Introduction à la logique

Qu’est ce que la logique ?

- On peut dire que la logique est une formalisation des lois qui font si un raisonnement est correct, au-delà de son contenu.

Exemple:

s’il neige, alors il fait froid.

Il neige,

donc il fait froid

- De la forme « si X alors Y »

Définitions de la logique

Définition 1

- La **logique** (du grec logikê, terme signifiant à la fois raison, langage, et raisonnement) est l'étude des règles formelles que doit respecter toute argumentation correcte.

Définition 2

- La **logique** peut se définir comme l'étude des «**raisonnements** » :

– Raisonnements « formels » : démonstrations mathématiques, preuves.

– Raisonnements « informels » : étude des discours en langues naturelles, utilisation des connaissances.

Définitions de la logique

- **Définition 3**

La logique est la science des arguments valides en vertu de leur forme uniquement.

Le but de la logique est de distinguer les arguments valides de part leur forme des autres. Pour cela, les logiciens ont construit des langages particuliers, des langages formels. Ces langages font complètement abstraction du contenu des expressions, pour ne retenir que leur structure logique.

Logique en informatique

La logique apparaît dans différents aspects de l'informatique :

- l'*analyse formelle de programmes* : formalisation des spécifications, preuves de programmes, optimisations…
- Aider les mathématiques dans l'établissement de preuves: assistants de preuves, démonstrateurs automatiques.

La logique est une base de **l'intelligence artificielle**:

- Représentation, acquisition et utilisation de connaissances (*systèmes experts ou à base de connaissances*) ;
- Résolution de problèmes (*robots, jeux*) ;
- Compréhension et traitement des langues naturelles (*linguistique informatique*).

Notions élémentaires de logique formelle

Une logique est constituée des éléments suivants:

- Langage : ce qui permet de définir les formules.
- Syntaxe ou système formel= système de calcul purement syntaxique sur les formules formé d'Axiomes et de règles d'inférence
- Sémantique : ce qui donne un sens aux formules

Définitions

- Une syntaxe: système de symboles et de règles pour les combiner sous formes de formules.

- Une sémantique : permet d'interpréter, c'est-à-dire d'attacher aux formules ainsi qu'aux symboles une signification.

- Un système de déduction permet de raisonner en construisant des démonstrations.

La logique classique

La logique comprend classiquement :

- la logique des propositions (aussi appelée calcul des propositions),
- la logique des prédicats.

Ces deux derniers points feront l'objet de notre cours.

Le calcul propositionnel

Qu’est ce qu’une proposition?

- Les propositions (contrairement aux paradoxes) sont des affirmations qui ne peuvent être que vraies ou fausses.

Exemples:

- "le ciel est bleu",
- "les coquelicots sont jaunes",
- "2+2=4",
- "2+2=5"

Logique propositionnelle

- On appelle **logique propositionnelle** la partie de la logique qui traite des *propositions*.

- L'un des buts de la logique propositionnelle est d'élaborer un calcul, que nous nommerons: *calcul propositionnel*.
- Cela entraîne que les propositions soient traitées comme des *variables*, désignées par des *lettres* (p, q, r, ...) et que l'on introduise des *opérations* permettant de combiner les *valeurs* de ces variables.

Intérêt de la logique propositionnelle

La logique propositionnelle permet:

- La représentation des propositions d'une manière formelle (le langage du calcul propositionnel)
- Déduction de nouvelles propositions à partir d'autres propositions (système déductif)
- Vérification de la validité des propositions d'une manière formelle (tables de vérité)

Les paradoxes

- Le **paradoxe** est une affirmation qui contient une contradiction logique (vraie et fausse en même temps), ou un raisonnement qui, bien que sans faille apparente, aboutit à une absurdité, ou encore une situation qui contredit l'intuition commune.

Paradoxe de l'autoréférence

- Considérons la phrase suivante : "cette phrase est fausse". Est-elle vraie ou fausse?

- Si cette phrase est vraie, alors elle est fausse. Et si elle est fausse de part son contenu, alors elle est vraie.

Paradoxe de Zénon d'Elée: La course à pied

- Pour atteindre la ligne d'arrivée, un coureur doit d'abord parcourir la moitié de la distance qui le sépare de cette ligne, puis la moitié de la distance restante, et ainsi de suite à l'infini. Le coureur ne terminera donc jamais la course.

- Ce paradoxe dit de "la dichotomie" (c'est à dire division par deux) veut montrer l'impossibilité du mouvement:si on fait la somme (infinie) de toutes les distances du problème on trouve 1: 1/2+1/4+1/8+... La limite de cette suite vaut 1. Mais une limite est ce vers quoi on tend sans jamais l'atteindre...

Le tiers exclu

- Toute proposition est soit vraie soit fausse.

Calcul propositionnel

Etude syntaxique

- Définir un langage revient à définir:

– Des symboles (alphabet)

– Des expressions (formules)

  - Soit $\mathcal{L}_p(\neg,\wedge)$ un langage propositionnel

L'alphabet

- L'alphabet de $\mathcal{L}_p(\neg,\wedge)$ se compose de trois classes de symboles:
- Les symboles des variables propositionnelles : lettres majuscules avec ou sans indices (A, B, C, A_1, A_2,…)
- Les symboles logiques ou connecteurs $\neg,\wedge$
- Les symboles impropres '**(**' et '**)**'

Les formules

- L'ensemble des formules de $\mathcal{L}_p(\neg,\wedge)$ est défini inductivement de la manière suivante:

– Toute variable propositionnelle est une formule atomique (simple)

– Si α et β sont deux formules alors $\neg\alpha$, $\alpha\wedge\beta$ sont des formules composées.

- Les lettres grecques sont utilisées pour représenter les noms des formules.

- **Remarque:**
  - Le connecteur $\wedge$ est binaire et le connecteur $\neg$ est unaire.

Autres connecteurs

- $\alpha \vee \beta =_{def} \neg(\neg\alpha \wedge \neg\beta)$
- $\alpha \rightarrow \beta =_{def} \neg(\alpha \wedge \neg\beta)$
- $\alpha \leftrightarrow \beta =_{def} (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$
- $=_{def} \neg(\alpha \wedge \neg\beta) \wedge \neg(\beta \wedge \neg\alpha)$

Priorités

(	)	La plus à gauche, puis la plus interne
¬		Le plus interne d'abord
∧	∨	Le plus à gauche d'abord
→	↔	Le plus à gauche d'abord

Exemples:

- $\neg(\neg(\neg A)))$ s'écrira $\neg\neg\neg A$
- $((\alpha\wedge\beta)\vee \gamma)$ s'écrira $\alpha\wedge\beta\vee \gamma$
- $(\alpha \rightarrow \beta) \leftrightarrow \gamma$ s'écrira $\alpha \rightarrow \beta \leftrightarrow \gamma$
- $(\alpha \wedge \beta) \leftrightarrow \gamma$ s'écrira $\alpha \wedge \beta \leftrightarrow \gamma$
- $(\alpha \vee (\beta \leftrightarrow \gamma))$ s'écrira $\alpha \vee (\beta \leftrightarrow \gamma)$

Formule bien formée

On définit une formule bien formée (fbf) par :

- P (variable propositionnelle)
- $(A \vee B)$
- $(A \wedge B)$
- $(\neg A)$
- $(A \rightarrow \mathrm{B})$: si A alors B
- $(A \leftrightarrow \mathrm{B})$

Où A et B sont des variables propositionnelles ou des formules bien formées.

Arbre syntaxique

- On peut dessiner pour toute formule A de la logique propositionnelle un arbre de construction = un arbre de formation = un arbre syntaxique:
  - La racine de l'arbre correspond à la formule A.
  - Les nœuds de cet arbre sont des formules.
  - Les feuilles de l'arbre correspondront aux atomes propositionnels.
  - Tout nœud (à l'exception des feuilles) a un ou deux successeurs.

Exemple:

$$
\begin{array}{ccc}
 & (\neg p \rightarrow q) & \\
 & / \qquad\qquad \backslash & \\
\neg p & & q \\
| & & \\
p & &
\end{array}
$$

FBF et arbre syntaxique

- Pour toute formule — toute expression bien formée — il existe un et en seul arbre syntaxique : la manière de formation d'une formule est uniquement déterminée par sa syntaxe.
- Tout essai de dessiner un arbre syntaxique pour une non-formule va échouer.

Le système déductif

Deux règles pour chaque connecteur:

- Une règle d'introduction
- Une règle d'élimination

Règles du $\wedge$

Règle d'introduction

$$\frac{\alpha \qquad \beta}{\alpha \wedge \beta} \; I\wedge$$

Règles du $\wedge$

Règle d'élimination

$$\frac{\alpha \wedge \beta}{\alpha}\ E\wedge \qquad \frac{\alpha \wedge \beta}{\beta}\ E\wedge$$

Règles du ¬

Règle d'introduction

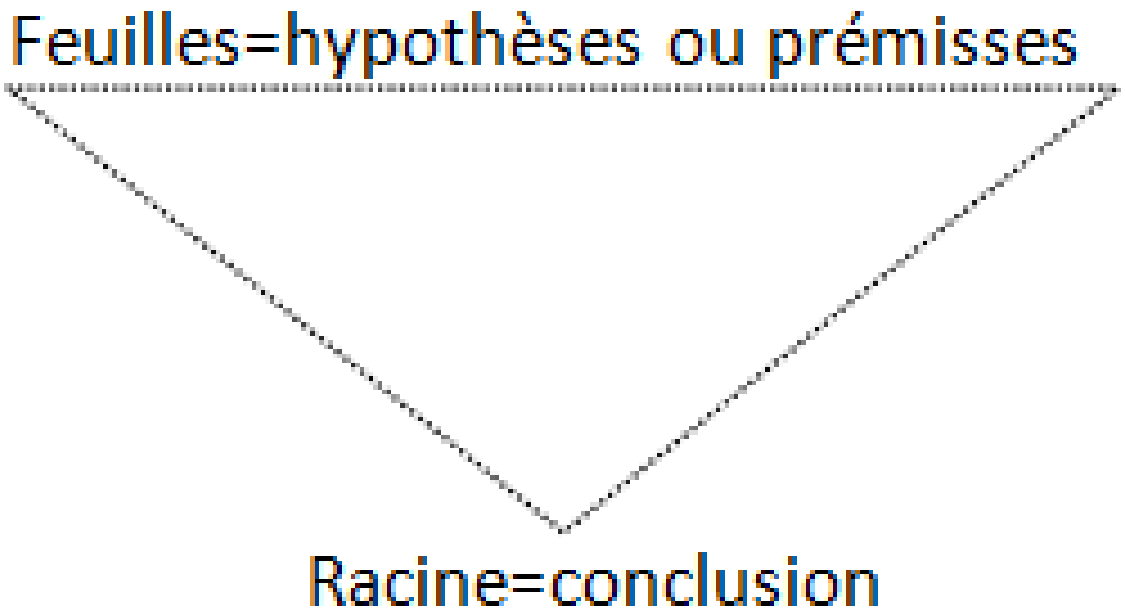
$$\frac{\begin{array}{c}\alpha \\ \vdots \\ \bot\end{array}}{\neg\alpha}\ \neg\mathrm{I}$$

Règles du ¬

Règle d'élimination

$$\frac{\neg\ \neg\alpha}{\alpha}\ E\neg$$

Déduction



Notations

- Γ est un ensemble de formules et β une formule
- $\Gamma \vdash \beta$: il existe une déduction β dont les prémisses non éliminées sont dans l'ensemble Γ.
- Cas particulier: $\Gamma = \emptyset$, $\vdash \beta$ est appelé un théorème du calcul propositionnel.
- $\{\alpha_1, \alpha_2, \ldots, \alpha_n\} \vdash \beta$ est équivalent à $\alpha_1, \alpha_2, \ldots, \alpha_n \vdash \beta$
- $\Gamma \cup \alpha \vdash \beta$ est équivalent à $\Gamma, \alpha \vdash \beta$

Etude sémantique

Calcul propositionnel

Étude sémantique

- la sémantique renvoie à l'étude de la façon dont les formules dénotent des valeurs de vérité.

- Dans la logique classique que nous étudions ici, il n'y a que deux valeurs de vérité, le vrai et le faux.

Table de vérité

- C'est un énorme avantage de n'avoir que deux valeurs. En effet, si n propositions p1, p2, ..., pn entrent dans la définition d'un même "*monde possible*", celui-ci est complètement caractérisé par une situation des valeurs de vérité de ces propositions parmi 2^n possibles.
- Ainsi si nous avons 3 propositions: p, q, r, elles déterminent un ensemble de 2×2×2 situations a priori possibles, qui seront modélisées sous forme d'une table appelée **Table de vérité**.

- La table de vérité d'une formule α contenant n variables propositionnelles $A_1,\ldots, A_n$ est composée de 2^n+1 lignes et (n+m) colonnes: n colonnes pour les n variables propositionnelles et m colonnes pour les m sous-formules de α.
- Chaque ligne est appelée **une instanciation**.
- On définit une fonction de valuation $\mathcal{V}$: $\{A_1,\ldots,A_n\} \rightarrow \{ V,F \}$
- On dit que α est satisfiable s'il existe une valuation pour laquelle α est vraie.
- Nous associons à chaque connecteur du langage $\mathcal{L}_p(\neg,\wedge)$ une table de vérité.

Table du connecteur ¬

A	¬A
V	F
F	V

Table des connecteurs $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

A	B	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
V	V	V	V	V	V
V	F	F	V	F	F
F	V	F	V	V	F
F	F	F	F	V	V

Remarque:

- Si α et β sont des formules alors:
- $\alpha \wedge \beta = V \Leftrightarrow \alpha = V$ et $\beta = V$
- $\alpha \vee \beta = F \Leftrightarrow \alpha = F$ et $\beta = F$
- $\alpha \rightarrow \beta = F \Leftrightarrow \alpha = V$ et $\beta = F$
- $\alpha \rightarrow \beta = V \Leftrightarrow \alpha = F$ ou $\beta = V$
- $\alpha \leftrightarrow \beta = V \Leftrightarrow (\alpha = V$ et $\beta = V)$ ou $(\alpha = F$ et $\beta = F)$: α et β ont la même valeur de vérité

Tautologie

Une formule α est appelée **Tautologie** (et on note $\models \alpha$) si quelle que soit la valuation V on a V(α)=vrai. En d'autres termes, dans la table de vérité il n'y a que la valeur **vrai** comme résultat.

Tautologie, antilogie

- **Remarque:**
- Si α et β sont deux formules alors:
- $\models \alpha \Leftrightarrow$ toutes les valuations satisfont α
- $\models \neg\alpha \Leftrightarrow$ aucune valuation ne satisfait α, α est une **antilogie.**
- $\not\models \alpha \Leftrightarrow \alpha$ n'est pas une tautologie (il existe au moins une valuation qui ne satisfait pas α.
- $\not\models \alpha$ (non tautologie) $\not\Rightarrow$ antilogie
- $\models \alpha\wedge\beta \Leftrightarrow \models \alpha$ et $\models \beta$
- ($\models \alpha\rightarrow \beta$ et $\models \alpha$) $\Rightarrow \Vdash \beta$
- $\models \alpha\rightarrow \beta \Leftrightarrow$ (pour chaque instanciation si α=v alors β=v)
- $\models \alpha\leftrightarrow \beta \Leftrightarrow$ (pour chaque instanciation α et β ont les mêmes valeurs de vérité)

Exemple:

- 1-Donner la nature (tautologie, antilogie, autre) de:

- $A \vee \neg A$
- $A \wedge \neg A$

Equivalence logique

Soient deux formules logiques α et β. On dira que α est logiquement équivalente à β, et on note

$\alpha \equiv \beta$, si α et β ont les mêmes valeurs de vérité pour les mêmes instanciations.

- **Remarque:**
- $\equiv$ définit une relation d'équivalence sur l'ensemble des formules de $\mathcal{L}_p(\neg,\wedge)$
- $\alpha \equiv \beta$ ssi $\Vdash \alpha \leftrightarrow \beta$

Corollaire

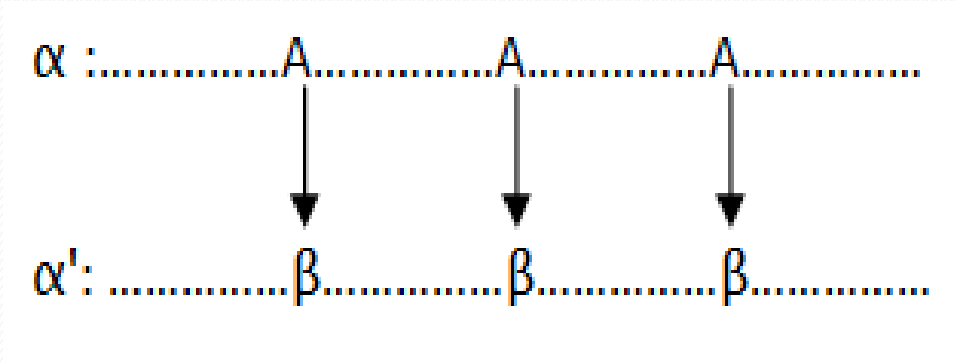
- Toute formule est logiquement équivalente à une formule qui ne contient que les connecteurs $(\neg,\wedge)$. On dira alors que $\{\neg,\wedge\}$est un système complet de connecteurs SCC.

- Un ensemble Ω de connecteurs est un SCC si toutes les formules de la forme $\neg\alpha$, $\alpha\wedge\beta$ peuvent s'exprimer à l'aide des connecteurs de Ω.

Théorème de substitution

- Soient α une formule contenant la variable propositionnelle A, et α' la formule obtenue en substituant toutes les occurrences de A par la formule β.

Alors : $\models \alpha \Rightarrow \models \alpha'$



A:variable propositionnelle

β:Formule

Exemple

α: $A \wedge B \rightarrow A \vee \neg B$,

α': $A \wedge \neg B \rightarrow A \vee \neg\neg B$

- Montrer que $\models \alpha$
- Montrer que $\models \alpha'$ en utilisant le théorème de substitution

Théorème de remplacement

Soient :

α une formule contenant une sous-formule β

et

α' la formule obtenue en remplaçant β par β'.

Alors:

$$\beta \equiv \beta' \quad \Rightarrow \quad \alpha \equiv \alpha'$$

Comment montrer que $\alpha \equiv \beta$?

- Table de vérité: $\alpha \equiv \beta$ ssi elles ont les mêmes valeurs de vérité
- $\alpha \equiv \beta$ ssi $\vDash \alpha \leftrightarrow \beta$
- $\alpha \equiv \beta$ ssi $(\alpha = v \Leftrightarrow \beta = v)$
- $\alpha \equiv \alpha_1 \equiv \alpha_2 \equiv \ldots\ldots\ldots\ldots \equiv \alpha_n \equiv \beta$ (Théorème de remplacement n fois)

- Aucun ensemble de connecteurs monaires ne peut former un SCC
- Il existe des SCC à deux connecteurs qui sont des SCC:

  $\{\neg, \vee\}, \{\neg, \rightarrow\}$
- Il existe des SCC à trois connecteurs ou plus.
- Il existe des SCC dont l'un des connecteurs est d'arité supérieure à 2.
- Il existe exactement deux SCC à un seul connecteur binaire. Ces connecteurs sont appelés barres de Shaffer. Notons ce connecteur "|".

Pour exprimer le ¬:

A	¬A	A|A
v	f	f
f	v	v

Pour exprimer le $\wedge$:

A	B	A|B	A|B	A|B	A|B
v	v	f	f	f	f
v	f	v	v	f	f
f	v	v	f	v	f
f	f	v	v	v	v

$\neg(A \wedge B)$	$\neg B$	$\neg A$	$\neg(A \vee B)$

Nous obtenons deux nouveaux connecteurs:

- $A \uparrow B \equiv \neg(A \wedge B)$ (Nand)
- $A \downarrow B \equiv \neg(A \vee B)$ (Nor)

1. $\{\uparrow\}$ est un SCC:

- $\neg\alpha \equiv \alpha \uparrow \alpha$
- $\alpha \wedge \beta \equiv \neg(\alpha \uparrow \beta) \equiv (\alpha \uparrow \beta) \uparrow (\alpha \uparrow \beta)$

2. $\{\downarrow\}$ est un SCC:

- $\neg\alpha \equiv \alpha \downarrow \alpha$
- $\alpha \vee \beta \equiv \neg(\alpha \downarrow \beta) \equiv (\alpha \downarrow \beta) \downarrow (\alpha \downarrow \beta)$

Exercice:

- Montrer:
- $\alpha \wedge \beta \equiv (\alpha \downarrow \alpha) \downarrow (\beta \downarrow \beta)$
- $\alpha \vee \beta \equiv (\alpha \uparrow \alpha) \uparrow (\beta \uparrow \beta)$

Les formes normales

Littéraux, conjonction de deux littéraux

- Un littéral = une variable ou une négation de variable

  $p \quad , \quad \neg p$

- Etant données deux variables propositionnelles A et B on peut construire les conjonctions de littéraux suivantes:

<u>Conjonctions de deux littéraux</u>

- $\varphi_1 = A \wedge B$
- $\varphi_2 = \neg A \wedge B$
- $\varphi_3 = A \wedge \neg B$
- $\varphi_4 = \neg A \wedge \neg B$

Disjonction de deux littéraux

Etant données deux variables propositionnelles A et B on peut construire les disjonction de littéraux:

<u>Disjonctions de littéraux</u>

- $\Psi_1 = A \vee B$
- $\Psi_2 = \neg A \vee B$
- $\Psi_3 = A \vee \neg B$
- $\Psi_4 = \neg A \vee \neg B$

Les formes normales

Considérons le littéral $\tilde{A}_i = Ai$ ou $\neg Ai$

FNC Forme Normale Conjonctive

- $\varphi = \tilde{A}_1 \wedge \tilde{A}_2 \wedge \ldots\ldots\ldots \wedge \tilde{A}_n,$

- φ=Vrai ssi $\tilde{A}_i = \begin{cases} \text{vrai si } \tilde{A}i = Ai \\ \text{faux si } \tilde{A}i = \neg Ai \end{cases}$

FND Forme Normale Disjonctive

- $\Psi = \tilde{A}_1 \vee \tilde{A}_2 \vee \ldots\ldots\ldots \vee \tilde{A}_n$,

- Ψ=Faux ssi $\tilde{A}_i = \begin{cases} \text{faux si } \tilde{A}i = Ai \\ \text{vrai si } \tilde{A}i = \neg Ai \end{cases}$

Définitions:

- On appelle forme normale disjonctive (FND) toute disjonction de conjonction de littéraux.
- On appelle forme normale conjonctive (FNC) toute conjonction de disjonction de littéraux.

Théorème:

- Toute formule α est logiquement équivalente à une FND (respectivement à une FNC)

Comment construire une FND?

Soit α une formule contenant N variables $A_1,\ldots,A_n$.

- Etablir la table de vérité de α
- Déterminer l'ensemble des lignes $I=\{I_1,\ldots,I_m\}$ de la table de vérité telles que α =vrai

Pour chaque ligne $i \in I$, on détermine $\varphi_i = \tilde{A}_1 \wedge \tilde{A}_2 \wedge \ldots \wedge \tilde{A}_n$

où $\tilde{A}_i = \begin{cases} A_i & \text{si } A_i \text{ vrai,} \\ \neg A_i & \text{si } A_i \text{ faux} \end{cases}$

- $\varphi = \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_m$
- Si α est une antilogie alors $\varphi = A_i \wedge \neg A_i$

Exemple FND

A	B	$A \uparrow B$	φ_i
v	v	f	
v	f	v	$\varphi_1 = A \wedge \neg B$
f	v	v	$\varphi_2 = \neg A \wedge B$
f	f	v	$\varphi_3 = \neg A \wedge \neg B$

$$\varphi = (A \wedge \neg B) \vee (\neg A \wedge B) \vee (\neg A \wedge \neg B)$$

Comment construire une FNC?

On adopte le même raisonnement en considérant les lignes où α=Faux:

- $\Psi_i = \tilde{A}_1 \vee \tilde{A}_2 \vee \ldots \vee \tilde{A}_n$

où $\tilde{A}_i = \begin{cases} \neg A_i & \text{si } A_i \text{ vrai,} \\ A_i & \text{si } A_i \text{ faux} \end{cases}$

- $\Psi = \Psi_1 \wedge \Psi_2 \wedge \ldots \wedge \Psi_m$
- Si α est une tautologie alors $\Psi = A_i \vee \neg A_i$

Exemple FNC

A	B	$A \downarrow B$	Ψ_i
v	v	f	$\Psi_1 = \neg A \vee \neg B$
v	f	f	$\Psi_2 = \neg A \vee B$
f	v	f	$\Psi_3 = A \vee \neg B$
f	f	v	

$$\Psi = (\neg A \vee \neg B) \wedge (\neg A \vee B) \wedge (A \vee \neg B)$$

Relation entre la syntaxique et la sémantique

Consistance et complétude

Théorème de consistance

- Soient $\alpha_1, \alpha_2, \ldots, \alpha_n$ et α des formules de $\mathcal{L}_p(\neg, \wedge)$. Alors:

$$\alpha_1, \alpha_2, \ldots, \alpha_n \vdash \alpha \Rightarrow \models (\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \rightarrow \alpha$$

- Cas particulier: $\vdash \alpha \Rightarrow \models \alpha$

Conséquence

- Les quatre règles du calcul propositionnel du langage $L_p(\neg, \wedge)$ ne peuvent pas produire une inconsistance ou bien sont consistantes.

Théorème de complétude

- Soient $\alpha_1, \alpha_2, \ldots, \alpha_n$ et α des formules de $\mathcal{L}_p(\neg, \wedge)$. Alors:

$$\models (\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \rightarrow \alpha \Rightarrow \alpha_1, \alpha_2, \ldots, \alpha_n \vdash \alpha$$

- Cas particulier: $\models \alpha \Rightarrow \vdash \alpha$

Définitions

- Un ensemble est dit inconsistant s'il permet de produire une inconsistance.
- Un ensemble est dit satisfaisable (ou satisfiable) s'il existe une instanciation qui vérifie toutes les formules (i.e. les formules peuvent être vraies en même temps).
- Un ensemble est dit inconsistant s'il permet de produire une contradiction $\perp$.

- **Lemme 1**

$\alpha_1, \alpha_2, \ldots, \alpha_n \vdash \alpha \Leftrightarrow \{ \alpha_1, \alpha_2, \ldots, \alpha_n, \neg\alpha\}$ est inconsistant.

- **Lemme 2**

Tout ensemble de formules consistant est un ensemble satisfiable.

- **Contraposée**

$\{ \alpha_1, \alpha_2, \ldots, \alpha_n, \neg\alpha\}$ non satisfiable $\Rightarrow$ inconsistant

- **Résultat**

Un ensemble E sera dit inconsistant s'il existe un couple de littéraux de la forme $(A, \neg A)$, sinon E sera dit satisfiable.

Algorithme de réfutation

- Soit Γ un ensemble de formules et soient les transformations suivantes (clauses):

$\{\neg\neg\alpha\} \longrightarrow \{\alpha\}$

$\{ \alpha \wedge \beta \} \longrightarrow \{ \alpha, \beta \}$

$\{ \neg(\alpha \wedge \beta) \} \longrightarrow \{\neg\alpha\}$

$\{ \neg(\alpha \wedge \beta) \} \longrightarrow \{\neg\beta\}$

- On appelle ces transformations les clauses associées aux connecteurs $\neg, \wedge$.
- On applique les clauses sur l'ensemble Γ jusqu'à obtenir uniquement des ensembles de littéraux (éliminer tous les connecteurs).
- L'application des clauses permet de former un arbre dont la racine est l'ensemble Γ et les feuilles sont des ensembles de littéraux.

- La méthode de construction de l'arbre aboutit à une situation d'arrêt, puisque le nombre de connecteurs diminue à chaque étape jusqu'à l'obtention des littéraux.
- A la fin de l'algorithme, deux cas peuvent se présenter:
  - Si tous les ensembles de littéraux sont inconsistants alors Γ est inconsistant.
  - S'il existe au moins un ensemble de littéraux satisfaisable alors Γ est satisfaisable.

Conclusion

$$\alpha_1, \alpha_2, \ldots, \alpha_n \vdash \alpha$$

$$\Leftrightarrow$$

$$\models (\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n) \rightarrow \alpha$$

$$\Leftrightarrow$$

$\{ \alpha_1, \alpha_2, \ldots, \alpha_n, \neg\alpha \}$ est inconsistant

Le calcul propositionnel est décidable

Le calcul des prédicats

Tout homme est mortel

Or Socrate est un homme

Donc Socrate est mortel

motivations

- Nécessité de dépasser la logique propositionnelle :
  - Exprimer des énoncés vérifiés pour un ensemble de variables
  - introduire des objets
  - introduire les relations vérifiées par les objets
  - introduire la quantification

Apports

- introduction des Prédicats dont la validité dépend des variables et des constantes qu'on leur applique
  - Ex: Homme(x), Mortel(x)

- Exprimer qu'un ensemble de variables vérifie un prédicat
  - Ex: $\forall x$ Mortel(x)

Éléments du langage

- Symboles:
  - Variables:x,y,z,x_1,….
  - Constantes:a,b,c,a_1,….
  - Fonctions:f,g,f_1,…Arité de la fonction=nombre d'arguments
  - Prédicats: P,Q,R,P_1,… Arité du prédicat=nombre d'arguments
  - Symbole d'égalité '=':prédicat prédéfini d'arité 2
  - Symboles logiques $\neg$, $\wedge$, $\forall$ (quantifieur universel)
  - symboles impropres '(' ')' ','

Éléments du langage

- Termes:
  - Toute variable est un terme
  - Toute constante est un terme
  - Si f est une fonction d'arité n et $t_1,\ldots t_n$ des termes alors $f(t_1,\ldots, t_n)$ est un terme

Éléments du langage

- Formules:
  - Si P est une fonction d'arité n et $t_1,\ldots t_n$ des termes alors $P(t_1,\ldots, t_n)$ est une formule atomique
  - Toute formule atomique est une formule
  - Si $\mathcal{A}$ et $\mathcal{B}$ sont des formules, alors $\neg\mathcal{A}$, $\mathcal{A}\wedge\mathcal{B}$, $\forall x\,\mathcal{A}$ sont des formules

Autres symboles et priorités

- Le quantifieur existentiel $\exists$:

$$\exists x\, \mathscr{A}(x) \quad =_{def} \quad \neg\forall x\, \neg\mathscr{A}(x)$$

- Connecteurs $\vee$, $\rightarrow$, $\leftrightarrow$

Priorités

$$\begin{array}{c} (\;) \\ \neg \\ \vee \quad \wedge \\ \forall \quad \exists \\ \rightarrow \quad \leftrightarrow \end{array}$$

Variable libre, liée

$\forall x \quad \boxed{\mathscr{A}(x)} \Rightarrow$ champs de x

- Toute occurrence d'une variable dans son champs est dite « liée», elle est dite « libre » sinon
- Une variable est dite « libre » si elle a au moins une occurrence libre
- Une variable est dite « liée » si elle a au moins une occurrence liée
- Une variable peut être à la fois libre et liée

Occurrence libre

Occurrence liée

Occurrence liée

Occurrence libre

Champs de ∃y

Occurrence libre

$$\forall x\,((P(x) \vee Q(x, y)) \rightarrow (\exists y(R(y) \wedge Q(x, y))) \wedge S(u,y, x)$$

Champs de ∀x

Occurrence liée

Occurrence liée

Occurrence liée

Occurrence libre

Formule fermée

- Une formule α est dite « fermée » si toutes les occurrences des variables de la formule sont liées

- Si α est une formule contenant les variables $x_1,\ldots,x_n$, on appelle « fermeture de α» la formule:

$$\forall x_1,\ldots, \forall x_n\ \alpha$$

$$(P(x) \vee Q(x, y)) \rightarrow (R(y) \wedge Q(x, y) \wedge S(u,y, x))$$

Fermeture de cette formule?

$$\forall x\ \forall y\ \forall u\ ((P(x) \vee Q(x, y)) \rightarrow (R(y) \wedge Q(x, y) \wedge S(u,y, x)))$$

Terme libre

Soient α(x) une formule, t un terme et x une variable

On dira que le terme t est libre pour x (ou libre d'être substitué pour x) dans α si aucune variable de t ne devient liée après substitution dans α(t)

- Tout terme ne contenant pas de variables est libre pour toute variable dans toute formule
- x est libre pour x dans toute formule

α: $\forall x\ ((P(x) \vee Q(x, y))$

Est-ce que $t=f(x,y)$ est libre pour y dans α ?

$\forall x\ ((P(x) \vee Q(x, y))$

Occ libre de y:substituer par t

$\forall x\ ((P(x) \vee Q(x, f(x,y)\ ))$

Cette occurence de x devient liée

f(x,y) non libre pour y dans α

Étude syntaxique: le système déductif

- Axiome: $\forall x \quad x=x$
- Règles de $(\neg,\wedge)$
- Règle de particularisation $\dfrac{\forall x\ \alpha(x)}{\alpha(t)}$ t *libre pour* x *dans* α
- Règle de généralisation

$$\frac{\alpha(x)}{\forall x\ \alpha(x)}\ x\ n'apparaitpas\ libre\ dans\ les\ prémisses\ non\ éliminées\ au\ dessus\ de\ \alpha(x)$$

- Règle de remplacement $\dfrac{t1 = t2 \quad \alpha(t1)}{\alpha(t2)}$ $t1$ *et* $t2$ *libres pour* x *dans* α

Axiome

Axiome:

$$\forall x \quad x=x$$

Règle de particularisation

(E∀):

$$\frac{\forall x\ \alpha(x)}{\alpha(t)}\ t\ libre\ pour\ x\ dans\ \alpha$$

Cas particulier: t=x

$$\frac{\forall x\ \alpha(x)}{\alpha(x)}$$ x libre pour x

Règle de généralisation

$$\frac{\alpha(x)}{\forall x\ \alpha(x)}\ x\ n'apparaitpas\ libre\ dans\ les\ prémisses\ non\ éliminées\ au\ dessus\ de\ \alpha(x)$$

Règle de remplacement

$$\frac{t1 = t2 \quad \alpha(t1)}{\alpha(t2)} \; t1 \; et \; t2 \; libres \; pour \; x \; dans \; \alpha$$

Déduction

Soient Γ un ensemble de formules et α une formule

On dira que α est déduite de Γ et on note $\Gamma \vdash \alpha$ s'il existe une déduction telle que:

- α est la conclusion
- Les seules prémisses non éliminées (sauf l'axiome) sont dans Γ
- Les règles utilisées sont celles du système déductif défini précédemment

Etude sémantique: les interprétations

Soit $\mathcal{D}$ un ensemble appelé « domaine d'interprétation »

l'interprétation I est une correspondance telle que:

- Symbole de constante ---> élément de $\mathcal{D}$
- Symbole de fonction d'arité n--->fonction $\mathcal{F} : \mathcal{D}^n \rightarrow \mathcal{D}$
- Symbole de prédicat d'arité n--->relation $\mathcal{R} : \mathcal{D}^n \rightarrow \{v,f\}$
- Prédicat '=' ---> relation Diag $\mathcal{D}$={ (d,d)/ d $\in \mathcal{D}$ }

Valuation

Soit I une interprétation de domaine $\mathcal{D}$ du langage $\mathcal{L}$

- On appelle valuation $\mathcal{V}$ des variables toute fonction de l'ensemble des variables dans l'ensemble $\mathcal{D}$:

$\mathcal{V} : \mathcal{V}ar \text{ --->} \mathcal{D}$

$\mathcal{V}ar$: Ensemble de variables de $\mathcal{L}$

Formule satisfaite par une valuation pour une interprétation I

Soient $\mathcal{L}$ un langage, I une interprétation de domaine $\mathcal{D}$, et $\mathcal{V}$ une valuation définie sur $\mathcal{D}$

- On dit qu'une formule α est satisfaite par $\mathcal{V}$ pour I et on note $I \models \alpha\ [\mathcal{V}]$ si l'interprétation de la formule α est vraie lorsqu'on associe aux connecteurs leurs tables de vérité et le quantifieur universel est interprété « quel que soit l'élément du domaine $\mathcal{D}$ »

Formule satisfaisable pour une interprétation I

- On dira que la formule α est satisfaisable pour I s'il existe une valuation $\mathcal{V}$ telle que $I \models \alpha\ [\mathcal{V}]$

- On dira qu'une formule α est satisfaisable s'il existe une interprétation I sur le domaine $\mathcal{D}$ pour laquelle α est satisfaisable. i.e. il existe I et il existe $\mathcal{V}$ sur $\mathcal{D}$ telles que $I \models \alpha\ [\mathcal{V}]$

Formule valide pour I, universellement valide

- On dira qu'une formule α est valide pour I et on note $I \models \alpha$ si quelle que soit la valuation $\mathcal{V}$ on a $I \models \alpha\ [\mathcal{V}]$
- On dira que α est universellement valide et on note $\models \alpha$ si quelle que soit l'interprétation I on a $I \models \alpha$. i.e. quelle que soit l'interprétation I et quelle que soit la valuation $\mathcal{V}$ sur $\mathcal{D}$ on a $I \models \alpha\ [\mathcal{V}]$

Pour résumer…

- α satisfaisable : $\exists\ I,\ \exists\ \mathcal{V}\ /\ \ I \models \alpha\ [\mathcal{V}]$
- α satisfaisable pour I : $\exists\ \mathcal{V}\ /\ \ I \models \alpha\ [\mathcal{V}]$
- α est satisfaite par $\mathcal{V}$ pour I : $I \models \alpha\ [\mathcal{V}]$
- α est universellement valide: $\forall I, \forall \mathcal{V}\ \ I \models \alpha\ [\mathcal{V}]$ est vraie
- α est valide pour I : $\forall \mathcal{V}\ \ I \models \alpha\ [\mathcal{V}]$ est vraie

Remarques

- $\models \alpha$ ssi $\forall I, \forall \mathcal{V} \quad I \models \alpha[\mathcal{V}]$
- $\not\models \alpha$ ssi $\exists\ I, \exists\ \mathcal{V}\ /\quad I \not\models \alpha[\mathcal{V}]$
- $I \not\models \alpha$ ssi $\exists\ \mathcal{V}\ /\quad I \not\models \alpha[\mathcal{V}]$
- α non satisfaisable ssi $\forall I, \forall \mathcal{V} \quad I \not\models \alpha[\mathcal{V}]$

Satisfiabilité: définition formelle

On considère toutes les formes possibles que peut prendre une formule $\mathcal{A}$

- $I \models (xi=xj)[\mathcal{V}] \Leftrightarrow (\mathcal{V}(xi), \mathcal{V}(xj)) \in \text{Diag}_{\mathcal{D}}$
- $I \models (ti=tj)[\mathcal{V}] \Leftrightarrow (I(ti), I(tj)) \in \text{Diag}_{\mathcal{D}}$
- $I \models P(t_1,\ldots,tn)[\mathcal{V}] \Leftrightarrow (I(t_1),\ldots,I(tn)) \in I(P)$
- $I \models (\alpha \wedge \beta)[\mathcal{V}] \Leftrightarrow I \models \alpha[\mathcal{V}]$ et $I \models \beta[\mathcal{V}]$
- $I \models (\alpha \vee \beta)[\mathcal{V}] \Leftrightarrow I \models \alpha[\mathcal{V}]$ ou $I \models \beta[\mathcal{V}]$
- $I \models (\alpha \rightarrow \beta)[\mathcal{V}] \Leftrightarrow (I \models \alpha[\mathcal{V}] \Rightarrow I \models \beta[\mathcal{V}])$ ou bien si … alors
- $I \models (\alpha \leftrightarrow \beta)[\mathcal{V}] \Leftrightarrow (I \models \alpha[\mathcal{V}] \Leftrightarrow I \models \beta[\mathcal{V}])$ ou bien ssi
- $I \models (\forall x \alpha(x)) \Leftrightarrow \forall d \in \mathcal{D}, I \models \alpha[\mathcal{V}^{d}_{x}]$
- $I \models (\exists x \alpha(x)) \Leftrightarrow \exists d \in \mathcal{D}, I \models \alpha[\mathcal{V}^{d}_{x}]$

Remarque

- Si α est une formule contenant les variables libres $x_1,\ldots x_n$ alors:

α valide pour $I \Leftrightarrow \forall x_1,\ldots, \forall x_n\, \alpha$ valide pour I

i.e.

$$\forall \mathcal{V} \quad I \models \alpha[\mathcal{V}] \Leftrightarrow I \models \forall x_1,\ldots, \forall x_n\ \alpha$$

- Si α est une formule fermée alors α est soit valide pour I soit non valide pour I.

i.e.

$$\exists \mathcal{V} \quad I \models \alpha[\mathcal{V}] \Leftrightarrow \forall \mathcal{V} \quad I \models \alpha[\mathcal{V}] \Leftrightarrow I \models \alpha$$

Modèle

Soit Γ un ensemble de formules.

- On dira que l'interprétation $\mathcal{M}$ est un modèle de Γ si toutes les formules de Γ sont valides pour l'interprétation $\mathcal{M}$.

i.e.

$\forall \alpha \in \mathcal{M}$, $\mathcal{M} \models \alpha$

Théorème de consistance et complétude: définitions

- Soient Γ un ensemble de formules et α une formule

  Γ implique sémantiquement α, noté $\Gamma \models \alpha$ si:

  $\forall I, \forall \mathcal{V}\ (\forall \beta \in \Gamma\ I \models \beta\ [\mathcal{V}] \Rightarrow I \models \alpha[\mathcal{V}]\ )$

- Soient α et β deux formules

  α est sémantiquement équivalent à β (on note $\alpha \models\mid \beta$) si

  $\forall I, \forall \mathcal{V}\ (I \models \beta\ [\mathcal{V}]$ ssi $I \models \alpha[\mathcal{V}]\ )$

Remarques

- $\alpha \models \beta \Leftrightarrow \models \alpha \rightarrow \beta$
- $\alpha \models\mid \beta \Leftrightarrow \models \alpha \leftrightarrow \beta$
- $\alpha \models\mid \beta \Leftrightarrow (\alpha \models \beta \text{ et } \beta \models \alpha)$
- $\alpha \vdash\mid \beta \Leftrightarrow (\alpha \vdash \beta \text{ et } \beta \vdash \alpha)$

Théorème

- $\vdash \alpha \Leftrightarrow \models \alpha$ (théorème ssi universellement valide)
- $\Gamma \vdash \alpha \Leftrightarrow \Gamma \models \alpha$ (déduction ssi implication sémantique)
- $\alpha \vdash\dashv \beta \Leftrightarrow \alpha \models\dashv \beta$

Proposition:

Soit Γ un ensemble de formules et β une formule

$\Gamma \vdash \beta \Leftrightarrow \Gamma \cup \{\neg\beta\}$ n'admet pas de modèle

Décidabilité

Soit la déduction $\Gamma \vdash \beta$

Question: existe-t-il un programme (algorithme) qui effectue n'importe quelle déduction?

- il existe un algorithme semi-décidable:
  - Soit il se termine: on a une réponse
  - Soit il ne se termine pas: on ne peut rien dire
- Le calcul des prédicats est indécidable