LOGIQUE

Pourquoi étudier la logique dans un cursus d'informatique ?

"Le langage de la logique est l'outil naturel pour exprimer le comportement attendu - les objectifs - d'un programme, de la même manière qu'un langage de programmation permet d'exprimer les algorithmes réalisant ce comportement. Les techniques logiques jouent un rôle central dans l'automatisation de la programmation, ceci incluant la synthèse, la vérification, la correction, et la transformation de programmes. Les méthodes issues de la logique sont utiles dans de nombreux domaines de l'Intelligence Artificielle, incluant la planification, la représentation de connaissances, et le traitement de la langue naturelle. Des langages de programmation (comme LISP ou PROLOG) utilisant des formules de la logique comme programmes sont de plus en plus utilisés. La connaissance de la logique est devenue une nécessité quotidienne pour l'informaticien professionnel."

(extrait de la Préface de Z. Manna & R. Waldinger *The Logical Basis for Computer Programming* Addison-Wesley, 1985 - depuis cette date leur point de vue a été largement confirmé)

Ainsi, la logique a une place centrale dans les techniques *d'intelligence artificielle*, elle est l'un des cadres de référence des *bases de données relationnelles*, elle est très importante en *génie logiciel*, et, pour ne citer qu'un exemple d'application récente en France, le programme de conduite du métro METEOR a été réalisé par la méthode B, qui est basée sur la logique. Signalons enfin, même si ce cours s'intéresse principalement aux applications de la logique dans le cadre du "logiciel", que la logique des propositions est la base de la construction des circuits logiques qui constituent le cœur du "matériel".

Objectif principal de la logique

L'objectif principal de la logique est *d'étudier la validité des raisonnements* : des raisonnements humains dans tous les domaines - des raisonnements en mathématiques aux raisonnements dans la vie quotidienne (raisonnements "naturels") - ainsi que des raisonnements "artificiels" (calculs sur ordinateur), étudier la validité, la véracité, de propositions complexes formées avec des propositions élémentaires qui représentent — modélisent — des énoncés sur le monde, et pas la véracité de ces propositions élémentaires.

Exemple : Le diamètre de Vénus est < 10.000 kms
ou
le diamètre de Vénus est ≥ 10.000 kms

Cette proposition complexe - formée des deux propositions élémentaires A = "le diamètre de Vénus est < 10.000 kms" et B = "le diamètre de Vénus est ≥ 10.000 kms" est valide, c'est-à-dire toujours "vraie", ou encore : "vraie" quelle que soit la valeur de vérité des propositions élémentaires qui la composent. Inutile d'avoir des notions d'astronomie pour affirmer la validité d'une telle expression, nous utilisons une modélisation de l'énoncé sous la forme $(A \vee \neg A)$ qui est une proposition valide. (Nous n'aborderons les problèmes posés par une telle modélisation — tiers exclu, sens des connecteurs, choix des énoncés élémentaires, ... — que dans les TD au travers de quelques exemples).

Chapitre 1
Rappels de logique des propositions

1 Syntaxe

Définition des formules bien formées, ***fbf***, de la logique des propositions par induction :

(base) un ensemble Σ dénombrable de symboles propositionnels : {p, q, ……}
[un symbole particulier $\bot$ (absurde, FAUX, ...)]

(cons) A et B étant des fbf
$\neg A$, $(A \wedge B)$, [$(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$] sont des fbf

Ces fbf de la logique des propositions s'appellent aussi parfois simplement des propositions. Nous noterons par ***PROP(Σ)***, ou PROP lorsque Σ est implicite, l'ensemble des fbf construites à partir de Σ.

De nombreuses variantes syntaxiques existent (toutes équivalentes comme nous le verrons plus tard). Par exemple : ici on ne considère que les deux connecteurs $\wedge$ et $\neg$, les autres étant des abréviations (des macros) :
$\bot$ pour $(A \wedge \neg A)$,
$(A \vee B)$ pour $\neg(\neg A \wedge \neg B)$,
$(A \rightarrow B)$ pour $(\neg A \vee B)$,
$(A \leftrightarrow B)$ pour $((A \rightarrow B) \wedge (B \rightarrow A))$, donc pour $((\neg A \vee B) \wedge (\neg B \vee A))$

Ceci sera justifié plus tard (cf. le "formulaire de base" dans la section "sémantique").

Définition de quelques notions par induction

L'ensemble des symboles d'une fbf peut être défini rigoureusement en suivant la définition inductive des fbf :
SymbProp(P) = si P est un symbole propositionnel alors {P}
si P = $\neg$Q alors SymbProp(Q)
si P = (Q $\wedge$ R) alors SymbProp(Q) $\cup$ SymbProp(R)

De même, le nombre total d'occurrences de connecteurs apparaissant dans une fbf peut être défini par :
nombre-de-connecteurs(P) =
si P est un symbole propositionnel alors 0
si P = $\neg$Q alors nombre-de-connecteurs(Q) + 1
si P = (Q $\wedge$ R) alors nombre-de-connecteurs(Q) + nombre-de-connecteurs(R) + 1

On peut définir ainsi de nombreuses notions : ensemble des sous-fbf d'une fbf, profondeur d'une fbf, ensemble des connecteurs d'une fbf, ...
Une notion particulièrement importante est celle d'arborescence syntaxique d'une fbf.

Arborescence syntaxique

arbo(P) =

si P est un symbole propositionnel alors c'est une arborescence réduite à un sommet étiqueté par P

si P = ¬Q alors c'est une arborescence ayant sa racine étiquetée par ¬ et ayant un seul successeur qui est la racine de arbo(Q)

si P = (Q ∧ R) alors c'est une arborescence ayant sa racine étiquetée par ∧ et ayant pour successeur gauche la racine de arbo(Q) et pour successeur droit la racine de arbo(R).

On peut définir ARBO-PROP(Σ) comme étant l'ensemble des arborescences ayant leurs feuilles étiquetées sur Σ, leurs nœuds étiquetés par les connecteurs, et respectant les arités. On peut démontrer qu'il existe une bijection entre PROP(Σ) et ARBO-PROP(Σ).

Pour prouver que toute fbf a une certaine propriété P, il suffit de prouver :
(1) que tout symbole propositionnel a la propriété P,
(2) que si P = ¬Q et si Q a la propriété P alors P a également la propriété P,
(3) que si P = (Q ∧ R) et si Q et R ont la propriété P alors P a également la propriété P.
[Si la fbf ⊥ et tous les connecteurs sont dans le langage initial il faut également prouver : ⊥ a la propriété P, et idem (3) avec tous les connecteurs binaires.]

2 Sémantique

Pour pouvoir donner un "sens", c'est-à-dire une valeur prise dans un ensemble à deux éléments {0,1} (ou {*vrai*, *faux*} *avec vrai=1 et faux=0*), à une fbf quelconque nous allons suivre la définition par induction des fbf.

Interprétation des symboles propositionnels
Si Σ est un ensemble de symboles propositionnels une ***interprétation*** de Σ est une application *I* de Σ dans l'ensemble B ={0, 1} (ou dans{*vrai*, *faux*}).
Une telle application peut-être considérée comme la fonction caractéristique d'une partie de Σ, une telle partie s'appelant parfois un "monde possible". L'ensemble de ces applications est en bijection avec l'ensemble des parties de Σ.

L'interprétation des lettres autres que les symboles propositionnels apparaissant dans une fbf est *indépendante* de l'interprétation des symboles propositionnels.

Interprétation de ⊥ c'est 0 (ou *faux*), donc on interprète toujours de la même façon le symbole ⊥. De même, on interprète toujours de la même façon les connecteurs. Ces interprétations ne sont pas arbitraires, elles ont été choisies de telle sorte qu'elles puissent être utilisées pour modéliser des raisonnements.

Interprétation des connecteurs

On interprète les connecteurs comme des applications de B^2, ou de B, dans B, suivant qu'ils sont d'arité 2 ou 1.

NON, l'interprétation de $\neg$, est l'application de B dans B qui échange 0 et 1,

ET, l'interprétation de $\wedge$, est l'application de B^2 dans B qui vaut 1 ssi les deux arguments valent 1,

[nous donnons l'interprétation des autres connecteurs même si c'est redondant]

OU, l'interprétation de $\vee$, est l'application de B^2 dans B qui vaut 0 ssi les deux arguments valent 0,

SI-ALORS , l'interprétation de $\rightarrow$, est l'application de B^2 dans B qui vaut 0 ssi le premier argument vaut 1 et le second 0,

SSI, l'interprétation de $\leftrightarrow$, est l'application de B^2 dans B qui vaut 1 ssi les deux arguments ont la même valeur.

Une fois le cadre, restrictif (considérer exclusivement deux valeurs de vérité), choisi, si l'on passe en revue les 4 applications de B dans B, le choix de l'interprétation du symbole de négation semble naturel. De même, si l'on souhaite que l'interprétation du connecteur $\wedge$ permette de modéliser certains usages de la conjonction de coordination "et" de la langue naturelle, son interprétation semble naturelle. Pour se convaincre que le choix du sens de $\rightarrow$ n'est pas stupide, il suffit de considérer les 16 applications de B^2 dans B et de chercher celle qui correspondrait le moins mal au "si...alors..." déductif (i.e. si hypothèse alors conclusion) de la langue naturelle.

Interprétation d'une fbf

Une interprétation I étant donnée, on peut définir la valeur de vérité, ***v(P, I)***, d'une proposition P.
L'application v est définie par :

si P est un symbole propositionnel alors v(P, I) = I(P)
si P = $\perp$ alors v(P, I) = 0 pour tout I
si P = $\neg$Q alors v(P, I) = NON (v(Q, I))
si P = (Q$*$R) alors v(P, I) = OP(v(Q, I), v(R, I)) ; où OP = OU si $* = \vee$, OP = ET si $* = \wedge$, OP = SI-ALORS si $* = \rightarrow$, et OP = SSI si $* = \leftrightarrow$.

On peut dire que la logique des propositions a pour objectif d'étudier un ensemble d'énoncés simples Σ et les énoncés que l'on peut construire, avec les connecteurs, à partir de Σ. D'un point de vue intuitif une interprétation de ces énoncés est un "monde possible" dans lequel chaque énoncé est vrai ou faux.
Une interprétation qui donne la valeur 1 à une fbf est parfois appelée un modèle de cette fbf (respectivement contre-modèle si la valeur est 0).

Table de vérité

P étant une proposition ayant $\Sigma = \{p_1, p_2, ..., p_k\}$ pour ensemble de symboles la table de vérité de P est un tableau qui associe aux 2^k interprétations différentes I_j de Σ les 2^k valeurs de $v(P, I_j)$.

Définitions

Une fbf est ***satisfiable*** s'il existe au moins une interprétation dans laquelle elle est vraie

Une fbf est ***contingente*** s'il existe au moins une interprétation dans laquelle elle est vraie et une dans laquelle elle est fausse

Une fbf est ***valide*** si dans toute interprétation elle est vraie

Une fbf est ***insatisfiable*** si elle est fausse dans toute interprétation

Propriétés

P est satisfiable ssi ¬P n'est pas valide
P est insatisfiable ssi ¬P est valide
P est contingente ssi ¬P est contingente

3 Equivalence logique et formulaire de base

Définition

Deux fbf P et Q sont ***logiquement équivalentes*** ssi elles ont la même valeur pour toute interprétation i.e. pour toute interprétation I v(P, I) = v(Q, I).

On utilise la notation $P \equiv Q$.

Attention : $P \equiv Q$ n'est pas une fbf, il ne faut pas confondre $P \equiv Q$ et $(P \leftrightarrow Q)$

Théorème

$P \equiv Q$ ssi $P \leftrightarrow Q$ est valide

Il ne faut pas non plus confondre l'équivalence logique avec l'égalité de deux fbf, P = Q, qui est une notion syntaxique.

Formulaire de base

Idempotence de $\wedge$ et $\vee$

$(A \wedge A) \equiv A$ $\quad\quad$ $(A \vee A) \equiv A$

Associativité et commutativité de $\wedge$ et $\vee$

$((A \wedge B) \wedge C) \equiv (A \wedge (B \wedge C))$ $\quad\quad$ $((A \vee B) \vee C) \equiv (A \vee (B \vee C))$

$(A \wedge B) \equiv (B \wedge A)$ $\quad\quad$ $(A \vee B) \equiv (B \vee A)$

Distributivité

$((A \wedge (B \vee C) \equiv ((A \wedge B) \vee (A \wedge C))$ $\quad\quad$ $((A \vee (B \wedge C) \equiv ((A \vee B) \wedge (A \vee C))$

Double négation

$\neg\neg A \equiv A$

Lois de DeMorgan

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

Implication

$(A \rightarrow B) \equiv (\neg A \vee B)$ $\quad\quad$ $(A \leftrightarrow B) \equiv ((A \rightarrow B) \wedge (B \rightarrow A))$

$\neg A \equiv (A \rightarrow \bot)$

Absurde

$\bot \equiv (A \wedge \neg A)$

Définitions
{P1, P2, ..., Pk} est ***consistant*** ssi (P1∧(P2∧ ...(Pk-1∧Pk)...) est satisfiable
{P1, P2, ..., Pk} est ***inconsistant***, ou contradictoire, ssi (P1∧ (P2∧ ...(Pk-1∧Pk)...) est insatisfiable

4 Conséquence logique

Définition
Une fbf C est une ***conséquence logique*** de l'ensemble de fbf $\{H_1, ..., H_k\}$ ssi pour toute interprétation I telle que pour tout i $v(H_i, I) = 1$ (ou *vrai*) alors $v(C, I) = 1$ (ou *vrai*).
Dans ce cas on utilise la notation $\{H_1, ..., H_k\} \models C$.
Attention : $\{H_1, ..., H_k\} \models C$ n'est pas une fbf !

La notion de conséquence logique peut être considérée comme une modélisation d'un pas élémentaire de raisonnement.

Théorème
{H1, ..., Hk} |= C ssi
H1 ∧ ... ∧ Hk |= C ssi
H1 ∧ ... ∧ Hk → C valide ssi
H1 ∧... ∧Hk ∧ ¬C insatisfiable

Ce théorème permet de remplacer la pertinence d'un pas de raisonnement par la validité d'une fbf. (La démonstration du théorème est immédiate à partir des définitions.)