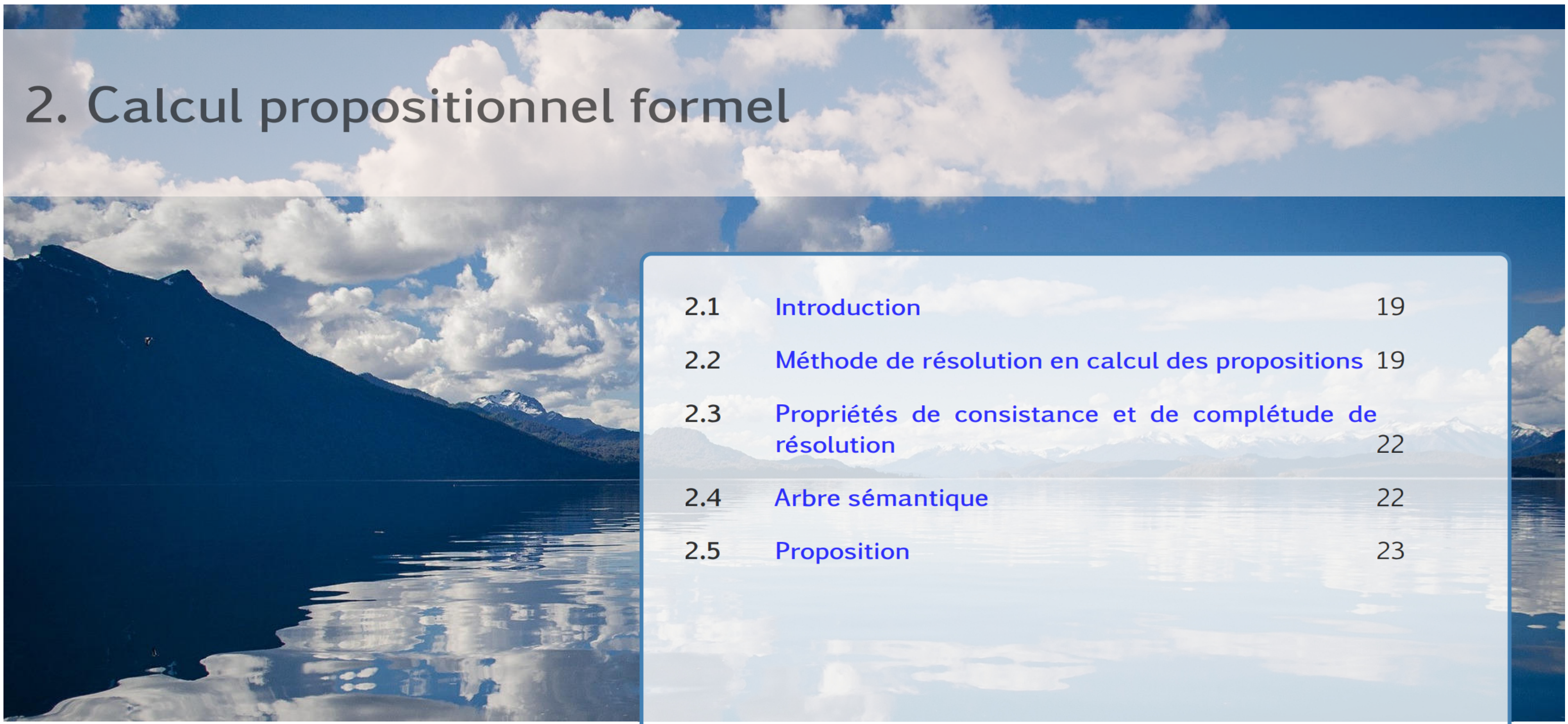


2. Calcul propositionnel formel

2.1 Introduction 19
2.2 Méthode de résolution en calcul des propositions 19
2.3 Propriétés de consistance et de complétude de résolution 22
2.4 Arbre sémantique 22
2.5 Proposition 23

Le chapitre "Calcul propositionnel formel" vise à :

- *Construire un ensemble de clauses à partir d'un ensemble de formules propositionnelles;*
- *Déterminer si un ensemble de clauses est satisfiable à l'aide de la résolution propositionnelle;*
- *Construire un arbre sémantique pour un ensemble de clauses;*
- *Déterminer si un ensemble de clauses est satisfiable à l'aide d'un arbre sémantique;*
- *Définir les concepts de consistance d'un système logique.*

2.1 Introduction

Les tables de vérité (approche sémantique) permettent de décider si une proposition est une tautologie, une contradiction ou bien encore qu'elle est conséquence logique d'une ou d'un ensemble de formules. Si le tableau de vérité s'est montré incontournable, son utilisation peut se révéler fastidieuse lorsque nombre de variables propositionnelles intervenant dans la formule est très grand. La théorie de la démonstration nous offre les moyens de vérifier la validité d'une formule et de déduire des formules à partir d'autres formules sans nous intéresser aux valeurs de vérité. D'où l'appellation d' étude syntaxique souvent donnée à la théorie de la démonstration.

2.2 Méthode de résolution en calcul des propositions

Le principe de la résolution est dû à J. Robinson (1965). Cette méthode de la théorie de la démonstration repose sur la règle d'inférence : la résolution . La résolution procède par réfutation : pour montrer qu'une formule α est valide, nous cherchons à montrer que sa négation $(\neg\alpha)$ est non satisfiable, et pour montrer qu'une formule β est une conséquence logique d'un ensemble de formules Γ, nous cherchons à montrer que l'ensemble $\Gamma \cup \{\neg\beta\}$ est non satisfiable. Cependant, la résolution exige de manipuler des formules sous forme clausale.

Définition 2.1
- Un *littéral* est une formule atomique ou bien la négation d'une formule atomique.
- Deux littéraux sont complémentaires si l'un est la négation de l'autre (P et $\neg P$ sont deux littéraux complémentaires).
- Une clause est une formule de la forme : $L_1 \vee L_2 ... \vee L_m$ où L_i est un littéral.

  Exemple 2.1 $Q \vee R \vee \neg S$ est une clause alors que $(Q \wedge R) \vee \neg S$ n'en est pas une.

  Une *clause unitaire* est une clause contenant un seul littéral.

- Une formule est sous forme clausale si elle est de la forme :$c_1 \wedge c_2 \wedge ... \wedge c_n$ où chaque c_i est une clause.

  Exemple 2.2 les formules $\alpha \equiv (Q \vee R \vee \neg S) \wedge (Q \vee \neg R)$ et $\beta \equiv \neg Q \wedge R$ sont sous forme clausale.

Définition 2.2
- **Clause résolvante**: Étant donné une clause c_i contenant le littéral L , et une clause c_j contenant le littéral $L^{'}$ complémentaire de L, la clause $c_i^{'} \vee c_j^{'}$ où $c_i^{'}$ désigne la clause c_i sans le littéral L , et où $c_j^{'}$ désigne la clause $c_j^{'}$ sans le littéral $L^{'}$ est appelée clause résolvante de c_i et c_j.

  Exemple 2.3 La clause résolvante de $\neg Q \vee R$ et de $P \vee Q$ est la clause $R \vee P$

- **Clause vide**: est la clause résolvante de deux clauses unitaires c_i et c_j formée l'une du littéral L et l'autre du littéral $L^{'}$ complémentaire de L . C'est une clause ne contenant aucun littéral, on la symbolise $\square$.

2.2.1 Proposition

Si c_r est une clause résolvante de deux clauses c_i et c_j alors $c_i, c_j \models c_r$

Exemple 2.4 $\neg Q \vee R, P \vee Q \models R \vee P$

2.2.2 Proposition

Si c_{r1} est une clause résolvante de deux clauses c_i et c_j de l'ensemble S_0 : $c_1, c_2, c_i, ..., c_j, ..., c_n$ alors $c_1, c_2, c_i, ..., c_j, ..., c_n \models c_{r1}$
De la même façon si c_{r2} est une clause résolvante de deux clauses de l'ensemble S_1: $S_0 \cup c_{r1}$, on a :
$c_1, c_2, c_i, ..., c_j, ..., c_n, c_{r1} \models c_{r2}$ où c_{r2} est une clause résolvante de deux clauses de $\{c_1, c_2, c_i, ..., c_n, c_{r1}\}$
$c_1, c_2, c_i, ..., c_j, ..., c_n, c_{r1}, c_{r2} \models c_{r3}$ où c_{r3} est une clause résolvante de deux clauses de $\{c_1, c_2, c_i, ..., c_n, c_{r1}, c_{r2}\}$

.
.

$\vdots$

$c_1, c_2, c_i, ..., c_j, ..., c_n, c_{r1}, c_{r2} ..., c_{ri-1} \vDash c_{ri}$ où c_{ri} est une clause résolvante de deux clauses de $\{c_1, c_2, c_i, ..., c_n, c_{r1}, .., c_{ri-1}\}$
On déduit $c_1, c_2, ..., c_n \vDash c_{ri}$

2.2.3 Proposition

Si l'ensemble $c_1, c_2, c_i, ..., c_n, c_{r1}, c_{r2}, ..., , c_{ri}$ contient une clause unitaire formée du seul littéral P et une clause unitaire formée du seul littéral $\neg P$, l'ensemble dans ce cas est non satisfiable, nous aurons : $c_1, c_2, c_i, ..., c_n, c_{r1}, c_{r2}, .., P, ..., \neg P, c_{ri} \vDash \square$

Nous déduisons qu'un ensemble de clauses est non satisfiable, si par application itérée de la résolution nous produisons la clause $\square$ à partir de cet ensemble.

2.2.4 Proposition

Si c_{r1} est une clause résolvante de deux clauses c_i et c_j de l'ensemble $c_1, c_2, ..., c_n$, alors : $\vDash c_1 \wedge c_2 \wedge ... \wedge c_n \Leftrightarrow c_1 \wedge c_2 \wedge ... \wedge c_n \wedge c_{r1}$

Exemple 2.5 $\vDash (\neg Q \vee R) \wedge (P \vee Q) \Leftrightarrow (\neg Q \vee R) \wedge (P \vee Q) \wedge (R \vee P)$

2.2.5 Proposition

Un ensemble S de clauses est non satisfiable si et seulement si $S \vDash \square$.

2.2.6 Proposition

Si S est un ensemble de clauses comprenant une clause unitaire $c_i = L$ et S_0 l'ensemble S obtenu en retirant de S: la clause c_i, toutes les clauses contenant L ainsi que les littéraux complémentaires de L dans les clauses où ils apparaissent, alors: S est satisfiable si et seulement si S_0 l'est.

Définition 2.3 $S_0 : \{\neg P \vee \neg Q \vee R, \neg P \vee Q \vee \neg R, P \vee R, \neg R\}$
$S^{'} : \{\neg P \vee \neg Q, P\}$

2.2.7 Proposition

Soit S un ensemble de clauses contenant un littéral L dont le complémentaire n'apparait dans aucune clause. On note $S^{'}$ l'ensemble obtenu en supprimant de S toutes les clauses où figure L. Alors S est satisfiable ssi $S^{'}$ l'est. L est appelé ***littéral pur***.

Exemple 2.6
$S : \{\neg P \vee \neg Q \wedge R, \neg P \vee Q, P \vee R, P \vee \neg Q\}$
$S^{'} : \{\neg P \vee Q, P \vee \neg Q\}$

Définition 2.4 Une déduction d'une clause c à partir d'un ensemble S de clauses ($S \vdash c$) est une séquence de clauses$c_1, c_2, ..., c_n$ dans laquelle chaque clause c_i est soit une clause de S , soit une clause obtenue par application de la résolution à deux clauses qui précèdent dans la séquence.

Définition 2.5 un ensemble S de clauses est inconsistant si et seulement si, il existe une déduction de la clause vide à partir de S ($S \vdash \square$).

La résolution procède par réfutation : pour montrer qu'une formule α est valide, nous cherchons à montrer que sa négation $\neg\alpha$ est non statisfiable, et pour montrer qu'une formule β est une conséquence logique d'un ensemble de formules Γ, nous cherchons à montrer que l'ensemble $\Gamma \cup \{\neg\beta\}$ est non satisfiable.

2.3 Propriétés de consistance et de complétude de résolution

Theorem 2.1 Étant donnés $S : \{c_1, c_2, ..., c_m\}$ un ensemble de clauses et c une clause, alors :Si $S \vdash c$ alors $S \vDash c$
Si $c = \square$ (c-à-d que l'ensemble S est inconsistant, alors S est est non satisfiable)

Theorem 2.2 Si un ensemble S de clauses est non satisfiable, il existe alors une déduction de la clause vide ($\square$) à partir de S.

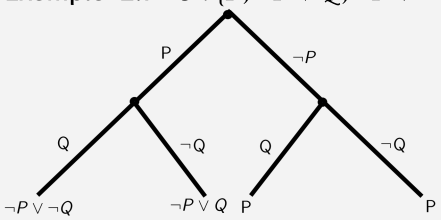
2.4 Arbre sémantique

Soit S un ensemble de clauses, et $VP = \{P_1, P_2, ..., P_k\}$ l'ensemble des variables propositionnelles de S. Un arbre sémantique T pour S est un arbre binaire tel que:

- Chaque arc de T est étiqueté par une variable propositionnelle ou la négation d'une variable propositionnelle;
- Les deux arcs issus d'un même nœud sont étiquetés par des littéraux complémentaires;
- Deux arcs issus d'une même branche ne sont pas étiquetés par des littéraux complémentaires;
- Les feuilles de l'arbre sont étiquetées par les clauses de S.

Un arbre sémantique T est dit complet ssi en plus des conditions exigées de l'arbre sémantique, on a la condition: chaque de branche de l'arbre sémantique contient **pour toute variable propositionnelle** soit le littéral P_i soit le littéral $\neg P_i$, sinon l'arbre est dit partiel.
T est formé de 2^k branches, et chaque branche est formée de k arcs. Un arbre sémantique complet pour un ensembles de clauses, peut être vu comme une représentation graphique du tableau de vérité des formules de cet ensemble. Chaque branche de l'arbre représente une ligne de la table de vérité et qui est une interprétation (I_i de S).

Exemple 2.7 $S : \{P, \neg P \lor Q, \neg P \lor \neg Q\}$



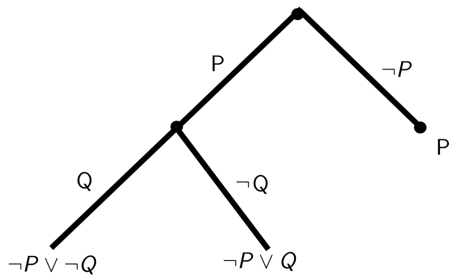
P	Q	$\neg P \lor Q$	$\neg P \lor \neg Q$	P
1	1	1	0	1
1	0	0	1	1
0	1	1	1	0
0	0	1	1	0

Exemple 2.8 (Exemple d'un arbre sémantique complet)

Définition 2.6 Un nœud N de T est appelé nœud d'échec si l'ensemble formé des littéraux associés aux arcs qui précèdent N falsifie au moins une clause de S ($I_n(c) = 0$).

2.4.1 Arbre sémantique clos

Un arbre sémantique T pour un ensemble de clauses S est **clos** , si et seulement si, toutes ses branches se terminent par des nœuds d'échec. Les feuilles de l'arbre sémantique précédent sont toutes des nœuds d'échec. De manière identique, chaque combinaison de P et Q du tableau de vérité associé à l'arbre falsifie au moins une formule de S.



Arbre sémantique clos

2.5 Proposition

Un ensemble S de clauses est non satisfiable, si et seulement si, il existe un arbre sémantique clos pour S .