Logique

Logique des propositions

- Manipule des formules auxquelles peuvent être assignées les valeurs Vrai (V) ou Faux (F)
- Les formules peuvent être de deux types
  – Formules atomiques ou propositions (indécomposables)
  – Formules composées obtenues avec des opérations logiques (ET noté $\wedge$, OU noté $\vee$, NOT noté $\neg$, si … alors noté $\Rightarrow$)
- Les propositions peuvent être exprimées dans un langage plus ou moins formel exemples:
- « Il pleut », « 1>0 »…
- But du calcul des propositions: partir d'un certain nombre de faits (formules tenues pour vraies) et en déduire des conséquences (autres formules vraies ou fausses)

Logique des propositions

- Les formules atomiques doivent être considérées comme sans signification intrinsèque :
- Si « il pleut » et « la route est mouillée » sont deux propositions atomiques elles n'ont en tant que telles aucun rapport entre elles.
- En revanche si les faits sont:
  - « il pleut » est Vrai
  - (« il pleut » ⇒ « la route est mouillée ») est Vrai
- Alors les manipulations formelles de la logique des propositions doivent permettre de conclure : « la route est mouillée » est Vrai

Logique des propositions

 Plus formellement :

- Soit $P=\{a,b,\ldots\}$ un ensemble de propositions.
- Prenant comme symboles de base $\vee$ et $\neg$ (convention de Whitehead et Russel en 1910, d'autres choix sont possibles), l'ensemble des formules F est défini comme :
  - Toute proposition est une formule
  - Si A est une formule, $(\neg A)$ est une formule
  - Si A et B sont des formules $(A \vee B)$ est une formule
- On introduit par ailleurs certaines abréviations :
  - $(A \wedge B)$ est une abréviation de $\neg(\neg A \vee \neg B)$
  - $(A \Rightarrow B)$ est une abréviation de $\neg A \vee B$ (comprendre $(\neg A) \vee B$, le $\neg$ étant prioritaire sur le $\vee$)
  - $(A \Leftrightarrow B)$ est une abréviation de $(A \Rightarrow B) \wedge (B \Rightarrow A)$

Tables de vérité : ∨, ¬

5 La définition des connecteurs logiques résulte des axiomes des algèbres de Boole, mais on peut les représenter (comme pour toute formule) par des tables listant tous les cas de vérité des variables de la formule (ici ces deux tables peuvent être prises pour définition de ∨ et ¬) :

A	B	A∨B
V	**V**	**V**
V	**F**	**V**
F	**V**	**V**
F	**F**	**F**

A	¬A
V	**F**
F	**V**

Tables de vérité : ∧

6 On peut alors en déduire les tables correspondant aux abréviations définies plus haut ici pour $A \wedge B$ défini comme $\neg(\neg A \vee \neg B)$

A	B	¬A	¬B	¬A∨¬B	¬(¬A∨¬B)
V	**V**	**F**	**F**	**F**	**V**
V	**F**	**F**	**V**	**V**	**F**
F	**V**	**V**	**F**	**V**	**F**
F	**F**	**V**	**V**	**V**	**F**

Tables de vérité, ⇒

Idem pour (A⇒B) défini comme ¬A∨B)

A	B	¬A	¬AvB
V	**V**	**F**	**V**
V	**F**	**F**	**F**
F	**V**	**V**	**V**
F	**F**	**V**	**V**

- Important : ne pas considérer le ⇒ comme une causalité (on rappelle que les propositions doivent être considérées comme sans signification intrinsèque)
- Le (si A alors B) ici noté (A⇒B) mais parfois noté (A⊃B) signifie simplement que B n'est pas plus faux que A

Signification du $\Rightarrow$

8 L'utilisation de $A \Rightarrow B$ énoncé comme «A implique B», «A entraîne B» ou «si A alors B» dans le cas où A est faux est troublant pour le logicien débutant.

- C'est pourtant nécessaire pour prendre en compte dans le formalisme de la logique les cas des raisonnements justes (implication vraie) sur des hypothèses fausses, qui peuvent aboutir à des conclusions vraies ou fausses !

A	B	$A \Rightarrow B$
V	**V**	**V**
V	**F**	**F**
F	**V**	**V**
F	**F**	**V**

- Par exemple pour 4 entiers a, b, c et d quelconques :
- $a=c \wedge b=d \Rightarrow a+b=c+d$ est VRAI
- $1=2 \wedge 3=4 \Rightarrow 4=6$ est VRAI
- $1=2 \wedge 4=3 \Rightarrow 5=5$ est VRAI
- Il est donc VRAI que le FAUX « implique » FAUX ou VRAI !

Un autre exemple de Faux ⇒ Faux

9 Souvenez-vous du raisonnement par récurrence : un propriété peut très bien être fausse mais héréditaire ($P_n \Rightarrow P_{n+1}$ est vrai)

- Exemple : 10^n+1est divisible par 9
- On suppose que 10^n+1 est divisible par 9 soit $10^n+1=9*k$
- Alors $10^{n+1}+1=10*10^n+1=10*(10^n+1)-10+1=10*9*k-9=9*(10*k-1)$ la propriété est donc héréditaire.
- Elle est pourtant tout ce qu'il y a de plus fausse…
- Bel exemple de Vrai (Faux⇒Faux) qui montre l'importance de l'initialisation dans le raisonnement par récurrence.

Signification du ⇒

- Lorsque A⇒B est VRAI on peut dire :

- « B n'est pas plus faux que A »
- «A implique B»
- «A entraîne B»
- «si A alors B»
- «B si A» (A est une condition suffisante pour B)
- «A seulement si B» (B est une condition nécessaire pour A)

A	B	A⇒B
V	**V**	**V**
V	**F**	**F**
F	**V**	**V**
F	**F**	**V**

Tables de vérité, ⇔

- Pour l'équivalence, notée comme $(A \Leftrightarrow B)$ est une définie comme une abréviation de $(A \Rightarrow B) \wedge (B \Rightarrow A)$.
- On a donc la table de vérité triviale suivante :

A	B	A⇒B	B⇒A	A⇔B
V	**V**	**V**	**V**	**V**
V	**F**	**F**	**V**	**F**
F	**V**	**V**	**F**	**F**
F	**F**	**V**	**V**	**V**

- L'équivalence n'est vraie que si les propositions A et B ont les mêmes valeurs de vérité.

Signification du ⇔

L'équivalence, (A⇔B) est donc :

- (A⇒B) : B si A
  (A est une condition suffisante pour B)
- et
- (B⇒A) :
  B seulement si A
  (A est une condition nécessaire pour B)

A	B	A⇒B	B⇒A	A⇔B
V	**V**	**V**	**V**	**V**
V	**F**	**F**	**V**	**F**
F	**V**	**V**	**F**	**F**
F	**F**	**V**	**V**	**V**

Soit : B si et seulement si (ssi) A (A est une condition nécessaire et suffisante (cns) pour B)

Ou A si et seulement si (ssi) B (B est une condition nécessaire et suffisante (cns) pour A)

Tautologies

On appelle tautologie (ou formule valide) une formule évaluée à Vrai quelle que soit la valeur des variables propositionnelles. Par exemple pour toute formule A, $(A \vee \neg A)$ est une tautologie.

- Deux formules A et B sont dites équivalentes si la formule $(A \Rightarrow B) \wedge (B \Rightarrow A)$, également notée $A \Leftrightarrow B$ ou $B \Leftrightarrow A$ est une tautologie. Par exemple $\neg(\neg A) \Leftrightarrow A$ est une tautologie.
- Noter que les tautologies généralisent aux formules la notion d'équivalence.
- Intuitivement les tautologies (dont les équivalences sont un cas particulier important) peuvent être vues comme des « formules vraies » dans le système formel.

Tautologies

A titre d'exemple si les faits sont que :

- A (exemple : « Il pleut ») est vrai
- Et que ($A \Rightarrow B$) (exemple : « si il pleut alors la route est mouillée ») est vrai

- Alors on doit pouvoir déduire que B (« la route est mouillée ») est vrai (car B ne peut être plus faux que A)

  - Par conséquent la formule $(A \wedge (A \Rightarrow B)) \Rightarrow B$ doit être une tautologie.

A	B	$A \Rightarrow B$
V	**V**	**V**
V	**F**	**F**
F	**V**	**V**
F	**F**	**V**

Tables de vérité

- Le fait que $(A \wedge (A \Rightarrow B)) \Rightarrow B$ soit une tautologie peut se vérifier à l'aide de la table de vérité

A	B	$A \Rightarrow B$	$(A \wedge (A \Rightarrow B))$	$(A \wedge (A \Rightarrow B)) \Rightarrow B$
V	**V**	**V**	**V**	**V**
V	**F**	**F**	**F**	**V**
F	**V**	**V**	**F**	**V**
F	**F**	**V**	**F**	**V**

Tables de vérité

- Les équivalences classiques en logique sont donc également des tautologies. Exemple pour la contraposée d'une proposition conditionnelle :

A	B	$A \Rightarrow B$	$\neg A$	$\neg B$	$\neg B \Rightarrow \neg A$	$(A \Rightarrow B) \Leftrightarrow (\neg B \Rightarrow \neg A)$
V	**V**	**V**	**F**	**F**	**V**	**V**
V	**F**	**F**	**F**	**V**	**F**	**V**
F	**V**	**V**	**V**	**F**	**V**	**V**
F	**F**	**V**	**V**	**V**	**V**	**V**

- Idem pour tous les théorèmes classiques de logique, dont la transitivité de l'implication : $((A \Rightarrow B) \wedge (B \Rightarrow C)) \Rightarrow (A \Rightarrow C)$

Tables de vérité

- Les équivalences classiques en logique sont donc également des tautologies. Exemple pour l'une des lois de Morgan :

A	B	AvB	¬A	¬B	¬(AvB)	¬A∧¬B	¬(AvB) ⇔(¬A∧¬B)
V	**V**	**V**	**F**	**F**	**F**	**F**	**V**
V	**F**	**V**	**F**	**V**	**F**	**F**	**V**
F	**V**	**V**	**V**	**F**	**F**	**F**	**V**
F	**F**	**F**	**V**	**V**	**V**	**V**	**V**

Récréation : formel et langage courant

P «Tu sais»

- Opérateur not : ne… pas. Donc not(P) : «Tu ne sais pas»
- Opérateur not : être sans. Donc not(P) : «Tu es sans savoir»
- Mais par ailleurs not(«Tu sais») se dit aussi «Tu ignores»
- Donc «Tu n'ignores pas» ⇔ not(not(P)) ⇔ P «Tu sais»
- «Tu n'es pas sans savoir» ⇔ not(not(P)) ⇔ P «Tu sais»
- «Tu es sans ignorer» ⇔ not(not(P))⇔P «Tu sais»
- «Tu n'es pas sans ignorer» ⇔ not(not(not(P)))⇔not(P) «Tu ne sais pas»
- «Tu n'es pas sans ne pas ignorer» ⇔not(not(not(not(P)))) ⇔ P «Tu sais»

Systèmes axiomatiques

- Il est possible, au lieu de construire la table de vérité de toute formule pour vérifier si elle est valide, d'utiliser des systèmes d'axiomes afin de déduire des formules valides à partir d'autres, en utilisant deux règles d'inférence :
- Modus Ponens : Si A est une tautologie (prémisse mineure) et $A \Rightarrow B$ est une tautologie (prémisse majeure) alors B est une tautologie.
- Substitution : Dans toute tautologie il est possible de remplacer une lettre représentant une formule, par une formule quelconque, à condition de remplacer toutes les occurrences de cette lettre par la même formule.

Systèmes axiomatiques

- A partir d'axiomes (tautologies prises comme point de départ) d'autres tautologies (théorèmes) peuvent déduites en utilisant les règles d'inférence.
- Par exemple l'axiomatique de Whitehead et Russel prend comme point de départ les axiomes suivants :

1. $(A \vee A) \Rightarrow A$
2. $B \Rightarrow (A \vee B)$
3. $(A \vee B) \Rightarrow (B \vee A)$
4. $(A \vee (B \vee C)) \Rightarrow (B \vee (A \vee C))$
5. $(B \Rightarrow C) \Rightarrow ((A \vee B) \Rightarrow (A \vee C))$

Systèmes axiomatiques

- Dans ce système axiomatique, on peut démontrer (par exemple) la formule suivante en tant que théorème $(B \Rightarrow C) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$: en effet :
- $(B \Rightarrow C) \Rightarrow ((A \vee B) \Rightarrow (A \vee C))$: axiome 5
- $(B \Rightarrow C) \Rightarrow ((\neg A \vee B) \Rightarrow (\neg A \vee C))$: substitution de A par $\neg A$
- $(B \Rightarrow C) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$: définition de $\Rightarrow$

Afin de formaliser plus clairement le raisonnement (processus de preuve de théorème), il est toutefois plus aisé de distinguer plusieurs niveaux de langage à l'aide de la logique des séquents.

Formalisation du raisonnement : Séquents

- Qu'est ce qu'une preuve ?
- Dans la plupart des cas ce qui est à prouver n'est pas une affirmation (qui à ce stade est une supposition), mais le fait que certaines hypothèses (appelées prémices ou suppositions) entraînent cette affirmation (appelée conclusion ou but).
- Cela est formalisé par un niveau d'abstraction plus élevé que la logique des propositions, appelé logique des séquents.
- Formellement un séquent s'écrit $\text{HYP} \vdash \text{BUT}$

et se lit « l'ensemble des prémices prises comme hypothèses entraîne la conclusion ou but ».

Formalisation du raisonnement : Séquents

23

- HYP et BUT sont (par exemple) des formules de la logique des propositions, les séquents ayant alors la forme : $H1, H2, ...Hn \vdash B1, B2..Bm$
- ce qui se lit comme « de H1 et H2 et… Hn on peut déduire B1 ou B2 ou… Bm »
- De fait une fois le séquent prouvé, la formule : $H1 \wedge H2 \wedge \ldots Hn \Rightarrow B1 \vee B2 \vee \ldots Bm$ est une formule valide (tautologie) de la logique des propositions dans le système formel considéré
- Au lieu de tout traiter au même niveau, la logique des séquents distingue donc le niveau sémantique où sont effectuées les déductions.

Formalisation du raisonnement : Séquents

- Ainsi le but des preuves formelles sera de déduire un séquent à partir d'autres séquents (certains pris comme départ en tant qu'axiomes) en utilisant des règles de transformations appelées règles d'inférence.
- Les règles d'inférence sont généralement écrites ainsi :

$$\frac{\begin{array}{c}\text{ANTECEDENT1}\\ \text{ANTECEDENT2}\\ \dots\\ \text{ANTECEDENTn}\end{array}}{\text{CONSEQUENCE}}$$

  - Les antécédents et la conséquence étant des séquents.
  - On dit que la preuve de la conséquence est dérivée de (i.e. résulte de), celle des antécédents.
  - De manière équivalente, on dit que la preuve de la conséquence peut être réduite à celle des antécédents (i.e. la preuve des antécédents est suffisante pour prouver la conséquence).

Formalisation du raisonnement : Séquents

$$\frac{\begin{array}{c}\text{ANTECEDENT1}\\ \text{ANTECEDENT2}\\ \ldots\\ \text{ANTECEDENTn}\end{array}}{\text{CONSEQUENCE}}$$

- Les règles d'inférence permettent donc de transformer les séquents afin de cheminer vers la preuve. Elles peuvent être utilisées :
- Vers l'avant (dérivation i.e. regroupement) si on a prouvé les antécédents, on a prouvé la conséquence.
- Plus souvent, vers l'arrière (réduction) pour prouver conséquence, il suffit de prouver antécédents (peut être plus facile)

Formalisation du raisonnement : Séquents

- Certaines règles sont génériques de la logique des propositions (vraies pour toutes propositions). Exemples :

$$\frac{H, P \vdash Q}{H \vdash P \Rightarrow Q} \qquad \frac{H \vdash P \Rightarrow Q}{H, P \vdash Q}$$

- Ces deux règles (qui signifient que les deux séquents sont équivalents), met bien en évidence les trois niveaux sémantiques :

$P \Rightarrow Q$

- Niveau formule : P implique Q (si P alors Q)

$H \vdash P \Rightarrow Q$

- Niveau séquent : de H on peut déduire P implique Q

$$\frac{H, P \vdash Q}{H \vdash P \Rightarrow Q}$$

- Niveau règle : pour prouver $H \vdash P \Rightarrow Q$ (conséquence) il suffit de prouver $H, P \vdash Q$ (antécédent) : interprétation arrière (réduction).

Formalisation du raisonnement : Séquents

$$\frac{H, P \vdash Q}{H \vdash P \Rightarrow Q}$$

$$\frac{H \vdash P \Rightarrow Q}{H, P \vdash Q}$$

- Ramener ces règles au seul niveau de la logique des propositions revient à dire que la formule $(H \wedge P \Rightarrow Q) \Leftrightarrow (H \Rightarrow (P \Rightarrow Q))$ est une tautologie ce que l'on peut vérifier :

H	P	Q	H∧P	P⇒Q	H∧P⇒Q	H⇒(P⇒Q)	(H∧P⇒Q)⇔(H⇒(P⇒Q))
V	V	V	V	V	V	V	V
V	V	F	V	F	F	F	V
V	F	V	F	V	V	V	V
V	F	F	F	V	V	V	V
F	V	V	F	V	V	V	V
F	V	F	F	F	V	V	V
F	F	V	F	V	V	V	V
F	F	F	F	V	V	V	V

Système formel, preuves

- La donnée d'un ensemble de règles d'inférences définit un système formel.
- Certaines de ces règles n'ont pas d'antécédent, ce sont de simples séquents qui sont les axiomes : $\overline{H \vdash P}$
- Certains axiomes peuvent même être de simples propositions : $\overline{\vdash P}$
- L'application des règles d'inférence permet de prouver de nouveaux séquents qui sont des théorèmes.
- On peut aussi prouver de nouvelles règles en montrant qu'ajouter les antécédents de la nouvelle règle au système formel existant permet de prouver sa conséquence.
- On peut alors ajouter ces règles dérivées et théorèmes aux règles de départ.

Systèmes incohérents, théorème de Gödel

- Un système formel est incohérent s'il permet de prouver deux séquents contradictoires $H \vdash P, H \vdash \neg P$
- On montre alors facilement qu'il permet de prouver tout et son contraire.
- Un système formel est dit incomplet si des formules n'y sont pas démontrables (dites indécidables).
- Un théorème majeur des mathématiques (Gödel 1931) montre que tout système formel assez grand pour contenir l'arithmétique des entiers est soit incohérent soit incomplet…
- Rajouter la formule indécidable (ou son contraire) dans l'axiomatique existante ne résout par le problème car l'axiomatique enrichie reste incomplète !

Logique des prédicats

- La logique des propositions ne permet pas d'exprimer des propositions générales concernant les éléments d'un ensemble, genre :
  - Tout entier naturel est positif ou nul
  - Il existe un ensemble ne contenant aucun élément
- La logique des prédicats est une extension de la logique des propositions permettant d'exprimer ce type de propositions.
- Dans cette optique les symboles de l'alphabet doivent être étendus en incluant les quantificateurs universel ($\forall$) et existentiel ($\exists$) qui doivent porter sur des variables qui doivent également faire partie de l'alphabet.

Logique des prédicats

- Plus précisément l'alphabet de la logique des prédicats comprend :
- Un ensemble de variables $x_1, x_2, \ldots$
- Un ensemble de symboles de fonction de 0, 1, 2… de ces variables, les symboles de fonction de 0 variables représentant des constantes. Exemples : x_1+1 (une variable) x_1+x_2 (2 variables), 1 (0 variables)…
- Un ensemble de symboles de prédicat (expressions logiques) de 0, 1, 2… de ces variables, les symboles de prédicat de 0 variables représentant des propositions. Exemples : $x_1=0$ (une variable), $x_1 \geq x_2$ (2 variables)…
- Les connecteurs : $\vee$, $\wedge$, $\neg$, $\Rightarrow$, $\Leftrightarrow$
- Les quantificateurs : $\forall, \exists$

Logique des prédicats

- La logique des prédicats distingue :
  - Les termes : toute variable est un terme ainsi que toute fonction de 0 variables (constante), ainsi que toute fonction de n termes.
  - Le B connaît la notion de terme sous le nom d'expression.
  - Les formules : tout prédicat de n termes est une formule, et si P et Q sont des formules $\neg P$, $P \vee Q$, $P \wedge Q$, $P \Rightarrow Q$, $P \Leftrightarrow Q$, $\forall x_i.P$, $\exists x_i.P$ sont des formules.
  - Le B ne distingue pas les prédicats et les formules.

Logique des prédicats

- Exemple : axiomes de l'arithmétique des nombres entiers de Péano (s(x) : successeur de) :
- $(\forall x)\neg(s(x)=0)$: aucun entier n'a 0 pour successeur
- $(\forall x\exists y)(\neg(x=0)\Rightarrow s(y)=x)$: tout entier non nul est successeur d'un entier
- $(\forall x,y)(s(x)=s(y))\Rightarrow x=y$: deux entiers de même successeur sont égaux
- $(\forall x)(x+0=x)$: 0 est l'élément neutre de l'addition
- $(\forall x,y)(x+s(y)=s(x+y))$: $x+(y+1)=(x+y)+1$
- $(\forall x)(x*0=0)$: 0 est élément absorbant de la multiplication
- $(\forall x,y)(x*s(y)=x*y+x)$: distributivité de * sur successeur

En vertu du théorème de Gödel, certaines formules sur les entiers sont indécidables dans l'axiomatique de Péano.

Syntaxe du langage B

- Le langage B reprend les notions de la logique des prédicats légèrement adaptée :
- On ne distingue par Prédicat et Formule, sont donc considérés comme prédicats : Prédicat, Prédicat∧Prédicat, Prédicat⇒Prédicat, ¬Prédicat, ∀Variable.Prédicat, [Substitution]Prédicat
- On introduit les substitutions, de la forme [Variable:=expression] ce qui signifie que toutes les occurrences libres de la variable sont remplacées par l'expression (notion équivalente à terme).
- Occurrence libre signifie que la variable n'est pas concernée par la portée d'un ∀ :
- Dans ∀y.(x+y-y=x), les variables y et z n'ont pas d'occurrence libre

Syntaxe du langage B

- Les notations sont étendues par l'introduction des définitions suivantes :
- $P \vee Q$ est défini par $\neg P \Rightarrow Q$
- $P \Leftrightarrow Q$ est défini par $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$
- $\exists x.P$ est défini par $\neg \forall x.P$
- Par conséquent les occurrences de variables concernées par le quantificateur $\exists$ne sont pas libres non plus.
- Les parenthèses ne font pas vraiment partie du langage mais servent à désambiguïser les prédicats ou expressions ou à forcer les priorités qui ont des niveaux définis par défauts (voir manuel de référence du langage).