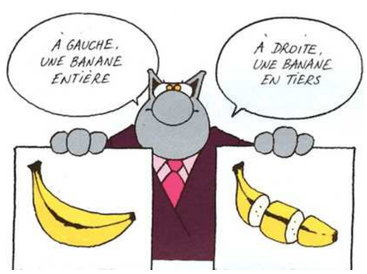
INF1130

Mathématiques pour l'informatique

Zied Zaier, PhD
Département d'informatique
Université du Québec à Montréal



Cours 1

LOGIQUE

CE DOCUMENT EST INSPIRÉ DES TRAVAUX DES PROFESSEURS KENNETH H. ROSEN ET TIMOTHY WALSH.

Contenu du présent document

- Introduction à la Mathématique Discrètes
- Étude des notions fondamentales de la logique.
- Notions de propositions logiques.
- Notions d'équivalences logiques.
- Notions fondamentales de la logique des prédicats.
- Notions de quantificateurs.
- Notions de variables liées.

Pourquoi dire math. DISCRÈTES?

Que représente des structures discrètes?

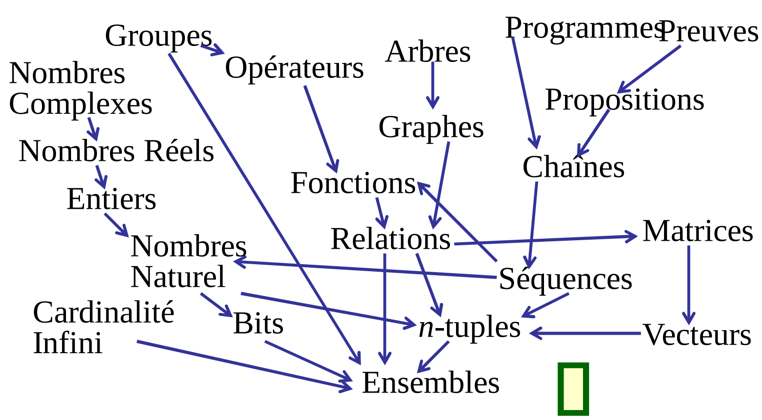
- "***Discrète***" – Signifie composées de parties distinctes, séparables. (À l'opposition de continues)
  discret VS continu ⇔ *digital VS analogique*
- "***Structures***" – Objets construits à partir d'objets plus simples selon certains patterns définis.
- "***Mathématiques Discrètes***" – Vouée à l'étude des objets mathématiques et des structures discrètes.

Quelques Structures Discrètes

- Propositions
- Prédicats
- Preuves
- Ensembles
- Fonctions
- Ordre
- Algorithmes
- Entiers
- Sommations
- Séquences
- Chaînes
- Permutations
- Combinaisons
- Relations
- Graphes
- Arbres
- Circuits Logiques
- Automate

Relations possibles entre Structures

- "$\longrightarrow$" $:\stackrel{\text{def}}{=}$ "Peu être définis en terme de"



Quelques notations utiles

$\neg p$	$p \wedge q$	$p \oplus q$	$p \rightarrow q$	$p \Leftrightarrow q$	$\forall x\, P(x)$
$\exists x\, P(x)$	$\{a_1, \cdots, a_n\}$	$\mathbf{Z}, \mathbf{N}, \mathbf{R}$	$\therefore$	$\{x \mid P(x)\}$	$x \notin S$
$\varnothing$	$S \subseteq T$	$\lvert S \rvert$	$A \cup B$	$\overline{A}$	$\bigcap_{i=1}^{n} A_i$
$f : A \rightarrow B$	$f^{-1}(x)$	$f \circ g$	$\lfloor x \rfloor$	$\sum_{\alpha \in S} a_\alpha$	$\prod_{i=1}^{n} a_i$
O, Ω, Θ	$\min, \max$	$a \nmid b$	$\gcd, \mathrm{lcm}$	$\bmod$	$a \equiv b \pmod m$
$(a_k \cdots a_0)_b$	$[a_{ij}]$	$\mathbf{A}^{\mathrm{T}}$	$\mathbf{A} \odot \mathbf{B}$	$\mathbf{A}^{[n]}$	$\binom{n}{r}$
$C(n; n_1, \cdots, n_m)$	$p(E \mid F)$	R^*	Δ	$[a]_R$	$\deg^+(v)$

Pourquoi étudier les Math. Discrètes?

- La base de tout traitement de l'information en format digital: *<u>Manipulations discrètes de structures discrètes représentées en mémoire.</u>*
- Le langage de base et la fondation conceptuelle des sciences informatiques.
- Les concepts des Math. Discrètes sont aussi utilisés en mathématique en général, en science, en ingénierie, en économie, biologie, *etc.*, …
- Un outil essentiel pour le raisonnement rationnel.

Utilisation des Math Discrètes en informatique

- Algorithmique et structures de données
- Langage de Programmation Compilateurs et interpréteurs.
- Réseautique
- Systèmes d'exploitation
- Architecture des ordinateurs
- Systèmes de gestion de base de données
- Cryptographie
- Correction d'erreurs dans le code
- Algorithmes en Graphisme et Animation, engins de jeux, *etc.*…

Objectifs du cours

- Vérifier la validité d'énoncés, de fonctions , d'arguments logiques simples (preuves).
- Vérifier la rectitude d'un algorithme.
- Création d'instances simples d'arguments logiques valides et d'algorithmes correctes.
- Comprendre les définitions et les propriétés de diverses structures discrètes.
- Lire correctement, représenter et analyser diverses structures discrètes utilisant des notations standards.

Fondements de la logique

*La Logique Mathématique permet la manipulation d'énoncés composés é*laborés.

Incluant:

- Un langage formel pour les exprimer.
- Une notation concise pour les exprimer.
- Une méthodologie pour raisonner de façon objective sur leur véracité ou leur fausseté.

Fondements de la logique: Survol

- Logique propositionnelle:
  - Définitions de bases.
  - Règles d'équivalence et dérivations.
- Logique des prédicats.
- Prédicats:
  - Expressions quantifiées de prédicats.
  - Équivalences et dérivations.

Logique propositionnelle (section 1.1)

La Logique Propositionnelle est la logique d'énoncés composés construits à partir d'énoncés plus simples en utilisant des opérateurs (connecteurs) Booléens*.*

Quelques applications en science informatique:

- Design de circuits electroniques digitaux .
- Exprimer des conditions dans des programmes.
- Requêtes dans une BD et des engins de recherche.

Définition: *Proposition*

Définition: Une *proposition* (dénotée *p*, *q*, *r*, …) est:

- Un énoncé (*i.e.*, une phrase déclarative)
  - *avec une sens bien défini*, (sans ambiguïté)
- Ayant une valeur de vérité soit *vraie* (**T**) ou *fausse* (**F**)
  - **Jamais** les deux, ni entre les deux.
    - Cependant, il est possible de *ne pas connaître* la valeur de vérité,
    - et, la valeur de vérité peut *dépendre* d'une situation particulière ou du contexte.
- Nous verrons plus tard, en étudiant la *théorie probabiliste* qu'il est possible d'assigner un *degré de certitude* (entre **T** ou **F**) à des propositions.
  - Pour l'instant pensez! VRAI/FAUX

Exemples de propositions

- “C’est nuageux.” (Dans une situation donnée.)
- “Ottawa est la capitale du Canada.”
- “1 + 2 = 3”

Exemples qui ne sont pas des propositions:

- “Quelle heure est-il?” (interrogation, question)
- “OH ! OH! OH!.” (sans signification)
- “Fait ce devoir !” (impératif, commande)
- “Roule 4-5 minutes, tourne à gauche...” (vague)
- “1 + 2” (expression sans valeur de vérité)

Opérateurs / Connecteurs

Un *opérateur* ou un *connecteur* combine une ou plusieurs expressions (*opérandes*) en une expression plus grande, plus élaborée.

- Opérateurs *Unaires* requièrent 1 opérande (*ex:* -3);
- Opérateurs *binaires* requièrent 2 opérandes (*ex:* 3×4).
- Opérateurs *propositionnels* ou *Booléens* s’appliquent sur des propositions (ou leur valeur de vérité) plutôt que sur des nombres.

Opérateurs Booléens populaires

Nom formel	Nom court	Parité	symbolee
Négation	NOT (NON)	Unaire	$\neg$
Conjonction	AND (ET)	Binaire	$\wedge$
Disjonction	OR (OU)	Binaire	$\vee$
OU-Exclusif	XOR (OUX)	Binaire	$\oplus$
Implication	IMPLIQUE	Binaire	$\rightarrow$
Biconditionnel	SSI	Binaire	$\leftrightarrow$

L'opérateur de Négation

L'opérateur de *négation* "¬" (*NOT*) transforme une prop. dans sa forme logique complémentaire (*négation*).

Ex: SI p = "J'ai les cheveux blanc."

ALORS $\neg p$ = "Je n'ai pas les cheveux blanc."

Table de vérité du NOT:

p	$\neg p$
T	F
F	T
Opérande	Résultat

T :≡ True; F :≡ False
":≡" "est défini comme"

L’opérateur de Conjonction

L’opérateur de *conjonction* “$\wedge$” (*AND, ET*) combine deux propositions pour former leur *conjonction logique*.

$\wedge$ND

Ex: p=“J’irai à Québec.” et q=“J’irai au Château Frontenac.”, alors $p \wedge q$=“J’ira à Québec et au Château Frontenac.”

Rappel: “$\wedge$” pointe vers le haut comme un “A”, et correspond au “$\wedge$ND”

Table de vérité (Conjonction)

- Notez qu’une conjonction $p_1 \wedge p_2 \wedge \ldots \wedge p_n$ de n propositions aura 2^n rangées dans sa table de vérité.

Opérandes

p	q	$p \wedge q$
F	F	F
F	T	F
T	F	F
T	T	T

- Aussi: Les opérations $\neg$ et $\wedge$ sont suffisantes pour déduire n’importe quelles tables de vérités Booléennes

L’opérateur de Disjonction

L’opérateur de *disjonction* “$\vee$” (*OU*) combine deux propositions pour former une *disjonction* logique.

p=“Mon ordinateur a une bonne carte graphique.”

q=“Mon ordinateur a un CPU performant.”

$p \vee q$=“Mon ordinateur a soit une bonne carte graphique, **or (ou)** mon ordinateur a un CPU performant.”

Table de vérité (Disjonction)

- Notez que $p \vee q$ signifie que p est VRAI, ou q est vrai, **ou les deux** sont vraies!
- Cette opération est aussi appelée *ou inclusif,* et **inclus** la possibilité que p et q soient VRAIES.

p	q	$p \vee q$
F	F	F
F	T	**T**
T	F	**T**
T	T	T

Notez la différence avec AND

Un exercise simple

Posons p = "Il a plue la nuit dernière",
q = "Le balai mécanique a lavée la rue cette nuit",
r = "La rue est mouillée ce matin."

Tranduisez chaque proposition:

$\neg p$ = "Il n'a pas plue la nuit dernière."

$r \wedge \neg p$ = "La rue est mouillée mais il n'a pas plue"

$\neg r \vee p \vee q$ = "Soit que la rue n'était pas mouillée, ou il a plue la nuit dernière, ou la rue a été lavée cette nuit."

L'opérateur OU Exclusif (*Exclusive Or)*

L'opérateur *OU-Exclusif* "$\oplus$" (*XOR*) combine deux propositions formant leur "OU exclusif" logique.

p = "J'obtiendrai un A+ dans ce cours,"

q = "J'abandonnerai ce cours,"

$p \oplus q$ = "Je vais soit avoir un A+ dans ce cours, ou j'abandonnerai (mais pas les deux)"

Table de vérité du OU-Exclusif

- Notez que $p \oplus q$ veut dire que p est vrai, ou q est vrai, mais pas les deux!
- Cette opération est appelée OU-*exclusif,* puisqu'elle **exclus** la possibilité que p et q soient VRAIES.

p	q	$p \oplus q$
F	F	F
F	T	T
T	F	T
T	T	**F**

Notez la différence avec le OR

L'ambiguïté du Language Naturel

Notez qu'en français le "ou" peut être <u>ambïgu</u> en regard des cas inclusif et exclusif

"Joe est un plombier ou Joe est un politicien." - $\vee$

"Joe a eu un A dans son cours de plomberie ou Joe a eu un B dans ce cours." - $\oplus$

p	q	p "or" q
F	F	F
F	T	T
T	F	T
T	T	F

Besoin du contexte pour désambiguïser le sens des propositions

L'opération *Implication*

L' *implication* $\overbrace{p}^{\text{Hypothèse}} \rightarrow \overbrace{q}^{\text{Conclusion}}$ signifie que *p* implique *q*.

Ex:, posons *p* = "Vous étudiez beaucoup."
q = "Vous obtenez une bonne note."

p → *q* = "Si vous étudiez beaucoup, vous obtiendrai Alors une bonne note."

La table de vérité de l'Implication

- *p* → *q* est **faux** <u>seulement</u> quand *p* est VRAI mais que *q* n'est pas VRAI (**not** true).
- *p* → *q* ne veut pas dire que *p* <u>a causé</u> *q*
- *p* → *q* ne requiert pas que *p* ou *q* **<u>soit VRAIE</u>**
- *EX:* "(1=0) → Dumbo l'éléphant vole" est VRAIE

p	*q*	*p*→*q*
F	F	T
F	T	T
T	F	**F** $\}$ Le <u>seul</u> cas FAUX
T	T	T

Pourquoi l’implication semble bizarre?

- Considérons une phrase comme,
  - “Si je regarde CNN demain, il tombera de la grêle”
- En logique, cette phrase est considérée VRAIE tant que je ne regarde pas CNN ou qu’il grêle.
- Mais, dans une conversation normale, ce genre d’affirmation est déconcertante et questionnable.

Comment résoudre cette inconsistance

- Une phrase en langage courant “SI *p* ALORS *q*” peut signifier implicitement:
  - “<u>Dans toutes les situations possibles</u>, *p* implique *q*.”
  - SI p est VRAI ALORS q est aussi VRAI.
  - SI p n’est pas VRAI ALORS q est FAUX.
  - Il existe une relation entre, l’hypothèse et la conclusion.
- Ce n’est pas le cas en logique.
  - $p \rightarrow q$ signifie $\neg p \vee q$
- Ex.: “Si vous avez 100 au final, vous aurez A+”
  - Si vous avez 100, vous vous attendez à avoir un A+ ($1 \rightarrow 1=1$)
  - Sinon, vous pourriez avoir un A+ quand même ($0 \rightarrow 1=1$)
  - Par contre, si vous avez un 100 et n’avez pas un A+, vous vous sentirez probablement lésé ($1 \rightarrow 0 = 0$)

Examples d'Implications

- "Si ce cours ne se termine jamais, Alors le soleil se lèvera demain." *True* or *False*?
- "Si lundi est un jour de la semaine, Alors Je suis un singe." *True* or *False*?
- "Si 1+1=6, Alors Obama est président." *True* or *False*?
- "Si la Suisse est un fromage, Alors je suis plus riche que Warren Buffet." *True* or *False*?

Sens de phrases en langage courant $p \rightarrow q$

- "p implique q"
- "Si p, Alors q"
- "Si p, q"
- "quand p, q"
- "chaque fois que p, q"
- "q si p"
- "q quand p"
- "q chaque fois que p"
- "p seulement si q"
- "p est suffisant pour que q"
- "q est nécessaire pour que p"
- "q découle de p"
- "q est supposé par p"

Réciproque, Inverse, Contraposée

Formes découlants de l'implication $p \rightarrow q$:

- Sa réciproque est: $q \rightarrow p$.
- Son *inverse* est: $\neg p \rightarrow \neg q$.
- Sa *contraposée*: $\neg q \rightarrow \neg\, p$.
- Une de ces formes a le même sens (même table de vérité) que $p \rightarrow q$. Laquelle ?

Contraposée

Comment en être certain?

Prouvons l'équivalence de $p \rightarrow q$ et sa contraposée par table de vérité:

p		q	$\neg q$		$\neg p$	$p \rightarrow q$	$\neg q \rightarrow \neg p$
F	$\rightarrow$	F	T	$\rightarrow$	T	T	T
F	$\rightarrow$	T	F	$\rightarrow$	T	T	T
T	$\not\rightarrow$	F	T	$\not\rightarrow$	F	F	F
T	$\rightarrow$	T	F	$\rightarrow$	F	T	T

L’opérateur *biconditionnel*

Une forme *biconditionelles* $p \leftrightarrow q$ est vraie si la proposition p est vraie et si *la proposition* q est vraie. Nous dirons p SSI q.

p = “Obama gagne les élections de 2008.”

q = “Obama sera président jusqu’en 2012.”

$p \leftrightarrow q$ = “Obama gagne les élections de 2008, SSI, Obama sera président jusqu’en 2012.”

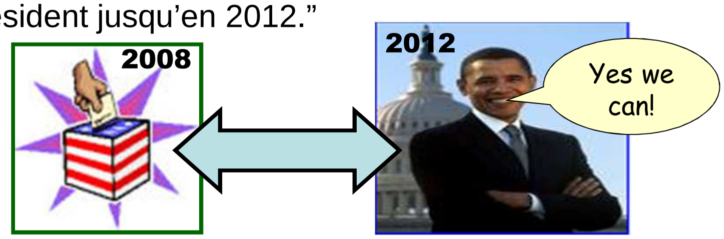


Table de vérité de la biconditionnelle

- $p \leftrightarrow q$ signifie que p et q ont la même valeur de vérité.
- Notez que cette table de vérité est l’opposée du XOR $\oplus$

  Donc, $p \leftrightarrow q$ est équivalent $\neg(p \oplus q)$

p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

- $p \leftrightarrow q$ ne veut pas dire que p et q sont VRAIES, ou que chacune est la cause de l’autre, ou découlent d’une cause commune.

Précédence des opérateurs

- Nous pouvons avoir des énoncés composés
  - $r \vee p \rightarrow q$
- Quel est l'ordre d'application des opérateurs logiques?
  - Les parenthèses spécifient l'ordre
  - $r \vee (p \rightarrow q)$: Implication en premier
- Si pas de parenthèses, la précédence des opérateurs intervient

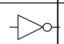
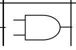
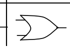
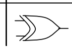
Opérateur	*Précédence*
$\neg$	1
$\wedge$	2
$\vee$	3
$\rightarrow$	4
$\leftrightarrow$	5

Opérations Booléennes (Sommaire)

- Table de vérité des opérateurs logiques vus jusqu'à maintenant.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
F	F	T	F	F	F	T	T
F	T	T	F	T	T	T	F
T	F	F	F	T	T	F	F
T	T	F	T	T	F	T	T

Quelques Notations Alternatives

Nom:	not	and	or	xor	implies	iff
Proposition logique:	$\neg$	$\wedge$	$\vee$	$\oplus$	$\rightarrow$	$\leftrightarrow$
Algèbre Booléenne:	$\overline{p}$	pq	$+$	$\oplus$		
C/C++/Java (wordwise):	!	&&	||	!=		==
C/C++/Java (bitwise):	~	&	|	^		
Portes Logiques:						

Opérations Logiques et Binaires

- Un *bit* est un binary (base 2) digit: 0 ou 1.
- Les bits sont aussi utilisés pour représenter des valeurs de vérité.
- Par convention:
  0 représente “FAUX”; 1 représente “VRAI”.
- *L' Algèbre Booléenne* est comme l'algèbre conventionnelle sauf que les variables sont logiques (bits), + est le “or”, et la multiplication est le “and”.

Chaîne de Bits

- *Une chaîne de Bits* de longueur *n* est une séquence ordonnée (séries, tuple) de $n \geq 0$ bits.
- Par convention, les chaînes de bits sont écrites généralement de gauche à droite:
  - BLPS "1001101010" est 1.
- Alors la chaîne de bits correspond en base 10 à: $1101_2 = 8+4+1 = 13$.

Opérations sur les Bits (Bitwise)

- Opérations booléennes peuvent être appliquées sur des chaînes de bits autant que des bits individuels.
- EX:
  0110110110
  <u>1100011101</u>
  1110111111 (Bit-wise OR)
  0100010100 (Bit-wise AND)
  1010101011 (Bit-wise XOR)

Récapitulations (sec. 1.1)

- Définitions sur les propositions
- Opérateurs sur les propositions logiques
  - Notations symboleiques.
  - Équivalents dans le langage courant.
  - Sens logique.
  - Tables de vérité.
- Proposition simples vs. composées.
- Notations alternatives.
- Bits et chaînes de bits.
- Section: 1.2
  - Équivalences propositionnelles.
  - Comment les prouver.

Équivalence Propositionnelle (sec. 1.2)

- Deux propositions composées syntaxiquement différentes peuvent être identiques du point de vue sémantique. Elles sont alors dites *équivalentes*. Il existe:
  - Diverses *règles **ou** lois d'équivalence*.
  - Des *dérivations symboliques* pour *prouver* les équivalences.

Tautologies et Contradictions

Une ***tautologie*** est une proposition composée qui est **VRAIE** et ce pour toutes les combinaisons de valeurs des propositions atomiques la composant.

Ex. $p \vee \neg p$ [Quelle est la table de vérité?]

Une ***contradiction*** est une proposition composée qui est **FAUSSE** dans tous les cas. *Ex.* $p \wedge \neg p$ [La table de vérité?]

Les autres propositions composées sont des *contingences*.

Équivalence logique

$$\boldsymbol{p \Leftrightarrow q}$$

- Les propositions composées p et q sont logiquement équivalentes (exprimée $p \Leftrightarrow q$) **SSI** p et q possèdent les mêmes valeurs logiques dans <u>toutes</u> les rangées de leur table de vérité.

Preuve d'équivalence par table de vérité

Ex. Prouvez que $p \vee q \Leftrightarrow \neg(\neg p \wedge \neg q)$.

p	q	$p \vee q$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$	$\neg(\neg p \wedge \neg q)$
F	F	F	T	T	T	F
F	T	T	T	F	F	T
T	F	T	F	T	F	T
T	T	T	F	F	F	T

Lois d'Équivalence

- Similaires aux identités arithmétiques utilisées en algèbre, mais appliquées aux équivalences propositionnelles.
- Peuvent être substituées entièrement ou en partie à des propositions complexes pour en prouver l'équivalence.

Lois d'équivalence- Exemples

- *Identité*: $p \wedge \mathbf{T} \Leftrightarrow p \qquad p \vee \mathbf{F} \Leftrightarrow p$
- *Domination*: $p \vee \mathbf{T} \Leftrightarrow \mathbf{T} \qquad p \wedge \mathbf{F} \Leftrightarrow \mathbf{F}$
- *Idempotence*: $p \vee p \Leftrightarrow p \qquad p \wedge p \Leftrightarrow p$
- *Double négation:* $\neg\neg p \Leftrightarrow p$
- *Commutativité:* $p \vee q \Leftrightarrow q \vee p \qquad p \wedge q \Leftrightarrow q \wedge p$
- *Associativité:* $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
  $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$

Autres lois d'équivalence

- *Distributivité*: $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$
  $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$
- *De Morgan*:
  $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$
  $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$
- *Tautologie/contradiction triviale* (Équivalences logiques utiles:ELU)
  $p \vee \neg p \Leftrightarrow \mathbf{T} \qquad p \wedge \neg p \Leftrightarrow \mathbf{F}$
  $(p \rightarrow q) \Leftrightarrow (\neg p \vee q)$

Preuve de tautologie

$[\neg p \wedge (p \vee q)] \rightarrow q$

$\Leftrightarrow [(\neg p \wedge p) \vee (\neg p \wedge q)] \rightarrow q$	Distributivité
$\Leftrightarrow [\mathrm{F} \vee (\neg p \wedge q)] \rightarrow q$	ELU
$\Leftrightarrow [\neg p \wedge q] \rightarrow q$	Identitité
$\Leftrightarrow \neg [\neg p \wedge q] \vee q$	ELU
$\Leftrightarrow [\neg(\neg p) \vee \neg q] \vee q$	DeMorgan
$\Leftrightarrow [p \vee \neg q] \vee q$	Double Négation
$\Leftrightarrow p \vee [\neg q \vee q]$	Associativité
$\Leftrightarrow p \vee [q \vee \neg q]$	Commutativité
$\Leftrightarrow p \vee \mathrm{T}$	ELU
$\Leftrightarrow \mathrm{T}$	Domination

Autre preuve d'équivalence logique

$\neg(p \vee (\neg p \wedge q)) \Leftrightarrow (\neg p \wedge \neg q)$

$\Leftrightarrow \neg p \wedge \neg(\neg p \wedge q)$	De Morgan
$\Leftrightarrow \neg p \wedge [\neg(\neg p) \vee \neg q]$	De Morgan
$\Leftrightarrow \neg p \wedge (p \vee \neg q)$	Double Négation
$\Leftrightarrow (\neg p \wedge p) \vee (\neg p \wedge \neg q)$	Distributivité
$\Leftrightarrow \mathrm{F} \vee (\neg p \wedge \neg q)$	ELU
$\Leftrightarrow (\neg p \wedge \neg q) \vee \mathrm{F}$	Commutativité
$\Leftrightarrow (\neg p \wedge \neg q)$	Identité

Définir des opérateurs par les Équivalences

Avec les équivalences, de nouveaux opérateurs peuvent être déduits en termes d'autres opérateurs:

- OU Exclusif: $p \oplus q \Leftrightarrow (p \lor q) \land \neg(p \land q)$
  $p \oplus q \Leftrightarrow (p \land \neg q) \lor (q \land \neg p)$
- Implication: $p \rightarrow q \Leftrightarrow \neg p \lor q$
- Biconditionnelle:

$$p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \land (q \rightarrow p)$$
$$p \leftrightarrow q \Leftrightarrow \neg(p \oplus q)$$

Revue: Logique Propositionnelle sec. (1.1-1.2)

- Propositions atomiques : p, q, r, …
- Opérateurs Booléens : $\neg \land \lor \oplus \rightarrow \leftrightarrow$
- Propositions composées:

  $s :\equiv (p \land \neg q) \lor r$
- Équivalences: $p \land \neg q \Leftrightarrow \neg(p \rightarrow q)$
- Preuves d'équivalences par:
  - Tables de vérité.
  - Dérivations symboliques. $p \Leftrightarrow q \Leftrightarrow r$ …

Logique des prédicats (sec.1.3)

- La *logique des prédicats* est une extension de la logique propositionnelle permettant un raisonnement concis sur diverses classes d'entités.
- La logique propositionnelle traite des *propositions* simples (phrases) comme des entités atomiques.
- Par contre, la *logique des prédicats* distingue le sujet d'une phrase de son prédicat.
  - Termes de la grammaire Française?

Sujets et Prédicats

- Dans la phrase "La vache rumine":
  - La phrase "La vache" correspond au *sujet* - l'o*bjet* ou l'e*ntité* par rapport auquel la phrase complète est construite.
  - La phrase "rumine" correspond au *prédicat*- une propriété qui est VRAIE pour le sujet.
- En logique des prédicats, un *prédicat* représente une fonction $P(\cdot)$ qui transpose les objets en propositions.
  - $P(x)$ = "x rumine" (ou x est un objet quelconque).

Conventions associées aux prédicats

- Convention: Variables en minuscules *x*, *y*, *z*... Dénotent les objets/entités; variables en majuscules *P*, *Q*, *R*… dénotent les prédicats.
- Il faut savoir que l'application d'un prédicat *P* à un objet *x* est une *proposition* *P*(*x*). Mais le prédicat *P* lui même (ex: *P*="rumine") n'est pas une proposition (pas une phrase complète).
  - *Ex:* si *P*(*x*) = "*x* est un nombre premier", *P*(3) est la *proposition* "3 est un nombre premier."
- En Java: les predicats sont des méthodes qui retournent des valeurs **`booléenne`**
  - **`someLinkedList.`*`isEmpty`*`()`**
  - ***`isPrime`*`(17)`**

Applications de la logique des prédicats

- Notation formelle pour écrire des *définitions* mathématiques, axiomes et théorêmes clairs, concis, et non ambigüs.
- La logique des prédicats peut avec une notation très simple (symboles de fonctions, opérateur "=" , des quantifieurs, et quelques règles de construction de preuves) définir de multiples systèmes mathématiques et prouver le bon fonctionnement de ce système.

Applications pratiques de la logique des prédicats

- C'est la base pour exprimer les spécifications formelles de systèmes complexes.

- Utiliser comme outil de validation automatique de preuves et dans de multiples systèmes en Intelligence Artificielle.
  - *Ex:* Systèmes de vérification automatique de programmes.
- Des énoncés en logique des prédicats sont supportés par des interpréteurs de requêtes de BD et autres librairies de classes
  - Ils sont perçus comme des outils de programmation.

Univers du discours

- La possibilité de distinguer des objets à partir de prédicats permet de faire des affirmations sur un ensemble d'objets en un seul coup.
- Ex: posons $P(x)$="$x+1>x$". Nous pouvons alors dire: "Pour tout nombre x, $P(x)$ est VRAI" au lieu de faire l'énumération: $(\mathbf{0}+1>\mathbf{0}) \wedge (\mathbf{1}+1>\mathbf{1}) \wedge (\mathbf{2}+1>\mathbf{2}) \wedge \ldots$
- La collection des valeurs de x est appelée l'univers du discours (domaine) de x.

Univers du discours

- Dans les langages de programmation le ud correspond à la notion de **type**. En JAVA, il existe deux catégories de type: : *type reference (*objets et arrays), et des *types primitifs* comme **`int`**, **`boolean`**, **`char`**, etc.
- Exemples de ud en Java:

  `int, char, int[][], Object, String, java.util.LinkedList, Exception,` etc.

Quantificateurs

- Les *quantificateurs* sont une notation permettant de *quantifier* (compter) *combien d'*objects dans l'univers du discours satisfont un prédicat donné.
- "$\forall$" POUR TOUT ou quantificateur *universel*.
  $\forall x\, P(x)$ Pour tout x, *P* est satisfait.
- "$\exists$" EXISTE or quantificateur *existentiel*.
  $\exists x\, P(x)$ il *existe* une valeur de *x* *tel que* $P(x)$ est VRAI.

Quantificateur Universel $\forall$

- Exemple:
  Avec le u.d. de x les étudiants en informatique.

- Posons S(x) le prédicat “x est dans le cours INF1130”
- Posons *P*(*x*) le prédicat “*x* étudient la logique des prédicats.”

- Avec le quantificateur *universel*, $\forall x$ (S(x) -> *P*(*x*)), est la *proposition:*
  - “Tous les étudiants en informatique du cours INF1130 étudient la logique des prédicats”
  - *i.e.*, “Tous les étudiants du cours INF1130 étudient la logique des prédicats.”
  - *i.e.*, “Chaque étudiant du cours INF1130, étudie la logique des prédicats.”

Quanficateur Existentiel $\exists$

- Exemple: Posons l’u.d. de x est: sièges au spectacle de Billy Talent
- Posons *P*(*x*) le *prédicat* “*x* est réservé”
  Alors la quantification *existentielle de P*(*x*), $\exists x$ *P*(*x*), donne la *proposition*:
  - “Quelque sièges sont réservés pour le spectacle.”
  - “Un siège est réservé pour le spectacle.”
  - “Au moins un siège est réservé pour le spectacle.”

Négations

- $\forall x$: Que représente $\neg\forall x$?
- $\exists x$: Que représente $\neg\exists x$?
- Que représente la négation de $\forall x\ P(x)$?
- Ou la négation de $\exists x\ P(x)$?
- $\neg(\forall x\ P(x))$: $\exists x\ \neg P(x)$
- $\neg\ (\exists x\ P(x))$: $\forall x\ \neg P(x)$

Variables libres ou liées

- Une expression comme *P*(*x*) possède une *variable libre x* (*x* est non définie).
- Un quantificateur ($\forall$ ou $\exists$) *opèrent* sur une expression ayant une ou plusieurs variables libres, et *lie* une ou plusieurs de ces variables, pour produire une expression ayant une ou plusieurs *variables liées*.

Exemples de liaisons

- $P(x,y)$ a 2 variables libres, x et y.
- $\forall x\ P(x,y)$ a 1 variable libre, et une variable liée.
- "$P(x)$, où x=3" permet de lier x.
- Une expression avec <u>aucune</u> variable libre est une proposition réelle.
- Une expression avec <u>une ou plus</u> variables libres ne reste qu'un prédicat:

  Ex: posons $Q(y) = \forall x\ P(x,y)$

Quantificateurs imbriqués (sec. 1.4)

Exemple: Posons l'u.d. de x et y sont des personnes.

Posons A(x,y)="x aime y" (prédicat: 2 f.v., x,y)

Alors $\exists y\ A(x,y)$ = "Quelqu'un est aimé par x." (prédicat: 1 f.v., x)

Alors $\forall x\ (\exists y\ A(x,y))$ =

"Chacun a quelqu'un qu'il aime."

(Une **Proposition** avec **0** variable libre.)

Exemple: Quantificateur

$\forall x$: Pour tout x

$\exists y$: Il existe un 'y'

Si C(*x*,*y*)="*x* compte sur *y*," s'exprime en français non ambigue :

$\forall x(\exists y\, C(x,y))$= Chacun a quelqu'un sur qui compter.

$\exists y(\forall x\, C(x,y))$= Il existe quelqu'un sur qui tous peuvent compter même lui-même.

$\exists x(\forall y\, C(x,y))$= Il existe quelqu'un qui compte sur tous même lui-même.

$\forall y(\exists x\, C(x,y))$= Tous ont quelqu'un qui compte sur eux.

$\forall x(\forall y\, C(x,y))$= Tous comptent sur tous même eux-mêmes.

D'autres conventions

- Parfois l'u.d. est restreint à l'intérieure de la quantification, *ex:*,
  - $\forall x>0\ P(x)$ correspond à "Pour tout $x > 0$, $P(x)$."
    $=\forall x\ (x>0 \rightarrow P(x))$
  - $\exists x>0\ P(x)$ correspond à "Il existe un $x > 0$ tel que $P(x)$."
    $=\exists x\ (x>0 \wedge P(x))$

Un peu plus sur la liaison

- $\forall x\ \exists x\ P(x)$ - x n'est pas une f.v. dans $\exists x\ P(x)$, alors la liaison $\forall x$ n'est pas utilisée.
- $(\forall x\ P(x)) \wedge Q(x)$ - La variable x est hors de la portée du quantificateur $\forall x$, donc x est une f.v.. Une proposition incomplète.
- $(\forall x\ P(x)) \wedge (\exists x\ Q(x))$ – Légale, puisque nous avons 2 x différents

Règles d'équivalence des quantificateurs

- Définitions des quantificateurs: Si l'u.d.=a,b,c,…
  $\forall x\ P(x) \Leftrightarrow P(\mathrm{a}) \wedge P(\mathrm{b}) \wedge P(\mathrm{c}) \wedge \ldots$
  $\exists x\ P(x) \Leftrightarrow P(\mathrm{a}) \vee P(\mathrm{b}) \vee P(\mathrm{c}) \vee \ldots$
- Nous pouvons alors prouver les règles:
  $\forall x\ P(x) \Leftrightarrow \neg\exists x\ \neg P(x)$
  $\exists x\ P(x) \Leftrightarrow \neg\forall x\ \neg P(x)$
- Quelle règle d'équivalence *propositionnelle* peut être utilisée?

DeMorgan's

D’autres règles d’équivalence

- $\forall x\, \forall y\, P(x,y) \Leftrightarrow \forall y\, \forall x\, P(x,y)$
  $\exists x\, \exists y\, P(x,y) \Leftrightarrow \exists y\, \exists x\, P(x,y)$
- $\forall x\, (P(x) \wedge Q(x)) \Leftrightarrow (\forall x\, P(x)) \wedge (\forall x\, Q(x))$
  $\exists x\, (P(x) \vee Q(x)) \Leftrightarrow (\exists x\, P(x)) \vee (\exists x\, Q(x))$
- Exercise:
  Voir à prouver ces propositions.
  – Quelles règles d’équivalences propositionnelles utilisées vous?

Exemple

- $\forall x(C(x) \vee \exists y(C(y) \wedge F(x,y)))$ où
  – C(x): “x a un portable”
  – F(x,y): “x et y sont amis”
  – U.d. de x et y sont les étudiants de l’école
- $\forall x(C(x) \vee \exists y(C(y) \wedge F(x,y)))$
  – Pour tout x, x a un portable OU
  – Il existe un y qui a aussi un portable ET x et y sont amis

Autre exemple

- Si <u>une personne est femme</u> et <u>est un parent</u>, alors <u>cette personne est la mère de quelqu'un</u>.
- F(x): "x est une femme", P(x): "x est un parent"
- M(x,y): "x est la mère de y"
- Pour chaque personne x, Si x est un parent, alors x est la mère de quelqu'un.
- $\forall x \, ((F(x) \wedge P(x)) \rightarrow \exists y M(x,y))$

Revue: Logique des prédicats (sec 1.3)

- Objets *x*, *y*, *z*, …
- Prédicats *P*, *Q*, *R*, … sont des fonctions qui associent les objets *x* aux propositions *P*(*x*).
- Prédicats multi-valués *P*(*x*, *y*).
- Quantificateurs: ($\forall x\, P(x)$) ="Pour tous *x*' *P*(*x*)."
  ($\exists x\, P(x)$)="Il existe des *x* tel que *P*(*x*)."