4\ DS\ 2$

× Logique propositionnelle : syntaxe, sémantiq, pb de satisfaisabilité (SAT

× Logique prédicats : syn, sm

Antqté au XIX^e s :

- ∅ avoir vie bonne, se conformer lois ℒ, raisnmt
  syllogismes

→ Histqt ∅ maths used syllogistiq pr DMQ TH.
→ deps Aristote, très peu évoluo.

XIX^e : Mathématisao de la logiq

- Georges Boole, The laws of Thought : invent° lgq mq.
- Gottlob Frege, Begriffsschrift quantificaσ

→ On parle de proposi° & pas quantificaσ, gestion de la portée des variables.

Dbt XX : crise fdmt maths

- Paradoxe de Bertrand Russel en th g ens :
  $U = \{x \mid x \notin x\}$

- $U \notin U \Rightarrow$, par défi° de U que $U \in U$.
- $U \in U \Rightarrow U \notin U$

$\boxed{U \in U \Leftrightarrow U \notin U.}$

Menace cohérence mathmqs, dbt physiq, industrie, éco

- Pgm Hilbert. → pgm finitiste.
  → user ens fini de pas Dmq ∀ maths.

Kurt Gödel | DMq pgm voué à échec

- 1° th incomplétude (1931) : ∃ proposi° φ de l'arithq q n'est pas démontrable & dt négation n'est pas démont.
- 2° ——— ——— : tt système logique contenant l'arithmtq est soit contradictoire, soit il ne pt dmq sa propre cohérence.

Gm ne pt pas discerner entre pptés ≠ objt infinis.

G a dmq' en 1929 que tt th de arithmtq est démontrable.

TH de complétude

G numérise g elt. G : Gödel

↳ nbrs pvnt reprst énoncés logiqs.
↳ arithmtq pvnt opérats.
↳ on pt représenter la logiq ds-m de arithmtq & la réduire à du calcul sr les nbrs.

Alan Turing (1936) thèse de master, lis th G :

→ modèle de calcul q formalise : machines de Turing.
→ exiber machine q pvnt exécuter les autres : machine programmable.
→ exibe un pb q n'a pas de sol° algorithmq : le pb de l'arrêt des machines de Turing (en corollaire, la logiq est indécidable).

Début informatiq.

** Avènement de l'informatiq

→ 1ers ordinateurs : 2GM.
→ logiq ↔ créa° ordis & logiciels

Syntaxe	Sémantiq
φ programmes	modèles (graphes, entiers, effets

** Pb log ?

- Vérifica° de modèle : φ vraie ds M fixé ?
- Satisfaisabilité : ∃ ?
- Tautologie ∃-t M φ est vraie ?
- Démonstra° : st φ vrai ds ∀ M q'c t démontra°.

M modèle

** Applica°s

VDM

→ en programma°, recherche de données
→ optimisa° pgrm, élim° code mort S
→ pb résolu°s / optimisa° s/ contraintes
→ programma° par contraintes : pb graphes, jeux
→ pg° synchrone : généra° de code de machines à états finis ∈ spécificats logiqs

T

ampl pvnt solveurs
→ dm autmq th.
→ véri. optimisa° pgrms
→ certifica° pgrms (logiq de Hoare)
pb Σ' dmch.

D

→ Dmst° contin' algorithms.

$\forall a,b \in \mathbb{N}, \exists d, r \in \mathbb{N}, r < b \wedge a = d \times b + r$

→ on obtient un algorithme.

↳ $\boxed{M}$ dialectica de Gödel

↳ CoQ Curry Howard de Bruijn

↳ extraire DM ↔ Algorithme

article ⇒ on the usual effectiveness of Logic in computer science.

⟶ la complexité descriptive : formules pb ds certains logiques.

⟶ logiques épistémiques.

⟶ constr° processeurs

C Logique propositionnelle.

LP ⇔ articulation vérité

→ énoncés vrais → cste des vrais.

→ condit° de vérité d'un énoncé.

→ dmq énoncé est bien vrai.

I/ Syntaxe

D s: des termes ou formules.

Les atomes :

- $\top$ et $\bot$: tautologie & absurde, true/false
- var. propositionnelles.

FF composites.

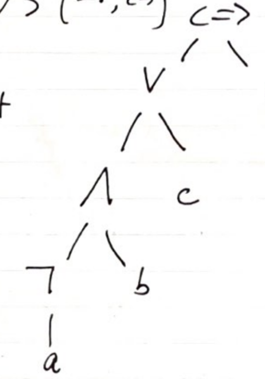
termes   Conjonction   Disjonction

Syntaxe concrète & Abstraite.

- Ordre de priorité : $\neg > \wedge > \vee > \{\Rightarrow, \Leftrightarrow\}$

$\neg a \wedge b \vee c \Leftrightarrow d$ dénotent

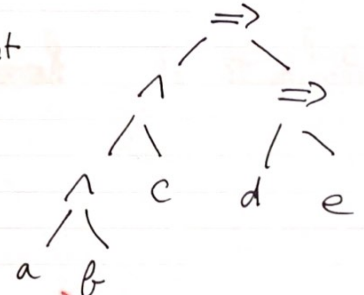
$((\neg a \wedge b) \vee c) \Leftrightarrow d$



- Associativité

$a \wedge b \wedge c \Rightarrow d \Rightarrow e$ dnt

$((a \wedge b) \wedge c) \Rightarrow (d \Rightarrow e)$



Définition inductive (récursives).

- Variables d'une formule $Var(\varphi)$

top $\top$
bottom $\bot$

$Var(\varphi) = \emptyset$

$Var(x) = \{x\}$

$Var(\neg\varphi) = Var(\varphi)$

$Var(\varphi \text{ op } \psi) = Var(\varphi) \cup Var(\psi)$

· Hauteur d'une formule $h(\varphi)$

· Substitu° $\not{f}$. $subs(\varphi, \sigma)$

$subst(\varphi, \sigma) = \varphi$ si $\varphi \in \{T, \perp\}$

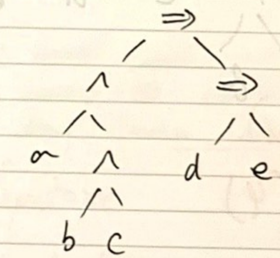
$subst(x, \sigma) = \sigma(x)$

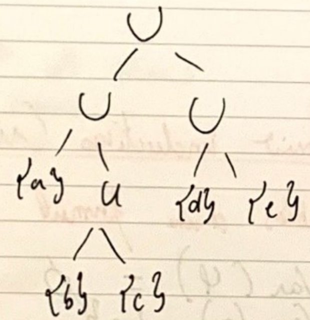
$subst(\neg\varphi, \sigma) = \neg\ subst(\varphi, \sigma)$

as définiθs st : homomorphismes (on se base sur S^{re} qq chose)

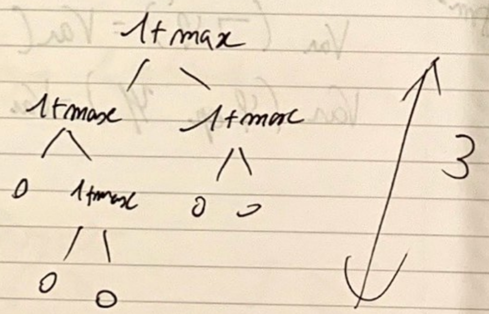
@ homomorphismes

vari





Hauteur



Le principe d'inducθ.

Inducθ structurelle.

ppté est vraie si $\forall \not{f}$!

: $P(\perp)$, $P(T)$, $P(x)$ $h=0$

· $P(\varphi)$ alors $P(\neg\varphi)$

· si $P(\varphi)$ et $P(\psi)$ alors $P(\varphi\ op\ \psi)$ pr $\forall$ op ds $\{\wedge, \vee, \Rightarrow, \Leftarrow\}$

Un cas particulier de récurrence.

$\Updownarrow$ inducθ.

Démq $\not{f}$ $\not{d}$ (IS).

(TH) Une $\not{f}$ φ ds $Prop(X)$ & tte subs σ si $\forall x$ dans X, ds $h(\sigma(x)) \leq N$ alors

$h(subs(\varphi, \sigma)) \leq h(\varphi) + N$

Dm Par inducθ sr S^{re} de φ :

· Cas $\varphi = \perp$ ou $\varphi = \perp$ ✓
si φ est $\perp$ ou T.

· Cas $\varphi = x$
si $\varphi = x$ alors $subs(\varphi, \sigma) = \sigma(x)$
et h

· Cas $\varphi = \neg\psi$
si $\varphi = \neg\psi$ alors $h(subs(\neg\psi, \sigma)) = h(\neg(subs(\psi, \sigma)))$
$= h(\psi, \sigma) + 1$
$= h(\psi) + 1 + N$
$= h(\neg\psi) + N$
$= h(\varphi) + N$

· Cas $\varphi = \psi_1\ op\ \psi_2$

Par inducθ, ppté est vraie pr tte $\not{f}$.

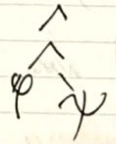
$h(\varphi, \psi) = 1 + max(~~h~~\ \varphi ⊕ \psi)$.

Syntaxe ⟶ homomorphismes ⇝ ppe induct°

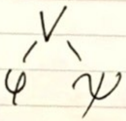
II) Sémantique

Stq intuitive connecteurs.

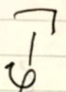
- $\top$
- $\bot$
- conjonc°



- disjonc°



- négat°



Variables? valuat°s?

0 → absurde, 1 → vérité.

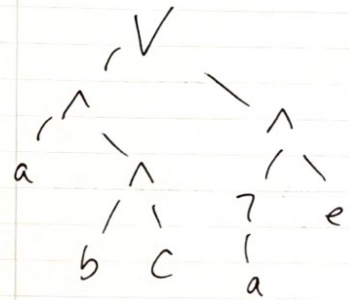
Stq formelle connect^{rs}

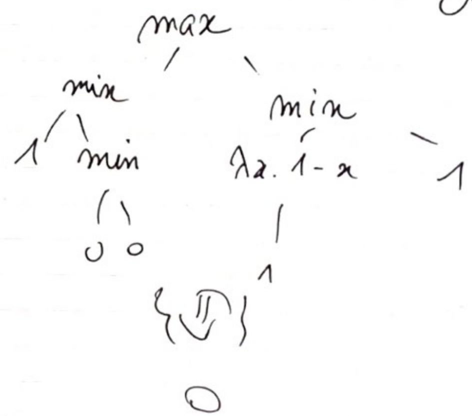
Évalu° $\mathcal{P}$: valuat° v,

$$[\![\top, v]\!] = 1$$
$$[\![\bot, v]\!] = 0$$
$$[\![x, v]\!] = v(x)$$
$$[\![\neg\varphi, v]\!] = 1 - [\![\varphi - v]\!]$$
$$[\![\varphi \wedge \psi, v]\!] = \min([\![\varphi, v]\!], [\![\psi, v]\!])$$
$$[\![\varphi \vee \psi, v]\!] = \max([\![\varphi, v]\!], [\![\psi, v]\!])$$

σ + valuat° st $a=1, b=0, c=0, d=1, e=1$

évaluat° de $[\![a \wedge b \wedge c \vee \neg a \wedge e, \sigma]\!] \rightsquigarrow \alpha$





$$\Lambda(x) = 1 - x$$

$\Leftrightarrow \lambda x.\ 1-x$

$$\alpha = \max([\![a \wedge b \wedge c, \sigma]\!], [\![\neg d \wedge e, \sigma]\!])$$

Stq ε table de vérité

a	b	¬a	a∨b	a∧b
0	0	1	0	0
0	1	1	1	0
1	0	0	1	0
1	1	0	1	1

connaître les valuat°s (conditions de vérité

min pour ∧, max pour ∨ et $\lambda x.\ 1-x$ pour ¬

Sig intuitive de $\Rightarrow$

la ff $\varphi \Rightarrow \psi$:

si φ (est vraie) alors ψ (est vraie.

Prop : Si le bord du carré est rouge alors son intérieur est gris.

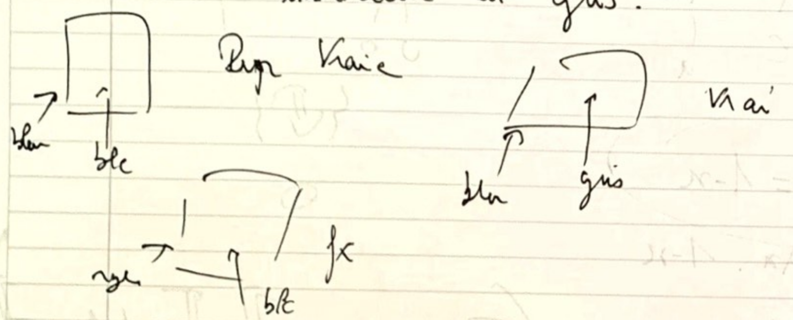


Table vér

φ	ψ	$\varphi \Rightarrow \psi$
0	0	1
0	1	1
1	0	0
1	1	1

TDV $\Leftrightarrow$

$\varphi \Leftrightarrow \psi = \varphi \Rightarrow \psi \wedge \psi \Rightarrow \varphi$.

φ	ψ	$\varphi \Rightarrow \psi$	$\psi \Rightarrow \varphi$	$\varphi \Leftrightarrow \psi$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	1	1	1

$\tilde{m}$ valeurs

a b . . . .

Construire TDV ff : $((a \Rightarrow b) \wedge c) \vee \neg a \wedge b$

2^3

a	b	c	$a \Rightarrow b$	$(a \Rightarrow b) \wedge c$	$\neg a$	$\neg a \wedge b$	$((a \Rightarrow b) \wedge c) \vee$ ... vr
0	0	0	1	0	1	0	0
0	0	1	1	1	1	0	1
0	1	0	1	~~1~~ 0	1	1	1
0	1	1	1	~~1~~ 1	1	1	1
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	1	0	0	0	.
1	1	1	1	1	0	0	1

Nbr lignes : 2^n qd n : nbr variables.

Satisfaisabilité & Tautologie

$\boxed{S}$ Une ff φ est satisfiable s'il $\exists$ une valuation v tq $[\![\varphi, v]\!] = 1$.

On dit que v satisfait φ.

Tautologie

Une ff φ est t. si $\forall$ tte valuaϑ v, $[\![\varphi, v]\!] = 1$

On note $\models \varphi$ le pt φ soit tautologie

Éqvla & cong s'mtq

équivalentes sémantiqs

$\varphi \equiv \psi$ lsq tte valuaϑ v, $[\![\varphi, v]\!]$, $[\![\psi, v]\!]$.

tq'
= syntaxiq en arbre
= sémtq en compt

on pt internaliser ds logiq l'éqvla sémtq.

$\varphi \equiv \psi$ ssi $\models \varphi \Leftrightarrow \psi$.

$\varphi_1, \dots \varphi_n \models \psi$ ssi $\models \varphi_1 \Rightarrow$ ___

Congruence

$\equiv$.

Reflexivité $\varphi \equiv \varphi$

symétrie $\varphi \equiv \psi$ impliq $\psi \equiv \varphi$

Transitive $\varphi \equiv \psi$ et $\psi \equiv \theta$ impliq $\varphi \equiv \theta$.

- elle commute ap les connecteurs.

Ptés Algébriqs

Commut.

Associat

Distribt

Identité $\varphi \vee \bot \equiv \varphi$, $\psi \wedge \top \equiv \psi$

Zéro $\varphi \vee \top \equiv \top$, $\varphi \wedge \bot \equiv \bot$

Itempotence $\varphi \vee \varphi \equiv \varphi$, $\varphi \wedge \varphi \equiv \varphi$

Abϑ $\varphi \vee (\varphi \wedge \psi) \equiv \varphi$, $\varphi \wedge (\varphi \wedge \psi) \equiv \varphi$

Complémentaϑ $\varphi \vee \neg \varphi \equiv \top$, $\varphi \wedge \neg\varphi \equiv \bot$

Double négaϑ $\neg\neg \varphi \equiv \varphi$

de Morgan $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$,

~~Défini~~ $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$.

Définiϑ $\varphi \Rightarrow \psi \equiv \neg\varphi \vee \psi$,

$\varphi \Leftrightarrow \psi \equiv \varphi \Rightarrow \psi \wedge \psi \Rightarrow \varphi$.

Complétude équationnelle.

Axiomatisaϑ de compétence.

si $\varphi \equiv \psi$, on pt réécrire

ap équaϑ données ds axiomatisaϑ

** Système complet de connecteurs

(SC) si pr tte φ, $\exists$ ψ. tq

- ψ n'est construite de connect de φ
- $\varphi \equiv \psi$.

@ SC $\{\vee, \neg\}$, $\{\wedge, \neg\}$, $\{\Rightarrow, \neg\}$, $\{\Rightarrow, \bot\}$

@ $\{\neg, \vee\}$.

transformation $\overline{\cdot}$ (homomorphisme) $h(\ )$

- $\overline{\bot} = \neg(x \vee \neg x)$
- $\overline{\top} = x \vee \neg x$
- $\overline{x} = x$
- $\overline{\varphi \vee \psi} = \overline{\varphi} \vee \overline{\psi}$
- $\overline{\varphi \wedge \psi} = \neg(\neg\overline{\varphi} \vee \neg\overline{\psi})$
- $\overline{\varphi \Rightarrow \psi} = \neg\overline{\varphi} \vee \overline{\psi}$
- $\overline{\varphi \Leftrightarrow \psi} = \neg(\neg(\neg\overline{\varphi} \vee \overline{\psi}) \vee \neg(\neg\overline{\psi} \vee \overline{\varphi}))$

Donc $\{\neg, \vee\}$ forme un système complets de connecteurs.

Démonstration
Par induction

- $\overline{\varphi}$ ne contient q $\neg$ et $\vee$.
- $\varphi \equiv \overline{\varphi}$.