TD #21 — Logique

Exercice 1. *Syntaxe & sémantique d'expressions arithmétiques*

1. Définissez une syntaxe des *expressions arithmétiques* sur un ensemble de variable $\mathcal{V}$ avec deux constructeurs $+$ et $\times$. (Vous pouvez définir cette syntaxe directement sous la forme d'un `type ae` OCaml.)
2. Définissez la sémantique usuelle pour les expressions arithmétiques entières, sous la forme d'une fonction OCaml de type `ae -> ··· -> int`, où l'on assimilera les valeurs de type `int` aux entiers, et où l'on ignorera tout problème lié à la finitude de ces valeurs.
3. De même pour les expressions arithmétiques sur $\mathbb{F}_{257}$, le corps fini à 257 éléments, où l'on assimilera les valeurs de type `int` entre `0` et `256` aux éléments de $\mathbb{F}_{257}$, et ignorera tout problème lié au dépassement de capacité lors de calcul avec des valeurs de type `int`.

Exercice 2. *Quelques équivalences sémantiques classiques*

En utilisant des tables de vérité, montrez les équivalences sémantiques suivantes pour la sémantique par tables de vérité de la logique propositionnelle :

1. $\neg\varphi \equiv \varphi \to \bot$
2. $\neg(\varphi_1 \wedge \varphi_2) \equiv \neg\varphi_1 \vee \neg\varphi_2$
3. $\neg(\varphi_1 \vee \varphi_2) \equiv \neg\varphi_1 \wedge \neg\varphi_2$
4. $\neg\neg\varphi \equiv \varphi$
5. $\varphi_1 \to \varphi_2 \equiv \neg\varphi_2 \to \neg\varphi_1$
6. $\varphi_1 \vee (\varphi_2 \wedge \varphi_3) \equiv (\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \varphi_3)$
7. $\varphi_1 \wedge (\varphi_2 \vee \varphi_3) \equiv (\varphi_1 \wedge \varphi_2) \vee (\varphi_1 \wedge \varphi_3)$

où φ, φ_1, φ_2 désignent des formules quelconques sur un ensemble de variable arbitraire.

Exercice 3. *Quelques tautologies classiques*

Montrez les tautologies suivantes pour la sémantique par tables de vérité de la logique propositionnelle :

1. $\vDash \neg(\varphi \wedge \neg\varphi)$
2. $\vDash \varphi \vee \neg\varphi$
3. $\vDash \bot \to \varphi$
4. $\vDash (\neg\varphi \to \varphi) \to \varphi$

où φ etc.

Exercice 4. *Quelques propriétés des modèles pour la sémantique classique*
En utilisant la définition alternative de la sémantique par tables de vérité vue en cours, montrez que, :

1. $\mathbb{M}(\neg\varphi) = \mathbb{V} \setminus \mathbb{M}(\varphi)$
2. $\mathbb{M}(\varphi_1 \wedge \varphi_2) = \mathbb{M}(\varphi_1) \cap \mathbb{M}(\varphi_2)$
3. $\mathbb{M}(\varphi_1 \vee \varphi_2) = \mathbb{M}(\varphi_1) \cup \mathbb{M}(\varphi_2)$

où φ etc.

Exercice 5. *Quelques conséquences sémantiques*
Montrez les conséquences sémantiques suivantes (ou non) pour la sémantique par tables de vérité de la logique propositionnelle :

1. $\varphi_1 \to \varphi_2 \not\models \varphi_2$
2. $\{\varphi_1, \varphi_1 \to \varphi_2\} \models \varphi_2$
3. $\{\varphi_1, \neg\varphi_1\} \models \varphi_2$

où φ_1 etc.

Exercice 6. *Universalité de $\bar{\wedge}$ pour la sémantique T.V.*

1. Montrez que le connecteur binaire infixe $\bar{\wedge}$ (« non-et », ou « NAND ») dont la table de vérité pour la sémantique par tables de vérité est :

   — $a \bar{\wedge} b = 0$ si $a = b = 1$, et 1 sinon ;

   est *universel* pour la sémantique par tables de vérité de la logique propositionnelle. C'est à dire, montrez que toute formule propositionnelle s'écrivant avec les connecteurs $\neg$, $\wedge$, $\vee$, $\to$ est sémantiquement équivalente (pour la sémantique T.V. usuelle de ces derniers) à une formule utilisant uniquement le connecteur $\bar{\wedge}$.

Remarque. On peut montrer la même propriété pour le connecteur binaire infixe $\bar{\vee}$ (« non-ou », ou « NOR ») dont la table de vérité est $a \bar{\vee} b = 1$ si $a = b = 0$, et 0 sinon. L'universalité de ces connecteurs peut être intéressante dans le cadre de l'*électronique des circuits* : il suffit de disposer de *portes logiques* NAND ou NOR pour pouvoir implémenter l'évaluation de n'importe quelle formule propositionnelle pour la sémantique T.V. (dans ce cadre, on parlerait plutôt de *circuits* ou *fonctions booléennes*).

Exercice 7. *Additionneur 2-bit*

1. Donnez trois formules φ_0, φ_1, φ_2 définies sur les variables x_0, x_1, y_0, y_1 telles que pour x & y deux entiers naturels $\in [\![0, 3]\!]$ et σ une affectation pour laquelle x_i (resp. y_i) vaut vrai ssi. le i-ème bit de x (resp. y) vaut 1 dans son écriture en base deux, l'on a toujours $[\![\varphi_i]\!]_\sigma$ valant vrai ssi. le i-ème bit de l'écriture en base deux de $x + y$ vaut 1.