Cours 5

Sémantique : sans tables de vérité

Les formules valides expriment les lois de la logique. Construire des tables de vérité est un bon moyen pour découvrir si des formules sont valides. Mais c'est un moyen qui a ses limites :

> Pour une formule à 100 variables, la table de vérité aura $2^{100} \approx 1.26 * 10^{30}$ lignes et le calcul de cette table dépasse les possibilités de toute machine. Avec un processeur 10 Ghtz générant et évaluant une ligne de table de vérité par cycle a besoin de plus de 10^{13} années pour vérifier toute la table.

4.1 Décidabilité algorithmique du calcul propositionnel

Une logique est décidable s'il existe un algorithme (calcul réalisable sur un ordinateur qui termine toujours pour toute donnée) qui permet de savoir pour chaque formule si elle est une tautologie(valide) ou pas.

Théorème 1. *Le calcul propositionnel est décidable.*

Démonstration. Méthode des tables de vérité : calculer la table de vérité prenant en argument les symboles propositionnels de F et calculer pour chaque valuation(assignation) possible la valeur de F. Coût : $O(2^n)$ avec n le nombre de variables de F.

4.2 Substitution et remplacement

4.2.1 Substitution

Une substitution est une application de l'ensemble des variables dans l'ensemble des formules. L'application d'une substitution σ à une formule, consiste à remplacer dans la formule toute variable x par la formule $\sigma(x)$. Soit A une formule, on note A_σ l'application de la substitution σ à la formule A.

Lemme 1 (propriété des substitutions)**.** *Soit A une formule, v une assignation et σ une substitution.*
On a : $[A\sigma]_v = [A]_w$ où pour toute variable $x, w(x) = [\sigma(x)]_v$.

Exercice. Montrer le Lemme 1 par induction structurelle.

Théorème 2. *L'application d'une substitution à une formule valide donne une formule valide.*

Exemple 1. *Soit A la formule $\neg(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$. Cette formule est valide. Soit σ la substitution suivante : $\sigma(p) = (a \vee b); \sigma(q) = (c \wedge d)$. La formule $A\sigma$ est $\neg((a \vee b) \wedge (c \wedge d)) \Leftrightarrow (\neg(a \vee b) \vee \neg(c \wedge d))$. Cette formule est aussi valide.*

4.2.2 Remplacement

Définition 1 (Remplacement)**.** *Soient A, B, C, D des formules. La formule D est obtenue en remplaçant des occurrences de A dans C par B, s'il existe une formule E et une variable x telles que, $C = E < x := A >$ et $D = E < x := B >$.*

Théorème 3. *Soit C une formule et D la formule obtenue en remplaçant dans C, des occurrences de la formule A par la formule B. On a : $(A \Leftrightarrow B) \models (C \Leftrightarrow D)$.*

Preuve. Par définition du remplacement, il existe une formule E et une variable x telles que, $C = E < x := A >$ et $D = E < x := B >$. Supposons que v est une assignation modèle de $(A \Leftrightarrow B)$. Nous avons donc $[A]_v = [B]_v$. D'après le lemme 1 :
— $[C]_v = [E]_w$ où w est identique à v sauf que $w(x) = [A]_v$.
— $[D]_v = [E]_{w'}$ où w' est identique à v sauf que $w'(x) = [B]_v$.
Puisque $[A]_v = [B]_v$, les assignations w et w' sont identiques, donc $[C]_v = [D]_v$. Par suite v est modèle de $(C \Leftrightarrow D)$.

Corollaire 4. *Soit C une formule et D la formule obtenue en remplaçant dans C, une occurrence de la formule A par la formule B. On a : si $A \equiv B$ alors $C \equiv D$.*

Preuve. Si $A \equiv B$, alors la formule $(A \Leftrightarrow B)$ est valide, donc la formule $(C \Leftrightarrow D)$ également puisqu'elle est, d'après le théorème ci-dessus, la conséquence de $(A \Leftrightarrow B)$, par suite $C \equiv D$.

Exemple 2. *1. D'après le théorème 3 : $p \Leftrightarrow q \models (p \vee (\boxed{\boldsymbol{p}} \Rightarrow r)) \Leftrightarrow (p \vee (\boxed{\boldsymbol{q}} \Rightarrow r))$*
2. D'après le corollaire 4 : $(\neg(p \vee q) \Rightarrow (\boxed{\neg(p \vee q)} \vee r)) \equiv (\neg(p \vee q) \Rightarrow (\boxed{(\neg p \wedge \neg q)} \vee r))$

La différence entre substitution et remplacement est qu'une substitution remplace un ensemble de variables par des formules alors qu'un remplacement remplace les *occurrences* de certaines formules par une autre formule en utilisant des substitutions.

4.3 Transformation en somme de monômes

4.3.1 Formes normales

Littéral, monôme, clause

Un littéral est une variable ou la négation d'une variable.
Par exemple $x, y, \bar{z}$ sont des littéraux.
Un monôme est un produit de littéraux .
Une clause est une somme de littéraux.
Par exemple, $x\bar{y}z$ est un monôme dont l'unique modèle est $x = 1; y = 0; z = 1$.
Le monôme $x\bar{y}z\bar{x}$ comporte une variable et sa négation : il vaut 0.
Par exemple $x + \bar{y} + z$ est une clause dont l'unique contre modèle est $x = 0; y = 1; z = 0$.
La clause $x + \bar{y} + z + \bar{x}$ comporte une variable et sa négation : elle vaut 1.

Forme normale disjonctive

Une formule est une forme normale disjonctive (FND) ssi elle est une somme de monômes.
Par exemple $xy + \bar{x}\bar{y}z$ est une FND, qui a comme modèles $x = 1; y = 1$ et $x = 0; y = 0; z = 1$.

L'intérêt des *FND* est de mettre en évidence ces modèles.

Forme normale conjonctive

Une formule est une forme normale conjonctive (FNC) ssi elle est un produit de clauses.
Par exemple $(x + y) \cdot (\bar{x} + \bar{y} + z)$ est une FNC, qui a deux contre-modèles $x = 0; y = 0$ et $x = 1; y = 1; z = 0$.

L'intérêt des FNC est de mettre en évidence ces contre-modèles.

Par convention, on considère que 0 et 1 sont des sommes de monômes et des produits de clauses.

4.3.2 Étapes de Transformation en somme de monômes

Pour transformer une formule en une somme *équivalente* de monômes, on applique les transformations :

1. Élimination des équivalences **2.** Élimination des implications
3. Déplacement des négations **4.** Distribution des produits sur les sommes

Éliminer une équivalence

Éliminer une équivalence, c'est remplacer une occurrence de $A \Leftrightarrow B$ par
(a) $(\overline{A} + B) \cdot (\overline{B} + A)$ ou **(b)** $A \cdot B + \overline{A} \cdot \overline{B}$

Éliminer une implication

Éliminer une implication, c'est remplacer une occurrence de $A \Rightarrow B$ par $\overline{A} + B$.

Déplacer les négations

Déplacer les négations, c'est remplacer une occurrence de

(a) $\overline{\overline{A}}$ par A
(b) $\overline{A + B}$ par $\overline{A} \cdot \overline{B}$
(c) $\overline{A \cdot B}$ par $\overline{A} + \overline{B}$

En appliquant ces trois transformations dans l'ordre indiqué, il est clair que la formule initiale a été transformée en une formule équivalente avec les seules opérations $\wedge, \vee, \neg$ et les négations ayant comme seul opérande des variables : **la formule est dite sous forme normale.** Par exemple $(\bar{x} + y) \cdot \bar{z}$ est une formule sous forme normale conjonctive.

Somme de monômes

En distribuant (tous) les produits sur les sommes, on obtient une somme de monômes.

Par exemple : $\bar{x}\bar{z} + y\bar{z}$ est une somme de monômes(FND).

L'ordre des transformations n'est pas important : en effectuant les transformations dans un ordre quelconque, on obtient finalement une somme de monômes.

En pratique

Il est recommandé de remplacer directement $\neg(A \Rightarrow B)$ par $A \wedge \neg B$ et $\neg(A \Leftrightarrow B)$ par $A\bar{B} + \bar{A}B$. En pratique, il est capital de simplifier (le plus tôt possible) de la façon suivante

- Un produit qui, soit comporte une formule et sa négation, soit comporte un 0, est remplacé par 0
- Une somme qui, soit comporte une formule et sa négation, soit comporte un 1, est remplacé par 1
- Remplacer $\bar{1}$ par 0 et $\bar{0}$ par 1
- Enlever les 0 des sommes et les 1 des produits
- Appliquer les simplifications $x + xy = x, x(x+y) = x, x + \bar{x}y = x + y$
- Appliquer l'idempotence de la somme et du produit.

Utilisation des sommes de monômes

La transformation par équivalence d'une formule en une somme de monômes, peut être utilisée pour déterminer si une formule est valide ou non.

On transforme $\neg A$ en une somme de monômes équivalente à $\neg A$. Si $\neg A \equiv 0$ alors $A \equiv 1$ donc A est valide, sinon, toutes simplifications étant faites, $\neg A$ est égal à une somme de monômes non nuls, qui nous donnent des modèles de $\neg A$, donc des contre-modèles de A.

1. Soit $A = (p \Rightarrow (q \Rightarrow r)) \Rightarrow (p \cdot q \Rightarrow r)$. Déterminer si A est valide par transformation de $\neg A$ en somme de monômes.

$$\begin{array}{ll}
\neg A & \\
\equiv (p \Rightarrow (q \Rightarrow r)) \cdot \overline{(p \cdot q \Rightarrow r)} & \text{Par l'identité } \neg(B \Rightarrow C) = B \wedge \neg C \\
\equiv (\bar{p} + \bar{q} + r) \cdot \overline{(p \cdot q \Rightarrow r)} & \text{En éliminant deux implications} \\
\equiv (\bar{p} + \bar{q} + r) \cdot (pq\bar{r}) & \text{Par l'identité } \neg(B \Rightarrow C) = B \wedge \neg C \\
\equiv (\bar{p}pq\bar{r}) + (\bar{q}pq\bar{r}) + (rpq\bar{r}) & \text{Par distributivité du produit sur la somme} \\
\equiv 0 & \text{Car chaque monôme vaut 0}
\end{array}$$

Donc $\neg A = 0$ et $A = 1$, c'est-à-dire A est valide (tautologie).

2. Soit $A = (a \Rightarrow b) \cdot c + a \cdot d$. Déterminer si A est valide par transformation de $\neg A$ en somme de monômes.

$$\begin{array}{ll}
\neg A & \\
\equiv \overline{(a \Rightarrow b) \cdot c} \cdot \overline{a \cdot d} & \text{Par déplacement des négations} \\
\equiv (\overline{(a \Rightarrow b)} + \bar{c}) \cdot (\bar{a} + \bar{d}) & \text{Par déplacement des négations} \\
\equiv (a\bar{b} + \bar{c}) \cdot (\bar{a} + \bar{d}) & \text{Par l'identité } \neg(B \Rightarrow C) = B \wedge \neg C \\
\equiv a\bar{b}\bar{a} \; + \; a\bar{b}\bar{d} \; + \; \bar{c}\bar{a} \; + \; \bar{c}\bar{d} & \text{Par distributivité du produit sur la somme} \\
\equiv a\bar{b}\bar{d} + \bar{c}\bar{a} \; + \; \bar{c}\bar{d} & \text{Par simplification}
\end{array}$$

On obtient 3 modèles de $\neg A$ qui sont les mêmes que les contre-modèles de A. Donc A n'est pas valide.

4.4 Transformation en produit de sommes de littéraux

On procède exactement comme pour la transformation en somme de monômes, sauf que l'on applique la distributivité (inhabituelle) de la somme sur le produit, autrement dit on remplace toute sous-formule

$$A + (B \cdot C) \text{ par } (A + B) \cdot (A + C) \text{ et toute sous-formule } (B \cdot C) + A \text{ par } (B + A) \cdot (C + A).$$

Transformons la formule $A = (a \Rightarrow b) \cdot c + a \cdot d$ en produit de clauses.

$$\begin{array}{rcl}
A & \equiv & (\bar{a} + b)c + ad \\
& \equiv & (\bar{a} + b + a)(\bar{a} + b + d)(c + a)(c + d) \\
& \equiv & (\bar{a} + b + d)(c + a)(c + d)
\end{array}$$

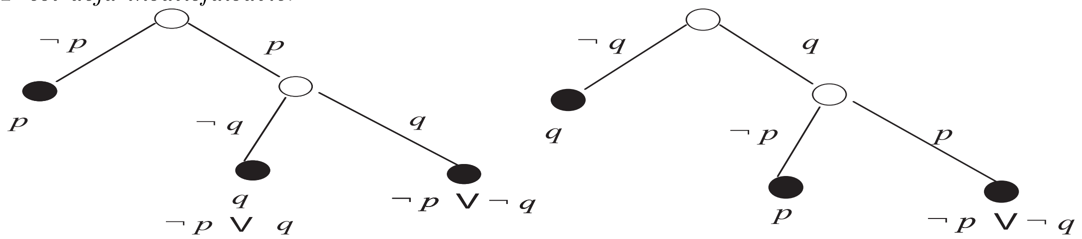
Exercice. Transformer cette formule en FNC : $F = \neg((p_1 \vee (p_2 \Rightarrow p_3)) \wedge \neg(p_4 \vee p_5) \vee p_6)$.

4.5 Arbre sémantique

Un arbre sémantique est associé à une énumération p_k des variables propositionnelles. Un arbre sémantique est un arbre binaire dont les arêtes sont étiquetées par des littéraux, et pour chaque noeud de niveau i, l'arête gauche partant de ce noeud est étiquetée par $\neg p_i$ et l'arête droite partant de ce noeud est étiquetée par p_i.

A chaque noeud n d'un arbre sémantique est associée une interprétation partielle I_n des variables propositionnelles : si n est un noeud de niveau i, c'est l'interprétation partielle définie en prenant la valeur 1 pour tous les littéraux qui étiquettent les arêtes menant de la racine au noeud n.

Exemple 3. *Considérons l'ensemble de clauses* $\Gamma = \{p, q, \neg p \vee q, \neg p \vee \neg q\}$. *On trouvera ci-dessous un arbre sémantique associé à l'énumération* $\{p, q\}$ *et un arbre sémantique associé à l'énumération* $\{q, p\}$. *On remarquera que l'arbre sémantique peut avoir moins de noeuds fermés qu'il y a de clauses dans* Γ, *lorsqu'un sous-ensemble strict de clauses de* Γ *est déjà insatisfaisable.*



Définition 2. *Soit A un arbre sémantique et* Γ *un ensemble fini de clauses.*

(i) *Un noeud n de l'arbre sémantique A réfute la clause C de* Γ *si* $I_n(C) = 0$.

(ii) *Un noeud n est un noeud d'échec pour* Γ *si :*
- *il existe une clause C de F qui est réfutée par n,*
- *pour tout noeud n situé sur le chemin menant de la racine de l'arbre à n, il n'existe aucune clause C de* Γ *qui soit réfutée par n.*

(iii) *Un arbre sémantique A est fermé pour* Γ *si, sur toute branche, il existe un noeud d'échec pour* Γ

Proposition 1. *Si F est en FNC, alors F est insatisfaisable ssi F admet un arbre sémantique fermé.*

L'ordre des variables influe sur la construction de l'arbre sémantique. Rechercher l'ordre permettant de construire l'arbre sémantique de taille minimum (le moins de sommets possible) est un problème difficile.

Un arbre sémantique est dit complet si chaque branche contient une fois chaque littéral. Il est dit partiel dans le cas contraire. De la même manière que la table de vérité de F contient 2^n lignes, un arbre sémantique complet comprend 2^n feuilles. Chaque ligne correspond à une branche. Donc il peut être vu comme une représentation graphique de la table de vérité.

4.6 Algorithme de Quine

L'algorithme de Quine est une amélioration des arbres sémantiques. A chaque noeud de l'arbre binaire, on réalise une évaluation partielle en prenant en compte tous les atomes dont la valeur est déterminée. Si cette évaluation permet de conclure directement, on ne poursuit pas la construction à partir de ce noeud. **Simplifications de Quine**

1.	$a \wedge \bot \equiv \bot \wedge a \equiv \bot$.	**4.**	$a \vee \bot \equiv \bot \vee a \equiv a$.	**7.**	$\bot \Rightarrow a \equiv \top \equiv a \Rightarrow \top$.
2.	$a \wedge \top \equiv \top \wedge a \equiv a$.	**5.**	$a \Leftrightarrow \bot \equiv \bot \Leftrightarrow a \equiv \neg a$.	**8.**	$a \Rightarrow \bot \equiv \neg a$.
3.	$a \vee \top \equiv \top \vee a \equiv \top$.	**6.**	$a \Leftrightarrow \top \equiv \top \Leftrightarrow a \equiv a$.	**9.**	$\top \Rightarrow a \equiv a$.

Exemple. Appliquer l'algorithme de Quine à la formule suivante :

$$((c \Rightarrow ((b \vee a) \wedge d)) \wedge (b \Leftrightarrow (a \wedge (c \vee d))) \wedge (c \Rightarrow a)) \vee ((b \wedge a) \Rightarrow d)$$