Introduction

- La logique est utile dans beaucoup de domaines :
  - Conception de circuits.
  - Preuves de programmes.
  - Programmation logique.
  - Simulation de raisonnements en intelligence artificielle.
  - ...
- Nous n'utiliserons que le *calcul des propositions* :
  bien que limité, c'est la première étape dans la définition de la logique et du raisonnement.
- Le *calcul des prédicats* qui englobe le *calcul des propositions* et qui permet une formalisation achevée du raisonnement mathématique, ne sera pas abordé.

Introduction

- Une *proposition* est une affirmation du type "il pleut" ou "2+2=3" .
  On peut lui affecter une valeur de vérité : *vrai* ou *faux*.
- Une proposition ne contient ni des variables,
  ni des quantificateurs.
- Un *prédicat* est une proposition dont la valeur de vérité dépend de variables.
  Par exemple “L'étudiant.e XY habite à Paris”
  est vrai ou faux en fonction de l'étudiant.
  En calcul des prédicats on utilise les *quantificateurs* :
  “Tout étudiant.e habite à Paris”.
- Le calcul des propositions traite du raisonnement sur les propositions. Il définit les règles de déduction qui relient les propositions, tout ceci *indépendamment* de leur contenu.
- On ne traite que des valeurs booléennes {vrai, faux} ou {1,0}.

Syntaxe et sémantique

Dans les théories de la logique mathématique, en particulier en calcul des propositions, on considère deux aspects :

- La syntaxe, où l'on définit le langage du calcul des propositions par les *règles d'écriture des formules*.
- La sémantique qui détermine les règles d'*interprétation des formules*. On attribue des valeurs de vérité (vrai/faux) aux propositions élémentaires et on explique comment les connecteurs se comportent vis-à-vis de ces valeurs de vérité. On exprime souvent ce comportement par une table de vérité.

Exemple de syntaxe et sémantique

On considère les phrases dans une langue :

- La syntaxe fixe les règles d'écriture des phrases.

  PHRASE = (SUJET | VERBE | COMPLÉMENT)
- La sémantique permet l'interprétation des phrases.
- *Exemples :*
  1. "le chat boit son lait"
  2. "le fermier conduit un troupeau"
  3. "le lait boit son chat"
  4. "un troupeau conduit le fermier"

  Une phrase dont la syntaxe est correcte, n'a pas nécessairement un sens

Les constituants du langage

La syntaxe du calcul des propositions utilise

- Les **variables propositionnelles** ou **propositions atomiques**.
  Notées $p_1, p_2, \ldots$ ou p, q, r,...

- Les **opérateurs** ou **connecteurs**.
  Ils permettent la construction de propositions plus complexes.

$\neg$	*non*	négation
$\wedge$	*et*	conjonction
$\vee$	*ou*	disjonction
$\rightarrow$ ou $\Rightarrow$	*implique*	implication
$\leftrightarrow$ ou $\Leftrightarrow$	*équivaut*	équivalence

- La ponctuation "(" et ")", les parenthèses permettent de lever les ambiguïtés.

Ces éléments constituent *l'alphabet* du calcul propositionnel.

Les formules propositionnelles

Grâce à cet alphabet, on peut construire l'ensemble des mots qui est l'ensemble des suites finies d'éléments de l'alphabet.

L'ensemble des **formules** ou **expressions bien formées** du calcul des propositions est le plus petit ensemble de mots tel que

1. les variables propositionnelles, ou atomes, sont des formules ;
2. si A est une formule, alors $\neg A$ est une formule ;
3. si A et B sont des formules, alors $(A * B)$ est une formule,
   où $*$ est l'un des connecteurs binaires, $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

Exemple

- Considérons la formule $(((\neg a \vee b) \wedge c) \rightarrow (\neg\neg a \wedge \neg b))$

  On omet en général les parenthèses extrêmes des formules, d'où

  $$((\neg a \vee b) \wedge c) \rightarrow (\neg\neg a \wedge \neg b)$$

- Vérification de la cohérence des parenthèses :
  On attribue un poids +1 à la parenthèse ouvrante, un poids -1 à la parenthèse fermante et 0 aux autres symboles.
  La somme des poids d'une formule est alors nulle.
  Attention : ceci ne suffit pas à garantir que la formule est syntaxiquement correcte.

- Les parenthèses sont importantes pour lever les ambiguïtés lorsque l'on utilise des connecteurs binaires.

  Exemple : comment interpréter $p \rightarrow q \rightarrow r$?
  Soit $(p \rightarrow q) \rightarrow r$, soit $p \rightarrow (q \rightarrow r)$.

Distribution de vérité

- Une distribution de vérité δ est une application de l'ensemble des variables propositionnelles dans l'ensemble des valeurs de vérité $\{0, 1\}$.

  Exemple : $\delta \ : \ \{a, b\} \longrightarrow \{0, 1\}$,
  or comme $\delta(a) = 0$ ou 1, et de même pour $\delta(b)$, il y a $2^2 = 4$ distributions de vérité possible :

a	0	0	1	1
b	0	1	0	1

- Une distribution δ étant fixée, on définit $\delta(F)$, ou $\text{val}(F, \delta)$, pour toute formule F, à partir des tables de vérité.

  Une distribution de vérité δ se prolonge ainsi en une application de l'ensemble des formules dans $\{0, 1\}$.

Tables de vérité des connecteurs

- On peut écrire $\delta(\neg a)$:

a	$\neg a$
1	0
0	1

- Les tables de vérité des connecteurs binaires sont

a	b	$a \wedge b$	$a \vee b$	$a \rightarrow b$	$a \leftrightarrow b$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

- On peut ainsi donner le comportement associé à chaque formule :
  - $\neg a$ prend la valeur 1 si et seulement si a prend la valeur 0.
  - $a \rightarrow b$ prend la valeur 0 si et seulement si
    a prend la valeur 1 et b prend la valeur 0.

    $\vdots$

Sémantique : valeurs d'une formule

- Une distribution δ donnée est un modèle de F si $\delta(F) = 1$.

  Exemple : la distribution $\delta(a) = 1$ et $\delta(b) = 0$ est un modèle pour la formule $F = a \vee b$.

- Une formule F est une tautologie si pour toute distribution δ, on a $\delta(F) = 1$.

  On dit aussi que F est **valide**. On note $\models F$.

  Exemple : La formule $a \vee \neg a$ est une tautologie, donc $\models (a \vee \neg a)$

- Une formule F est une antilogie si pour toute distribution δ, on a $\delta(F) = 0$.

  On dit aussi que F est une **contradiction** ou **insatisfaisable**.

  Exemple : La formule $a \wedge \neg a$ est une antilogie.

Sémantique : valeurs d'une formule

- Si la formule F prend au moins une fois la valeur 1, on dit que F est **satisfaisable**.
- Deux formules F et G sont équivalentes
  si et seulement si pour toute distribution δ on a $\delta(F) = \delta(G)$.
  On note "$F \,\underline{eq}\, G$" la propriété
  "les formule F et G sont équivalentes".
- Attention : ne pas confondre "$\leftrightarrow$" et $\underline{eq}$.
  Le premier est un symbole du *langage formel*,
  le second un symbole du *métalangage*.

  Ainsi $(F \leftrightarrow G)$ est une formule,
  tandis que "$F \,\underline{eq}\, G$" énonce une propriété des formules F et G.
  Et "$F \,\underline{eq}\, G$" est un énoncé équivalent à $\models F \leftrightarrow G$.

Exemple 1

Évaluons $A = ((\neg a \vee b) \wedge c) \rightarrow (\neg\neg a \wedge \neg b)$.

On pose $F = ((\neg a \vee b) \wedge c)$ et $G = (\neg\neg a \wedge \neg b)$.

a	b	c	$\neg a \vee b$	F	G	$F \rightarrow G$
0	0	0	1	0	0	1
0	0	1	1	1	0	0
0	1	0	1	0	0	1
0	1	1	1	1	0	0
1	0	0	0	0	1	1
1	0	1	0	0	1	1
1	1	0	1	0	0	1
1	1	1	1	1	0	0

La formule A est satisfaisable et la distribution $(0, 0, 0)$ est un modèle de A.

Exemple 2

Évaluons $B = ((p \vee r) \wedge (q \vee \neg r)) \rightarrow (p \vee q)$.

On pose $F = (p \vee r) \wedge (q \vee \neg r)$.

p	q	r	p ∨ r	q ∨¬ r	F	p∨q	*B*
0	0	0	0	1	0	0	1
0	0	1	1	0	0	0	1
0	1	0	0	1	0	1	1
0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1
1	0	1	1	0	0	1	1
1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1

La formule B est une tautologie : $\models ((p \vee r) \wedge (q \vee \neg r)) \rightarrow (p \vee q)$

Quelques tautologies

Pour toutes formules A, B :

$\models (A \rightarrow A)$ *identité*

$\models (A \vee \neg A)$ *tiers exclus*

$\models (A \rightarrow (A \vee B))$

$\models (A \wedge \neg A) \rightarrow B$

$\models ((A \wedge B) \rightarrow A)$

$\models [A \rightarrow (B \rightarrow A)]$

Pour toutes formules A, B et C :

$\models (A \rightarrow B) \rightarrow [(B \rightarrow C) \rightarrow (A \rightarrow C)]$ *exemple de syllogisme*

$\models \{[A \rightarrow (B \rightarrow C)] \rightarrow [(A \rightarrow B) \rightarrow (A \rightarrow C)]\}$

$\models \{(\neg B \rightarrow \neg A) \rightarrow [(\neg B \rightarrow A) \rightarrow B)]\}$

Formules équivalentes

Lois de DE MORGAN

Pour toutes formules A et B :

- On lit «*non(A et B) équivaut à (non A) ou (non B)*»

$$\neg(A \wedge B) \ \underline{eq} \ (\neg A) \vee (\neg B)$$

- On lit «*non(A ou B) équivaut à (non A) et (non B)*»

$$\neg(A \vee B) \ \underline{eq} \ (\neg A) \wedge (\neg B)$$

Formules équivalentes

Pour toutes formules propositionnelles p, q et r :

- $(\neg\neg p) \ \underline{eq} \ p$ *idempotence de* $\neg$
- $((p \wedge q) \vee r)) \ \underline{eq} \ ((p \vee r) \wedge (q \vee r))$ *distributivité de* $\vee$ *par rapport à* $\wedge$
- $((p \vee q) \wedge r) \ \underline{eq} \ ((p \wedge r) \vee (q \wedge r))$ *distributivité du* $\wedge$ *par rapport à* $\vee$
- $(p \rightarrow q) \ \underline{eq} \ (\neg q \rightarrow \neg p)$ *contraposition*
- $(p \rightarrow q) \ \underline{eq} \ (\neg p \vee q)$
- $\neg(p \rightarrow q) \ \underline{eq} \ (p \wedge \neg q)$
- $(p \leftrightarrow q) \ \underline{eq} \ ((p \rightarrow q) \wedge (q \rightarrow p))$
- $(p \leftrightarrow q) \ \underline{eq} \ (p \wedge q) \vee (\neg p \wedge \neg q)$

Systèmes complets de connecteurs

Un système de connecteurs est dit complet si toute formule valide (syntaxe correcte) du calcul des propositions peut se démontrer à partir des connecteurs du système.

Exemple 1 : le système $\{\neg, \wedge, \vee\}$ est un système complet de connecteurs.
En effet, les équivalences citées permettent de montrer ceci.

Exemple 2 : grâce aux Lois de De Morgan, les systèmes $\{\neg, \wedge\}$ et $\{\neg, \vee\}$ sont complets.

Exemple 3 : comme $(p \vee q)$ $\underline{eq}$ $(\neg p \rightarrow q)$ le système $\{\neg, \rightarrow\}$ est complet aussi.

Application : Lors de la construction de circuits logiques ceci permet d'utiliser un nombre limité de “circuits de base”.

Sous formules

- Les règles syntaxiques d'écriture des formules du calcul propositionnel permettent de construire des formules.
- On a souvent besoin de simplifier une formule ou de l'écrire sous une forme normalisée (circuits logiques, démonstration automatique).
- On introduit la notion de sous formule :
  - Si A est une formule, alors $\neg A$ est une formule.
    On dit que "$\neg$" est le connecteur principal de $\neg A$
    et que A est la sous-formule (immédiate) de $\neg A$.
  - Si A et B sont des formules, alors $(A * B)$ est une formule,
    où $*$ est l'un des connecteurs binaires, $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.
    On dit que «$*$» est le connecteur principal de $(A * B)$
    et que A et B sont les deux sous-formules (immédiates) de $(A * B)$.
- Par induction :
  Les sous-formules des sous-formules d'une formule A sont des sous-formules de A.

Sous formules

- Si B est une sous-formule de A, et si B' $\underline{eq}$ B,
alors la formule A', obtenue en remplaçant B par B' dans A,
est équivalente à A.

- *Exemple :* Soit $A = ((\neg a \wedge b) \vee c) \rightarrow (\neg\neg a \vee \neg b)$.
On sait que $(\neg\neg a)$ $\underline{eq}$ a.
Donc A $\underline{eq}$ $[((\neg a \wedge b) \vee c) \rightarrow (a \vee \neg b)]$.

- Ceci peut permettre de simplifier les formules,
i.e. de les remplacer par des formules équivalentes plus courtes.

Formes normales

- On appelle littéral ou atome une formule qui est soit une variable propositionnelle, soit la négation d'un telle variable :
si p et q sont des variables propositionnelles, alors p et $\neg q$ sont des atomes.

- Une formule est sous forme normale *disjonctive* (FND) si elle est écrite sous forme de disjonctions de conjonctions de littéraux.

  Exemples : p, $(p \vee (\neg q))$, $(p \wedge q) \vee (\neg q)$ sont des FND.
Mais pas $\neg(p \vee q)$, ni $p \vee (q \wedge (r \vee s))$.

- Une formule est sous forme normale *conjonctive* (FNC) si elle est écrite sous forme de conjonctions de disjonctions de littéraux.

  Exemples : q, $(p \wedge (\neg q))$, $(p \vee q) \wedge (\neg q)$ sont des FND.
Mais pas $\neg(p \wedge (\neg q))$, ni $p \wedge (q \vee (r \wedge s))$.

Formes normales

- Toute formule du calcul propositionnel est équivalente à une formule sous forme normale *disjonctive* et aussi à une formule sous forme normale *conjonctive*.
- Pour convertir une formule en forme normale, on utilise les lois de De Morgan, la distributivité des opérations $\vee$ et $\wedge$, l'une par rapport à l'autre, et l'idempotence de $\neg$.
- L'écriture sous forme normale peut agrandir la formule de manière exponentielle.

  Exemple : $(p_1 \vee q_1) \wedge (p_2 \vee q_2)$ est sous FNC et il y a 2 termes.

  Sa FND comporte 2^2 termes :

$$(p_1 \wedge p_2) \vee (q_1 \wedge p_2) \vee (p_1 \wedge q_2) \vee (q_1 \wedge q_2)$$

- Grâce aux tables de vérité, on peut facilement trouver les FND et FNC.

Construction de formes normales disjonctives par table de vérité

- On se donne la formule F grâce à sa table de vérité.
- On cherche les distributions de vérité δ qui sont un modèle pour F, c'est-à-dire les lignes où l'on trouve un “1”.
- Pour chaque modèle de F, on écrit la **conjonction** des littéraux tels que l'on obtient “1”.

  Exemple : Si $\delta(a) = 0$ et $\delta(b) = 1$ est un modèle, alors la conjonction qui est vraie pour cette distribution s'écrit $(\neg a) \wedge b$.
- On écrit ensuite la **disjonction** de toutes les conjonctions.
- Cette méthode garantit la construction d'une formule, sous **forme normale disjonctive**, équivalente à F.

FND par table de vérité. Exemple

Soit la formule $A = ((\neg a \vee b) \wedge c) \rightarrow ((\neg\neg a) \wedge (\neg b))$.

On pose $F = ((\neg a \vee b) \wedge c)$ et $G = ((\neg\neg a) \wedge (\neg b))$.

a	b	c	$\neg a \vee b$	F	G	$A = F \rightarrow G$
0	0	0	1	0	0	1
0	0	1	1	1	0	0
0	1	0	1	0	0	1
0	1	1	1	1	0	0
1	0	0	0	0	1	1
1	0	1	0	0	1	1
1	1	0	1	0	0	1
1	1	1	1	1	0	0

On obtient la FND

$$(\neg a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge \neg b \wedge c) \vee (a \wedge b \wedge \neg c)$$

Simplification des FND

- On constate que la FND de la formule A est assez longue. Souvent il est possible de simplifier l'expression obtenue.
- La FND de A est

  $$(\neg a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge \neg b \wedge c) \vee (a \wedge b \wedge \neg c)$$

- Il y a 5 termes :
  - les termes 1, 2, 3 et 5 contiennent tous $\neg c$, que l'on peut mettre en facteur ;
  - la disjonction de ces 4 termes équivaut à

    $$\neg c \wedge [(\neg a \wedge \neg b) \vee (\neg a \wedge b) \vee (a \wedge \neg b) \vee (a \wedge b)] \ ;$$

  - ce qui est équivalent à $\neg c$;
  - la formule A est donc équivalente à $\neg c \vee (a \wedge \neg b \wedge c)$;
  - Finalement

    $$A \ \underline{eq} \ (\neg c \vee (a \wedge \neg b)) \ .$$

Construction de formes normales conjonctives par table de vérité

- On se donne la formule F grâce à sa table de vérité.
- On cherche les distributions de vérité δ qui ne sont pas un modèle pour F, c'est-à-dire les lignes où l'on trouve un "0".
- Pour chaque telle distribution, on écrit la **disjonction** des littéraux tels que l'on obtient "0".

  Exemple : Si $\delta(a) = 0$ et $\delta(b) = 1$, alors la disjonction qui est fausse pour cette distribution s'écrit $a \vee (\neg b)$.
- On écrit ensuite la **conjonction** de toutes les disjonctions.
- Cette méthode garantit la construction d'une formule, sous **forme normale conjonctive**, équivalente à F.

FNC par table de vérité. Exemple

Soit la formule $A = ((\neg a \vee b) \wedge c) \to ((\neg\neg a) \wedge (\neg b))$.

On pose $F = ((\neg a \vee b) \wedge c)$ et $G = ((\neg\neg a) \wedge (\neg b))$.

a	b	c	$\neg a \vee b$	F	G	$A = F \to G$
0	0	0	1	0	0	1
0	0	1	1	1	0	0
0	1	0	1	0	0	1
0	1	1	1	1	0	0
1	0	0	0	0	1	1
1	0	1	0	0	1	1
1	1	0	1	0	0	1
1	1	1	1	1	0	0

On obtient la FNC

$$(a \vee b \vee \neg c) \wedge (a \vee \neg b \vee \neg c) \wedge (\neg a \vee \neg b \vee \neg c)$$

Simplification des FNC

- La FNC de A est

$$(a \vee b \vee \neg c) \wedge (a \vee \neg b \vee \neg c) \wedge (\neg a \vee \neg b \vee \neg c)$$

- Il y a 3 termes :
  - pour les termes 1 et 2 :

$$(a \vee b \vee \neg c) \wedge (a \vee \neg b \vee \neg c) \ \underline{eq} \ (a \vee \neg c)$$

  - la formule A est donc équivalente à

$$(a \vee \neg c) \wedge (\neg a \vee \neg b \vee \neg c)$$

  - Finalement $\quad A \ \underline{eq} \ (a \vee \neg c) \wedge (\neg b \vee \neg c)$
- En factorisant la FNC, on retrouve la FND $(\neg c \vee (a \wedge \neg b))$.
- La simplification des FND et FND n'est pas toujours facile, on utilise des méthodes spécifiques comme les *tables de Karnaugh*.

Algèbre de Boole

- Le nom d'**algèbre de Boole** ou **calcul booléen** est en l'honneur du mathématicien britannique George Boole (1815-1864), considéré comme créateur de la logique moderne.
- Le calcul booléen appliqué au calcul des propositions permet une approche algébrique pour traiter les formules logiques.
- On introduit les opérateurs arithmétiques binaires "+" pour $\vee$ et "." pour $\wedge$. L'opérateur $\bar{a}$ représente l'opération unaire $\neg a$ du calcul des propositions.

  Exemple : $(a \vee \neg c) \wedge (\neg a \vee \neg b \vee \neg c)$ s'écrit $(a + \bar{c}).(\bar{a} + \bar{b} + \bar{c})$
- La structure d'algèbre de Boole s'applique dans d'autres cadres, pas étudiés dans ce cours.
  Un exemple est l'*algèbre de Boole des parties d'un ensemble* E, $\mathcal{P}(E)$. L'opération"+" est alors la réunion $\cup$, l'opération "." est l'intersection $\cap$ et $\bar{A}$ désigne le complément de A dans E.