Cours 5

Calcul propositionnel : sémantique (suite et fin) et déduction par coupure

Substitutions

Notation : $F[p_1, \ldots, p_n]$ indique qu'une formule F ne comporte pas de variables en dehors de $p_1, \ldots, p_n$.

Proposition

Soit $F[p_1, .., p_n]$ une formule et φ une valuation, alors la valeur $\varphi(F)$ ne dépend que de la valeur de φ sur $p_1, .., p_n$.

Proposition

Soient F et G deux formules et φ une valuation, alors la valeur de $F[G/p]$ pour φ est égale à la valeur de F pour une valuation φ' telle que $\varphi'(p) = \varphi(G)$ et $\varphi'(q) = \varphi(q)$ pour tout $q \in \mathbb{V} \setminus \{p\}$.

Substitutions

Corollaire

Soit F, F', G, G' des formules et $p \in \mathbb{V}$:

- *si F est une tautologie alors $F[G/p]$ est une tautologie*
- *si $F \equiv F'$ alors $F[G/p] \equiv F'[G/p]$*
- *si $G \equiv G'$ alors $F[G/p] \equiv F[G'/p]$*

Exemples :

- pour toutes formules F, G, H, $((F \Rightarrow (G \Rightarrow H)) \Rightarrow ((F \Rightarrow G) \Rightarrow (F \Rightarrow H)))$ est une tautologie.
- $\neg(F \wedge G) \equiv (\neg F \vee \neg G)$

Formes normales

Théorème

On a une bijection entre les applications de $\{0,1\}^n \to \{0,1\}$ *et l'ensemble des formules sur* $p_1, .., p_n$ *quotienté par* $\equiv$.

Représentants particuliers : formes normales

- Forme normale disjonctive (FND) : $F = G_1 \vee .. \vee G_k$ et $G_i = (B_{i,1} \wedge .. \wedge B_{i,n_i})$ avec $B_{i,j} = p$ ou $\neg p$ (*littéral*).
- Forme normale conjonctive (FNC) : $F = G_1 \wedge .. \wedge G_k$ et $G_i = (B_{i,1} \vee .. \vee B_{i,n_i})$ avec $B_{i,j}$ un litéral.

Théorème

Tout formule est équivalente à une formule sous FND et une formule sous FNC.

Remarque : on s'autorise à supprimer des parenthèses quand cela ne change pas la classe d'équivalence.

Forme normale

Comment calculer une forme normale ?

- à partir de la table de vérité
- par manipulation symbolique

Systèmes complets de connecteurs

Définition

Un système complet de connecteurs est un ensemble de connecteurs qui permet d'engendrer toutes les applications de $\{0,1\}^n \longrightarrow \{0,1\}$

Exemple : $\{\neg, \wedge, \vee\}$, $\{\neg, \vee\}$, $\{\text{nand}\}$, $\{\text{nor}\}$.

Théorème de compacité

Définition

Un ensemble Σ de formules est *finiment satisfiable* si tout sous-ensemble fini de Σ est satisfiable.

Théorème (de compacité)

Σ satisfiable si et seulement si Σ finiment satisfiable.

Systèmes de déduction : la déduction par coupure

Système de déduction (rappel)

Un système formel est constitué

- d'une syntaxe
  - un alphabet A
  - une procédé de formation des formules : $\mathcal{F} \subseteq A^*$
- d'une sémantique :

$$\Sigma \models F$$

- d'un système de déduction
  - un ensemble d'axiomes
  - un ensemble fini de règles de déduction

$$\Sigma \vdash F$$

Vocabulaire : le système de déduction

- est dit *correct* si $\Sigma \vdash F$ implique $\Sigma \models F$,
- est dit *complet* si $\Sigma \models F$ implique $\Sigma \vdash F$.

La déduction par coupure

On se limite à un sous ensemble des formules : *les clauses*.

Définition

Une clause est une disjonction de littéraux.

Il s'agit d'un sous ensemble suffisamment représentatif :

Théorème

Toute formule est sémantiquement équivalente à une conjonction de clauses.

Exemple

- $p \vee \neg q \vee r$

Vocabulaire et notations

Nous noterons les clauses sous la forme d'un ensemble de littéraux en distinguant les variables *négatives* (apparaissant avec une négation) des variables *positives* (apparaissant sans négation).

$$C = (\Gamma, \Delta)$$

Γ : variables propositionnelles négatives
Δ : variables propositionnelles positives

Exemples

- $p \vee \neg q \vee r$ sera noté $(\{q\}, \{p, r\})$.
- $p \vee \neg q \vee \neg q \vee p$ sera noté $(\{q\}, \{p\})$.

Vocabulaire et notations

$$C = (\Gamma, \Delta)$$

Cas particuliers :

- $\Delta = \emptyset$: clause négative
- $\Gamma = \emptyset$: clause positive
- $\Delta = \Gamma = \emptyset$: clause vide, notée $\square$

Remarque : une clause $(\{a_1, \ldots, a_n\}, \{b_1, \ldots, b_m\})$ est équivalente à

$$(a_1 \wedge \cdots \wedge a_n) \Rightarrow (b_1 \vee \cdots \vee b_m)$$

Règle de coupure

Définition

Soit $C_1 = (\Gamma_1, \Delta_1)$ et $C_2 = (\Gamma_2, \Delta_2)$, et $p \in \Delta_1 \cap \Gamma_2$.
$C = (\Gamma, \Delta)$ se *déduit par coupure sur p* si

$$\Gamma = \Gamma_1 \cup (\Gamma_2 \setminus \{p\}) \text{ et } \Delta = \Delta_2 \cup (\Delta_1 \setminus \{p\})$$

On note :

$$\frac{C_1 \qquad C_2}{C}$$

Preuve par coupure

Définition (Preuve par coupure)

Soit S un ensemble de clauses. L'ensemble des clauses C *prouvables par coupure à partir de* S (noté $S \vdash C$) est définie inductivement par

- pour toute clause C

$$\text{si } C \in S \text{ alors } S \vdash C$$

- pour toutes clauses C_1, C_2 et C

$$\text{si } S \vdash C_1,\ S \vdash C_2 \text{ et } \frac{C_1 \qquad C_2}{C} \text{ alors } S \vdash C$$

Réfutation par coupure de S : $S \vdash \square$

Correction

Lemme

Si $\dfrac{C_1 \qquad C_2}{C}$ *alors* $\{C_1, C_2\} \models C$.

Théorème

Si $S \vdash C$, *alors* $S \models C$.

Corollaire

Si $S \vdash \square$ *alors* S *n'est pas satisfiable.*

Complétude

Lemme

Soit S un ensemble de clauses non satisfiables qui ne contient pas $\square$. Alors il existe p, C_1 et C_2 tels que $p \in \Gamma_1 \cap \Delta_2$.

Définition (Résolvant)

Avec les notations précédentes et S_p le sous-ensemble des clauses de S contenant p, on appelle *résolvant* de S_p (noté $Res(S_p)$) l'ensemble des clauses obtenues à partir de deux clauses de S_p par coupure sur p.

Remarque : Si S est non satisfiable et ne contient pas $\square$, alors il existe p tel que $Res(S_p) \neq \emptyset$.

Complétude

Lemme

S satisfiable ssi $(S \setminus S_p) \cup Res(S_p)$ satisfiable.

Lemme

Si S est fini et $S \models C$, alors $S \vdash C$.

Théorème

Si $S \models C$, alors $S \vdash C$.

Plan

1. Sémantique : suite et fin
2. Systèmes de déduction : la déduction par coupure