Mathématiques appliquées à l'informatique
3 – Logique

Bertrand LIAUDET

SOMMAIRE

Dernière édition : mars 2017

LOGIQUE DES PROPOSITIONS

Références

Mathématique pour l'informatique – BTS SIO – Dunod – 2015 : Chapitre 1, pp. 3-33.

Méthodes mathématiques pour l'informatique – IUT-Licence-Ecole d'ingénieurs-CNAM – Dunond 2013.

Notions de Logique – Philippe Thiry – De Boeeck Université – 1997

Proposition et logique d'ordre 0

Définitions

Proposition : énoncé ayant un sens et une valeur de vérité

Une proposition est un énoncé **ayant un sens** et qui est **vrai ou faux.**

« Bonjour » et « 8 » ne sont pas des propositions.

« Il fait beau », « 8 = 4*2 » , « 10 = 5+6 » sont des propositions. La dernière est fausse !

« Je vais gagner à la loterie » n'est pas une proposition : c'est un futur et c'est peu prédictible même si c'est très probablement faux.

« x <- 3 » n'est pas une proposition : c'est une affectation (ou assignation), c'est-à-dire une action. Une action n'est pas une proposition, elle n'a pas de valeur de vérité. Mais elle a été réalisée ou pas.

x=3, en tant qu'expression mathématique, n'est pas une proposition. En effet, x n'est pas défini. On ne sait donc pas de quoi on parle et on ne peut donc pas dire si c'est vrai ou faux.

x=0 et x=9 n'est pas une proposition. Elle semble fausse quel que soit x, mais à condition de définir x comme étant par exemple un réel. Avec une autre définition, on pourrait concevoir que ce soit vrai (par exemple quand on fait une preuve par 9).

On verra avec la logique des prédicats comment transformer ces énoncés en propositions.

Notation

p, q, P, Q

- ***Exemples***

On peut écrire :

p= « il fait beau »

q = « il fait beau et il y a du vent »

Proposition atomique et proposition complexe

Une proposition atomique est vraie ou fausse et ne comporte pas de parties qui soit vraies ou fausse.

Une proposition complexe est vraie ou fausse et comporte des parties qui sont vraies ou fausses.

➢ ***Exemples***

« Il fait beau » : est une proposition atomique.

« Le ciel bleu » n'est pas une proposition.

« Le soleil brille et il y a du vent » est une proposition complexe constituée de 2 propositions simples : « le soleil brille » et « il y a du vent ».

Utilisation en informatique

➢ ***Expression booléenne***

En informatique, les expressions booléennes sont des propositions :

Dans le code suivant :

```
a=-5;
if(a<0):
    print("négatif");
```

L'expression (a<0) est une expression booléenne : c'est une proposition qui est vraie ou fausse.

Elle est évaluée : selon la valeur de a, elle est vraie ou fausse et on fait ou pas le travail.

Soit le code suivant :

```
n=0;
while n<10:
    n=n+1;
    print(n);
```

L'expression (n<10) est une expression booléenne : c'est une proposition.

Elle est évaluée : selon la valeur de N, elle est vraie ou fausse et on fait ou pas le travail.

Table de vérité

Une table de vérité est un tableau dans lequel on met en colonne des propositions et en ligne leur valeur de vérité possibles : vrai ou faux. Ces tables seront utilisés avec les connecteurs logiques.

Il fait beau
Vrai
Faux

Table de vérité et algèbre booléenne

Dans l'algèbre booléenne :

1 = vrai

0 = faux

La table de vérité devient :

Il fait beau
1
0

Logique des propositions : logique d'ordre 0

La logique des propositions s'occupe de propositions et des opérations que l'on peut effectuer sur ces propositions.

Elle se nomme aussi logique d'ordre 0.

Remarque épistémologique : la falsifiabilité

Pour Popper, une proposition dont on dit qu'elle est vraie est scientifique si elle peut être réfutée, c'est-à-dire si on peut prouver sa négation. On dit qu'une proposition scientifique soit être « falsifiable ». Les propositions qui ne sont pas falsifiables sont des croyances.

Par exemple : « Le Mont Blanc est la plus haute montagne de France » est une proposition scientifique. Pour prouver le contraire, il suffit de trouver une montagne plus haute.

Autre exemple : « les extraterrestres n'existent pas » est une proposition qui peut être vraie ou fausse. Si on considère qu'elle est vraie, alors son contraire est « les extraterrestres existent ». Cette proposition est « facilement » prouvable : il suffit d'en trouver un ! Comme si je disais : « il n'existe pas de dragon ». Jusqu'à preuve du contraire, c'est vrai et scientifique !

Par contre, si on affirme » les extraterrestres existent », on affirme une proposition qui n'est pas scientifique et qui est donc une croyance. En effet, comment prouver le contraire ? Comment prouver qu'ils n'existent pas ? Il faudrait parcourir l'univers tout entier dans tous ses recoins ! Impossible.

Donc la seule proposition scientifique possible est « les extraterrestres n'existent pas » au moins jusqu'à preuve du contraire.

Exercices

- ***Quels sont les énoncés qui sont des propositions :***
  - Je crois aux extraterrestres.
  - Il a dit « à demain » et il est sorti
  - Faites les exercices !
  - Il peut et il ne pleut pas
  - La lune est un morceau de gruyère
  - La plus vielle maison de Paris
  - S'il pleut alors il ne pleut pas

- ***Dans les énoncés précédents :***
  - Quelles sont les propositions complexes ? Ecrivez les sous la forme : p1= « … »
  - Ecrivez les propositions atomiques constituants les propositions complexes sous la forme : a= « … »

- ***Codez la boucle ci-dessous en python***

```
n=0;
while n<10:
    n=n+1;
    print(n);
```

- ***En python, affichez la valeur de vérité de :***
  - 3>5,
  - 2==4-2,
  - 2>4,
  - 2==2+2.

Les opérateurs logiques

Présentation

Il existe de nombreux opérateurs logiques (opérateurs ou connecteurs).

Par exemple, à partir de la proposition

p = « il fait beau »

on peut construire sa négation :

non(p) = « il ne fait pas beau ».

A partir d'une deuxième proposition :

q = « il y a du vent »

on peut construire la conjonction des deux propositions :

r = « il ne fait pas beau et il y a du vent »

Cette dernière proposition pourra s'écrire en formalisme logique :

r = non(p) et q

On va étudier les opérateurs suivants :

Négation

Equivalence

Et

Ou

Ou exclusif

Implication.

On sera amené à faire des tables de vérité pour chaque proposition, atomique ou complexe.

Négation : ¬P

Formalisme

non(P), not(P), !P, ¬P, $\overline{P}$

Langage naturel

P= « il fait beau »

nonP = « il ne fait pas beau »

nonP, c'est la négation : « ne … pas »

Proposition atomique et proposition complexe

Une proposition négative est toujours une proposition complexe.

« Il ne fait pas beau » est une proposition complexe.

« Il fait beau » est une proposition atomique.

Table de vérité

La table de vérité permet de calculer la valeur de vérité pour une opération donnée (ici la négation) en partant des valeurs possibles pour la proposition de départ.

Valeur de vérité	
P	non (P)
Vrai	Faux
Faux	Vrai

Algèbre booléenne	
P	non (P)
1	0
0	1

Exemples

P : « PI est un entier » non P : « PI n'est pas un entier »

Q : 3 > 4 non Q : 3 <= 4

Code

```
a=2>3;      a;
b= not a;   b;
```

Conjonction : P et Q

Formalisme

P et Q, P and Q, $P \wedge Q$, P*Q, P.Q, PQ, $P \cap Q$

Table de vérité

Valeur de vérité		
P	Q	P et Q
Vrai	Vrai	Vrai
Vrai	Faux	Faux
Faux	Vrai	Faux
Faux	Faux	Faux

Algèbre booléenne		
P	Q	P et Q
1	1	1
1	0	0
0	1	0
O	0	0

Langage naturel

P et Q veut dire les deux en même temps. Si P et Q sont des ensembles (les tigres et les lions), les deux en même temps veut dire l'intersection des deux ensembles.

Exemples

« x > 0 et x <10 » est vrai ou pas en fonction de x.

« 3 >2 et 5 >6 » est vrai

« 2 >3 et 5>6 » est faux

Il fait beau et il fait chaud

Il fait chaud et il y a du vent

Proposition atomique et proposition complexe

Une proposition conjonctive est toujours une proposition complexe.

p = « Il fait chaud et il y a du vent » est une proposition complexe.

p est constitué de 2 propositions atomiques :

q = « Il fait chaud»

r = « Il y a du vent »

p =q et r

Exercices

1. Tester les cas ci-dessus en python.

Disjonction : P ou Q

Formalisme

P ou Q, P or Q, P∨Q, P+Q, P ∪ Q

Table de vérité

Valeur de vérité		
P	Q	P ou Q
Vrai	Vrai	Vrai
Vrai	Faux	Vrai
Faux	Vrai	Vrai
Faux	Faux	Faux

Algèbre booléenne		
P	Q	P ou Q
1	1	1
1	0	1
0	1	1
O	0	0

Langage naturel

P ou Q veut dire l'un ou l'autre ou les deux en même temps. Si P et Q sont des ensembles (les tigres et les lions), l'un ou l'autre veut dire l'union des deux ensembles.

Exemples

« x > 10 ou x <0 » est vrai ou pas en fonction de x.

« 3 >2 ou 7 >6 » est vrai

« 3 >2 ou 5 >6 » est vrai

« 3 >4 ou 5 >6 » est faux

Il fait beau ou il pleut

Il fait chaud ou il y a du vent

Proposition atomique et proposition complexe

Une proposition conjonctive est toujours une proposition complexe.

p = « Il fait chaud ou il pleut» est une proposition complexe.

p est constitué de 2 propositions atomiques :

q = « Il fait chaud»

r = « Il pleut»

p =q ou r

Exercices

1. Tester les cas ci-dessus en python. Pour la première proposition, tester tous les cas en python.

Table de vérité et proposition complexe

Présentation

Une table de vérité permet d'analyser la valeur de vérité d'une proposition complexe en partant des différents cas possibles pour chaque proposition atomique.

Exemple

P = Il fait beau et il ne pleut pas

Q= il fait beau

R = il pleut

P = Q et nonR

- ***Table de vérité***

Valeur de vérité			
Q	R	nonR	P = Q et nonR
Vrai	Vrai	Faux	Faux
Vrai	Faux	Vrai	Vrai
Faux	Vrai	Faux	Faux
Faux	Faux	Vrai	Faux

Algèbre booléenne			
Q	R	nonR	P = Q et nonR
1	1	0	0
1	0	1	1
0	1	0	0
0	0	1	0

Equivalence : P ⇔Q

Formalisme

P ⇔ Q

Table de vérité

Valeur de vérité		
P	Q	P ⇔ Q
Vrai	Vrai	Vrai
Vrai	Faux	Faux
Faux	Vrai	Faux
Faux	Faux	Vrai

Algèbre booléenne		
P	Q	P ⇔ Q
1	1	1
1	0	0
0	1	0
O	0	1

Langage naturel

P ⇔ Q veut dire que quand on a l'un, on à l'autre et quand on n'a pas l'un on n'a pas l'autre non plus.

Les deux propositions vont ensemble.

Ca peut se rapprocher de l'égalité : dire que deux proposition sont équivalentes, c'est comparer leur valeur de vérité et constater qu'elles sont identiques dans tous les cas.

Exemples

- « 4x - 8 = 0 <=> x = 2 » est vrai : je peux passer d'une proposition à l'autre
- « 4x – 8 = 0 <=> x = 1 » est faux : je ne peux passer d'une proposition à l'autre
- « 4 + 4 = 8 <=> 1 < 2 » est vrai : je peux passer d'une proposition à l'autre, même si ça semble contre intuitif. Du vrai je peux toujours passer à du vrai.
- « 4 + 4 = 9 <=> 1 < 2 » est faux : du faux, je ne peux pas passer à du vrai.
- « La Terre est un cube <=> Les chats ont 6 pattes» est vrai : du faux je peux toujours passer à du faux. C'est très contre-intuitif puisqu'on peu

Exercices

1. Tester les cas ci-dessus en python. Par exemple pour le premier cas : x=2 ; (4*x - 8==0)==(x==2). Tester en changeant la valeur de a.
2. A l'aide d'une table de vérité, démontrer le principe de la double négation : P ⇔ non(non(P))

Ou exclusif : P xor Q

Formalisme

P xor Q, P ⊕ Q

Table de vérité

Valeur de vérité		
P	Q	P xor Q
Vrai	Vrai	Faux
Vrai	Faux	Vrai
Faux	Vrai	Vrai
Faux	Faux	Faux

Algèbre booléenne		
P	Q	P xor Q
1	1	0
1	0	1
0	1	1
O	0	0

Langage naturel

P xor Q veut dire l'un ou l'autre mais pas les deux, comme dans l'option « fromage ou dessert » dans certains menus de restaurant.

Exemples

« x > 0 xor x < 10 » est vrai en fonction de x et y

« x >= 0 xor x< 0 » est vrai

« 3 >2 xor 6 >5 » est faux

« 3 >3 xor 5>6 » est vrai

Exercices

1. Tester les cas ci-dessus en python. Pour la première proposition, tester tous les cas en python. En python, on utilise l'opérateur « ^ » pour faire un xor à condition de mettre chaque partie autour du « ^ » entre parenthèses.

Implication : =>

Formalisme

P implique Q, Si P alors Q, P⇒Q, P => Q

Table de vérité

Valeur de vérité		
P	Q	P⇒Q
Vrai	Vrai	Vrai
Vrai	Faux	Faux
Faux	Vrai	Vrai
Faux	Faux	Vrai

Algèbre booléenne		
P	Q	P⇒Q
1	1	1
1	0	0
0	1	1
O	0	1

Langage naturel : implication logique et implication physique

Dans le langage naturel, l'implication se retrouve avec les mots « implique », « donc », « si… alors ».

L'interprétation la plus simple est de dire : « **P donc Q** » ou « **si P alors Q** ».

Elle se traduit par le fait qu'on ne peut pas avoir P sans avoir Q : non (P et non Q).

Par exemple : « **Le temps est sec donc il ne pleut pas** » est vrai. On ne peut pas avoir un temps sec et de la pluie.

L'implication en langage naturel est compliquée à comprendre car le langage naturel mélange l'**implication logique** (l'un en même temps que l'autre) et l'implication comme conséquence chronologique (l'un après l'autre), qu'on peut appeler « **implication physique** » ou « causalité ».

L'implication physique dit que si la pomme se décroche de l'arbre, alors elle va tomber par terre. C'est celle des lois de la physique et de la chimie.

« Le temps est sec donc il ne pleut pas » est une implication logique. C'est vrai dans ce sens, c'est faux dans l'autre.

« si je cours vite longtemps alors je serai fatigué » est une implication comme conséquence chronologique.

L'implication est sujette à beaucoup de discussions car il arrive souvent qu'on prenne les causes pour des conséquences et les conséquences pour des causes.

Doit-on dire **« je pense donc je suis » ou « je suis donc je pense » ?** C'est la première qui est vrai.

Propriétés issues du langage naturel

- ***P => Q <=> nonP ou Q***

  P => Q <=> on ne peut pas avoir P et non Q

  <=> non (P et nonQ)

  <=> nonP ou Q

- ***La contraposé : P=>Q ⇔ nonQ => nonP***

  Je parle donc j'existe

  ⇔ je n'existe pas donc je ne parle pas ⇔ nonQ => nonP

Du faux, on peut tout déduire

Le faux implique le vrai !

C'est étrange mais si on cherche à changer les valeurs de la table de résultats, on a trois options dont aucune n'est satisfaisante :

P	Q	P⇒Q	Option1 = et	Option2= ⇔	Option3 : empêche la contraposé
1	1	1	1	1	1
1	0	0	0	0	0
0	1	1	0	0	1
0	0	1	0	1	0

Exercices

1. Démontrer l'équivalence de la contraposition sans passer par les tables de vérité.
2. A l'aide d'une table de vérité, démontrer que P => Q ⇔ non(P ou Q)
3. A l'aide d'une table de vérité, démontrer que P => Q ⇔ nonQ => nonP
4. Traduisez en propositions et implications les phrases suivantes :
   - je pense donc je suis.
   - Le temps est sec donc il ne pleut pas
   - Si je cours vite longtemps alors je serai fatigué
   - Le cours se déroule si le professeur est arrivé.
   - Si l'accusé est coupable il n'a pas d'alibi.
   - Pour que les étudiants réussissent, il faut qu'ils étudient.
5. Dans les deux derniers cas, faites le tableau de vérité et interpréter tous les cas.

<u>Spéculations (facultatif !) : relation entre causalité logique et causalité physique</u>

- ***Physique => Logique***

  Soit P = penser et E = être

  P => E est une implication logique

  E -> P est une implication physique. On utilise le « -> » pour distinguer les 2 implications. Ce formalisme n'est pas standard.

  La « -> » peut se comprendre comme « est une condition nécessaire mais pas forcément suffisante de ». L'Etre est une condition de la Pensée.

  E -> P, c'est l'un ensuite l'autre. Donc E_{t1} -> P_{t2} avec t2 > t1

  Donc, si j'ai P, j'ai forcément E (ou au moins je l'avais).

  On arrive donc à la propriété : (E -> P) => (P => E)

- ***Extrapolations***

  On pourrait avoir envie de passer d'une => à une <=> : (E -> P) <=> (P => E)

  Cette hypothèse est plus spéculative.

  Il faut noter que dans ce cas, par la contraposé, on aura aussi :

  (nonE => nonP) <=> (nonP -> nonE)

  Ce qu'on formuler ainsi : l'absence de pensée est une condition nécessaire à l'absence d'être.

  C'est compliqué, mais c'est plutôt vrai.

  [réflexions chez les logiciens]

- ***Autres exemples***

  Soit M = manger et F = avoir faim

  M => F (en supposant qu'on ne mange pas sans faim !)

  On en déduit que F -> M : la faim est une condition nécessaire au fait de manger (en supposant toujours qu'on ne mange pas sans faim.

  Soit S = le temps est sec et P = il pleut.

  S => nonP

  On ne déduit que nonP -> S : l'absence de pluie est une condition nécessaire à un temps sec.

- ***Exercices***

  Chercher l'implication physique sur les autres exemples ci-dessus.

  Chercher l'implication physique par la contraposé et l'équivalence entre logique et physique.

  Donner du sens aux résultats trouvés. Bonne chance ! ☺

Généralisation : les 16 opérateurs binaires

Quand on fait la table de vérité d'une opération binaire, on obtient un résultat qu'on peut lire comme un nombre binaire à 4 chiffre : P et Q = 1 0 0 0.

Valeur de vérité		
P	Q	P et Q
1	1	1
1	0	0
0	1	0
0	0	0

Il y a donc autant de résultats possibles que de codages possibles sur 4 binaires = 2^4 = 16

On peut donc faire le tableau des résultats pour les 16 cas possibles :

P	Q		et		P			xor	ou		⇔			Q	=>		
P	Q	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Cela nous montre qu'il y a beaucoup d'autres opérateurs que ceux qui sont intuitifs pour nous. On pourrait imaginer des théories sur et avec ces opérateurs.

Priorité des opérateurs

Les parenthèses sont prioritaires sur les opérateurs.

L'opérateur unaire de négation est prioritaire sur les opérateurs binaires

L'opérateur « et » est prioritaire sur l'opérateur « ou »

Les opérateurs de logique propositionnelle (et, ou, implique, etc.) sont prioritaires sur les opérateurs de comparaison arithmétique (==, >, >, etc)

Exercices

1. Que vaut False and False or True ?
2. Que vaut False and (False or True) ?
3. Que vaut 4==4 and True ?
4. Que vaut 4== (4 and True) ?
5. Testez les expressions ci-dessus en python.

Exercices récapitulatifs

Traduire en langage symbolique les expressions suivantes :

- Le monde n'a ni commencement ni fin.
- Ce quadrilatère est un carré seulement si c'est un losange.
- Si le règlement est très strict mais est le même pour tout le monde alors les étudiants ont tort de le contester.
- Qu'il pleuve ou non, je prend mon parapluie.

Propriétés, démonstration et raisonnement

Propriétés : tautologies

Les propriétés sont des expressions logiques qui sont toujours vraies.

On les appelle des « tautologies ».

Tableaux des principales propriétés :

Principe d'identité	P ⇔ P
Principe de non contradiction	non (P ⇔ non(P))
Double négation	P ⇔ non(non(P))
Commutativité du « et »	P et Q ⇔ Q et P
Commutativité du « ou »	P ou Q ⇔ Q ou P
Associativité du « et »	(P et Q) et R ⇔ P et (Q et R)
Associativité du « ou »	(P ou Q) ou R ⇔ P ou (Q ou R)
Distributivité du « ou » sur le « et »	P ou (Q et R) ⇔ (P ou Q) et (P ou R)
Distributivité du « et » sur le « ou »	P et (Q ou R) ⇔ (P et Q) ou (P et R)
Loi de Morgan	non(P et Q) ⇔ non(P) ou non(Q)
Loi de Morgan	non(P ou Q) ⇔ non()P et non(Q)
Implication et disjonction	(P ⇒ Q) ⇔ non (P et nonQ) ⇔ nonP ou Q

Démonstration des propriétés

Pour démontrer les propriétés, on utilise un tableau de vérité et on calcule la valeur de vérité de la propriété à démontrer.

Dans tous les cas pour les propositions de départ, la tautologie doit être vraie.

non(P ou Q) ⇔ non(P) et non(Q)

- ***Démonstration :***

P	non (P)	non(non(P))	**P ⇔ non(non(P))**
Vrai	Faux	Vrai	Vrai
Vrai	Faux	Vrai	Vrai
Faux	Vrai	Faux	Vrai
Faux	Vrai	Faux	Vrai

Distributivité du « ou » sur le « et »

T = P ou (Q et R) ⇔ (P ou Q) et (P ou R)

- ***Démonstration :***

P	Q	R	Q et R	P ou (Q et R)	P ou Q	P ou R	(P ou Q) et (P ou R)	**T**
Vrai	Vrai	Vrai	Vrai	**Vrai**	Vrai	Vrai	**Vrai**	**Vrai**
Vrai	Vrai	Faux	Faux	**Vrai**	Vrai	Vrai	**Vrai**	**Vrai**
Vrai	Faux	Vrai	Faux	**Vrai**	Vrai	Vrai	**Vrai**	**Vrai**
Vrai	Faux	Faux	Faux	**Vrai**	Vrai	Vrai	**Vrai**	**Vrai**
Faux	Vrai	Vrai	Vrai	**Vrai**	Vrai	Vrai	**Vrai**	**Vrai**
Faux	Vrai	Faux	Faux	**Faux**	Vrai	Faux	**Faux**	**Vrai**
Faux	Faux	Vrai	Faux	**Faux**	Faux	Vrai	**Faux**	**Vrai**
Faux	Faux	Faux	Faux	**Faux**	Faux	Faux	**Faux**	**Vrai**

Exercices

- Démontrer les lois de Morgan
- Démontrer l'équivalence entre implication et disjonction : P => Q <=>   nonP ou Q

Forme normale disjonctive

Définition

On appelle proposition conjonctive (PC) une proposition constituée de propositions atomiques (atome) ou de leurs négations reliées entre elles uniquement par des conjonctions (et).

« p et non q et r » est une proposition conjonctive.

On appelle forme normale disjonctive (FND) une proposition constituée de propositions conjonctives (PC) reliées entre elles uniquement par des disjonctions (ou).

« (p et non q et r) ou (non p et q) » est une FND.

Une FND ne doit pas présenter un atome (p, q, etc.) et sa négation dans une PC.

« p ou (non p et q) » n'est pas une FND car p est une proposition atomique et non p se trouve dans une PC. La FND correspondante est : « p ou q ». Il suffit d'appliquer la distributivité du ou sur le et pour arriver au résultat.

Propriété

Toute proposition peut s'écrire sous la forme d'une FND.

Méthode pour construire une FND

Traduire tous les connecteurs en « non », « et » et « ou ».

Eliminer les doubles négations.

Utiliser les lois de Morgan pour que les négations portent sur des atomes.

Utiliser les lois de la distributivité pour « sortir » les « ou » des parenthèses et « entrer » les « et » dans les parenthèses.

Propriétés utiles pour construire une FND

P ou (Q et nonQ) ⇔ P ou Faux ⇔ P

P et (Q ou nonQ) ⇔ P et Vrai ⇔ P

a=>b en FND

a=>b ⇔ non(a) ou b

a xor b en FND

a xor b ⇔ (non(a) et b) ou (a et non(b)) : ce n'est pas intuitif.

On peut partir de la table de vérité d'un « ou » et se demander ce qu'il faudrait faire comme opération pour obtenir le résultat.

On peut partir d'une première opération : P ou Q.

Il reste à annuler la première ligne.

On cherche (P ? Q) tel que (P ou Q) et (P ? Q) = P xor Q

P	Q	P ou Q	P ? Q	(P ou Q) et (P ? Q) = P xor Q
1	1	1	0	0
1	0	1	1	1
0	1	1	1	1
0	0	0	1	0

On déduit la table de vérité de P ? Q

On compare ensuite P ? Q aux opérations possibles « et » et « ou » et on constate que (P ?Q) c'est non (P et Q)

P	Q	P ou Q	P et Q	P ? Q = non (P et Q)
1	1	1	1	0
1	0	1	0	1
0	1	1	0	1
0	0	0	0	1

On a donc P xor Q = (P ou Q) et non (P et Q)

Il reste à transformer cet énoncé en FND.

(P ou Q) et non (P et Q) ⇔

(P ou Q) et (nonP ou nonQ) ⇔

(P et (nonP ou nonQ)) ou (Q et (nonP ou nonQ)) ⇔

((P et nonP) ou (P et nonQ)) ou ((Q et nonP) ou (Q et nonQ)) ⇔

(P et nonQ) ou (nonP et Q)

Vérification avec une table de vérité :

P	Q	nonP	nonQ	P et nonQ	nonP et Q	(P et nonQ) ou (nonP et Q)
1	1	0	0	0	0	0
1	0	0	1	1	1	1
0	1	1	0	0	1	1
0	0	1	1	0	0	0

Raisonnement (ou syllogisme)

Le modus ponens : (p+ p=>q) => q

Un raisonnement, une méthode pour produire une proposition à partir de 2 autres propositions.

La forme de base de tout raisonnement est la suivante :

(p + p=>q) => q

p=>q est appelée : proposition majeure ou Majeure

p est appelée : proposition mineure ou Mineure

q est appelée : Conclusion

La forme de base (p + p=>q) => q est appelée : « modus ponens » ou encore « règle du détachement » : c'est une tautologie.

Remarque

Etant donné qu'on utilise des implications logiques, la difficulté sera de ne pas les confondre avec des implications physiques et donc de ne pas confondre les causes et les conséquences !

Dérivés

Il existe de nombreuses dérivés à la forme de base.

- ***Dérivé par la contraposé : le modus tollens : (non q + p => q) => non p***

Par la contraposé, on a : p=>q <=> non q => non p

donc (non q + non q => non p) => non p

donc (non q + p => q) => non p

« modus tollens ».

- ***Raisonnement par l'absurde : (p => q) <=> non (p + non q)***

Le raisonnement par l'absurde consiste à vérifier que p + non q est faux

Exercices

Démontrer la validité des raisonnements. Commencer par exprimer les propositions atomiques qu'ils recèlent, puis les propositions complexes et faites les déductions.

- Il suffit qu'il pleuve pour que l'escargot sorte, or l'escargot sort. Donc il pleut.
- Pour ne pas rater cet exercice, il faut garder la tête froide. Or vous ne perdez pas la tête. Donc vous ne raterez pas cet exercice.
- Si mon raisonnement est valide, je ne raterai pas mon exercice. Or ou bien je rate mon exercice, ou bien la logique est facile. Et précisément la logique n'est pas facile, donc mon raisonnement n'est pas valide.

Applications aux circuits booléens

Tableaux de Karnaugh

Utilité

Ces tableaux permettent de produire des formes normales disjonctives (FND) plus facilement.

Tableau à 2 variables : a et b

Le tableau a 4 cases. On croise les valeurs de vérité de a et de b.

	b	$\overline{b}$
a	$a\,b$	$a\,\overline{b}$
$\overline{a}$	$\overline{a}\,b$	$\overline{a}\,\overline{b}$

	b	$\overline{b}$
a	$a\,b$	$a\,\overline{b}$
$\overline{a}$	$\overline{a}\,b$	$\overline{a}\,\overline{b}$

$a + \overline{b}$

Quand on a une expression, par exemple : $ab + \overline{b}$, on colorie la case du ab et la colonne du $\overline{b}$.

A partir de là, on analyse le résultat graphique pour produire la FND la plus simple. Ici on voit que la ligne du a est pleine ainsi que la colonne du b, donc on peut écrire :

$ab + \overline{b} = a + \overline{b}$

On aurait pu trouver ce résultat par calcul, mais ça va plus vite avec le tableau.

Par calcul :

$ab + \overline{b}$

$\Leftrightarrow (a + \overline{b})$ et $(b + \overline{b})$

$\Leftrightarrow (a + \overline{b})$ et $(b + \overline{b})$

$\Leftrightarrow (a + \overline{b})$ et Vrai

$\Leftrightarrow (a + \overline{b})$

Tableau à 3 variables : a, b et c

Le tableau aura 8 cases : on croise les valeurs de vérité de a avec celles des couples possibles entre b et c : $b\,c$, $b\,\overline{c}$, $\overline{b}\,\overline{c}$, $\overline{b}\,c$ dans cet ordre là par convention.

	$b\,c$	$b\,\overline{c}$	$\overline{b}\,\overline{c}$	$\overline{b}\,c$
a	$a\,b\,c$	$a\,b\,\overline{c}$	$a\,\overline{b}\,\overline{c}$	$a\,\overline{b}\,c$
$\overline{a}$	$\overline{a}\,b\,c$	$\overline{a}\,b\,\overline{c}$	$\overline{a}\,\overline{b}\,\overline{c}$	$\overline{a}\,\overline{b}\,c$

La première ligne correspond à « a », la deuxième à « non a »

Les deux premières colonnes correspondent à « b »

Les deux dernières colonnes correspondent à « non b »

Les deux colonnes du milieu correspondent à « non c »

La première et la dernière colonne correspondent à « c »

- ***Utilisation***

  Soit l'expression : non(a)b + abc + non(c)b + non(a)non(b)

  On remplit le tableau en conséquence :

	bc	$b\bar{c}$	$\bar{b}\bar{c}$	$\bar{b}c$
a	abc	$ab\bar{c}$	$a\bar{b}\bar{c}$	$a\bar{b}c$
$\bar{a}$	$\bar{a}bc$	$\bar{a}b\bar{c}$	$\bar{a}\bar{b}\bar{c}$	$\bar{a}\bar{b}c$

Quelle FND peut-on produire à partir du tableau ?

On a la ligne du non(a) et les colonnes du (b)

Donc

non(a)b + abc + non(c)b + non(a)non(b)

⇔ non(a) + b

<u>Tableau à 4 variables : a, b, c et d</u>

Le tableau aura 16 cases : on croise les valeurs de vérité de du couple ab avec celles du couple cd.

L'utilisation est la même que pour un tableau à 3 variables.

<u>Exercices</u>

Soit A = ab + non(c) et B=non(a) + bc

Déterminer la FND et non(A) et de non(B) en utilisant un tableau de Karnaugh, puis sans tableau.

Soit A=non(c)a+non(b)c+abd et B=non(a)b+non(b)c+non(c)a

Déterminer la FND et non(A) et de non(B) en utilisant un tableau de Karnaugh, puis sans tableau.

Karnaugh – Sudoku

Trouver la composition de la bibliothèque, sachant qu'elle contient :

- 52 livres d'histoire dont 27 en anglais,
- 34 livres reliés dont 3 d'histoire en français,
- 46 livres en anglais, la moitié étant brochée,
- 20 livres de littérature en français, et
- 31 livres brochés de littérature.

Circuit à relais

L'intérêt de la FND et du tableau de Karnaugh est de réaliser plus facilement des circuits électroniques.

Notion de relais

Les relais peuvent être ouverts ou fermés et ainsi laisser ou non passer le courant.

Un relais ouvert vaut 1. Un relais fermé vaut 0.

Circuit ET

Le courant circule de l'entrée E vers la sortie S en passant par 2 relais A et B.

```
E --------------------- A ----------------------- B -----------------------S
```

Le courant passe si A et B sont ouverts. La fonction du circuit est f(A,B) = A et B

Circuit OU

Le courant circule de l'entrée E vers la sortie S en passant par 2 relais A et B.

```
                      |--------------A ------------|
E --------------------- |                            |-----------------------S
                      |--------------B ------------|
```

Le courant passe si A ou B sont ouverts. La fonction du circuit est f(A,B) = A ou B

Circuit complexe en OU

```
                      |-----A-------------non B----|
E --------------------- |--A----------B---------C----|-----------------------S
                      |-------A------------non A---|
```

La fonction du circuit est f(A,B,C) = (A et non B) ou (A et B et C) ou (A et non A)

Circuit complexe en ET

```
                      |--A--|    |-----B----|    |--non A--|
E --------------------- |     |-----|          |------|--non B--|-----------------------S
                      |--C--|    |--non C--|    |-----C-----|
```

La fonction du circuit est f(A,B,C) = (A ou C) et (B ou non C) et (non A ou non B ou C)

Simplification

L'objectif est de fabriquer un circuit le plus simple possible.

Dans nos 2 exemples complexes, on peut simplifier le circuit. Pour ça, il suffit de trouver la FND correspondant à la fonction du circuit.

Pour cela, on peut passer par un tableau de Karnaugh.

Ensuite, on peut reconstruire le circuit de façon simplifiée.

Exercice

Simplifier les deux circuits complexes présentés ci-dessus.

Circuit électronique

Circuit de l’additionneur bit a bit

On additionne x et y ainsi qu’un report d’addition : c

Le résultat vaut Z, la retenue r

L’objectif est de construire le circuit permettant de faire cette addition.

Table de vérité l’additionneur 1 bit

Entrées **Sorties**

x	**y**	**c**	**Z**	**r**
1	1	1	1	1
1	0	1	0	1
0	1	1	0	1
0	0	1	1	0
1	1	0	0	1
1	0	0	1	0
0	1	0	1	0
0	0	0	0	0

$Z = (x\ y\ c) + (\neg x\ \neg y\ c) + (x\ \neg y\ \neg c) + (\neg x\ y\ \neg c)$

$r = (x\ y\ c) + (x\ \neg y\ c) + (\neg x\ y\ c) + (x\ y\ \neg c)$

FND de Z et de r

$r = (x\ y\ c) + (x\ \neg y\ c) + (\neg x\ y\ c) + (x\ y\ \neg c)$

	$x\,y$	$x\,\bar{y}$	$\bar{x}\,\bar{y}$	$\bar{x}\,y$
c	$c\,x\,y$	$c\,x\,\bar{y}$	$c\,\bar{x}\,\bar{y}$	$c\,\bar{x}\,y$
$\bar{c}$	$\bar{c}\,x\,y$	$\bar{c}\,x\,\bar{y}$	$\bar{c}\,\bar{x}\,\bar{y}$	$\bar{c}\,\bar{x}\,y$

r= xy + xc + yc

$Z = (x\ y\ c) + (\neg x\ \neg y\ c) + (x\ \neg y\ \neg c) + (\neg x\ y\ \neg c)$

	$x\,y$	$x\,\bar{y}$	$\bar{x}\,\bar{y}$	$\bar{x}\,y$
c	$c\,x\,y$	$c\,x\,\bar{y}$	$c\,\bar{x}\,\bar{y}$	$c\,\bar{x}\,y$
$\bar{c}$	$\bar{c}\,x\,y$	$\bar{c}\,x\,\bar{y}$	$\bar{c}\,\bar{x}\,\bar{y}$	$\bar{c}\,\bar{x}\,y$

Cette configuration du tableau de Karnaugh correspond à une situation particulière qu’on n’a pas abordée : le ou exclusif

$Z = x \oplus y \oplus c$

- ***Exercice***

Prouver avec une table de vérité que :

$x \oplus y \oplus c \Leftrightarrow$

$(x\ y\ c)\ +\ (\neg x\ \neg y\ c)\ +\ (x\ \neg y\ \neg c)\ +\ (\neg x\ y\ \neg c)$

Transformation supplémentaire

$r = xy + xc + yc$

$\Leftrightarrow r = xy + c\ (x + y)$

$\Leftrightarrow r = xy + c\ (x \oplus y)$

Cette dernière équivalence n'a rien d'évident !

L'intérêt de cette formulation est qu'on trouve une opération commune entre Z et r, ce qui facilitera la construction du circuit.

- ***Exercice***

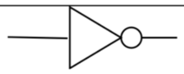
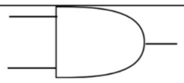
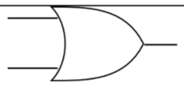
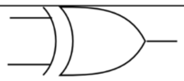
Prouver avec une table de vérité que :

$xy + c\ (x + y) \Leftrightarrow$

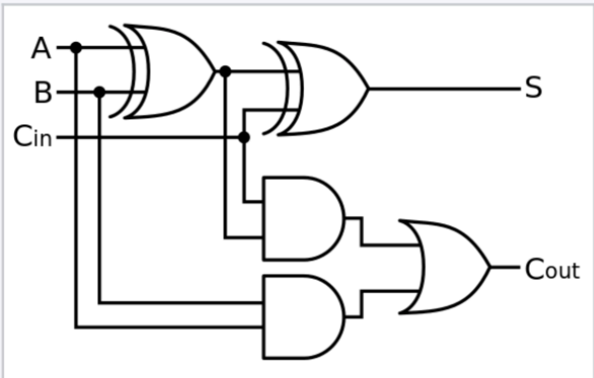
$xy + c\ (x \oplus y)$

Circuit électronique de l'additionneur 1 bit

- ***Les 4 portes logique de base des circuits électroniques***

NON(a)		$\bar{a}$
AND(a, b)		$a + b$
OR(a, b)		$a.b$
XOR(a, b)		$a \oplus b$

- ***Construction du circuit***



Un additionneur complet 1 bit.

On a :

$S = A \oplus B \oplus Cin$

$Cout = AB + Cin\ (A \oplus B)$

$\Leftrightarrow Cout = (A \oplus B)\ Cin + AB$

<u>On commence par $A \oplus B$: premier XOR</u>

$S = A \oplus B \oplus Cin$

$Cout = (A \oplus B)\ Cin + AB$

<u>Ensuite on fait $\oplus$ Cin : deuxième XOR</u>

$S = A \oplus B \oplus Cin$

<u>Ensuite on fait $(A \oplus B)$ Cin : premier ET</u>

$Cout = (A \oplus B)\ Cin + AB$

<u>Ensuite on fait AB : deuxième ET</u>

$Cout = (A \oplus B)\ Cin + AB$

<u>Enfin on fait le OU restant :</u>

$Cout = (A \oplus B)\ Cin + AB$

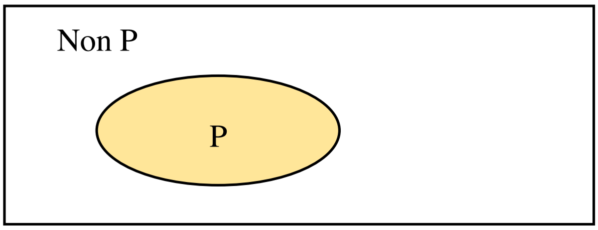
- ***Exercice***

Faire la table de vérité du circuit

LOGIQUE DES PREDICATS

Formalisme ensembliste

Le TOUT



Prédicat et logique d'ordre 1

Variables et prédicats

Proposition

Une proposition est un énoncé **ayant un sens** et qui est **vrai ou faux.**

Notion de variable

La logique des prédicats fait intervenir des variables dans les énoncés.

Une variable correspond à un objet non déterminé.

- ***Exemples***

x, dans « x est un chat », est une variable.

x, dans l'énoncé « x = 3 » est une variable.

Notion de prédicat

Le prédicat, c'est ce qu'on affirme de la variable.

En général, c'est l'attribut du sujet si le verbe est être.

Dans l'énoncé « x est un chat », un « chat », ou « chat » est le prédicat.

Dans l'énoncé « x mange », « mange » est le prédicat. On peut aussi dire « x est mangeant ».

Notion de constante

Une variable correspond à un objet non déterminé.

Une constante correspond à un objet déterminé.

« Tom est un chat » est une proposition vraie. « Tom» est une constante (le chat du dessin animé). Chat est un prédicat.

Tom est un Chat concret. On dit aussi que c'est une « instance » de Chat.

Formalisme

Les prédicats s'écrivent : P, Q, etc. En majuscule.

Les variables s'écrivent : x, y, z, etc. Fin de l'alphabet, en minuscule.

« x est un chat » s'écrit : Cx, C étant le prédicat « Chat ».

Les constantes s'écrivent : a, b, c, etc. Début de l'alphabet, en minuscule.

« Tom est un chat » s'écrit Ca, C étant le prédicat « Chat » et « a » étant l'objet Tom.

Variable mathématique (ou logique) et variable informatique

➢ ***Variable mathématique***

Une variable mathématique a les caractéristiques suivantes :

- **Elle n'est pas forcément déterminée** : quand j'écris $x^5+x^4+x^3+x^2+x=2$, il n'est pas certain que je puisse trouver la valeur de x !
- **Elle peut avoir de 0 à N valeurs** : quand j'écris $4x^2 + x + 3$, la théorie nous dit que x aura 0 valeurs (pas de solution), 1 valeur (une seule solution) ou 2 valeurs possibles (2 solutions).
- **Les valeurs possibles d'une variable ne change jamais**. Par contre, on peut diminuer le nombre de valeurs possibles. Si on dit $x^2=1$, on a deux valeurs possibles : 1 et -1. Si on ajoute $x>0$, il ne reste qu'une valeur possible : 1.

➢ ***Variable mathématique***

Une variable informatique à des caractéristiques totalement opposées !

- **Elle est toujours déterminée** : quand on déclare une variable, elle est forcément dans un état physique auquel correspond une certaine valeur selon l'interprétation qu'on en a (c'est-à-dire selon son type). Même la valeur « undef » correspond à une valeur possible !
- **Elle ne peut avoir qu'une seule valeur** : quand variable est dans un état unique qui correspond à sa valeur unique.
- **La valeur d'une variable change à chaque affectation**. En cours de programme, on peut changer la valeur des variables.

Proposition et énoncé

« Px », « x est un chat », « x=3 » sont des énoncés.

Etant donné qu'ils incluent une variable, on ne sait pas s'ils sont vrais ou faux.

Px n'est donc pas une proposition. Pas plus que « x est un chat », ni « x=3 ».

Quantificateurs

Définition

Pour transformer un énoncé avec un prédicat en proposition, on introduit 2 quantificateurs :

- Le symbole ∀ signifie « pour tout » : c'est le quantificateur universel.
- Le symbole ∃ signifie « il existe » : c'est le quantificateur existentiel.

Propositions prédicatives

Pour passer d'un énoncé prédicatif à une proposition, on précède l'énoncé par un quantificateur associé à une variable :

« $\forall$x Px » est un proposition qui se lit « pour tout x, x est un P » ou « pour tout x, P de x est vrai »

« $\exists$x Px » est une proposition qui se lit « il existe au moins un x tel que x est un P » ou « il existe au moins un x tel que P de x soit vrai ».

Formalisme : virgules et parenthèses

Pour faciliter la lecture, on peut mettre des virgules ou des parenthèses entre les termes d'une proposition :

« $\forall$x, Px », « ($\forall$x)Px », « $\forall$x (Px) »

Exemples

On peut trouver beaucoup d'exemples qu'on veut de proposition simple et vraie avec le quantificateur existentiel :

« Il existe x tel que x est un Chat » : « $\exists$x Cx »

« Il existe x tel que x = 3 » : « $\exists$x, x=3 »

Dans ce dernier cas, il faudrait préciser : « $\exists$x, x$\in$ N et x=3 » qu'on peut aussi écrire : « $\exists$x $\in$ N, x=3 ». On pourrait aussi écrire, mais ce n'est pas l'usage : « $\exists$x, Nx et x=3 »

Il est plus difficile de trouver des propositions simples et vraies avec le quantificateur universel. On a toutefois des propriétés mathématiques simples qu'on peut exprimer :

« Pour tout x, x^2 >=0 » : « $\forall$x $\in$ N, x^2>=3 »

Construction de proposition simple : 1 seule variable, 1 seul quantificateur

Exemples

- ***Comment traduire : « x > 3,5 »***

  $\exists$x, x $\in$ R $\wedge$ x > 3,5 : « il existe au moins un x tel que x $\in$ R et x > 3,5 »

  Cette proposition est vraie

  Cette proposition s'écrit habituellement : « $\exists$x $\in$ R, x > 3,5 » et se dit : « il existe au moins un x appartenant à R tel que x > 3,5 »

- ***Comment traduire : « x^2 > x »***

  « $\forall$x, x $\in$ R => x^2 > x » c'est-à-dire :« pout tout x, si x appartient à R, alors x^2 > x »

  Cette proposition est vraie.

  Cette proposition s'écrit habituellement : « $\forall$x $\in$ R, x^2 > x » et se dit « pour tout x appartenant à R, x^2 > x »

- ***Comment traduire : certains hommes sont chauves ? avec $\exists$ et $\wedge$***

  C : prédicat Chauve, H : prédicat Homme

  $\exists$x, Hx $\wedge$ Cx : il existe x tel que x est un Homme et x est Chauve.

- ***Comment traduire : tous les hommes sont mortels ? avec ∀ et =>***

  M : prédicat Mortel, H : prédicat Homme

  $\forall x$, Hx => Mx : « pour tout x, si x est un homme alors x est mortel »

<u>Formalisme</u>

Plutôt que : « $\exists x, x \in R \wedge x > 3{,}5$ », on écrit « $\exists x \in R, x > 3{,}5$ » qui traduit : « Il existe un réel tel que x >3,5 »

Plutôt que : « $\forall x, x \in R \Rightarrow x^2 > x$ », on écrit « $\forall x \in R, x^2 > x$ » qui traduit : « pour tout x appartenant à R, $x^2 > x$ »

- ***Généralisation :***

  Soit C, le prédicat Chat, N le prédicat Noir.

  On peut écrire : « $\exists x \in C$, Nx » pour dire « il existe un chat noir ».

  De même, pout dire « tous les hommes sont mortels», on peut écrire : $\forall x \in H$, Mx

<u>Constat</u>

Le quantificateur existentiel ∃ est en général associé à une conjonction : ∧

Le quantificateur universel ∀ est en général associé à une implication : =>

Construction de propositions complexes : plusieurs variables, plusieurs quantificateurs

<u>Exemples</u>

$\forall x \in R, \exists y \in R, y > x$:

Traduction : « pour tout x appartenant à R, il existe au moins un y appartenant à R tel que x>y »

Cette proposition est vraie.

On peut aussi l'écrire ainsi :

$\forall x, \exists y, (x \in R \wedge y \in R) \Rightarrow y > x$

Traduction : « pour tout x, il existe au moins un y tel que si x appartient à R et y appartient à R, alors y >x »

<u>Règle d'écriture</u>

Tous les quantificateurs doivent être placés avant les prédicats.

<u>On peut avoir des prédicats unaires, ou binaires ou plus</u>

Prédicat Préférer : x préfère y : Pxy

Prédidat PréférerA : x préfère y à z : Pxyz

<u>Commutativité des quantificateurs</u>

$\forall x \in R, \exists y \in R, y > x \quad \Leftrightarrow \quad \exists y \in R, \forall x \in R, y > x$

Traduction : : « il existe au moins un y appartenant à R tel que, pour tout x appartenant à R, x>y »

$\forall x, \exists y, (x \in R \wedge y \in R) \Rightarrow y > x \;\Leftrightarrow\; \exists y, \forall x, (x \in R \wedge y \in R) \Rightarrow y > x$

Traduction : « il existe au moins un y tel que, pour tout x, si x appartient à R et y appartient à R, alors y >x »

Exercices

Traduisez les énoncés suivants du langage naturel au langage du calcul des prédicats

Une porte est ouverte ou fermé. On prendra : Px : x est un porte. Ox : x est ouvert. Fx : x est fermé.

Toutes les vérités sont mathématiques. On prendra : Vx : x est une vérité. Mx : x est mathématiques.

Les élèves préfèrent le cours de python au cours de logique formelle. On prendra : Ex : x est un élève. Cx : x est un cours. Pxyz : x préfère y à z.

Tout ce qui brille n'est pas en or.

Il y a des peines et des plaisirs, mais aucune peine n'est un plaisir.

Il y a des bonnes actions qui ne sont pas récompensées mais aucune mauvaise action n'est récompensée.

Déduction immédiate

Notion de déduction immédiate

La déduction immédiate, c'est ce qui permet de passer d'une proposition à une autre, sans passer par une proposition intermédiaire.

La base de la déduction immédiate, c'est le principe d'équivalence

$A \Leftrightarrow \neg(\neg A)$

Ce qui nous amène aux lois de Morgan :

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

On va donc s'intéresser à la négation des quantificateurs.

Négation des quantificateurs

Approche par le langage naturel

Quelle est la négation de « tous les hommes sont mortels » ?

« Il existe un homme qui n'est pas mortels »

Le « tous » est transformé en « il existe »

Le prédicat est nié.

Propriété

$\neg (\forall x, Px) \Leftrightarrow \exists x, \neg Px$

$\neg (\exists x, Px) \Leftrightarrow \forall x, \neg Px$

$\neg (\forall x, \exists y, Pxy) \Leftrightarrow \exists x, \forall y, \neg Pxy$: ceci étant généralisable quelque soit le nombre de quantificateurs.

Exemples

- ***C : Chat, N : Noir***

  $\neg$ ($\exists x \in C$, Nx) $\Leftrightarrow$ $\forall x \in C$, $\neg$Nx

  Dire « il est faux qu'il existe un chat noir », c'est dire « aucun chat n'est noir »

- ***$\neg$ ($\forall x \in C$, Nx) $\Leftrightarrow$ $\exists x \in C$, $\neg$ Nx***

  Dire « il est faux que tous les chats sont noirs », c'est dire « il existe un chat non noir ».

- ***H : Homme, M : Mortel***

  $\neg$ ($\forall$x, Hx => Mx) $\Leftrightarrow$ $\exists$x, Hx $\wedge$ $\neg$Mx

  Dire « il est faut que tous les hommes sont mortels « , c'est dire « il existe un homme qui n'est pas mortel ».

Contraposé

Rappel

La contraposé : P=>Q ⇔ nonQ => nonP

Je parle donc j'existe ⇔ je n'existe pas donc je ne parle pas

Contraposé et quantificateur universel

Dire « Tous les hommes sont mortels », c'est dire la même chose que « Tous les immortels sont inhumains ».

$\forall x, Hx \Rightarrow Mx \quad \Leftrightarrow \quad \forall x, \neg Mx \Rightarrow \neg Hx$

Carré logique

Le carré logique résume dans un schéma les relations et les déductions que l'on peut faire à partir des quatre types de propositions possibles :

Universelle affirmative, Universelles négatives, Particulières affirmatives et Particulières négatives.

Les diagonales représentent la contradiction : $\neg Ua \Leftrightarrow Pn$ et $\neg Un \Leftrightarrow Pa$

Attention, le contraire, ou le subcontraire, ce n'est pas la même chose que le contradictoire : contraires et subcontraires n'expriment pas la véritable contradiction.

Les termes (toujours, jamais, parfois oui, parfois non) concernent la logique temporelle.

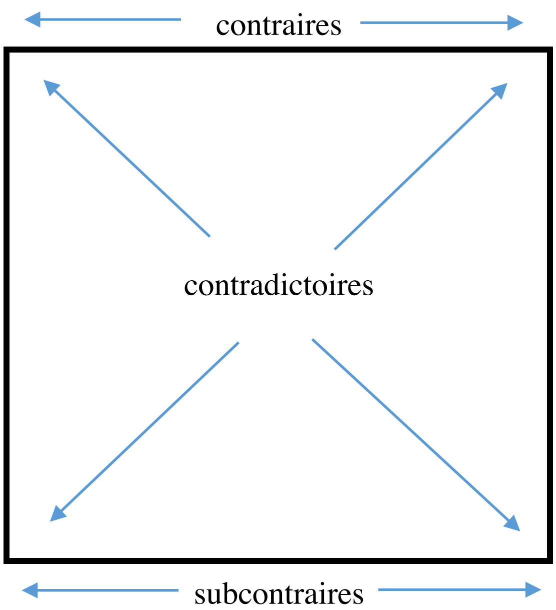
Les termes (obligatoire, interdit, permis, facultatif) concernent l'éthique.

Les termes (nécessaire, impossible, possible, contingent) concernent la modalité (le pouvoir être).

Tous ces termes sont différemment interprétés par les logiciens !

$\forall x\ (Px \Rightarrow Qx)$
Universelle affirmative
Tous les chats sont gris
Toujours
Obligatoire
Nécessaire $\Leftrightarrow$
Ne peut pas ne pas être
Codé A

$\forall x\ (Px \Rightarrow \neg Qx)$
Universelle négative
Aucun chat n'est gris
Jamais
Interdit
Impossible $\Leftrightarrow$
Ne peut pas être
Codé E



Subalternes
$Ua \Rightarrow Pa$
$Pa \subset Pb$

Subalternes
$Un \Rightarrow Pn$
$Pn \subset Un$

$\exists x\ (Px \Rightarrow Qx)$
Particulière affirmative
Certains chats sont gris
Parfois oui
Permis=Autorisé
Possible $\Leftrightarrow$
Peut être
Codé I

$\exists x\ (Px \Rightarrow \neg Qx)$
Particulière négative
Certains chats ne sont pas gris
Parfois non
Facultatif
Contingent $\Leftrightarrow$
Peut ne pas être
Codé O

Exercices

Trouvez la négation

Dans l'exercice précédent, donnez la négation des énoncés trouvés en langage du calcul des prédicats.

Traduisez cette négation en langage naturel.

Parmi les énoncés suivants, trouvez les équivalents et les opposés (contradictoire)

Pour argumenter de façon logique, on commencera par traduire les énoncés la langue du calcul des prédicats.

Les amours heureuses sont imaginaires.

Les amours imaginaires sont heureuses.

Les amours malheureuses ne sont pas imaginaires

Il n'y a pas d'amours heureuses qui ne soient pas imaginaires

Il n'y a d'imaginaire que les amours heureuses

Toutes les amours imaginaires sont heureuses

Raisonnement

Rappel : la règle du détachement : (p+ p=>q) => q

Un raisonnement, une méthode pour produire une proposition à partir de 2 autres propositions.

La règle fondamentale reste celle du détachement (ou modus ponens) :

(p + p=>q) => q

p=>q est appelée : proposition majeure ou Majeure

p est appelée : proposition mineure ou Mineure

q est appelée : Conclusion

Application aux prédicats

Présentation

Les raisonnements fonctionnent de la même manière en logique des propositions et en logique des prédicats.

Il y a 4 règles de raisonnements classique qu'on peut traduire sous forme de prédicat.

Ces raisonnements se construisent avec 3 propositions : 1 Majeure, une Mineure et une Conclusion.

Chacune de ces propositions peut correspondre à un des coins du carré logique et donc avoir un code.

Les raisonnements avaient été classés par des moyens mnémotechnique : 1 mot avec les 3 voyelles correspondant à chaque proposition.

BARBARA

Le syllogisme BARBARA est constitué de trois propositions A
Tous les humains sont mortels (A)
Les Français sont des humains (A)
Donc les Français sont mortels (A)

$\forall x, Hx \Rightarrow Mx$
$+ \forall x, Fx \Rightarrow Hx$
$= \forall x, Fx \Rightarrow Mx$

DARII

Le syllogisme DARII est constitué d'une proposition A et de deux propositions I
Tous les humains sont mortels (A)
Je suis un humain (I)
Donc je suis mortel(I)

a : instance de H valant moi !
$\forall x, Hx \Rightarrow Mx$
$+ Ha$
$= Ma$

CELARENT

Le syllogisme CELARENT est constitué d'une proposition A et de deux propositions E
Aucun humain n'est immortel (E)
Les français sont humains (A)
Donc aucun français n'est immortel (E)

$\forall x, Hx \Rightarrow \neg Ix$
$+ \forall x, Fx \Rightarrow Hx$
$= \forall x, Fx \Rightarrow \neg Ix$

FERIO

Le syllogisme FERIO est constitué d'une proposition E, d'une I et d'une O
Aucun humain n'est immortel
Je suis un humain
Donc je suis ne suis pas immortels

a : instance de H valant moi !
$\forall x, Hx \Rightarrow \neg Ix$

+ Ha

= ¬ Ia

Diagrammes

Ces raisonnements se représentent avec des ensembles en mixant les diagrammes de VENN, les diagrammes d'EULER et les instances dans les ensembles.