LOG : Logique des propositions

Définition : Formules du calcul des propositions (Inductive)
Soient P un ensemble de symboles (Variables propositionnelles), $C = \{\mathcal{V}, \mathcal{F}, \neg, \wedge, \vee, \Rightarrow, \Leftrightarrow\}$ l'ensemble des connecteurs logiques et $D = \{(,)\}$. L'ensemble des formules de la logique des propositions construite sur P, noté $Prop(P)$ est défini inductivement par :

- La base $B = P \cup \{\mathcal{V}, \mathcal{F}\}$
- Les opérations : $\forall p, q \in Prop(P)$:
$$(p \vee q) \quad (p \Rightarrow q) \quad (p \wedge q) \quad (p \Leftrightarrow q) \quad \neg p$$
  sont dans $Prop(P)$

Remarque :
On peut parfois juste utiliser $\neg$ et $\vee$:

- $x \Rightarrow y$ est une abréviation de $(\neg x) \vee x$
- $x \wedge y$ pour $\neg(\neg x \vee \neg y)$
- $x \Leftrightarrow y$ pour $(x \Rightarrow y) \wedge (y \Rightarrow x)$

Définition : Formules du calcul des propositions (Par les grammaires)
Soit P un ensemble, une formule de calcul des propositions de P est un mot engendré par la grammaire :

$$G = (\{X\}, \{\mathcal{V}, \mathcal{F}, a, \neg, \vee, \wedge, \Rightarrow, \Leftrightarrow, (,)\}, \rightarrow, X)$$

$$X \rightarrow \mathcal{V}|\mathcal{F}|a|\neg X|X \vee X|X \wedge X|X \Rightarrow X|X \Leftrightarrow X|(X)$$

où a est un élément quelconque de P. On définit $Prop(P) = L(G)$

Remarque :
Cette grammaire est ambigüe et ne tient pas compte de l'ordre priorité des connecteurs logiques.
Priorité décroissante des CL : $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

Définition :
On définit une application [] de $Prop(P)$ vers $\mathbb{F}$ de la façon suivante :

- $[\mathcal{V}] = 1$; $[\mathcal{F}] = 0$
- $[x] = x \quad (x \in P)$
- $[\neg\alpha] = \overline{[\alpha]} \quad (\alpha \in Prop(P))$
- $\alpha \vee \beta] = [\alpha + [\beta] \quad (\alpha \text{ et } \beta \in Prop(P))$
- $[\alpha \wedge \beta] = [\alpha].[\beta] \quad (\alpha \text{ et } \beta \in Prop(P))$
- $[\alpha \Rightarrow \beta] = [\alpha] \Rightarrow [\beta] \quad (\alpha \text{ et } \beta \in Prop(P))$
- $[\alpha \Leftrightarrow \beta] = [\alpha] \Leftrightarrow [\beta] \quad (\alpha \text{ et } \beta \in Prop(P))$

Définition : Valuation des variables propositionnelles
On appelle valuation des variables propositionnelles une application $\delta : P \rightarrow \{0, 1\}$

Définition : Sémantique
Soit une formule α de $Prop(P)$ et δ une valuation de P. La sémantique de α sur la valuation δ, notée $[\alpha](\delta)$ (ou $\delta(\alpha)$) est la valeur obtenue en remplaçant dans $[\alpha]$ chaque variable propositionnelle x par $\delta(x)$

Définitions :
Soient α une formule de $Prop(P)$, δ une valuation et $\mathcal{A}$ un ensemble de formules de $Prop(P)$.

- δ est un **modèle** de α si et seulement si $[\alpha](\delta) = 1$
- α est une **tautologie** si et seulement si $\forall \delta, [\alpha](\delta) = 1$
- α et β sont **sémantiquement équivalentes** si et seulement si $\alpha \Leftrightarrow \beta$ est une tautologie.
- α est **contradictoire** si et seulement si α ne possède pas de modèle.
- δ est un **modèle de** $\mathcal{A}$ si et seulement si δ est un modèle de chacune des formules de $\mathcal{A}$, i.e. si $\forall \alpha \in \mathcal{A}, [\alpha](\delta) = 1$
- $\mathcal{A}$ est contradictoire si et seulement si $\mathcal{A}$ ne possède pas de modèle, i.e. si $\forall \delta, \exists \alpha \in \mathcal{A}, [\alpha](\delta) = 0$

Définition : Déduction sémantique
Soit $\mathcal{A}$ un ensemble de formules de $Prop(P)$ et α une formule de $Prop(P)$, on dit que α se déduit sémantiquement de $\mathcal{A}$ si et seulement si tout modèle de $\mathcal{A}$ est un modèle de α. On note : $\mathcal{A} \models \alpha$
Si $\mathcal{A} = \emptyset$, on note $\models \alpha$ (Ce qui veut dire que α est une tautologie)

Propriétés :
Soient $\mathcal{A}, \mathcal{B} \subset Prop(P)$ et $\alpha, \beta \in Prop(P)$.

- Si $\mathcal{A} \subset \mathcal{B}$ alors pour toute proposition α de $Prop(P)$, si $\mathcal{A} \models \alpha$ alors $\mathcal{B} \models \beta$
- $\mathcal{A} \models \alpha$ si et seulement si l'ensemble $\mathcal{A} \cup \{\neg\alpha\}$ est contradictoire
- *Lemme du détachement :* $\mathcal{A} \cup \{\alpha\} \models \mathcal{B}$ si et seulement si $\mathcal{A} \models \alpha \Rightarrow \beta$

Théorème de compacité (ou de finitude)
Soient $\mathcal{A} \subset Prop(P)$ et $\alpha \in Prop(P)$

- Forme 1 : $\mathcal{A}$ admet un modèle si et seulement si tout sous-ensemble fini de $\mathcal{A}$ admet un modèle
- Forme 2 : α se déduit sémantiquement de $\mathcal{A}$ si et seulement si α se déduit sémantiquement d'un sous ensemble fini de $\mathcal{A}$
- Forme 3 : $\mathcal{A}$ est contradictoire si et seulement si l'un de ses sous ensembles finis est contradictoire

Définition : Système formel
Un système formel est un triplet $(E, \mathcal{A}, \mathcal{R})$ où

- E est un ensemble non vide de formules (bien formées)
- $\mathcal{A} \subset E$, l'ensemble des axiomes
- $\mathcal{R}$ est un ensemble de règles de déduction de la forme $\dfrac{e_1, \ldots, e_n}{e_{n+1}}(r)$ où les e_i sont des formules de E et r est le nom de la règle. On lit "e_{n+1} se déduit de $e_1, \ldots, e_n$ par la règle r"

Définition : Démonstration dans un système formel
Soit $S = (E, \mathcal{A}, \mathcal{R})$ un système formel et $\mathcal{H}$ un sous ensemble de E. Une démonstration de S avec hypothèses dans $\mathcal{H}$ est une suite finie $e_1, \ldots, e_n$ de formules de E telles que :

- Soit $e_i \in \mathcal{A}$ (e_i est un axiome)
- Soit $e_i \in \mathcal{H}$ (e_i est une hypothèse)
- Soit e_i est telle qu'il existe une règle de déduction $r \in \mathcal{R}$ et des indices j_1 à j_k tous strictement inférieurs à i vérifiant : $\dfrac{e_{j_1}, \ldots, e_{j_k}}{e_i}(r)$

Notation :
On note $\mathcal{H} \vdash_{\mathcal{S}} e_n$ et on dit que e_n se démontre dans $\mathcal{S}$ en utilisant les hypothèses de $\mathcal{H}$.
Si $\mathcal{H} = \emptyset$, on note $\vdash_S e$ et on dit que e est un **théorème** de $\mathcal{S}$

Proposition :
Soit $\mathcal{S} = (E, \mathcal{A}, \mathcal{R})$ un système formel, on a :

- Si $\mathcal{H} \subset \mathcal{H}'$, alors $\mathcal{H} \vdash_{\mathcal{S}} e$ implique $\mathcal{H}' \vdash_{\mathcal{S}} e$
- Si $\mathcal{H}' \vdash_{\mathcal{S}} e$ et $\mathcal{H}'' \cup \{e\} \vdash_{\mathcal{S}} e'$, alors $\mathcal{H}' \cup \mathcal{H}'' \vdash_{\mathcal{S}} e'$

Définition : Validité, complétude Soit $\mathcal{S} = (Prop(P), \mathcal{A}, \mathcal{R})$ un système formel sur le calcul des propositions.

- $\mathcal{S}$ est un système formel **valide** si et seulement si pour tout $\mathcal{H}$ tel que $\mathcal{H} \subset Prop(P)$ et pour tout formule $\alpha \in Prop(P)$ on a :
$$\mathcal{H} \vdash_{\mathcal{S}} \alpha \text{ implique } \mathcal{H} \models \alpha$$
- $\mathcal{S}$ est un système formel **complet** si et seulement si pour tout $\mathcal{H}$ tel que $\mathcal{H} \subset Prop(P)$ et pour tout formule $\alpha \in Prop(P)$ on a :
$$\mathcal{H} \vdash_{\mathcal{S}} \alpha \text{ équivaut à } \mathcal{H} \models \alpha$$

Proposition :
Soit $\mathcal{S} = (Prop(P), \mathcal{A}, \mathcal{R}$ un système formel du calcul des propositions, $\mathcal{S}$ est valide si et seulement si :

- Les axiomes de $\mathcal{S}$ sont des tautologies
- Pour chaque règle de la forme $\dfrac{e_1,\dots,e_n}{e_{n+1}}(r)$ de $\mathcal{S}$ on a $e_1,\dots,e_n \models e_{n+1}$. On dit que chaque règle est valide.

Définition : Atome
Soit P un ensemble de variables propositionnelles, un atome (ou littéral) construit sur P est soit une variable propositionnelle, soit la négation d'une variable propositionnelle.

Définition : Soit P un ensemble de variables propositionnelles, une clause sur P est une formule de $Prop(P)$ de la forme $a_1 \vee \dots \vee a_k \vee \neg a_{k+1} \vee \dots \vee \neg a_{k+r}$ où les a_i sont des éléments tous **distincts** de P. On note $CL(P)$ l'ensemble des clauses sur P

Notations : Partant d'une clause $c = a_1 \vee \dots \vee a_k \vee \neg a_{k+1} \vee \dots \vee \neg a_{k+r}$, on peut la noter :

- Sous forme implicative :
  - $c = (a_1 \vee \dots \vee a_k) \Leftarrow (a_{k+1} \wedge \dots \wedge a_{k+r}$
  - $c = (a_{k+1} \wedge \dots \wedge a_{k+r} \Rightarrow (a_1 \vee \dots \vee a_k)$
- Sous forme ensembliste : $c = (\{a_1,\dots,a_k\},\{a_{k+1},\dots,a_{k+r}\}) = (c^+,c^-)$

Si $k = 0$ et $r = 0$ la clause n'a pas d'atomes, on la note $\square$

Définition : Subsomption
Soient deux clauses c et c' définies sur P, on dit que c subsume c' si et seulement si une des 2 conditions équivalentes suivantes est vérifiée :

- Condition syntaxique : Tout littéral de c apparaît dans c'
- Condition sémantique : $\{c\} \models c'$ (i.e. pour toute valuation δ de P, $[c](\delta) = 1$ implique $[c'](\delta) = 1$)

Proposition :
Soit α une formule de $Prop(P)$, α est sémantiquement équivalente à une formule β de la forme $\beta = c_1 \wedge \dots \wedge c_n$ où les c_i sont des causes.

Définition :
Mettre une formule α sous forme clausale c'est trouver un ensemble de clauses $C(\alpha)$ dont la conjonction est équivalente à α

Proposition :
Soient α et β deux formules de $Prop(P)$, soient $\mathcal{C}(\alpha)$ et $\mathcal{C}(\beta)$ les ensembles de clauses associées respectivement à α et β, on a :

- $\mathcal{C}(\alpha \wedge \beta) = \mathcal{C}(\alpha) \cup \mathcal{C}(\beta)$
- $\mathcal{C}(\alpha \vee \beta) = \mathcal{C}(\alpha) \otimes \mathcal{C}(\beta)$ où $E \otimes E' = \{c \vee c' | c \in E, c' \in E'\}$

Proposition :
Soit α une formule de $Prop(P)$, α est une tautologie si et seulement si l'ensemble $\mathcal{C}(\alpha)$ obtenue par l'algorithme de mise sous forme clausale est égal à $\emptyset$

Définition : Système formel de Robinson
Soit P un ensemble de variables propositionnelles et $\mathcal{C}(P)$ l'ensemble des clauses construites sur P. Le système formel $(\mathcal{C}(P),\emptyset,\mathcal{R})$ où

$$\mathcal{R} = \left\{ \frac{c_1 \vee a \vee c_2, c_3 \vee \neg a \vee c_4}{c_1 \vee c_2 \vee c_3 \vee c_4} \ (Resolution), \frac{c_1 \vee a \vee c_2 \vee a \vee c_3}{c_1 \vee a \vee c_2 \vee c_3} \ (Factorisation \ \ +), \frac{c_1 \vee \neg a \vee c_2 \vee \neg a \vee c_3}{c_1 \vee \neg a \vee c_2 \vee c_3} \ (Factorisation \ \ -) \right\}$$

est un système basé sur la règle de résolution.

Proposition :
Le système formel de Robinson est valide

Démonstration :

- Les règles de factorisation sont trivialement valides.
- La règle de résolution est valide, en effet soit δ telle que $[c_1 \vee a \vee c_2](\delta) = 1$ et $[c_3 \vee \neg a \vee c_4](\delta) = 1$
  On a donc $[c_1 \vee c_2](\delta) = 1$ ou $[a](\delta) = 1$
  - Si $[c_1 \vee c_2](\delta) = 1$ alors $[c_1 \vee c_2 \vee c_3 \vee c_4](\delta) = 1$
  - Si $[a](\delta) = 1$ alors $[\neg a](a) = 0$ or $[c_3 \vee \neg a \vee c_4](\delta) = 1$ donc $[c_3 \vee c_4](\delta) = 1$ et ainsi $[c_1 \vee c_2 \vee c_3 \vee c_4](\delta) = 1$

Théorème de Robinson :

Soit $\mathcal{C}$ un ensemble de clauses. $\mathcal{C}$ est contradictoire si et seulement si il existe une démonstration de $\square$ avec hypothèses dans $\mathcal{C}$. On a donc les équivalences suivantes :

$$\mathcal{A} \models \alpha$$

$$ssi$$

$$\mathcal{C}(\mathcal{A}) \cup \mathcal{C}(\alpha) \ \ est \ \ contradictoire$$

$$ssi$$

$$\mathcal{C}(\alpha) \cup \mathcal{C}(\neg\alpha) \vdash_{Resolution} \square$$