Chapitre 8

Logique propositionnelle

Sommaire.

Les propositions marquées de ★ sont au programme de colles.

1 Formules propositionnelles.

1.1 Syntaxe.

Définition 1: Logique propositionnelle.

Soit $\mathbb{V}$ un ensemble fini (ou dénombrable) de symboles appelés variables propositionnelles.
On définit inductivement l'ensemble des formules propositionnelles sur $\mathbb{V}$:

- $\bot$ et $\top$ sont des expressions logiques, **Faux** et **Vrai** respectivement.
- p est une variable propositionnelle de $\mathbb{V}$.
- À partir de φ et ψ deux formules, on peut construire :
  - $(\varphi \wedge \psi)$ (conjonction).
  - $(\varphi \vee \psi)$ (disjonction).
  - $(\neg\varphi)$ (négation).

Ici, φ et ψ désigneront toujours des formules.
Toute formule propositionnelle peut être représentée par un arbre : avec les variables propositionnelles en tant que feuilles, et les constructeurs en tant que noeuds internes.

1.2 Sémantique.

Définition 2: Valuation.

Une **valuation** sur $\mathbb{V}$ est une application $v : \mathbb{V} \to \{0, 1\}$.
On étend cette application aux formules propositionnelles :
Soient φ, ψ des formules propositionnelles. On définit inductivement $v(\varphi)$ tel que :

- $v(\bot) = 0$.
- $v(\top) = 1$.
- $v(\varphi) = v(\varphi)$ si $\varphi \in \mathbb{V}$.
- $v(\varphi \wedge \psi) = v(\varphi) \times v(\psi)$.
- $v(\varphi \vee \psi) = v(\varphi) + v(\psi) - v(\varphi) \times v(\psi)$.
- $v(\neg\varphi) = 1 - v(\varphi)$.

Ici, v désignera toujours une valuation.

Définition 3: Équivalence logique. ★

Deux formules φ et ψ sont **sémantiquement équivalentes** si pour toute valuation v sur $\mathbb{V}$, $v(\varphi) = v(\psi)$.
On note alors $\varphi \equiv \psi$. Ainsi, $\equiv$ est une relation d'équivalence sur les formules.

Remarque: Dans la pratique, on compare les tables de vérité de φ et ψ.

Définition 4: Autres constructeurs.

Il existe des liens logiques qui s'expriment à partir de ceux de base :

- L'implication $\varphi \to \psi \;\equiv\; \neg\varphi \vee \psi$.
- L'équivalence $\varphi \leftrightarrow \psi \;\equiv\; \varphi \to \psi \wedge \psi \to \varphi$.
- Vrai : $\top \;\equiv\; \varphi \vee \neg\varphi$.
- Faux : $\bot \;\equiv\; \varphi \wedge \neg\varphi$.

Proposition 5: Lois de De Morgan. ★

Soient φ et ψ deux formules logiques. Alors :
- $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$.
- $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$.

Preuve :

On le montre facilement en comparant les tables de vérités.

1.3 Satisfiabilité.

Définition 6: Modèles.

Soit φ une formule sur $\mathbb{V}$. Une valuation $v : \mathbb{V} \to \mathscr{B}$ est un **modèle** de φ si $v(\varphi) = 1$.
On note alors $v \models \varphi$.
On dit alors qu'une formule est **satisfiable** si elle admet un modèle.

Une formule φ pour laquelle toute valuation est un modèle est une tautologie, on note $\models \varphi$.
Si aucune valuation n'en est un modèle, φ est une antilogie, on note $\not\models \varphi$.

Définition 7: Conséquence logique.

Soient φ, ψ deux formules.
On dit que φ est en **conséquence logique** de ψ, et on note $\psi \models \varphi$ si tout modèle de ψ est modèle de φ.
On étend cette notation à un ensemble Γ de formules, dans ce cas, on dit que φ est une **conséquence logique** de Γ si φ est en **conséquence logique** de toute formule de Γ.

Définition 8: Équisatisfiabilité

Deux formules φ et ψ sont **équisatisfiables** si φ est satisfiable si et seulement si ψ l'est.

2 Formes normales.

2.1 Formes normales négatives.

Définition 9: Littéral.

On appelle **littéral** une variable propositionnelle ou sa négation.

Définition 10: Construction. ★

Une formule est dite en **forme normale négative (FNN)** si ses négations ne s'appliquent qu'aux variables.
Pour une formule φ, on construit sa forme normale négative $\mathrm{nnF}(\varphi)$ inductivement de la manière suivante :
- $\mathrm{nnF}(\varphi) = \varphi$ si c'est un littéral.
- $\mathrm{nnF}(\neg\neg\varphi) = \mathrm{nnF}(\varphi)$
- $\mathrm{nnF}(\varphi \wedge \psi) = \mathrm{nnF}(\varphi) \wedge \mathrm{nnF}(\psi)$
- $\mathrm{nnF}(\varphi \vee \psi) = \mathrm{nnF}(\varphi) \vee \mathrm{nnF}(\psi)$
- $\mathrm{nnF}(\neg(\varphi \vee \psi)) = \mathrm{nnF}(\neg\varphi) \wedge \mathrm{nnF}(\neg\psi)$
- $\mathrm{nnF}(\neg(\varphi \wedge \psi)) = \mathrm{nnF}(\neg\varphi) \vee \mathrm{nnF}(\neg\psi)$

Proposition 11: Existence. ★

Pour toute formule φ, $\mathrm{nnF}(\varphi)$ est sous forme normale négative et $\mathrm{nnF}(\varphi) \equiv \varphi$.

Preuve :

Par induction sur les formules propositionnelles.
Cas de base. Soit φ un littéral. $\mathrm{nnF}(\varphi) = \varphi$ sous FNN et $\mathrm{nnF}(\varphi) \equiv \varphi$.
Hérédité: Soient φ, ψ telles que la propriété soit vraie sur elles-mêmes et leurs négations.
Soit v une valuation de φ et ψ.
On a $\mathrm{nnF}(\varphi \wedge \psi) = \mathrm{nnF}(\varphi) \wedge \mathrm{nnF}(\psi)$ donc c'est bien sous forme normale négative par hypothèse.
De plus, $v \vDash \mathrm{nnF}(\varphi \wedge \psi) = \iff v \vDash \mathrm{nnF}(\varphi) \wedge \mathrm{nnF}(\psi) \iff v \vDash \varphi$ et $v \vDash \psi \iff v \vDash \varphi \wedge \psi$.

On a $\mathrm{nnF}(\neg(\varphi \wedge \psi)) = \mathrm{nnF}(\neg\varphi) \vee \mathrm{nnF}(\neg\psi)$ donc c'est bien sous forme normale négative par hypothèse.
De plus, $v \vDash \mathrm{nnF}(\neg(\varphi \wedge \psi)) \Leftrightarrow v \vDash \mathrm{nnF}(\neg\varphi) \vee \mathrm{nnF}(\neg\psi) \Leftrightarrow v \vDash \neg\varphi$ ou $v \vDash \neg\psi \Leftrightarrow v \vDash \neg\varphi \vee \neg\psi \Leftrightarrow v \vDash \neg(\varphi \wedge \psi)$.
Même raisonnement pour la disjonction.
Par théorème d'induction, c'est vrai pour toute formule φ.

2.2 Formes normales conjonctives.

Définition 12: Problème SAT.

Le problème SAT prend une formule en entrée et répond à la question : "Cette formule est-elle satisfiable ?".

Définition 13: Clause.

Une **clause** est une disjonction de littéraux.

Définition 14: Forme normale conjonctive. ★

Une formule est en **forme normale conjonctive (FNC)** si elle est une conjonction de clauses.
On définit inductivement la mise sous FNC de φ en $\mathrm{cnF}(\varphi)$ par :

- $\mathrm{cnF}(\varphi) = \varphi$ si φ littéral.
- $\mathrm{cnF}(\varphi \vee \psi) = \varphi \vee \psi$ si φ, ψ littéraux.
- $\mathrm{cnF}(\varphi \wedge \psi) = \mathrm{cnF}(\varphi) \wedge \mathrm{cnF}(\psi)$.
- $\mathrm{cnF}(\varphi \vee (\psi \wedge \psi')) = \mathrm{cnF}(\varphi \vee \psi) \wedge \mathrm{cnF}(\varphi \wedge \psi')$.
- $\mathrm{cnF}(\varphi \vee (\psi \vee \psi')) = \mathrm{cnF}(\varphi \vee \mathrm{cnF}(\psi \vee \psi'))$.

Proposition 15

Si φ est une formule sous FNN, $\mathrm{cnF}(\varphi)$ est sous FNC et $\mathrm{cnF}(\varphi) \equiv \varphi$.

Preuve :

Même principe de preuve que pour la FNN.

Proposition 16

Si φ est sous FNN, on peut construire une FNC équisatisfiable à φ en temps linéaire.

Preuve :

La preuve existe dans le cours, elle est trop longue et horrible.

2.3 Algorithme de Quine

Définition 17: Substitution.

Soit φ une formule sur un esemble $\{p_1, , ..., p_n\}$ et soient $\{\varphi_1, ..., \varphi_n\}$ des formules.
La substitution des φ_i aux p_i est la formule obtenue en remplaçant simultanément chaque p_i par φ_i.
On la note $\varphi[\varphi_1/p_i, ..., \varphi_n/p_n]$.

La substitution se définit inductivement :

- $\varphi[\varphi_i/p_i] = \varphi_i$ si $\varphi = p_i$.
- $\varphi[\varphi_1/p_1, ..., \varphi_n/p_n] = \neg\varphi'[...]$ si $\varphi = \neg\varphi'$.
- $\varphi[...] = \varphi_1[...] \wedge \varphi_2[...]$ si $\varphi = \varphi_1 \wedge \varphi_2$.
- $\varphi[...] = \varphi_1[...] \vee \varphi_2[...]$ si $\varphi = \varphi_1 \vee \varphi_2$.

Proposition 18

Une substitution dans une tautologie donne une tautologie.

Preuve :

Soit φ sur $\{p_1, ..., p_n\}$ et $\{\varphi_1, ..., \varphi_n\}$ des formules sur $\mathbb{V}$.
Soit v une valuation sur $\mathbb{V}$ et ω sur $\{p_1, ..., p_n\}$: $\omega(p_i) = v(\varphi_i)$.
Montrons que $\omega(\varphi) = v(\varphi[...])$.
Cas de base. Trivial si $\varphi = \top$ ou $\varphi = \bot$.
Si $\varphi = p_i$, alors $\varphi[...] = \varphi_i$ et $\omega(\varphi) = \omega(p_i) = v(\varphi_i)$.
Hérédité.
Si $\varphi = \neg\varphi'$, $\omega(\varphi) = \omega(\neg\varphi') = \neg\omega(\varphi') = \neg v(\varphi'[...]) = v(\neg\varphi'[...]) = v(\varphi[...])$.
Si $\varphi = \varphi_1 \vee \varphi_2$, $\omega(\varphi) = \omega(\varphi_1 \vee \varphi_2) = \omega(\varphi_1) \vee \omega(\varphi_2) = v(\varphi_1[...]) \vee v(\varphi_2[...]) = v(\varphi_1[...] \vee \varphi_2[...]) = v(\varphi)$.
De même pour la conjonction, avec φ_1, φ_2 vérifiant l'hypothèse.
Par principe d'induction structurelle, la propriété est vérifiée.

Supposons φ une tautologie. Soit v une valuation de la formule substituée., il existe ω telle que $\omega(\varphi) = v(\varphi[...])$.
Comme φ est tautologie, $w(\varphi) = 1$ donc $v(\varphi[...]) = 1$ donc $v \models \varphi[...]$, c'est une tautologie.

Définition 19: Algorithme de Quine.

Entrée: φ sous FNC.
Sortie: 1 si φ est satisfiable, 0 sinon.
1. Simplifier les clauses.
2. Si φ est une conjonction sur $\varnothing$, renvoyer 1.
3. Si φ contient $\bot$, renvoyer 0.
4. Choisir la prochaine variable p dans l'une des clauses :
    - Si $\mathrm{Quine}(\varphi[\bot/p])$, renvoyer 1, sinon renvoyer $\mathrm{Quine}(\varphi[\top/p])$.

Étape 1:

- Si la clause est $\top$, la supprimer.
- Tiers-exclu : les clauses contenant des littéraux opposés sont supprimées.
- Fusion : supprimer les doublons de littéraux.
- Si une clause en contient une autre, on la supprime.
- Si une clause contient $\bot$, le supprimer.

Terminaison: Toutes les opérations s'effectuent en temps fini.
Il y a un nombre fini d'appels récursifs : variant d'appel donnée par le nombre de variables apparaissant dans la formule.

Correction: assurée par le tiers-exclu.

2.4 Forme normale disjonctive.

Définition 20: Conjonction élémentaire.

Une **conjonction élémentaire** est une formule sans disjonctions.

Définition 21: Forme normale disjonctive. ★

Une formule est une **forme normale disjonctive (FND)** si c'est une disjonction de conjonctions élémentaires.
Pour passer de φ sous FNN à $\mathrm{dnF}(\varphi)$ sous FND, on procède par induction :

- $\mathrm{dnF}(\varphi) = \varphi$ si φ est littéral.
- $\mathrm{dnF}(\varphi) = \varphi$ si $\varphi = l \wedge l'$ avec l, l' littéraux.
- $\mathrm{dnF}(\varphi \vee \psi) = \mathrm{dnF}(\varphi) \vee \mathrm{dnF}(\psi)$.
- $\mathrm{dnF}(\varphi \wedge (\psi \vee \psi')) = \mathrm{dnF}(\varphi \wedge \psi) \vee \mathrm{dnF}(\varphi \wedge \psi')$.
- $\mathrm{dnF}(\varphi \wedge (\psi \wedge \psi')) = \mathrm{dnF}(\varphi \wedge \mathrm{dnF}(\psi \wedge \psi'))$.

Proposition 22

Si φ est sous FNN, $\mathrm{dnF}(\varphi)$ est sous FND et $\mathrm{dnF}(\varphi) \equiv \varphi$.

Preuve :

Pour tout modèle v de φ, on construit :

$$\varphi_v = \bigwedge_{p \in \mathbb{V}} l_p \quad \text{où} \quad l_p = \begin{cases} p & \text{si } v(p) = 1 \\ \neg p & \text{sinon} \end{cases}.$$

On pose alors ψ la disjonction des φ_v pour tout modèle v de φ.
On obtient alors ψ sous FND et $\psi \equiv \varphi$.

Définition 23

Une FND est complète si chaque variable est représentée une unique fois dans chaque conjonction élémentaire.

3 Logique des prédicats.

Définition 24

Pour chaque instance de logique du premier ordre, on introduit un ensemble de symboles:
— Une infinité de symboles de variables.
— Des symboles de constantes, éléments particuliers du domaine d'interprétation.
— Des symboles de fonctions, transforment les tuples d'éléments.
— Des symboles de prédicats, expriment des propriétés sur les éléments.

Exemple 25

Un groupe peut être décrit par:
— Constantes: e le neutre.
— Fonctions: le produit, l'inverse.
— Prédicat: l'égalité.

Familles des ensembles:
— Constantes: $\varnothing$
— Fonctions: $\cup$, $\cap$, $\setminus$
— Prédicats: $\in$, $\subset$, $=$.

Définition 26: Termes.

À partir des fonctions, des constantes et des variables, on définit inductivement les **termes**:
— Assertions: constantes et variables.
— Règles d'inférence: si f est une fonction d'arité n, et $t_1, ..., t_n$ des termes, $f(t_1, ..., t_n)$ est un terme.

Définition 27: Atomes.

Un **atome** (ou formule atomique) est l'application d'un prédicat n-aire à une suite de n termes.

Définition 28: Formules.

Les **formules** de la logique des prédicats sont construites inductivement:
— Assertions: atomes.
— Règles d'inférence: $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$.

Exemple 29: Groupes abéliens.

Dans un groupe abélien, on voudrait que les formules suivantes soient vraies:

$$\forall x.\ x + 0 = x, \quad \forall x.\ \exists y.\ x + y = 0, \quad \forall x.\ \forall y.\ x + y = y + x$$

Définition 30: Occurences.

Une occurence d'une variable x dans une formule F est une position de cette variable dans F.
Une occurence de x est liée dans F si dans la branche qui aboutit à cette occurence on rencontre une quantification de x. Sinon elle est libre.

Exemple 31

Dans $\exists x.\ P(x, y)$, x est liée et y est libre.

Définition 32

Deux formules F et G sont α-équivalentes si on peut passer de l'une à l'autre en renommant des variables liées.