logique mathématique

F. BOUDJERIDA Univ de Jijel
2015- 2016

Objectif de la logique mathématique est de savoir trois notions :

- Logique propositionnelle
- Logique des prédicat
- Calculabilité

Introduction

- Le mot logique vient du grecque « logos » qui signifie parole discours.
- Logique est la science qui étude les règles qui doivent respecter tout raisonnement valide et qui permet de distinguer un raisonnement valide, d'un raisonnement qui ne l'est pas
- Logique mathématique elle s'intéresse à l'organisation et à la cohérence du discours mathématique c-à-d aux notions de validité et de presse.

Introduction

Il y a deux types de langages :

1. Langage naturel
2. Langage formel

Langage naturel

- Est le langage que nous utilisons dans la vie ; de tous les jours
- Ce langage a deux inconvénients majeurs qu'on utilise en mathématique ;
  - la complexité des phrases qui rend les choses plus compliqué, il faux plusieurs lignes incompréhensible pour dire quelque chose qui peut se résumer une simple équation
  - les ambigüités du langage qui peuvent conduire à des erreurs

Langage naturel

Alors l’idée est d’utiliser un langage symbolique permet de simplifier de choses en mathématique.

Langage formel

Lorsque l'on définit on doit définir deux choses qui caractérisent ce langage :

- syntaxe
- sémantique

Langage formel
→ syntaxe

- **Un alphabet** : un ensemble de symboles
- **Une grammaire** : un ensemble de règles qui définit quels mots appartiennent au langage formel

Mot : est une suite ordonnée de symboles, ces symboles appartenant à l'alphabet

- Un langage formel est un ensemble des mots de longueur finie par un alphabet et une grammaire

chapitre 1

logique propositionnelle

Le langage propositionnel

- **L'alphabet :**
  - d'un ensemble, fini ou dénombrable, de variables *propositionnelles* notées p , q , r , ...
  - de deux constantes : vrai (notée T) et faux (notée $\bot$) ;
  - d'un ensemble de *connecteurs logiques* : et (noté $\wedge$), ou (noté $\vee$),non (noté ¬), implique (noté →), équivalent (noté ↔) ;
  - les parenthèses : (,).

.

Le langage propositionnel

- **La grammaire**

définie par les deux règles suivantes :

a) Les variables propositionnelles sont des formules.
b) T et $\perp$ sont des formules.
c) Si A et B sont des formules alors
   $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $\neg A$ et $(A \leftrightarrow B)$
d) une expression n'est une formule FBF que si elle écrite conformément aux règles (a) et (b) et (c)

Remarque

L'appartenance à l'ensemble des formules est décidable

(ie il existe un algorithme permettant de décider si un mot quelconque est une formule)

Exemple :

$(P \vee Q)$
$(P1 \rightarrow (P2 \vee P3))$
ce sont des formules bien formés selon les règles

$(P1 \rightarrow (P2 \neg P3))$
$(PQ \vee)$
ce ne sont pas des FBF selon les règles : $\neg$ est uni et $P \vee Q$

Ambiguïtés

L'utilisation de la *notation infixe* s'accompagne de problèmes de lecture :
Comment lire $\varphi 1 \wedge \varphi 2 \vee \varphi 3$? $\varphi 1 \rightarrow \varphi 2 \rightarrow \varphi 3$?

Pour lever les ambiguïtés, on utilise les parenthèses ou des règles de priorité entre opérateurs :

- si *op*1 a une plus grande précédence (priorité) que *op*2 alors
  *e*1 *op*1 *e*2 *op2* *e*3 est équivalent à ((*e*1 *op1* *e*2) *op*2 *e*3)

Par ex :2 × 3 + 5 = (2 × 3) + 5 = 11 ≠ 2 × (3 + 5) = 16

- si *op*2 a une plus grande précédence (priorité) que *op*1 alors
  *e*1 *op*1 *e*2 *op2* *e*3 est équivalent à (*e*1 *op1* (*e*2 *op*2 *e*3))

Par ex :2 + 3 × 5 = 2 + (3 × 5) = 17.

- si *op* est associatif à gauche alors *e*1 *op* *e*2 *op* *e*3 est équivalent à
  ((*e*1*op* *e*2) *op* *e*3)

Par ex : 10/2/5 = (10/2)/5 = 1 ≠ 10/(2/5) .

Une fois ces règles fixées, à chaque formule correspond un et un seul arbre de lecture.

Règles de précédence

- Ordre de précédence $\prec$ sur les opérateurs :
  $\leftrightarrow \prec \rightarrow \prec \vee \prec \wedge \prec \neg$
  et associativité à gauche pour $\leftrightarrow$, $\vee$, $\wedge$ et à droite pour $\rightarrow$.

Exemples

$p \vee q \wedge r$ se lit $p \vee (q \wedge r)$
$p \rightarrow q \rightarrow p$ se lit $p \rightarrow (q \rightarrow p)$
$p \vee q \rightarrow r$ se lit $(p \vee q) \rightarrow r$
$\neg p \wedge q$ se lit $(\neg p) \wedge q$
$p \rightarrow q \wedge r \rightarrow s$ se lit $p \rightarrow ((q \wedge r) \rightarrow s)$

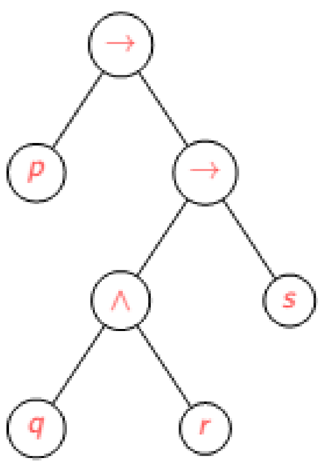
Les parenthèses permettent de contrecarrer ces règles, si elles ne conviennent pas. Elles permettent aussi de rendre une formule plus lisible, ou de ne pas devoir retenir les règles de précédence

Arbre correspondant à une formule

La formule $p \longrightarrow q \wedge r \longrightarrow s$ est donc équivalente à

$$p \longrightarrow ((q \wedge r) \longrightarrow s)$$

et donc son arbre de lecture est :



Profondeur d'une formule

La *profondeur d'une formule* φ est la profondeur de son arbre de lecture associée , notée $A\ \varphi$.

Elle se définit de manière inductive comme suit :

- cas de base : si $\varphi = p\ , T,\ \perp$ où p est une proposition alors
  $Prof(\varphi) = 0$;
- cas inductif : si $\varphi = \varphi\ 1 op\ \varphi\ 2$ alors
  $Prof(\varphi) = 1 + \mathbf{max}(Prof(\varphi\ 1), Prof(\varphi\ 2))$;
  si $\varphi = (\varphi\ 1)$ alors $Prof(\varphi) = Prof(\varphi\ 1)$.

Sémantique: validité, satisfiabilité

Valuation

La sémantique associé une fonction de valuation ϑ telle que

$$\vartheta: \quad Vp \rightarrow \{0; 1\} \qquad \begin{array}{l} 0 - F \\ 1 - V \end{array}$$

Soit on dit valuation ou interprétation

- Grace à cela on pourra calculer la valeur de vérité de la formule par rapport à cette interprétation.

Remarque

si F contient n variables propositionnelles alors il existe exactement 2^n interprétations de F.

Définition

- Une formule α est **valide** est on note $\models \alpha$ si elle est vraie pour toute interprétation (on dit aussi que c'est une **tautologie**).
- Elle est **satisfiable** s'il existe en moins une interprétation pour laquelle elle est vraie
- et elle est **contradictoire** ou **insatisfiable** sinon, c'est-à-dire quand la formule est fausse quelle que soit l'interprétation.

Définition

- On étend ces notions à un ensemble E fini ou infini de formules:
- L'ensemble E est valide si pour toute interprétation I et toute formule $\alpha \in$ E on a

  $$V_I(\alpha)=v.$$
- L'ensemble E est satisfiable s'il existe une interprétation I telle que pour toute formule $\alpha \in$ E on a $V_I(\alpha)=v.$
- L'ensemble E est insatisfiable si pour toute interprétation I, il existe une formule $\alpha \in$ E telle que $V_I(\alpha)=f.$

Remarques

- Dire que deux formules F et G sont **valides** est équivalent à dire que l'ensemble {F,G} est valide. Par contre si deux formules sont satisfiables, cela ne veut pas dire que l'ensemble {F,G} est satisfiable.
  - Il suffit de prendre {x,¬ x} qui est insatisfiable car une interprétation I ne peut pas rendre vrai à la fois x et ¬ x alors que, pour chaque formule prise séparément, on peut trouver une interprétation qui la rend vraie.

Remarques

- De manière duale si F et G sont **insatisfiables**, il en est de même de l'ensemble {F,G} mais le contraire n'est pas vrai, un ensemble de formules peut être insatisfiable alors que chaque formule individuellement est satisfiable.
- Un ensemble de formules **valide** est a fortiori satisfiable. Comme la valeur d'une formule est vraie (resp. fausse) si et seulement si la valeur de sa négation est fausse (resp. vraie), on en déduit une correspondance entre validité et insatisfiabilité.

Théorème

- La formule F est **valide** (resp. **insatisfiable**) ssi la formule ¬ F est insatisfiable (resp. valide).
- L'ensemble fini de formules $\{A_1,\ldots,A_n\}$ est **valide** (resp. **satisfiable**, resp. **insatisfiable**) ssi la formule $A_1 \wedge \ldots \wedge A_n$ est valide (resp. satisfiable, resp. insatisfiable)

Définition(Modèle d'une formule)

- Un **modèle** d'une formule F est une interprétation I des variables de cette formule qui rend la formule vraie c'est-à-dire telle que $V_I(F)=1$.

Exemple *Soit la formule P définie comme* $\neg(x \wedge y) \rightarrow \neg y$

– *L'interprétation* $\{x \mapsto 1; y \mapsto 0\}$ *est un modèle, par contre* $\{x \mapsto 0; y \mapsto 1\}$ *n'en est pas un.*

Définition (Conséquence logique $\models$)

- *On dira que P est une* **conséquence logique** *d'un ensemble de formules $\{A_1,\ldots,A_n\}$ et on écrira* $\boldsymbol{A_1,\ldots,A_n \models P}$ *si tout modèle de $\{A_1,\ldots,A_n\}$ est aussi un modèle de P.*

Remarque

- Si A est une tautologie, on peut écrire $\vDash A$.
- Si $A\rightarrow B$ est une tautologie($\vDash A\rightarrow B$), on dit de A qu'elle implique logiquement B.
- Si $A\leftrightarrow B$ est une tautologie($\vDash A\leftrightarrow B$), alors A et B sont dites logiquement équivalentes et on note $A\equiv B$.

Proposition

Pour toutes formules $A_1,\ldots,A_n$ et P, on a les propriétés suivantes:

- ***$A_1,\ldots,A_n \models P$ ssi $A_1 \wedge \ldots \wedge A_n \rightarrow P$ est valide***
- ***si $\{A_1,\ldots,A_n\}$ est insatisfiable alors pour toute formule P, on a $A_1,\ldots,A_n \models P$***
- ***$A_1,\ldots,A_n \models P$ ssi l'ensemble de formules $\{A_1,\ldots,A_n,\neg P\}$ est insatisfiable.***
-

Théorème

Si $\models A$ et $\models A \rightarrow B$ alors $\models B$

Vérité et substitution

- Soit β une formule où figure la variable propositionnelle p, et soit β' la formule obtenue à partir de β en substituant à p en toutes ses occurrences une formule α.
- La validité et l'insatisfiabilité qui sont des propriétés de toutes les interprétations ne changent pas lorsque l'on effectue une substitution dans une formule, un ensemble de formules et ne change pas la notion de conséquence logique.

Vérité et substitution

- Si une formule β est valide (resp. insatisfiable) alors β' est valide (resp. insatisfiable).
- Si un ensemble de formules E est valide (resp. insatisfiable) alors

$E[p \leftarrow \alpha] =^{def} \{\beta' [p \leftarrow \alpha] | \beta' \in E\}$ est valide (resp. insatisfiable).

- Si $A_1, \ldots, A_n \vDash B$ alors

$A_1[x \leftarrow \alpha], \ldots, A_n[x \leftarrow \alpha] \vDash B[x \leftarrow \alpha]$.

Équivalences booléennes

- On utilise plusieurs connecteurs booléens mais il y a des équivalences entre ces connecteurs.
- On écrira $P \equiv Q$ si $P \vDash Q$ et $Q \vDash P$ ce qui revient aussi à dire que les valeurs de vérité de *P* et *Q* coïncident pour toute interprétation
- **Lois algébriques :l**'ensemble des booléens avec les opérations de conjonction et de disjonction forme ce que l'on appelle une *Algèbre de Boole*.

Équivalences booléennes

La conjonction et la disjonction sont associatifs et commutatifs :

- $P \wedge Q \equiv Q \wedge P \quad P \vee Q \equiv Q \vee P$
- $(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$

  $(P \vee Q) \vee R \equiv P \vee (Q \vee R)$

1 et 0 sont des éléments neutres ou absorbants :

- $P \wedge 1 \equiv P \quad P \vee 1 \equiv 1 \quad P \vee 0 \equiv P \quad P \wedge 0 \equiv 0$

Les lois de Morgan établissent le comportement de la négation par rapport aux autres connecteurs :

- $\neg 0 \equiv 1 \quad \neg 1 \equiv 0 \quad \neg\neg P \equiv P$
- $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$
- $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$

Équivalence entre connecteurs.

Certains connecteurs peuvent se définir en fonction d'autres :

- $\neg P \equiv P \rightarrow 0$
- $P \rightarrow Q \equiv \neg P \vee Q$
- $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$

  $\equiv (P \wedge Q) \vee (\neg P \wedge \neg Q)$

Formes normales

- Les équivalences mises en évidence ci-dessus montrent que la même propriété peut s'exprimer de plusieurs manières différentes.
- On a par exemple déjà établi dans le paragraphe précédent que l'on pouvait se passer des connecteurs $\rightarrow$ et $\leftrightarrow$ en utilisant à la place les connecteurs $\neg$, $\vee$ et $\wedge$. On constate que dans le cas de l'équivalence, la formule transformée est deux fois plus grosse que la formule initiale, donc l'élimination du connecteur peut avoir un coût important.

Forme normale de négation(FNN)

Les lois de Morgan rappelées dans la section

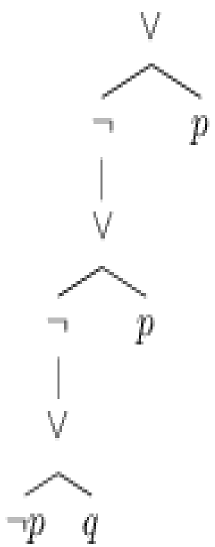
permettent de distribuer la négation vers les variables propositionnelles.

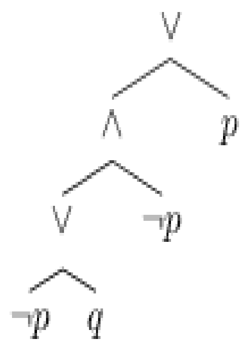
- **Définition 1 (Littéral)** On appelle littéral une formule qui est soit une formule atomique(variables propositionnelles x), soit la négation d'une formule atomique (variables propositionnelles ¬ x).
- **Proposition 1** Toute formule propositionnelle est équivalente à une formule propositionnelle qui ne comporte pas de connecteur ¬ sauf dans un littéral.
- Une telle formule est dite en forme normale de négation.

Forme normale de négation(FNN)

Exemple Soit la formule $((p \to q) \to p) \to p$ dans laquelle on élimine les implications, cette formule devient : $\neg (\neg (\neg p \vee q) \vee p) \vee p$.

- On peut ensuite repousser les négations en commençant par l'extérieur:





On obtient la formule :
$((\neg p \vee q) \wedge \neg p)) \vee p$
qui peut se simplifier en
$\neg p \vee p$ puis en 1.

Forme normale conjonctive (FNC)

- **Définition 2 (Clause)** On appelle clause une formule propositionnelle qui n'utilise que des littéraux et la disjonction.
- Comme la disjonction est associative et commutative, la clause peut toujours s'écrire de la forme $l_1 \vee l_2 \vee \dots \vee l_n$ avec l_i un littéral et on peut réarranger les littéraux dans l'ordre que l'on veut (par exemple en ordonnant les variables propositionnelles, puis pour chaque variable en mettant d'abord les littéraux positifs puis les négatifs). Par ailleurs, on a $P \vee P \equiv P$ et $P \vee \neg P \equiv 1$.
- On peut se ramener à une clause dans laquelle chaque variable propositionnelle apparaît au plus une fois sous forme positive ou négative.
- Par convention, la clause associée à un ensemble vide de littéraux est interprétée comme la formule fausse 0.

Forme normale conjonctive (FNC)

Proposition 3

Soient L_1 et L_2 des ensembles de littéraux, si $L_1 \subseteq L_2$ alors la clause correspondant à L_2 est une conséquence logique de la clause correspondant à L_1.

Définition 3 (FNC)

Une formule est dite en forme normale conjonctive si elle s'écrit comme une conjonction de clauses.

Proposition 4

Toute formule P admet une forme normale conjonctive, c'est-à-dire qu'il existe une formule Q équivalente à P et qui est en forme normale conjonctive.

Forme normale conjonctive (FNC)

Exemple 2 Soit la formule P : $(p \rightarrow q) \rightarrow (\neg p \wedge q)$, on commence par écrire la table de vérité :

- Les lignes qui nous intéressent sont la première et la dernière et on doit exprimer une formule qui dit exactement que l'on n'est pas dans un de ces deux cas.
- La formule qui dit que l'on n'est pas dans le cas de la ligne 1 est $\neg p \vee \neg q$ et la formule qui dit que l'on n'est pas dans le cas de la ligne 4 est $p \vee q$.

p	q	$\neg p \wedge q$	$p \rightarrow q$	$(p \rightarrow q) \rightarrow (\neg p \wedge q)$
1	1	0	1	0
1	0	0	0	1
0	1	1	1	1
0	0	0	1	0

La forme normale conjonctive est donc $(\neg p \vee \neg q) \wedge (p \vee q)$.

Forme normale disjonctive(FND)

Définition 4 (FND)

Une conjonction élémentaire est une formule uniquement composée de littéraux et de conjonctions.

Une formule P est en forme normale disjonctive si elle s'écrit comme une disjonction de conjonctions élémentaires.

Proposition 6 Pour toute formule propositionnelle P, il existe une formule Q équivalente en forme normale disjonctive.

Preuve: Comme précédemment, on peut construire la table de vérité de la formule P par rapport aux variables propositionnelles $\{p_1,\ldots,p_n\}$. On s'intéresse à l'ensemble des n-uplets $(b_1,\ldots,b_n)$, pour lesquels la table de vérité donne la valeur 1. Pour chaque ligne on construit une conjonction élémentaire $l_1 \wedge l_2 \wedge \ldots \wedge l_n$ avec $l_i=p_i$ si $b_i=1$ et $l_i=\neg\, p_i$ si $b_i=0$. Il suffit ensuite de prendre la disjonction de ces formules pour trouver une formule équivalente à P.

Forme normale disjonctive(FND)

Exemple 3 En reprenant la formule P de l'exemple 2, on obtient la forme normale disjonctive : $(p\wedge \neg q) \vee (\neg p \wedge q)$

- Si la formule est en forme normale disjonctive alors il est facile de trouver un modèle. On suppose que les branches de la disjonction sont en forme "simplifiée" c'est-à-dire qu'une variable propositionnelle apparaît dans un seul littéral. Il suffit de choisir une des branches de la disjonction. Pour chaque variable propositionnelle x qui apparaît dans un littéral *l* on prend dans le modèle $x\mapsto 1$ si l=x et $x\mapsto 0$ si $l=\neg x$. Il peut arriver qu'il n'y ait aucune composante en forme simplifiée: c'est le cas où la formule initiale est équivalente à 0 et donc il n'y a pas de modèle.
- Cependant, la mise en forme normale n'est pas forcément la meilleure manière de trouver un modèle.
- En effet, cette forme normale peut être exponentiellement plus grosse que la formule initiale.

L’aspect syntaxique

- L’approche sémantique(la méthode de table de vérité) ne convient pas pour faire des raisonnement sur des formules propositionnelles.
- L’approche syntaxique déductive a pour but de calculer les conséquences logiques par l’application de règles d’inférences.

L'aspect syntaxique

- Pour établir la validité des formules, on introduit un système de déduction qui permet de déduire qu'une formule est une conséquence logique d'un ensemble de formules, et on écrit :

$$T \vdash P$$

Ensemble des formules Une formule

- L'interprétation de cette relation est que si on a dériver $T \vdash P$ et que les formules dans T sont vrais alors la formule p est vraie.

T : un ensemble de formules que l'on pourra écrire $H_1, H_2, \ldots, H_n$ et sont appelés les hypothèses(les prémisses)

P : la conclusion

L’aspect syntaxique

- Les preuves en mathématique on utilise des règles qui nous permettent étape par étape de déduire la conclusion à partir de prémisses, ces règles sont appelées les règles d’inférences qui sont de la forme suivante :

$$\frac{h1,h2,\dots.hn}{C}$$

les prémisses

la conclusion

- On peut se lire sous les hypothèses h1,h2,….hn on peut déduire C.

La déduction naturelle

- La déduction naturelle a été développé par Gentzen (1934)
- Dans cette méthode il y a des hypothèses et les règles d'inférences on notera

$$\frac{h1,h2,\ldots.hn}{C}$$

- Le fait qu'on peut produire c à partir de h1,h2,….hn (inférence directe)

Règles d'inférences

1) Modus ponens(MP) :

$A \rightarrow B, A \vdash B$ ou $\dfrac{A \rightarrow B,\ A}{B}$

2) Modus tollens(MT) :

$A \rightarrow B, \neg B \vdash \neg A$ ou $\dfrac{A \rightarrow B,\ \neg B}{\neg A}$

3) Introduction de la conjonction($i\wedge$) :

$A, B \vdash A \wedge B$ $\dfrac{A, B}{A \wedge B}$

Règles d'inférences

4) Introduction de la disjonction :

(i1∨) $\dfrac{A}{A \vee B}$ **(i2∨)** $\dfrac{B}{A \vee B}$

5) Elimination de la conjonction

(e1∧) $\dfrac{A \wedge B}{A}$ **(e2∧)** $\dfrac{A \wedge B}{B}$

6) Elimination de la disjonction

(e∨) $\dfrac{T \vdash A \vee B \qquad T, A \vdash C \qquad T, B \vdash C}{C}$

Règles d'inférences

Exemple : $A \vee B \vdash B \vee A$?

$$(e\vee)\ \dfrac{A \vee B \text{ (hyp)} \qquad \mathbf{(i2\vee)}\ \dfrac{A}{B \vee A} \qquad \mathbf{(i1\vee)}\ \dfrac{B}{B \vee A}}{B \vee A}$$

Donc $A \vee B \vdash B \vee A$

7) Introduction de l'implication

$$\dfrac{T, A \vdash B}{T, \vdash A \rightarrow B}$$

Remarques

- Si on a une déduction naturelle de B sous un ensemble d'hypothèse dont éventuellement l'hypothèse A, on peut contenir une déduction naturelle de ($A \rightarrow B$)
- Dans la pratique, on appliquera la règle, on filtrera le n° de A que l'on souhaite retirer des hypothèses et on indiquera leur n° à droite de la barre on dit alors que l'hypothèse a déchargé.
- on tire alors que la formule $A \rightarrow B$ est correcte et elle est un loi logique ou une théorème.

Règles d'inférences

Théorème(formule démontrable) :Qui est le dernier élément d'une démonstration sans hypothèses.

8) Elimination de l'implication

$$\frac{T \vdash A \rightarrow B \quad T \vdash A}{T \vdash B}$$

9) Négation Soit $\perp$ la formule appelé l'absorbe (c'est une formule qui dans toutes ses interprétations prend faux)

- **Introduction de négation**

$$\frac{T, A \vdash \perp}{\neg A}$$

- **Elimination de la négation**

$$\frac{\textbf{(i}\neg\textbf{)}\ A \qquad \neg A \qquad \textbf{(e}\neg\textbf{)}}{\perp}$$

- **Le principe efalso**

$$\frac{\perp}{A}$$

- **Le principe de double négation**

$$\frac{\neg A \vdash \perp}{A}$$

Théorie formelle

- Une théorie formelle T est définie par :
- Un alphabet
- Un ensemble des FBF
- Un ensemble d’axiomes
- Un ou plusieurs règles d’inférences

Théorie formelle

- Une preuve dans T est une séquence de formules $\alpha_1, \alpha_2, \ldots, \alpha_n$ telles que tout α_i est soit un axiome soit obtenue à partir de certaines des formules qui précédent dans la séquence par application des règles d'inférences.
- Axiome : une formule que l'on admet qu'elle est vraie(on ne la démontre pas).
- Une formule α admet une preuve dans T(α est la dernière ligne d'une preuve) on dit de α qu'elle est démontrable ou que α est un théorème de T est on note $\vdash \alpha$

Exemple de théorie pour le logique propositionnelle

On définit une théorie comme suit :

1) L'alphabet :

- ensemble de variables propositionnelles notées P,Q, …
- les connecteurs logiques $\neg$ et $\rightarrow$
- les parenthèses

exemples : démontrer que $\{\neg , \rightarrow\}$ forme un système complet.

Définition Un ensemble T de connecteurs est complet si, étant donnée une formule F du calcul propositionnel, on peut trouver une formule F' dans laquelle n'interviennent que les connecteurs de T et telle que $F \equiv F'$

Exemple de théorie pour le logique propositionnelle

2) Les formules bien formées

- Les vp sont FBF
- Si α et β sont FBF alors

$$\left.\begin{array}{c}(\neg\alpha)\\(\alpha\rightarrow\beta)\end{array}\right\}\text{sont des FBF}$$

- **Les axiomes**

A1 : $(\alpha\rightarrow(\beta\rightarrow\alpha))$

A2 : $(\alpha\rightarrow(\beta\rightarrow\gamma))\rightarrow(\alpha\rightarrow\beta)\rightarrow(\alpha\rightarrow\gamma)$

A3 : $(\neg\alpha\rightarrow\neg\beta)\rightarrow(\beta\rightarrow\alpha)$

- **Règles d'inférences**

Comme règle d'inférence unique le Modus Ponens

Exemple de théorie pour le logique propositionnelle

Exep1 : démontrer $\vdash \alpha \rightarrow \alpha$

b0 : $(\alpha \rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha)) \rightarrow ((\alpha \rightarrow (\alpha \rightarrow \alpha)) \rightarrow (\alpha \rightarrow \alpha))$ (A2)

b1 : $\alpha \rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha)$ (A1)

b2 : $(\alpha \rightarrow (\alpha \rightarrow \alpha)) \rightarrow (\alpha \rightarrow \alpha)$, (MP b0, b1)

b3 : $\alpha \rightarrow (\alpha \rightarrow \alpha)$, (A1)

b4 : $\alpha \rightarrow \alpha$ (MP b2, b3)

Exep2 : démontrer $F \rightarrow H, F \rightarrow (H \rightarrow G) \vdash F \rightarrow G$

Théorème de déduction :

Si $\alpha_1, \alpha_2, \ldots, \alpha_n \vdash \beta$ alors $\alpha_1, \alpha_2, \ldots, \alpha_{n-1} \vdash \alpha_n \to \beta$

Propriétés de consistance et de complétude :

Dans le calcul propositionnel, il y a deux aspects :

- **Aspect sémantique (théorie des modèles) :**Qui introduit les notions de satisfiabilité, de tautologie et de conséquences logiques.
- **Aspect syntaxique (théorie de démonstration):** On s'intéresse à montrer qu'une formule est un théorie ou qu'une formule déduite à partir d'un ensemble de formules**.**
- Existe-t-il un rapport entres les théorèmes et les tautologies ?
- Existe-t-il un rapport entres la notion de conséquence logique et celle de déduction ?

Théorème de consistance

- Toutes formules démontrable est une tautologie Si $\vdash \beta$ alors $\models \beta$
- Un système d'inférence est **consistant** si tous les théorèmes sont des conséquences logiques des formules des axiomes   (tout ce qui est démontrable est « vrai »).
- Ou de manière équivalence, si on peut produire *G* à partir de l'ensemble de formules *F*, noté $F \vdash G$, alors G est une conséquence de *F* ($F \models G$)

Théorème de complétude

- Si $\models \beta$ alors $\vdash \beta$
- Un système d'inférence est **complet** s'il permet de produire *toutes* les conséquence logiques des axiomes, c'est-à-dire si toute conséquence des axiomes est un théorème (tout ce qui est vrai est démontrable).
- Ou, de manière équivalente, $F \models G$ entraîne

  $F \vdash G$

Le système d'inférence naturel est consistant et complet.

Principe de résolution de Robinson

- Cette règle d'inférence a été inventée par Robinson en 1965 . Elle est surtout utilisée dans les systèmes de preuve automatique (démonstrateurs de théorème, systèmes de programmation logique). C'est l'unique règle du système , donc il n'y a pas de problème de choix de la bonne règle à appliquer. Par contre on peut l'appliquer de multiples manières.
- Cette règle travaille uniquement sur des clauses. Il faut donc commencer par tout mettre en forme FNC $C1 \wedge C2 \wedge \ldots \wedge Cn$ où chaque clause Ci est de la forme $L1 \vee \cdots \vee Lm$. Les Lj sont des littéraux de la forme V ou $\neg V$ où V est une variable.

Principe de résolution de Robinson

La règle elle-même s’énonce comme suit :

- Si on a deux clauses

  C1 = (p v L1 v L2 v ... v Lm),

  C2 = (¬p v M1 v M2v..v Mn)

- on peut déduire le résolvant de C1 et C2 qui est la clause L 1 v L 2 v ... v L m v M1 v M2 v ... v Mn.
- En d’autre termes, si un littéral et son complément apparaissent dans deux clauses on peut déduire une nouvelle clause à partir de celles-ci.

Exemple

. Application du principe de résolution

$C1 = (p \vee q \vee \neg r \vee s)$

$C2 = (q \vee \neg p \vee t)$

Résolution sur p et $\neg p$.

$Rés(C1,C2)= (q \vee \neg r \vee s \vee q \vee t)$,

Exemple2

. On veut prouver

$\{(p\vee t) \rightarrow q , r\rightarrow (t \vee s) , (q \wedge t)\rightarrow u , p , \neg s , r \} \vDash u$

On met les formules sous forme de clauses :

1. $(p \vee t) \rightarrow q \equiv \neg (p \vee t) \vee q \equiv (\neg p \wedge \neg t) \vee q$

   $\equiv (\neg p \vee q) \wedge (\neg t \vee q)$

 ce qui donne deux clauses : $(\neg p \vee q)$ et $(\neg t \vee q)$

2. $r \rightarrow (t \vee s) \equiv \neg r \vee (t \vee s) \equiv \neg r \vee t \vee s$

3. $q \wedge t) \rightarrow u \equiv \neg(q \wedge t) \vee u \equiv (\neg q \vee \neg t) \vee u \equiv \neg q \vee \neg t \vee u$

Exemple2

On applique ensuite le principe de résolution à partir des six clauses obtenues :

1. $\neg p \vee q$
2. $\neg t \vee q$
3. $\neg r \vee t \vee s$
4. $\neg q \vee \neg t \vee u$
5. p
6. $\neg s$
7. r
8. q (rés(1,5) sur p)
9. $t \vee s$ (rés(3,7) sur r)
10. t (rés(6,9) sur s)
11. $\neg t \vee u$ (rés(4,8) sur q)
12. u (rés(10,11) sur t)

Exemple2

On ne peut pas appliquer le principe sur deux variables en même temps. Si on a

$C1 = (p \vee \neg q \vee \neg r \vee s)$

$C2 = (q \vee \neg p \vee t)$

on peut déduire

$\neg q \vee \neg r \vee s \vee q \vee t$ (rés sur p) ou $p \vee \neg r \vee s \vee \neg p \vee t$ (rés sur q)

mais pas $\neg r \vee s \vee t$! ("rés " sur p et q)

L'utilisation pratique du principe de résolution se base sur le théorème suivant :

Théorème

Le résolvant C est satisfiable ssi les clauses parentes C1 et C2 le sont (simultanément)

À partir de ce théorème on peut créer une procédure de résolution pour vérifier si un ensemble *S* de clauses est satisfiable.

- On définit $S0:=S$
- On calcule une séquence $S1$, $S2$, . . . avec la règle

  (a) soient $C1$ et $C2$ deux clauses de Si dont le résolvant est C
  (b) on définit $S_{i+1} := Si \cup \{C\}$
  (c) si au cours du processus on trouve un $C = []$ alors $S0$ était inconsistant
  (d) si $S_{i+1} = Si$ pour tout choix de $C1$ et $C2$, alors $S0$ était satisfiable

Théorème

- Une dérivation de [] à partir de S s'appelle une **réfutation** de *S*.
- La technique de ***preuve par* réfutation** consiste à prouver que *A* est une conséquence logique de *S* en réfutant $S0 = S \cup \{\neg A\}$.

Exemple

Soit $S = \{p, \neg p \vee q, \neg p \vee \neg q \vee r\}$. On veut montrer que $S \models r$.

1. $S0 = \{p, \neg p \vee q, \neg p \vee \neg q \vee r, \neg r\}$
2. $S1 = \{p, \neg p \vee q, \neg p \vee \neg q \vee r, \neg r, \neg p \vee \neg q\}$
3. $S2 = \{p, \neg p \vee q, \neg p \vee \neg q \vee r, \neg r, \neg p \vee \neg q, \neg p\}$
4. $S3 = \{p, \neg p \vee q, \neg p \vee \neg q \vee r, \neg r, \neg p \vee \neg q, \neg p, [\,]\}$

donc $S \models r$