Logique

NFP108/SEMS
2017-2018

Systèmes formels
logique propositionnelle (partie 1)

O. Pons

10 octobre 2018

Logique des propositions (syntaxe)

- Alphabet $\mathcal{A}$:
  - Ensemble d'atomes (ou variables propositionnelles) :

$$\mathcal{P} = \{p, q, r, \ldots\}$$

  - Ensemble de connecteurs :

non :	$\neg$		
et :	$\wedge$	ou :	$\vee$
implique :	$\Rightarrow$	équivalent	$\Leftrightarrow$
faux universel :	$\perp$	Vrai universel :	T

  - les symboles (et)

- $\wedge\wedge\wedge$, $\vee\wedge p(r$ et $(p\wedge q)\Rightarrow r$
  - mots sur cet alphabet,
  - mais seul le dernier est

    une **formule de la logique propositionnelle**

Logique des propositions (syntaxe)

- Ensemble des formules sur $\mathcal{A}$:

  le plus petit ensemble $\mathcal{F}$ tel que :

  1. toute variable propositionnelle est dans $\mathcal{F}$
  2. Si $f \in \mathcal{F}$ *alors* $\neg f \in \mathcal{F}$
  3. Si f *et* $g \in \mathcal{F}$ *alors* $(f \propto g) \in \mathcal{F}$ *pour* $\propto \; \equiv \; \wedge \, , \, \vee \, , \, \Rightarrow \, , \, \Leftrightarrow$

C'est une **Définition par induction**

Logique des propositions (syntaxe)

- On peut faire des **démonstration par induction** sur les formules :
  Pour prouver une propriétés $\mathcal{P}$ sur les formules il suffit :
  1. prouver que $\mathcal{P}$ est vérifiée par les variables
  2. prouver que si $\mathcal{P}$ est verifies par$f$ alors elle l'est aussi par $\neg f$
  3. prouver que si $\mathcal{P}$ est vérifiée par des formules f_1 et f_1 alors elle est aussi vérifiée par $(f_1 \propto f_2)$ où $\propto \ \equiv \wedge , \vee , \Rightarrow , \Leftrightarrow$
- Exemple : prouver que toute formule a le même nombre de parenthèses ouvrantes que de parenthèses fermantes

Logique des propositions (syntaxe)

- Autre définition
  - On défini la suite récurrente $\mathcal{F}_n$:
    - $\mathcal{F}_0 = P$
    - $\mathcal{F}_{n+1} = \mathcal{F}_n \cup \{\neg f, f \in \mathcal{F}^n\} \cup \{(f \propto g), f \in \mathcal{F}_n, g \in F_n,\}$
      $\propto \; \equiv \wedge, \vee, \Rightarrow, \Leftrightarrow$
  - Elle est croissante.

$\mathcal{F} = \bigcup_{n \in \mathbb{N}} \mathcal{F}_n$

Logique propositionnelle (sémantique 1)

- **Domaine d´interprétation** : $\mathbb{B} = \{v, f\}$
- Interprétation des variables : **valuation** : $\rho \in \mathcal{P} \to \mathbb{B}$
- Interprétation des connecteurs : associer à chaque opérateur élémentaire o une interprétation : $[\![o]\!] \in \mathbb{B} \times \mathbb{B} \to \mathbb{B}(ou\ \mathbb{B} \to \mathbb{B})$ décrite par une table de vérité.

x	y	x $[\![\wedge]\!]$ y	x $[\![\vee]\!]$ y	x $[\![\Rightarrow]\!]$ y
v	v	v	v	v
v	f	f	v	f
f	v	f	v	v
f	f	f	f	v

x	$[\![\neg]\!]$ x
v	f
f	v

Logique propositionnelle (sémantique 2)

- Interprétation des formules :
  dépend d'une valuation $\rho \in \mathcal{P} \to \mathbb{B}$

$$
\begin{array}{lcl}
[\![\Phi]\!] & : & (\mathcal{P} \to \mathbb{B}) \to \mathbb{B} \\
\\
[\![\bot]\!]_\rho & = & \mathsf{f} \\
[\![x]\!]_\rho & = & \rho(x) \\
[\![\neg\Phi]\!]_\rho & = & [\![\neg]\!][\![\Phi]\!]_\rho \\
[\![\Phi_1 o \Phi_2]\!]_\rho & = & [\![\Phi_1]\!]_\rho [\![o]\!] [\![\Phi_2]\!]_\rho,\ o \in \{\wedge, \vee, \Rightarrow\}
\end{array}
$$

- Exemple :

  Si $\rho = \{p_1 \mapsto v, p_2 \mapsto f\}$,

$$
\begin{array}{lcl}
[\![(p_1 \vee p_2) \Rightarrow \neg p_1]\!] & = & (v[\![\vee]\!]f)[\![\Rightarrow]\!]([\![\neg]\!]v) \\
 & = & \mathsf{v}[\![\Rightarrow]\!]f \\
 & = & \mathsf{f}
\end{array}
$$

Logique propositionnelle (sémantique 3)

- Tables de vérité d'une formule.

p_1	p_2	$(p_1 \vee p_2)$	$\neg p_1$	$(p_1 \vee p_2) \Rightarrow \neg p_1$
v	v	v	f	f
v	f	v	f	f
f	v	v	v	v
f	f	f	v	v

- pour 2 variables, il y a 2^2 fonctions de valuations possibles donc 4 « mondes » possibles.
  Chaque ligne de la table de vérité correspond à un de ces mondes.
- pour n variable on aurait 2^n valuations et donc une table de vérité de 2^n lignes
- que penser de la décidabilité ?

Logique propositionnelle (sémantique 4)

- Une formule Φ est :
  - **Satisfiable** : si il existe une valutation ρ tel que $[\![\Phi]\!]_\rho = v$
  - **Une tautologie** : si $[\Phi]\!]_\rho = v$ pour toute valuation ρ
    Notation : $\models \Phi$
  - **Insatisfiable** (antilogie) : si $[\![\Phi]\!]_\rho = f$ pour toute valuation ρ
- Un ensemble de formule Δ est **satisfiable** si il existe une valuation qui satisfait toutes les formules de Δ
- Un ensemble de formule $\Delta = \{f_1, f_2, \ldots f_n\}$ est **satisfiable** si la formule $f_1 \wedge f_2 \wedge \ldots \wedge f_n$ l'est.

Logique propositionnelle (sémantique 5)

- Deux formules Φ et Ψ sont **équivalentes** si et seulement si, pour tout valuation ρ, $[\![\Phi]\!]_\rho = [\![\Psi]\!]_\rho$
  Autement dit si leurs tables de vérité sont semblables.
- on notera $\Phi \equiv \Psi$
  Attention : $\equiv$ n'est pas un symbole du langage de la logique des proposition mais du méta-langage.
- $\Phi \equiv \Psi$ ssi $\Phi \Leftrightarrow \Psi$
- $\Phi \Leftrightarrow \Psi \equiv \Phi{\Rightarrow}\Psi{\wedge}\Psi{\Rightarrow}\Phi$
- toute tautologie est equivalente à $\top$
- toute antilogie est equivalente à $\bot$

Logique propositionnelle (sémantique 6)

Principales equivalences (à verifier en exercice)

- commutativité de $\wedge$ et $\vee$: $p \wedge q \equiv q \wedge p$, $p \vee q \equiv q \vee p$
- associativité de $\wedge$ et $\vee$: $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$, $(p \vee q) \vee r \equiv p \vee (q \vee r)$
- idempotence de $\wedge$ et $\vee$: $p \wedge p \equiv p$, $p \vee p \equiv p$
- distributivité de $\wedge$ sur $\vee$: $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
- distributivité de $\vee$ sur $\wedge$: $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
- éléments neutres : $p \wedge \top \equiv p$, $p \vee \bot \equiv p$
- absorption : $p \wedge \bot \equiv \bot$, $p \vee \top \equiv \top$
- contradiction : $p \wedge \neg p \equiv \bot$
- tiers exclus : $p \vee \neg p \equiv \top$
- double négation : $\neg\neg p \equiv p$
- lois de De Morgan : $\neg(p \wedge q) \equiv \neg p \vee \neg q$, $\neg(p \vee q) \equiv \neg p \wedge \neg q$
- implication : $(p \Rightarrow q) \equiv \neg p \vee q$
- contraposée : $(p \Rightarrow q) \equiv (\neg q \Rightarrow \neg p)$
- autres absorptions : $p \vee (p \wedge q) \equiv p, p \wedge (p \vee q) \equiv p$

Logique propositionnelle (sémantique 7)

- Φ conséquence (sémantique) d'un ensemble de formule Δ :

  si toute valuation qui satisfait toutes les formules de Δ satisfait Φ
  Notation :$\Delta \models \ F$

- **Théorème :**
  $\Delta \models \Phi$ Si et seulement Si $\Delta \cup \{\neg\Phi\}$ est insatisfiable.

Logique propositionnelle (déduction)

- Système de déduction :
  - Des axiomes (feuilles de l'arbre de preuve)
  - Des règles (feuilles ou noeuds de l'arbre)
- Différent systèmes de déduction :
  - À la Hilbert
  - Déduction Naturelle
  - Calcul des séquents

Logique propositionnelle (déduction à la Hilbert)

- Familles d'axiomes (pour toutes formules F, G et H)
  - Pour l'implication
    - K : $F \Rightarrow (G \Rightarrow F)$
    - S : $(F \Rightarrow (G \Rightarrow H)) \Rightarrow ((F \Rightarrow G) \Rightarrow (F \Rightarrow H))$
  - Pour la conjonction
    - $F \Rightarrow (G \Rightarrow F \wedge G)$
    - $F \wedge G \Rightarrow F$ et $F \wedge G \Rightarrow G$
  - Pour la disjonction
    - $F \Rightarrow F \vee G$ et $G \Rightarrow F \vee G$
    - $F \vee G \Rightarrow ((F \Rightarrow H) \Rightarrow ((G \Rightarrow H) \Rightarrow H))$ (par cas)
  - Faux et négation
    - $\bot \Rightarrow F$
    - $A => ((\neg A \Rightarrow \bot)$ et $(A \Rightarrow \bot)$
- Règles de déduction
  - $$\frac{F \Rightarrow G \quad F}{G}\ MP$$
  - $$\frac{}{A}\ Ax$$ ou A est une instance d'un des schéma d'axiome précédent

Logique propositionnelle (déduction à la Hilbert)

$$\dfrac{\dfrac{\dfrac{}{(p\Rightarrow((p\Rightarrow p)\Rightarrow p))\Rightarrow((p\Rightarrow(p\Rightarrow p))\Rightarrow(p\Rightarrow p))}\,Ax \quad \dfrac{}{(p\Rightarrow((p\Rightarrow p)\Rightarrow p)}\,Ax}{p\Rightarrow(p\Rightarrow p))\Rightarrow(p\Rightarrow p)}\,MP \quad \dfrac{}{p\Rightarrow(p\Rightarrow p)}\,Ax}{p\Rightarrow p}\,MP$$

Pénible, mais utile pour la méta-théorie

Logique propositionnelle (déduction à la Hilbert)

- Correction
  - $\vdash_H F$ implique $\models F$
- Complétude
  - $\models F$ implique $\vdash_H F$

Logique propositionnelle (déduction naturelle)

- Des **jugement** : $\Gamma \vdash \phi$,
  Intuitivement :
  "On peut démontrer ϕ à partir des formule de Γ".
- Des règles

$$\frac{\textit{ensemble de premisses}}{\textit{conclusion}}\ \text{Nom de regle}$$

  Prémisses et conclusion : sont des jugements.

- Construire des arbres de preuves : combiner les règles.

Logique propositionnelle (déduction naturelle)

$$\frac{}{\Gamma, \phi \vdash \phi}\ Ax$$

$$\frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}\ \wedge_i \qquad \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi}\ \wedge_{e1} \qquad \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \psi}\ \wedge_{e2}$$

$$\frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \Rightarrow \psi}\ \Rightarrow_i \qquad \frac{\Gamma \vdash \phi \Rightarrow \psi \quad \Gamma \vdash \phi}{\Gamma \vdash \psi}\ \Rightarrow_e$$

$$\frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi}\ \vee_{i1} \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \phi \vee \psi}\ \vee_{i2} \qquad \frac{\Gamma \vdash \phi \vee \psi \quad \Gamma, \phi \vdash \theta \quad \Gamma, \psi \vdash \theta}{\Gamma \vdash \theta}\ \vee_{e1}$$

$$\frac{\Gamma, \neg\phi \vdash \perp}{\Gamma \vdash \phi}\ \perp_e$$

$$\frac{\Gamma, \phi \vdash \perp}{\Gamma \vdash \neg\phi}\ \neg_i \qquad \frac{\Gamma \vdash \phi \quad \Gamma \vdash \neg\phi}{\Gamma \vdash \psi}\ \neg_e$$

Logique propositionnelle (déduction naturelle)

- Exemples

$$\dfrac{\dfrac{\dfrac{\dfrac{}{(A\Rightarrow\bot), A\vdash A}\,Ax \qquad \dfrac{}{(A\Rightarrow\bot), A\vdash(A\Rightarrow\bot)}\,Ax}{(A\Rightarrow\bot), A\vdash\bot}\,\Rightarrow_e}{(A\Rightarrow\bot)\vdash\neg A}\,\neg_i}{\vdash(A\Rightarrow\bot)\Rightarrow\neg A}\,\Rightarrow_i$$

On montre de même : $\vdash\neg A\Rightarrow(A\Rightarrow\bot)$

- Donc $\neg\phi$ abréviation pour $\phi\Rightarrow\bot$.

Logique propositionnelle (déduction naturelle)

- autre notation :

$$
\begin{array}{lr}
\vdash (A \Rightarrow \bot) \Rightarrow \neg A & \Rightarrow_i \\
(A \Rightarrow \bot) \vdash \neg A & \neg_i \\
(A \Rightarrow \bot), A \vdash \bot & \Rightarrow_e (1)(2) \\
(1)(A \Rightarrow \bot), A \vdash A & Ax \\
(2)(A \Rightarrow \bot), A \vdash A \Rightarrow \bot & Ax
\end{array}
$$

Logique propositionnelle (déduction naturelle)

- Démo en utilisant :
  `http://cedric.cnam.fr/~pons/DEDUCTION/deduction1.html`

Logique propositionnelle (déduction naturelle)

- Règles dérivées :
  - règle de coupure

$$\frac{\Gamma; A \vdash B \quad \Gamma \vdash A}{\Gamma \vdash B}\ \textit{coupure}$$

  - preuve :

$$\frac{\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} \Rightarrow_i \qquad \Gamma \vdash A}{\Gamma \vdash B} \Rightarrow_e$$

  - ou

$$\begin{array}{ll} \Gamma \vdash B & \Rightarrow_e (1)(2) \\ (1) \Gamma \vdash A \Rightarrow B & \Rightarrow_i \\ \Gamma, A \vdash B & \text{prémisse gauche} \\ (2) \Gamma \vdash A & \text{prémisse droite} \end{array}$$

Un Rapport (évidemment très neutre ;-) sur la reforme des retraites prévoit que :

1. Si la reforme n'est pas équitable, elle devra est promue par un intense matraquage de "communication".
2. si le Medef s'en mêle, la reforme ne sera pas équitable.
3. Si la reforme est équitable, il n'y aura pas de mouvement social.

La reforme est proposée peut après. Il y a un mouvement social. Les médias en déduisent que le Medef s'en est mêlé tandis que l'opinion publique pense avoir subit une intense campagne de "matraquage communicatif".

1. Modéliser le problème
2. Montrer que l'opinion publique a raison. (preuve en déduction naturelle).
3. quelles sont les hypothèses utiles
4. Montrer par des valeur de vérités bien choisie que les médias se trompent.

1. on introduit les propositions atomiques suivantes :
   - E : "la reforme est équitable",
   - C : "il y a une intense campagne de matraquage de communication"
   - M : "le Medef s'en mèle"
   - S : "il y a un mouvement social".

   Le rapport énnonce donc :

   1.1 $\neg E \Rightarrow C$
   1.2 $M \Rightarrow \neg E$
   1.3 $E \Rightarrow \neg S$

   Le fait qu'il y ait un mouvement social s'ennonce : S

La conclusion des médias est :

$$(\neg E \Rightarrow C), (M \Rightarrow \neg E), (E \Rightarrow \neg S), S \vdash M$$

ou ce qui est équivalent :

$$\vdash (\neg E \Rightarrow C) \Rightarrow (M \Rightarrow \neg E) \Rightarrow (E \Rightarrow \neg S) \Rightarrow S \Rightarrow M$$

La conclusion de l'opinion publique est :

$$(\neg E \Rightarrow C), (M \Rightarrow \neg E), (E \Rightarrow \neg S), S \vdash C$$

ou ce qui est équivalent :

$$\vdash (\neg E \Rightarrow C) \Rightarrow (M \Rightarrow \neg E) \Rightarrow (E \Rightarrow \neg S) \Rightarrow S \Rightarrow C$$

2. Le "script de preuve" (ie. la suite de regle a appliquer) est la suivante :

```
(~E=>C)=>(M=>~E)=>(E=>~S)=>S=>C
IntroImp
IntroImp
IntroImp
IntroImp

ElimImp ~E
 Axiom

 IntroNoT
 elimNot S
   Axiom

   ElimImp E
     Axiom

     Axiom
```

3. La preuve n'ulise pas le fait que $M \rightarrow \ E$ les autres sont utiles.

4. On peut construire la table de vérité de la formule

$$(\neg E \Rightarrow C) \Rightarrow (M \Rightarrow \neg E) \Rightarrow (E \Rightarrow \neg S) \Rightarrow S \Rightarrow M$$

mais ce n'est pas nécessaire on cherche a montrer que cette formule n'est pas une tautologie. ie qu'il existe un choix v de valeurs de vérité tel que
$v[\neg E \Rightarrow C] = v[M \Rightarrow \neg E] = v[E \Rightarrow \neg S] = v[S] = 1$ et
$v[M] = 0$
On a donc les contrainte $v[M] = 0$ (ce qui entraîne
$v[M \Rightarrow \neg E] = 1$) $v[S] = 1$, $v[\neg E \Rightarrow C] = v[E \Rightarrow \neg S] = 1$
Pour avoir $v[E \Rightarrow \neg S] = 1$ avec $v[S] = 1$ il faut avoir
$v[E] = 0$. Comme on doit aussi avoir $v[\neg E \Rightarrow C] = 1$ in doit avoir $v[C] = 1$
En résumé si $(v[E], v[C], v[M], v[S]) = (0, 1, 0, 1)$ les quatre hypothèses sont vrais mais la conclusion des médias est fausse.

Logique propositionnelle (preuve constructive)

- Preuve constructive : n'utilise pas la règle du Tiers Exclu.
- La déduction naturelle constructive est correcte (mais pas complète) pour la logique propositionnelle.
- Exemple de tautologie non démontrable en déduction naturelle constructive :
  : formule de Peirce : $(((A \Rightarrow B) \Rightarrow A) \Rightarrow A)$

Logique propositionnelle (preuve constructive)

- Problème :
  "Montrer qu'il existe deux nombre irrationnels a et b telque a^b est rationnel".
- Démonstration :
  Considérons $\sqrt{3}^{\sqrt{2}}$
  - si $\sqrt{3}^{\sqrt{2}} \in \mathbb{Q}$ on pose $a = \sqrt{3}$ et $b = \sqrt{2}$
  - si $\sqrt{3}^{\sqrt{2}} \neg \in \mathbb{Q}$ on pose $a = \sqrt{3}^{\sqrt{2}}$ et $b = \sqrt{2}$ et donc $a^b = 3 \in \mathbb{Q}$

  Donc a et b existent, mais quelles sont leurs valeurs ?
  Notion de témoin.

Coq : un assistant de preuve en logique constructive

- Disponible : `http://coq.inria.fr/`
- Notations $\Rightarrow$ noté $\rightarrow$ (`->`)
- Parenthèsage :
  $A \rightarrow (B \rightarrow C)$ simplement noté $A \rightarrow B \rightarrow C$.
- But courant :
  un séquent $A_1, \ldots, A_n \vdash A$ s'affiche :

```
 Hn : An
...
 H1 : A1
 =========================
 A
```

Logique propositionnelle (déduction naturelle)

- Règle $\Rightarrow_i$

```
                                          H : A
 =====================                   =====================
  A->B                                     B
```

- Commande : `intros H.`
- Remarques
  - on peut enchaîner plusieurs introductions : `intros H1 H2.`
  - on peut laisser le système nommer les hypothèses : `intros.`

Coq : un assistant de preuve en logique constructive

- Règle $\Rightarrow_e$

```
  H : A->B                                  H : A->B
 =====================                     ==================
  B                                          A
```

- Commande : `apply H.`
- Remarques
  - contrairement a $\Rightarrow_e$, il y un seul sous arbre de preuve : il faut déjà avoir une preuve de $A \Rightarrow B$ dans son contexte
  - forcer Coq a générer un sous-but pour $A \Rightarrow B$, on peut taper `assert (A->B).`

Coq : un assistant de preuve en logique constructive

- Règle *Ax*

```
H : A
============================
  A
```

- Commande : `apply H` ou `exact H`
- Remarque
  on peut eviter de nommer l'hypothèse en utilisant la commande `assumption`

Coq : un assistant de preuve en logique constructive

- Règle $\wedge_i$

```
H : A                          H : A
H0 : B                         H0 : B
========================= ============================
 A /\ B                        A

                               H : A
                               H0 : B
                              ============================
                               B
```

- Commande : `split.`

Coq : un assistant de preuve en logique constructive

- Règle $\wedge_e$

```
 H : A /\ B                                 H : A
 =======================                    H0 : B
  C                                         ========================
                                            C
```

- Commande : `destruct H.`
- Remarque :
  on peut nommer les hypothèses introduites avec la syntaxe
  `destruct H as [H H0].`
- `elim H;intros.` garde la conjonction dans le contexte.

Coq : un assistant de preuve en logique constructive

- Règle $\vee_i 1$ et $\vee_i 2$

```
==========================          =======================
 A \/ B                                       A
```

- Commande : `left.`
- Remarque $\vee_i 2$ correspond a la commande `right.` ‘

Coq : un assistant de preuve en logique constructive

- Règle $\vee_e$

```
                                   H : A
                                   ===========================
                                   C

H : A \/ B                         H : B
========================          ===========================
 C                                 C
```

- Commande : `destruct H.`

Coq : un assistant de preuve en logique constructive

- Règle $\perp_e$

```
H : False
==============================
 A
```

- Commande : `elim H.`
- Remarque
- `elim H` peut aussi s'utiliser quand $H : A_1 \rightarrow \ldots A_n \rightarrow \mathit{False}$ (mais il faut alors décharger n sous-buts)

Coq : un assistant de preuve en logique constructive

- Règle $\neg_i$

```
                                          H : A
 =========================                ====================
   ~A                                      False
```

- Commande : `intros H.`
- Remarque
  - $\neg A$ sucre syntaxique pour $A \rightarrow \mathit{False}$.
  - intros. ne marche pas ici.

Coq : un assistant de preuve en logique constructive

- Règle $\neg_e$

```
                                    ==========================
                                     ~A


   ==========================        ==========================
     B                                 A
```

- Commande : `absurd A.`
- Remarque
  la commande est utile même quand B n'est pas egal a *False*.

Coq : un assistant de preuve en logique constructive

- Règle : Tiers Exclu
- Il faut ajouter un axiome (excluded middle) !

```
Variable em :  P:Prop , P \/ ~P.
```

```
                                          H : A
                                          ==========
                                            C

                                          H : ~A
  ============================            ==========
    C                                       C
```

- Commande : `destruct (em A).`