

## 9. Logique

## Logique propositionnelle

### Qu'est ce que la logique ?

La logique sert à étudier le raisonnement. C'est un domaine où l'on fait la distinction entre :

- \* la juxtaposition formelle des symboles : la **syntaxe**
- \* la signification des suites de symboles : la **sémantique**

Les modèles logiques sont définis à partir :

- \* de langages de symboles
- \* d'ensembles de règles syntaxiques.

Exemples :

- \* Un programme est une suite de symboles. Un langage de programmation est un ensemble de règles pour agencer des symboles. Un programme se plie ainsi à une syntaxe. Ensuite, on lui applique une sémantique afin d'interpréter ce que fait le programme.

Du point de vue syntaxique,  $3+4$  est une expression arithmétique. Sémantiquement, on y voit une addition qui pourra être ensuite évaluée à 7.

### L'algèbre de Boole $\mathbb{B}$

L'ensemble de Boole  $\mathbb{B}$  est défini par l'ensemble  $\{0,1\}$  muni des opérations suivantes

- \* Une addition définie par

+	0	1
0	0	1
1	1	1

- \* Une multiplication

.	0	1
0	0	0
1	0	1

- \* L'opération complémentaire définie par

x	$\bar{x}$
0	1
1	0

Une variable à valeur dans  $\mathbb{B}$  sera dite **booléenne**.

### Algèbre de Boole

- \* Plus généralement, une **algèbre de Boole** est constituée d'un ensemble  $E$  muni de
  - \* de deux éléments distincts  $\top$  (vrai) et  $\perp$  (faux) dans  $E$
  - \* de deux opérations binaires  $\cdot$  et  $+$  d'une opération unaire  $\bar{\phantom{x}}$  satisfaisant les propriétés suivantes
- \* les opérations  $\cdot$  et  $+$  sont
  - \* associatives,
  - \* commutatives,
  - \* distributives l'une par rapport à l'autre
  - \* idempotentes.
- \* pour tout  $x$  et tout  $y$  de  $E$  :
  - \*  $x \cdot (x + y) = x = (y \cdot x) + x$  (absorption)
  - \*  $x \cdot \perp = \perp$       \*  $x + \top = \top$
  - \*  $x + \perp = x$       \*  $x \cdot \bar{x} = \perp$
  - \*  $x \cdot \top = x$       \*  $x + \bar{x} = \top$
- \* Exemple :
  - \* Les parties d'un ensemble fini avec l'intersection et l'union.

### Fonctions booléennes à deux variables

- \* C'est l'ensemble des fonctions de  $\{0,1\}^2$  dans  $\{0,1\}$
- \* En logique on utilise
  - \* la conjonction  $\wedge$ ,
  - \* la disjonction  $\vee$
  - \* la négation  $\neg$
- \* En informatique on utilise plutôt
  - \* AND,
  - \* OR
  - \* NOT

## Implication

\* On définit l'opération  $\rightarrow$  par

$$p \rightarrow q = \neg p \vee q$$

p	q	$\neg p$	$p \rightarrow q$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

## Contraposée

$$p \rightarrow q = \neg q \rightarrow \neg p$$

\* Il suffit de remarquer que

$$p \rightarrow q = \neg p \vee q$$

$$= q \vee \neg p$$

$$= \neg(\neg q) \vee \neg p$$

$$= \neg q \rightarrow \neg p$$

\* Attention, c'est différent d'une **preuve par l'absurde**

\* Pour montrer  $p \rightarrow q$  on suppose ( $\neg p$  et  $q$ ) et on aboutit à une contradiction

## Equivalence

\* On définit l'opération  $\leftrightarrow$  par

$$p \leftrightarrow q = (p \rightarrow q) \wedge (q \rightarrow p)$$

p	q	$p \rightarrow q$	$q \rightarrow p$	$p \leftrightarrow q$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	1	1	1


## Fonctions booléennes à deux variables

x	y	0	1	$\neg$	x	y	$\vee$	$\wedge$	nor	$\leftrightarrow$	$\neg y$	$\leftarrow$	$\neg x$	$\rightarrow$	nand	1
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1

## Circuits

On peut construire des circuits électroniques qui implémentent les opérations logiques à partir des portes logiques classiques

\* Porte ET 

\* Porte OU 

\* Porte NON 

## Calcul propositionnel : syntaxe

\* On définit le calcul propositionnel comme on définit le calcul arithmétique.

\* Les propositions sont construites à l'aide de variables issues d'un ensemble  $V$  dénombrable et des symboles  $\{\neg, \vee, \wedge, (\cdot, \cdot), \rightarrow, \leftrightarrow\}$ , où  $\square$  est l'absurde, et  $\blacksquare$  est son contraire, le vrai.

\* Définition inductive de l'ensemble des formules  $F$  :

\* ( $\mathcal{B}$ ) Tout élément de  $V \cup \{\square, \blacksquare\}$  est une formule de  $F$

\* (I) si  $A$  et  $B \in F$ , alors

\*  $(A \vee B) \in F$

\*  $(\neg A) \in F$

\*  $(A \rightarrow B) \in F$

\*  $(A \wedge B) \in F$

\*  $(A \leftrightarrow B) \in F$

## Calcul propositionnel : sémantique

- \* Soit  $F$  une formule et  $I$  une application de l'ensemble  $V$  dans  $\mathbb{B}$ . On identifie les constantes booléennes  $0$  et  $1$  respectivement à faux et à vrai.  $I$  est alors appelée une interprétation.
- \* La **valeur de vérité** de  $F$  est donnée par  $\|F\|$  ainsi définie (on étend à toutes les formules l'application  $I$  définie pour l'instant que pour les éléments de  $V$ ).
  - \* si  $F = \blacksquare$ , alors  $\|F\| = 1$ ,
  - \* si  $F = \square$  alors  $\|F\| = 0$
  - \* si  $F = v \in V$ ,  $\|F\| = I(v)$
  - \* si  $F = \neg F'$ ,  $\|F\| = \|F'\|$
  - \* si  $F = F' \wedge F''$ ,  $\|F\| = \|F'\| \cdot \|F''\|$
  - \* si  $F = F' \vee F''$ ,  $\|F\| = \|F'\| + \|F''\|$
  - \* si  $F = F' \rightarrow F''$ ,  $\|F\| = \|F'\| + \|F''\|$
  - \* si  $F = F' \leftrightarrow F''$ ,  $\|F\| = (\|F'\| + \|F''\|) \cdot (\|F'\| + \|F''\|)$

## Vocabulaire

- \* Une formule  $F$  est **valide** si pour toute interprétation  $I$ ,  $\|F\| = 1$ . On dit que  $F$  est une **tautologie**.
- \* Une formule  $F$  est **satisfaisable** s'il existe une interprétation  $I$  telle que  $\|F\| = 1$ . On dit que  $I$  **satisfait**  $F$  et on le note :  $I \models F$ .
- \* Une formule  $F$  est **insatisfaisable** si pour toute interprétation  $I$ ,  $\|F\| = 0$ . On dit que  $F$  est une **antilogie**.

## Satisfaisabilité

**Théorème** La satisfaisabilité de la logique propositionnelle est décidable.

- \* Dit autrement, il existe un algorithme qui prend en entrée une formule donnée et qui répond en sortie si la formule est satisfaisable ou pas.
- \* L'algorithme naïf consiste en l'étude exhaustive de toutes les valeurs de vérité possibles de la formule, mais ce n'est pas efficace.
- \* D'autres algorithmes ne se basant pas sur l'interprétation (donc purement syntaxiques) peuvent être plus efficaces. C'est l'idée de la déduction naturelle, un ensemble de méthodes pour produire automatiquement des démonstrations.

## Equivalence

- \* Soient  $F$  et  $F'$  deux formules.  $F$  et  $F'$  sont dites **équivalentes** et on note  $F \equiv F'$  quand on a  $\|F\| = \|F'\|$  pour toute interprétation  $I$ .
- \* Exemple : Une loi logique de Morgan :  

$$\neg (p \vee q) \equiv (\neg p) \wedge (\neg q)$$
- \* Il y a 4 possibilités de définir des valeurs pour le couple  $(p, q)$ .
- \* Pour toutes les possibilités, les deux formules ont la même valeur de vérité. Elles sont donc équivalentes.

## Quelques « identités remarquables »

Soient deux variables booléennes  $p$  et  $q$ .

### \* élimination des flèches

$$* p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$$

$$* p \rightarrow q \equiv \neg p \vee q$$

### \* au sujet de la négation

$$* \neg(\neg p) \equiv p$$

$$* \neg(p \vee q) \equiv (\neg p) \wedge (\neg q)$$

$$* \neg(p \wedge q) \equiv (\neg p) \vee (\neg q)$$

## Quelques « identités remarquables »

Soient trois variables booléennes  $p$ ,  $q$  et  $r$ .

### \* au sujet de la disjonction

$$* p \vee q \equiv q \vee p$$

$$* p \vee (q \vee r) \equiv (p \vee q) \vee r$$

$$* p \vee \square \equiv p$$

$$* p \vee \blacksquare \equiv \blacksquare$$

### \* au sujet de la conjonction

$$* p \wedge q \equiv q \wedge p$$

$$* p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$$

$$* p \wedge \square \equiv \square$$

$$* p \wedge \blacksquare \equiv p$$

### \* distributivité de l'une par rapport à l'autre

$$* p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

$$* p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

## Des équivalences célèbres

Des équivalences logiques sont à la base des techniques de démonstration.

\* L'implication et sa **contraposée**  $(p \rightarrow q) = ((\neg q) \rightarrow (\neg p))$

\* Deux raisonnements par l'absurde

\*  $((\neg p) \rightarrow \square) = p$

En effet  $((\neg p) \rightarrow \square) = ((\neg(\neg p)) \vee \square) = p \vee \square = p$ .

\*  $((p \wedge (\neg q)) \rightarrow \square) = (p \rightarrow q)$

En effet  $((p \wedge (\neg q)) \rightarrow \square) = (\neg(p \wedge (\neg q))) \vee \square$   
 $= (\neg p) \vee q \vee \square$   
 $= (\neg p) \vee q$   
 $= (p \rightarrow q)$

## Table de vérité

Pour connaître la valeur de vérité d'une formule, on calcule de proche en proche la valeur de vérité des sous-formules qui la composent.

x	y	$y \rightarrow x$	$x \vee (y \rightarrow x)$	$\neg(x \vee (y \rightarrow x))$	$\neg(x \vee (y \rightarrow x)) \wedge y$
0	0	1	1	0	0
0	1	0	0	1	1
1	0	1	1	0	0
1	1	1	1	0	0

## Conséquence d'un ensemble de formules

\* Un ensemble de formules  $\langle F_1, F_2, \dots, F_n \rangle$  satisfait F et on note  $\langle F_1, F_2, \dots, F_n \rangle \models F$  quand on a pour toute interprétation I :

$$(\forall i, I(F_i) = 1) \Rightarrow I(F) = 1$$

\* On dit alors que F est une **conséquence** des  $\langle F_1, F_2, \dots, F_n \rangle$ .

**Théorème**  $\langle F_1, F_2, \dots, F_n \rangle \models F$  ssi  $(F_1 \wedge F_2 \wedge \dots \wedge F_n) \rightarrow F$  est une tautologie.

\* Exercice : la démonstration se fait par récurrence sur n.

## Modus ponens

Est-ce que  $\{p \rightarrow q, p\} \models q$  ?

\* On cherche à montrer si :  $((p \rightarrow q) \wedge p) \rightarrow q = \text{Vrai}$

\* On pourrait faire la table de vérité. On procède ici par identités remarquables:

$$\begin{aligned} ((p \rightarrow q) \wedge p) \rightarrow q &= \neg((p \rightarrow q) \wedge p) \vee q \\ &= \neg((\neg p \vee q) \wedge p) \vee q \\ &= \neg((\neg p \wedge p) \vee (q \wedge p)) \vee q \\ &= \neg(\square \vee (q \wedge p)) \vee q \\ &= \neg(q \wedge p) \vee q \\ &= \neg q \vee (\neg p) \vee q \\ &= (\neg p) \vee \blacksquare \\ &= \blacksquare \end{aligned}$$

Cette règle de déduction s'appelle le **modus-ponens**.

## Dédution

\* En informatique, on est très intéressé par la démonstration automatique. D'où la nécessité de formaliser le raisonnement, et plus précisément la déduction.

## Des fonctions booléennes aux propositions

p	q	r	$f(p,q,r)$	$g(p,q,r)$	$f(p,q,r)$
1	1	1	1	1	$pqr$
1	1	0	0	1	
1	0	1	0	1	
1	0	0	1	0	$p\bar{q}\bar{r}$
0	1	1	0	1	
0	1	0	1	0	$\bar{p}q\bar{r}$
0	0	1	1	0	$\bar{p}\bar{q}r$
0	0	0	0	0	

On en déduit que  $f(p,q,r) = pqr + p\bar{q}\bar{r} + \bar{p}q\bar{r} + \bar{p}\bar{q}r$

## Des fonctions booléennes aux propositions

p	q	r	f(p,q,r)	g(p,q,r)	f(p,q,r)	g(p,q,r)
1	1	1	1	1	p q r	p q r
1	1	0	0	1		p q $\bar{r}$
1	0	1	0	1		p $\bar{q}$ r
1	0	0	1	0	$\bar{p}$ q r	
0	1	1	0	1		$\bar{p}$ q r
0	1	0	1	0	$\bar{p}$ q $\bar{r}$	
0	0	1	1	0	$\bar{p}$ $\bar{q}$ r	
0	0	0	0	0		

On en déduit que  $g(p,q,r) = pqr + pqr + pqr + pqr$

## Simplification d'expressions logiques

Toute expression logique peut s'écrire comme une somme de produit, appelée forme normale disjonctive.

Question. Peut on trouver une forme minimale ?

$$\begin{aligned}
 g(p,q,r) &= pqr + pqr + pqr + pqr \\
 &= pqr + pqr \\
 &+ pqr + pqr \\
 &+ pqr + pqr \\
 &= pq + pr + qr
 \end{aligned}$$

Remarque. On définit de la même manière la forme normale conjonctive comme un produit de somme.

C'est toujours possible puisque

$$(p+q) = \bar{p} \cdot \bar{q}$$

exercice

## Logique des prédicats

### Limite de la logique des propositions

- \* Le calcul des propositions est simple mais ne peut pas rendre compte de la totalité des raisonnements.
- \* il ne permet pas faire allusion aux propriétés d'une variable.
- \* Il ne permet pas non plus de décrire des relations entre plusieurs variables.
- \* Il leur reconnaît simplement deux états, elles sont vraies ou elles sont fausses.
- \* On va donc étudier comment généraliser le calcul des propositions afin de mieux formaliser le raisonnement.
- \* On définit le calcul des prédicats (logique du 1er ordre).

## Informatique et logique du 1er ordre

- \* Parmi les innombrables usages de la logique du 1er ordre en informatique, citons-en deux importants :
- \* Les langages de programmation peuvent être typés ou fortement typés.
- \* On a besoin de la logique du 1er ordre pour décrire le principe même du typage.
- \* Les bases de données et particulièrement des langages comme SQL se fondent sur le calcul des prédicats.

## Calcul des prédicats

- \* En bref,
- \* Un prédicat p est une fonction d'un nombre fini de variables ou de constantes dans  $\mathbb{B}$ .
- \* Les constantes et les variables sont issues d'un même ensemble appelé le domaine.
- \* Exemple
- \* si p signifie « être pair ». Sur le domaine des entiers, p(2) est vrai mais p(3) est faux. Sur le domaine  $\{1\}$ , p(1)=0 car p(1) est toujours faux.
- \* Une interprétation sera la définition d'un domaine et de prédicats sur ce domaine.
- \* Apparaissent les quantificateurs,
- \* l'un existentiel, noté  $\exists$       \*  $\exists x p(x)$
- \* et l'autre universel, noté  $\forall$       \*  $\forall x p(x)$

## Calcul des prédicats : syntaxe

- \* On dispose :
  - \* d'un ensemble dénombrable de variables  $V = \{x, y, w, v, \dots\}$  sur un domaine  $\mathcal{D}$
  - \* d'un ensemble dénombrable de constantes  $C = \{c_1, c_2, \dots\} \subseteq \mathcal{D}$
  - \* des symboles  $\{ \neg, \wedge, \vee, \rightarrow, \leftrightarrow, (, ) \}$ .
- \* On se donne un ensemble fini de prédicats  $\mathcal{P}$  définis sur le même domaine.
  - \* Exemple : Une variable peut représenter n'importe quel étudiant, une constante représente précisément l'un d'eux, un prédicat pourrait être le fait d'être admis à un diplôme.
- \* Les variables et les constantes  $t_1, t_2, \dots, t_j$  seront appelées les termes d'un prédicat  $p$  d'arité  $j$  dans l'expression  $p(t_1, t_2, \dots, t_j)$ .
  - \*  $p(t_1, t_2, \dots, t_j)$  est appelé un **atome**.

## Calcul des prédicats : syntaxe

- \* Définition inductive de l'ensemble des formules  $F$  du calcul des prédicats :
  - \* (B) Les atomes sont des formules de  $F$
  - \* (I)
    - \* si  $f \in F$ , alors  $(\neg f) \in F$
    - \* si  $f, f' \in F$ , alors
      - \*  $(f \wedge f') \in F$       \*  $(f \rightarrow f') \in F$
      - \*  $(f \vee f') \in F$       \*  $(f \leftrightarrow f') \in F$
    - \* si  $f$  est une formule et  $x$  une variable, alors  $(\exists x) f \in F$
    - \* si  $f$  est une formule et  $x$  une variable, alors  $(\forall x) f \in F$

## Variables libres ou liées

- \* Chacune des fois où une variable  $x$  apparaît dans une formule  $F$  est appelée une **occurrence** de  $x$  dans  $F$ .
- \* On dit qu'une occurrence de  $x$  dans  $F$  est **liée** si un quantificateur porte sur cette occurrence de  $x$ . Dans le cas contraire, on parle d'occurrence **libre**.
- \* On peut construire les arbres d'expression de ces formules pour s'y retrouver entre occurrence liée et libre.
- \* Une variable est libre si toutes ses occurrences sont libres, une variable est liée si elle a au moins une occurrence liée. On parle aussi de variable muette dans ce dernier cas.
- \* Précédence des opérateurs (du plus prioritaire au moins prioritaire)
  - \*  $\neg$     $\forall$     $\exists$                       \*  $\rightarrow$     $\leftrightarrow$
  - \*  $\wedge$     $\vee$

## Exemples

- \*  $(\forall x)(\exists y)(p(x,y) \rightarrow (\exists x)q(x,y,c))$   
Toutes les occurrences de  $x$  et  $y$  dans  $p$  ou  $q$  sont liées mais celles de  $x$  ne sont pas liées au même quantificateur.
- \*  $(\forall x)(p(x,y) \rightarrow (\exists x)(\exists y)q(x,y,c))$   
Toutes les occurrences de  $x$  dans  $p$  ou  $q$  sont liées, celles de  $y$  dans  $p$  est libre.
- \*  $(\forall x)(u(x) \vee (\exists x)w(x))$   
L'occurrence de  $x$  dans  $u$  est liée au  $\forall$ , celle de  $x$  dans  $w$  est liée seulement au  $\exists$ .
- \* On a l'équivalence :  
 $(\forall x)(u(x) \vee (\exists x)(w(x))) \equiv (\forall x)(u(x) \vee (\exists y)(w(y)))$

## Calcul des prédicats : sémantique

- \* Une interprétation  $I$  est la donnée :
  - \* d'un domaine non vide  $\mathcal{D}$  éventuellement infini
  - \* d'une évaluation dans  $\mathcal{D}$  de chaque variable (susceptibles d'occurrences libres)
  - \* d'un ensemble  $\mathcal{P}$  de prédicats. Un prédicat d'arité  $i$  sera défini sur  $\mathcal{D}^i$ .
  - \* d'une application de  $\mathcal{P}$  dans l'ensemble des fonctions à valeur dans  $\mathbb{B}$ . A chaque  $p$  de  $\mathcal{P}$  d'arité  $i$ , correspond une fonction de  $\mathcal{D}^i$  dans  $\mathbb{B}$ .
- \* La valeur de la formule  $F$  sous l'interprétation  $I$  est notée  $v(I, F)$
- \* Pour l'instant, seuls les atomes  $A$  ont une valeur  $v(I, A)$  fixée par l'interprétation  $I$ .

## Calcul des prédicats : sémantique

- \* La valeur de vérité de  $F$  est donnée par  $v(I, F)$  ainsi définie : on étend à toutes les formules la valuation  $v(I, A)$  définie pour l'instant que pour les atomes  $A$ .
  - \* si  $F = A$ ,       $v(I, F) = v(I, A) \in \mathbb{B}$
  - \* si  $F = \neg F'$ ,     $v(I, F) = \overline{v(I, F')}$                       \* si  $F = F' \wedge F''$ ,  $v(I, F) = v(I, F') \cdot v(I, F'')$
  - \* si  $F = F' \rightarrow F''$ ,     $v(I, F) = \overline{v(I, F') \cdot v(I, F'')}$                       \* si  $F = F' \vee F''$ ,  $v(I, F) = v(I, F') + v(I, F'')$
  - \* si  $F = F' \leftrightarrow F''$ ,     $v(I, F) = \overline{(v(I, F') + v(I, F''))} \cdot (v(I, F') + v(I, F''))$
  - \* Appelons  $\Theta x d$  la formule  $\Theta$  où toutes les occurrences libres de  $x$  ont été remplacées par  $d$  fixé dans  $\mathcal{D}$ .
    - \* si  $F = \forall x \Theta$ ,     $v(I, F) = 1$  si pour tout  $d$  de  $\mathcal{D}$ ,  $v(I, \Theta x d) = 1$  et  $v(I, F) = 0$  sinon
    - \* si  $F = \exists x \Theta$ ,     $v(I, F) = 1$  s'il existe un  $d$  de  $\mathcal{D}$  tel que  $v(I, \Theta x d) = 1$ , et  $v(I, F) = 0$  sinon.

## Vocabulaire

- \* Une formule  $F$  est **valide** si, pour toute interprétation  $I$  :  $v(I, F) = 1$ 
  - \* On dit aussi que  $F$  est une **tautologie**.
- \* Une formule  $F$  est satisfiable si il existe une interprétation  $I$  telle que :  $v(I, F) = 1$ 
  - \* On dit que  $I$  satisfait  $F$  et on le note :  $I \models F$ .
  - \* Une telle interprétation  $I$  est appelée un **modèle** pour  $F$ .
- \* Une formule  $F$  est insatisfiable si, pour toute interprétation  $I$  on a :  $v(I, F) = 0$ 
  - \* On dit que  $F$  est une **antilogie**.
- \* Soient  $F$  et  $F'$  deux formules,  $F$  et  $F'$  sont **équivalentes** et on note  $F = F'$ ssi, pour toute interprétation  $I$ , on a :  $v(I, F) = v(I, F')$ .

## Exemple

\* Soit  $F$  la formule  $F = \forall x \forall y (p(x,y) \rightarrow \exists z (p(x,z) \wedge p(z,y)))$

\* En notant  $\Phi$  l'implication, on a  $F : \forall x \forall y (\Phi)$ .

\* Première interprétation  $I_1$  pour  $F$

- \* le domaine  $D1$  de  $I_1$  est l'ensemble des réels
- \* le prédicat binaire  $p$  est l'ordre  $<$

\* si  $p(x,y)$  est faux,  $\Phi$  est vraie.

\* si  $p(x,y)$  est vrai, alors  $x < y$ . Posons  $z = (x+y)/2$ . Alors,  $p(x,z)$  est vrai et  $p(z,y)$  aussi.

\* en conclusion,  $v(I_1, F) = 1$  et  $I_1$  constitue un modèle pour  $F$ .

\* Deuxième interprétation  $I_2$  pour  $F$

- \* le domaine  $D2$  de  $I_2$  est l'ensemble des entiers naturels
- \* le prédicat binaire  $p$  est encore l'ordre  $<$

\* si  $p(x,y)$  est faux,  $\Phi$  est vraie.

\* si  $p(x,y)$  est vrai, alors  $x < y$ . Il n'existe pas d'entiers entre  $x$  et  $y$  dès lors que  $x$  et  $y$  sont consécutifs.  $\Phi$  peut être fausse.

\* en conclusion,  $v(I_2, F) = 0$ .  $F$  ne saurait être valide.

## Négation et quantifications

\* Que deviennent les négations des formules quantifiées ?

$$* \neg (\forall x F) = \exists x (\neg F)$$

$$* \neg (\exists x \Phi) = \forall x (\neg \Phi)$$

\* Démonstration de la première équivalence :

\* Il faut démontrer que  $\neg (\forall x F) \leftrightarrow \exists x \neg F$  est valide.

\* Commençons par montrer que  $\neg (\forall x F) \rightarrow \exists x \neg F$  est valide.

\* Supposons que  $F$  est une tautologie,  $\forall x F$  est aussi une tautologie. Alors,  $\neg (\forall x F)$  est une antilogie. Et l'implication  $\neg (\forall x F) \rightarrow \exists x \neg F$  est bien une tautologie.

\* Supposons que  $F$  n'est pas une tautologie. Il existe une interprétation  $I$  pour laquelle  $F$  est fausse. Alors,  $\exists x \neg F$  est une tautologie. Et l'implication  $\neg (\forall x F) \rightarrow \exists x \neg F$  est une tautologie.

\* Reste à donner une démonstration analogue pour la réciproque.

## Exemple

\* On cherche à simplifier la formule  $F$  :

$$\neg (\forall x \exists y \exists z \exists t p(x,y,z,t))$$

\* On applique les règles de négation d'une expression quantifiée :

$$\begin{aligned} * F &= \exists x (\neg (\exists y \exists z \exists t p(x,y,z,t))) \\ &= \exists x \forall y (\neg (\exists z \exists t p(x,y,z,t))) \\ &= \exists x \forall y \forall z (\neg (\exists t p(x,y,z,t))) \\ &= \exists x \forall y \forall z \forall t \neg p(x,y,z,t) \end{aligned}$$

## Autres identités remarquables

\* On a les identités :

$$* \forall x (F \wedge G) = (\forall x F) \wedge (\forall x G)$$

$$* \exists x (F \vee G) = (\exists x F) \vee (\exists x G)$$

\* Par contre, n'inventez pas des identités qui n'existent pas !

$$* A : \forall x (F \vee G)$$

$$* B : (\forall x F) \vee (\forall x G)$$

$$* A \neq B$$

$$* C : \exists x (F \wedge G)$$

## Contreexemple

\* Prenons l'alphabet (français) pour domaine et les deux prédicats :

\*  $v(x) = 1$  si  $x$  est une voyelle

\*  $c(x) = 1$  si  $x$  est une consonne

\* Il n'est pas équivalent de dire :

\* « toute lettre de l'alphabet est une consonne ou une voyelle » et

\* « toutes les lettres de l'alphabet sont des consonnes ou toutes les lettres de l'alphabet sont des voyelles »

\* Il n'est pas non plus équivalent de dire :

\* « il existe dans l'alphabet au moins une consonne et une voyelle » et

\* « il existe dans l'alphabet une lettre à la fois consonne et voyelle ».