Logique : le calcul propositionnel

Sophie Pinchinat
sophie.pinchinat@irisa.fr

IRISA, Université de Rennes 1

UE LOG – année 2023-2024

1. Introduction
2. Syntaxe du calcul propositionnel
3. Sémantique du calcul propositionnel
4. Les deux problèmes VALIDE et SAT
5. Formes normales en calcul propositionnel
6. Systèmes de preuve du calcul propositionnel

Syntaxe

Définition

Le langage du calcul propositionnel est formé de :

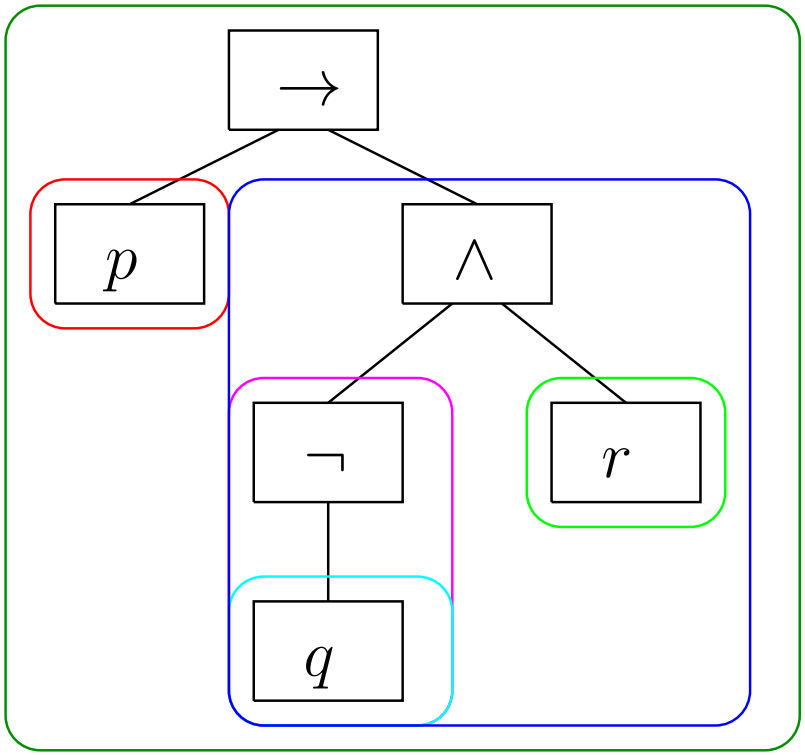
- un ensemble de variables propositionnelles/propositions
  $\text{Prop} = \{p, q, p_1, p_2, \ldots\}$
- les symboles $\top$ et $\bot$;
- connecteurs logiques $\{\neg, \vee, \wedge, \rightarrow\}$
- symboles auxiliaires : $\{(,), \sqcup\}$.

L'ensemble $\mathcal{L}$ des formules du calcul propositionnel est le plus petit ensemble tel que :

- $\top, \bot \in \mathcal{L}$ et $p \in \mathcal{L}$ pour tout $p \in \text{Prop}$,
- si $\varphi \in \mathcal{L}$, alors $\neg\varphi \in \mathcal{L}$,
- si $\varphi, \psi \in \mathcal{L}$, alors $\varphi \vee \psi, \varphi \wedge \psi, \varphi \rightarrow \psi$ appartiennent à $\mathcal{L}$.

Représentation arborescente des formules, notion de sous-formules

Arbre de $p \rightarrow (\neg q \wedge r)$:



Valuation et évaluation d'une formule (on se fixe un Prop)

Définition

Une valuation est une application ν : Prop $\to \{\texttt{faux}, \texttt{vrai}\}$.

Exemple

$[p \mapsto \texttt{faux}, q \mapsto \texttt{vrai}, r \mapsto \texttt{vrai}]$

Définition

Soit ν : Prop $\to \{\texttt{faux}, \texttt{vrai}\}$ une valuation qu'on étend au formules selon :

$$\nu(p) = \nu(p) \qquad \nu(\bot) = \texttt{faux} \qquad \nu(\top) = \texttt{vrai}$$

Valuation et évaluation d'une formule (on se fixe un Prop)

Définition

Une valuation est une application ν : Prop $\to \{\texttt{faux}, \texttt{vrai}\}$.

Exemple

$[p \mapsto \texttt{faux}, q \mapsto \texttt{vrai}, r \mapsto \texttt{vrai}]$

Définition

Soit ν : Prop $\to \{\texttt{faux}, \texttt{vrai}\}$ une valuation qu'on étend au formules selon :

$$\nu(p) = \nu(p) \qquad \nu(\bot) = \texttt{faux} \qquad \nu(\top) = \texttt{vrai}$$

$$\nu(\neg\varphi) = \texttt{vrai} \qquad \text{dès lors que} \quad \nu(\varphi) = \texttt{faux}$$

Valuation et évaluation d'une formule (on se fixe un Prop)

Définition

Une valuation est une application ν : Prop $\to \{\texttt{faux}, \texttt{vrai}\}$.

Exemple

$[p \mapsto \texttt{faux}, q \mapsto \texttt{vrai}, r \mapsto \texttt{vrai}]$

Définition

Soit ν : Prop $\to \{\texttt{faux}, \texttt{vrai}\}$ une valuation qu'on étend au formules selon :

$\nu(p) = \nu(p)$ $\quad$ $\nu(\bot) = \texttt{faux}$ $\quad$ $\nu(\top) = \texttt{vrai}$

$\nu(\neg\varphi) = \texttt{vrai}$ $\quad$ dès lors que $\quad$ $\nu(\varphi) = \texttt{faux}$

$\nu(\varphi \vee \psi) = \texttt{vrai}$ $\quad$ dès lors que $\quad$ $\nu(\varphi) = \texttt{vrai}$ ou $\nu(\psi) = \texttt{vrai}$

Valuation et évaluation d'une formule (on se fixe un Prop)

Définition

Une valuation est une application ν : Prop $\to \{\texttt{faux}, \texttt{vrai}\}$.

Exemple

$[p \mapsto \texttt{faux}, q \mapsto \texttt{vrai}, r \mapsto \texttt{vrai}]$

Définition

Soit ν : Prop $\to \{\texttt{faux}, \texttt{vrai}\}$ une valuation qu'on étend au formules selon :

$\nu(p) = \nu(p)$ $\quad$ $\nu(\bot) = \texttt{faux}$ $\quad$ $\nu(\top) = \texttt{vrai}$

$\nu(\neg\varphi) = \texttt{vrai}$ $\quad$ dès lors que $\quad$ $\nu(\varphi) = \texttt{faux}$

$\nu(\varphi \vee \psi) = \texttt{vrai}$ $\quad$ dès lors que $\quad$ $\nu(\varphi) = \texttt{vrai}$ ou $\nu(\psi) = \texttt{vrai}$

$\nu(\varphi \wedge \psi) = \texttt{vrai}$ $\quad$ dès lors que $\quad$ $\nu(\varphi) = \texttt{vrai}$ et $\nu(\psi) = \texttt{vrai}$

Valuation et évaluation d'une formule (on se fixe un Prop)

Définition

Une valuation est une application ν : Prop $\to \{\texttt{faux}, \texttt{vrai}\}$.

Exemple

$[p \mapsto \texttt{faux}, q \mapsto \texttt{vrai}, r \mapsto \texttt{vrai}]$

Définition

Soit ν : Prop $\to \{\texttt{faux}, \texttt{vrai}\}$ une valuation qu'on étend au formules selon :

$\nu(p) = \nu(p)$ $\quad$ $\nu(\bot) = \texttt{faux}$ $\quad$ $\nu(\top) = \texttt{vrai}$

$\nu(\neg\varphi) = \texttt{vrai}$ $\quad$ dès lors que $\quad$ $\nu(\varphi) = \texttt{faux}$

$\nu(\varphi \vee \psi) = \texttt{vrai}$ $\quad$ dès lors que $\quad$ $\nu(\varphi) = \texttt{vrai}$ ou $\nu(\psi) = \texttt{vrai}$

$\nu(\varphi \wedge \psi) = \texttt{vrai}$ $\quad$ dès lors que $\quad$ $\nu(\varphi) = \texttt{vrai}$ et $\nu(\psi) = \texttt{vrai}$

$\nu(\varphi \to \psi) = \texttt{faux}$ $\quad$ dès lors que $\quad$ $\nu(\varphi) = \texttt{vrai}$ et $\nu(\psi) = \texttt{faux}$

Tables de vérité

Pour évaluer une formule.

φ	$\neg\varphi$
vrai	faux
faux	vrai

φ	ψ	$\varphi \vee \psi$
vrai	vrai	vrai
vrai	faux	vrai
faux	vrai	vrai
faux	faux	faux

φ	ψ	$\varphi \wedge \psi$
vrai	vrai	vrai
vrai	faux	faux
faux	vrai	faux
faux	faux	faux

φ	ψ	$\varphi \rightarrow \psi$
vrai	vrai	vrai
vrai	faux	faux
faux	vrai	vrai
faux	faux	vrai

Ensemble Val_{Prop} des valuations de Prop

Définition

On note Val_{Prop} (ou Val), l'ensemble des valuations de Prop dans $\{\texttt{faux}, \texttt{vrai}\}$.

Exemple

L'ensemble Val peut être vu comme un tableau, où chaque ligne est une valuation.

	p_1	p_2	p_3
ν :	faux	faux	faux
	faux	faux	vrai
	faux	vrai	faux
	faux	vrai	vrai
	vrai	faux	faux
	vrai	faux	vrai
	vrai	vrai	faux
	vrai	vrai	vrai

Ensemble Val_{Prop} des valuations de Prop

Définition

On note Val_{Prop} (ou Val), l'ensemble des valuations de Prop dans $\{\texttt{faux}, \texttt{vrai}\}$.

Exemple

L'ensemble Val peut être vu comme un tableau, où chaque ligne est une valuation.

	p_1	p_2	p_3
ν :	faux	faux	faux
	faux	faux	vrai
	faux	vrai	faux
	faux	vrai	vrai
	vrai	faux	faux
	vrai	faux	vrai
	vrai	vrai	faux
	vrai	vrai	vrai

On note

$\nu \stackrel{\text{def}}{=} [p_1 \mapsto \texttt{faux}, p_2 \mapsto \texttt{faux}, p_3 \mapsto \texttt{faux}]$

cette valuation

Remarque

Si Prop a n éléments, alors Val a 2^n éléments.

Modèles d'une formule

Définition

Un modèle de φ est une valuation ν telle que $\nu(\varphi) = \texttt{vrai}$, noté $\nu \models \varphi$.

On désignera par $mod(\varphi) \subseteq Val$ l'ensemble des modèles de φ.

Exercice

Soit Prop $= \{p, q, r\}$. Quel est l'ensemble $mod((p \vee q) \wedge (p \vee \neg r))$, c-à-d. des valuations qui satisfont $(p \vee q) \wedge (p \vee \neg r)$?

Exercice

Que vaut $mod((p \vee q) \wedge (p \vee \neg r))$ si on prend Prop $= \{p, q, r, s\}$?

Satisfaisabilité, Tautologie

Définition

Une formule φ est satisfaisable si elle admet un modèle (c-à-d. s'il existe une valuation ν telle que $\nu \models \varphi$), sinon elle est insatisfaisable.

Exercice

- Proposer une formule satisfaisable et une formule insatisfaisable.
- Exprimer la propriété "φ est satisfaisable" en terme de $mod(\varphi)$.

Satisfaisabilité, Tautologie

Définition

Une formule φ est satisfaisable si elle admet un modèle (c-à-d. s'il existe une valuation ν telle que $\nu \models \varphi$), sinon elle est insatisfaisable.

Exercice

- Proposer une formule satisfaisable et une formule insatisfaisable.
- Exprimer la propriété "φ est satisfaisable" en terme de $mod(\varphi)$.

Définition

Une formule φ est valide/une tautologie, noté $\models \varphi$, si toute valuation est modèle de φ.

Exercice

- Proposer une formule valide et une formule non valide.
- Exprimer la propriété "$\models \varphi$" en terme de $mod(\varphi)$.

Propriétés des ensembles de modèles

Proposition

1. $mod(\varphi \vee \psi) = mod(\varphi) \cup mod(\psi)$;
2. $mod(\varphi \wedge \psi) = mod(\varphi) \cap mod(\psi)$;
3. $mod(\neg\varphi) = Val \setminus mod(\varphi)$;
4. $mod(\varphi \rightarrow \psi) = (Val \setminus mod(\varphi)) \cup mod(\psi)$;
5. $\models \varphi \rightarrow \psi$ ssi $mod(\varphi) \subseteq mod(\psi)$.

Exercice

Démontrer les Points 1-5 de la proposition ci-dessus.

Notion de formules équivalentes

Définition

Les formules φ et ψ sont équivalentes, noté $\varphi \equiv \psi$, lorsqu'elles ont les mêmes modèles : $mod(\varphi) = mod(\psi)$.

Proposition

$\equiv$ est une relation d'équivalence, c-à-d. :

- elle est réfléxive : pour toute $\varphi \in \mathcal{L}$, $\varphi \equiv \varphi$;
- elle est symétrique : pour toutes $\varphi, \psi \in \mathcal{L}$, $\varphi \equiv \psi$ implique $\psi \equiv \varphi$;
- elle est transitive :
  pour toutes $\varphi, \psi, \theta \in \mathcal{L}$, $\varphi \equiv \psi$ et $\psi \equiv \theta$ impliquent $\varphi \equiv \theta$;

Proposition

$$\varphi \equiv \psi \text{ ssi } \models \varphi \leftrightarrow \psi$$

Exercice

Montrer cette proposition.

Une sélection d'équivalences à justifier (1/2)

Soient $\varphi, \psi, \theta \in \mathcal{L}$,

(Associativité et commutativité de $\vee$)

- $(\varphi \vee \psi) \vee \theta \equiv \varphi \vee (\psi \vee \theta)$
- $\varphi \vee \psi \equiv \psi \vee \varphi$

(Associativité et commutativité de $\wedge$)

- $(\varphi \wedge \psi) \wedge \theta \equiv \varphi \wedge (\psi \wedge \theta)$
- $\varphi \wedge \psi \equiv \psi \wedge \varphi$

(Lois de distributivité entre $\vee$ et $\wedge$)

- $\varphi \wedge (\psi \vee \theta) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \theta)$
- $\varphi \vee (\psi \wedge \theta) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \theta)$

Une sélection d'équivalences à justifier (2/2)

(Négation)

- $\neg\neg\varphi \equiv \varphi$
- $\neg(\varphi \to \psi) \equiv \varphi \wedge \neg\psi$
- $\neg(\varphi \leftrightarrow \psi) \equiv (\varphi \wedge \neg\psi) \vee (\neg\varphi \wedge \psi)$

(Lois de De Morgan (1806-1871))

$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$ et $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$

(D'autres propriétés importantes)

- Contraposée : $(\varphi \to \psi) \equiv (\neg\psi \to \neg\varphi)$
- Exportation : $(\varphi \to (\psi \to \theta)) \equiv ((\varphi \wedge \psi) \to \theta)$
- Tiers exclu : $\models \varphi \vee (\neg\varphi)$ (vraie ou fausse, sans valeur intermédiaire)
- Non-contradiction : $\models \neg(\varphi \wedge \neg\varphi)$

Conséquence logique

Définition

Une formule φ est conséquence logique d'une formule ψ, noté $co\psi \models \varphi$, si $mod(\psi) \subseteq mod(\varphi)$ (c-à-d. tout modèle de ψ est un modèle de φ).

Exercice

Reprendre la signification des notations suivantes :

- $\nu \models \varphi$ (où ν est une valuation et φ une formule)
- $\psi \models \varphi$ (où ψ et φ sont des formules)
- $\models \varphi$ (où φ est une formule)

Conséquences logiques d'un ensemble de formules

Soit $\Gamma \subseteq \mathcal{L}$ un ensemble de formules de $\mathcal{L}$.

Définition

Un modèle de Γ est une valuation ν telle que pour chaque formule $\varphi \in \Gamma$, $\nu \models \varphi$.

Par extension, on note $mod(\Gamma)$ l'ensemble des modèles de Γ.

Définition

Γ est satisfaisable si $mod(\Gamma) \neq \emptyset$, sinon il est insatisfaisable/contradictoire).

Proposition

Si Γ est contradictoire, alors tout Σ tel que $\Gamma \subseteq \Sigma$ l'est aussi.

Exemple

L'ensemble $\{p \vee q, \neg p \vee r\}$ est satisfaisable.
L'ensemble $\{p, \neg p\}$ est contradictoire, et donc tout sur-ensemble de ce dernier, comme par exemple $\{p, \neg p, p \vee q\}$.

Conséquences logiques d'un ensemble de formules

Définition

Une formule φ est une conséquence logique de Γ, noté $\Gamma \models \varphi$, si $mod(\Gamma) \subseteq mod(\varphi)$.

On note $Conseq(\Gamma)$ l'ensemble des conséquences logiques de Γ.
Remarquons que $\Gamma \subseteq Conseq(\Gamma)$.

Exemple

$p \to s \in Conseq(\{(p \to s) \vee q, \neg q\})$,

ce qui s'écrit aussi $\{(p \to s) \vee q, \neg q\} \models p \to s$.

Remarque

Ne pas confondre les notations $\nu \models \varphi$ où ν est une valuation, et $\psi \models \varphi$ où ψ est une formule, et $\Gamma \models \varphi$ où Γ est une ensemble de formules.

Conséquences logiques d'un ensemble de formules

Remarque

Si un ensemble de formules Γ est vu comme un ensemble de contraintes à satisfaire, on peut en retirer celles qui sont conséquences logiques des autres sans changer les exigences que ces contraintes expriment.
Par exemple, on peut retirer de l'ensemble $\{p \wedge q, p \vee q, q\}$ les deux formules $p \vee q$ et q car $\{p \wedge q\} \models p \vee q$ et $\{p \wedge q\} \models q$.

Remarque

Deux ensembles incomparables de formules peuvent avoir le même ensemble de conséquences logique. Par exemple $\{p \wedge q\}$ et $\{p, q\}$.

Proposition

$\Gamma \models \varphi$ ssi $mod(\Gamma) = mod(\Gamma \cup \{\varphi\})$.

Exercice

Montrer la proposition ci-dessus.

Conséquences logiques : Propriétés

Proposition

$\Gamma \models \varphi$ ssi $\Gamma \cup \{\neg\varphi\}$ est insatisfaisable.

Proposition

Pour tous ensembles de formules Γ et Σ,

$$mod(\Gamma \cup \Sigma) = mod(\Gamma) \cap mod(\Sigma).$$

En particulier, si $\Sigma \subseteq \Gamma$ alors $mod(\Gamma) \subseteq mod(\Sigma)$.

Exercice

Trouver Γ et Σ tels que $mod(\Gamma) \subseteq mod(\Sigma)$ mais $\Sigma \not\subseteq \Gamma$.

Autres résultats

Proposition

Si $\Sigma \subseteq \Gamma$ et $\Sigma \models \varphi$ alors $\Gamma \models \varphi$.

Exercice

Montrer cette proposition.

Proposition

$\{\varphi_1, ...\varphi_n\} \models \varphi$ ssi $\models (\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow \varphi$.

Exercice

Montrer cette proposition.

Algorithmes pour le calcul propositionnel

Définition

Le problème VALIDE est le problème qui consiste à déterminer si une formule $\varphi \in \mathcal{L}$ donnée est valide.

Voici un algorithme "brute-force" (on esssaie tout) pour résoudre le problème VALIDE :

Algorithm 1 `bfVALIDE`(φ)

```
1: for all ν ∈ Val do
2:     if Eval(φ, ν) = faux then
3:         return "non"          ▷ on peut même donner ν comme témoin.
4:     end if
5: end for
6: return "oui"
```

- Le temps de calcul de `Eval`(φ, ν) est linéaire en la taille de φ (écrire l'algorithme).

Algorithmes pour le calcul propositionnel

Définition

Le problème VALIDE est le problème qui consiste à déterminer si une formule $\varphi \in \mathcal{L}$ donnée est valide.

Voici un algorithme "brute-force" (on esssaie tout) pour résoudre le problème VALIDE :

Algorithm 2 `bfVALIDE`(φ)

```
1: for all ν ∈ Val do
2:     if Eval(φ, ν) = faux then
3:         return "non"          ▷ on peut même donner ν comme témoin.
4:     end if
5: end for
6: return "oui"
```

- Le temps de calcul de `Eval`(φ, ν) est linéaire en la taille de φ (écrire l'algorithme).
- Le temps de calcul de `bfVALIDE` est exponentielle en la taille de φ dans le pire cas car on peut parcourir l'ensemble de toutes les valuations.

Lien entre validité et satisfaisabilité

Proposition

$\models \varphi$ (c-à-d. φ est valide) ssi $\neg\varphi$ est insatisfaisable.

Preuve $\models \varphi$

Lien entre validité et satisfaisabilité

Proposition

$\models \varphi$ (c-à-d. φ est valide) ssi $\neg\varphi$ est insatisfaisable.

Preuve $\models \varphi$
ssi $mod(\varphi) = Val$

Lien entre validité et satisfaisabilité

Proposition

$\models \varphi$ (c-à-d. φ est valide) ssi $\neg\varphi$ est insatisfaisable.

Preuve $\models \varphi$
ssi $mod(\varphi) = Val$
ssi $Val \setminus mod(\varphi) = \emptyset$

Lien entre validité et satisfaisabilité

Proposition

$\models \varphi$ (c-à-d. φ est valide) ssi $\neg\varphi$ est insatisfaisable.

Preuve $\models \varphi$
ssi $mod(\varphi) = Val$
ssi $Val \setminus mod(\varphi) = \emptyset$
ssi $mod(\neg\varphi) = \emptyset$

Lien entre validité et satisfaisabilité

Proposition

$\models \varphi$ (c-à-d. φ est valide) ssi $\neg\varphi$ est insatisfaisable.

Preuve $\models \varphi$
ssi $mod(\varphi) = Val$
ssi $Val \setminus mod(\varphi) = \emptyset$
ssi $mod(\neg\varphi) = \emptyset$
ssi $\neg\varphi$ n'est pas satisfaisable.

Lien entre validité et satisfaisabilité

Proposition

$\models \varphi$ (c-à-d. φ est valide) ssi $\neg\varphi$ est insatisfaisable.

Preuve $\models \varphi$
ssi $mod(\varphi) = Val$
ssi $Val \setminus mod(\varphi) = \emptyset$
ssi $mod(\neg\varphi) = \emptyset$
ssi $\neg\varphi$ n'est pas satisfaisable.

Lien entre validité et satisfaisabilité

Proposition

$\models \varphi$ (c-à-d. φ est valide) ssi $\neg\varphi$ est insatisfaisable.

Lien entre validité et satisfaisabilité

Proposition

$\models \varphi$ (c-à-d. φ est valide) ssi $\neg\varphi$ est insatisfaisable.

Donc la proposition ci-dessus peut aussi se lire :

φ est satisfaisable ssi $\neg\varphi$ n'est pas valide.

Cela donne un algorithme qui détermine si une formule donnée est satisfaisable :

Définition

Le problème SAT est le problème qui consiste à déterminer si une formule $\varphi \in \mathcal{L}$ donnée en entrée admet un modèle.

Un algorithme pour SAT avec l'entrée φ consiste à invoquer `bfVALIDE`$(\neg\varphi)$ et retourne "oui" ssi `bfVALIDE`$(\neg\varphi)$ retourne "non" et un modèle de φ.
Mais cet algorithme n'est pas efficace.
Il existe des outils pour le problème SAT, voir le terme "SAT solvers" sur le net, voir par exemple `http://www.satcompetition.org/`.
Nous utiliserons TouIST `https://www.irit.fr/touist/` pour nos TDs et expérimentations diverses.

Formes normales

La plupart des algorithmes pour VALIDE ou SAT débutent par une première phase de normalisation de la formule.

Notations

Si $\varphi_1, \varphi_2, \ldots, \varphi_n \in \mathcal{L}$, on utilise les notations :

$$\bigwedge_{i=1}^{n} \varphi_i = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$$

et

$$\bigvee_{i=1}^{n} \varphi_i = \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_n$$

On convient que : $\bigwedge_{i=1}^{0} \varphi_i = \top$ et $\bigvee_{i=1}^{0} \varphi_i = \bot$.

Littéraux, Clauses, Forme clausale

Définition

Un littéral ℓ est soit une proposition, soit la négation d'une proposition, c-à-d. de la forme p ou $\neg p$ (avec $p \in \text{Prop}$).

Définition

Une clause (disjonctive) est une formule de la forme : $C = \bigvee_{i=1}^{n} \ell_i$
où les ℓ_is sont des littéraux.
La clause sans aucun littéral, dite clause vide, est donc $\bot$.

Définition (Forme normale conjonctive)

Une formule est en forme normale conjonctive (FNC) si elle de la forme

$$\bigwedge_{j=1}^{m} C_j$$

où les C_j's sont des clauses non vides.

Exemple

La formule $(\neg p \vee q \vee r) \wedge (\neg q \vee p) \wedge s$ est en FNC, avec 3 clauses.

Existence de forme normale conjonctive

Proposition

Pour toute $\varphi \in \mathcal{L}$, il existe $\varphi' \in \mathcal{L}$ en FNC telle que $\varphi \equiv \varphi'$.

Preuve On propose l'algorithme suivant :

(1) Appliquer autant que possible la substitution

$$\text{de } (\varphi \to \psi) \text{ par } (\neg\varphi \vee \psi)$$

(2) Appliquer autant que possible les substitutions

$$\text{de } (\neg\neg\varphi) \text{ par } \varphi$$
$$\text{de } \neg(\varphi \wedge \psi) \text{ par } (\neg\varphi \vee \neg\psi)$$
$$\text{de } \neg(\varphi \vee \psi) \text{ par } (\neg\varphi \wedge \neg\psi)$$

(3) Appliquer autant que possible les substitutions

$$\text{de } \varphi \vee (\psi_1 \wedge \psi_2) \text{ par } (\varphi \vee \psi_1) \wedge (\varphi \vee \psi_2)$$
$$\text{de } (\psi_1 \wedge \psi_2) \vee \varphi \text{ par } (\psi_1 \vee \varphi) \wedge (\psi_2 \vee \varphi)$$

Exercice

Revoir votre TD2 où l'algorithme est détaillé, ainsi que la justification de sa terminaison et de sa correction.

Mise en forme normale conjonctive

Exemple

Soit $\varphi = (\neg(p \wedge (\neg q \vee (r \vee s)))) \wedge (p \vee q)$.

1. $\varphi \rightsquigarrow (\neg p \vee \neg(\neg q \vee (r \vee s))) \wedge (p \vee q)$
2. $\rightsquigarrow (\neg p \vee (\neg\neg q \wedge \neg(r \vee s))) \wedge (p \vee q)$
3. $\rightsquigarrow (\neg p \vee (q \wedge \neg r \wedge \neg s)) \wedge (p \vee q)$
4. $\rightsquigarrow (\neg p \vee q) \wedge (\neg p \vee \neg r) \wedge (\neg p \vee \neg s) \wedge (p \vee q)$

On en déduit qu'une forme normale conjonctive équivalente à φ est

$$(\neg p \vee q) \wedge (\neg p \vee \neg r) \wedge (\neg p \vee \neg s) \wedge (p \vee q)$$

Remarque

On peut écrire une formule en FNC comme un ensemble de clause, dite forme clausale. Par exemple, $(\neg p \vee q) \wedge (\neg p \vee \neg r) \wedge (\neg p \vee \neg s) \wedge (p \vee q)$ est représentée par $\{\{\neg p, q\}, \{\neg p, \neg r\}, \{\neg p, \neg s\}, \{p, q\}\}$.
On peut aussi utiliser une représentation intermédiaire (aussi appelée forme clausale tant elle est proche) avec un ensemble de clause.
Dans notre exemple, on obtient $\{\neg p \vee q, \neg p \vee \neg r, \neg p \vee \neg s, p \vee q\}$.

Amélioration de l'algorithme de mise en FNC

On peut simplifier une formule en FNC en préservant l'ensemble des modèles :

- Les clauses comportant deux littéraux opposés sont équivalentes à la formule $\top$ (voir la propriétés du tiers-exclu).
  Elles peuvent donc être retirées de l'ensemble, comme par ex. la clause $p \vee q \vee \neg r \vee \neg q$.
- On peut aussi supprimer les répétitions d'un littéral au sein d'une même clause, puisque $\ell \vee \ell \equiv \ell$.
- Si dans un ensemble de clauses on a une clause C' sous formule d'une autre clause C, p. ex. $p \vee q$ est sous-formule de $p \vee q \vee r$, alors clairement $C' \models C$, de sorte que la clause C peut être supprimée de l'ensemble.

Systèmes formels

Définition

Un système formel est la donnée :

- d'un langage (les propositions), et des formules sur ce langage;
- d'axiomes : des formules dont on part (les hypothèses);
- de règles d'inférence permettant de déduire une nouvelle formule (conclusion) à partir d'un ensemble de formules (axiomes ou formules déjà déduites).

Exemple (de règle)

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi}$$

Si on sait que φ est vraie et que ψ est vraie, alors on peut en déduire que $\varphi \wedge \psi$ est vraie.

Exemple (d'autres règles)

$$\frac{\varphi}{\varphi \vee \psi} \qquad \frac{\psi}{\varphi \vee \psi}$$

Un système formel particulier : la résolution

Introduite en 1965 par Robinson, la résolution est un système formel qui utilise des formules en FNC.

Le système de résolution n'a pas d'axiome, et comporte deux règles d'inférence.

Definition (Le système de résolution)

- La règle de coupure

$$\frac{C \vee p \quad C' \vee \neg p}{C \vee C'}$$

$$\frac{p \quad \neg p}{\bot} \text{ (cas particulier)}$$

- La règle de factorisation

$$\frac{C \vee \ell \vee \ell}{C \vee \ell}$$

Exemple de résolution

$$\frac{C \vee p \quad C' \vee \neg p}{C \vee C'} \text{ coupure} \qquad \text{et} \qquad \frac{C \vee \ell \vee \ell}{C \vee \ell} \text{ factorisation}$$

Exemple

Soit $\mathcal{C} = \{(p \vee r \vee s), (r \vee \neg s), \neg r, \neg p\}$.

Voici une dérivation par résolution de cet ensemble de clauses :

$$\dfrac{\dfrac{\dfrac{\dfrac{p \vee r \vee s \quad r \vee \neg s}{p \vee r \vee r}\ \text{coupure}}{p \vee r}\ \text{factorisation} \qquad \neg r}{p}\ \text{coupure} \qquad \neg p}{\bot}\ \text{coupure}$$

Système de résolution

Définition

On note $\varphi \vdash_R \psi$ lorsque qu'une forme clausale équivalente à φ permet, par application des règles de résolution (coupure ou factorisation), de dériver la formule ψ.

Exemple

Soit $\varphi = (\neg p \wedge \neg r \wedge \neg s) \vee (\neg r \wedge s) \vee r \vee p$.

L'ensemble de clauses associé à $\neg\varphi$ est $\mathcal{C} = \{(p \vee r \vee s), (r \vee \neg s), \neg r, \neg p\}$, or :

$$\dfrac{\dfrac{\dfrac{\dfrac{p \vee r \vee s \quad r \vee \neg s}{p \vee r \vee r}\ \text{coupure}}{p \vee r}\ \text{factorisation} \qquad \neg r}{p}\ \text{coupure} \qquad \neg p}{\bot}\ \text{coupure}$$

On a donc $\neg\varphi \vdash_R \bot$.

Définition

On notera $\vdash_R \varphi$ lorsque $\neg\varphi \vdash_R \bot$.

Propriétés du système de résolution

Deux questions fondamentales sont systématiquement posées pour relier les "preuves" d'un système formel S avec la sémantique :

1. Correction : si $\vdash_S \varphi$, alors $\models \varphi$

   c-à-d. ce qu'on déduit est correct.

2. Complétude : si $\models \varphi$, alors $\vdash_S \varphi$

   c-à-d. ce qui est peut être déduit avec le système formel.

Pour le cas où le système formel est le système R de résolution, on a :

Théorème (Correction et Complétude du système de résolution)

- La résolution est correcte : si $\vdash_R \psi$, alors $\models \psi$ (voir plus loin)
- La résolution est complète : si $\models \psi$ alors $\vdash_R \psi$ (admis dans ce cours)

Propriétés du système de résolution

Théorème (Correction et Complétude de la résolution)

1. La résolution est correcte : si $\vdash_R \varphi$, alors $\models \varphi$
2. La résolution est complète : si $\models \varphi$ alors $\vdash_R \varphi$ (admis)

On rappelle que $\vdash_R \varphi$ signifie $\neg\varphi \vdash_R \bot$. Paraphrasons les Points 1. et 2. :

1. Si $\neg\varphi \vdash_R \bot$, alors $\neg\varphi$ n'est pas satisfaisable, et ainsi φ est valide.
2. si φ est valide, alors $\neg\varphi \vdash_R \bot$, c-à-d. en partant d'une FNC équivalente à φ, il est possible de dériver $\bot$ en appliquant les règles de résolution.

Pour montrer le Point 1., on montre un résultat plus général :

Proposition

Si $\varphi_1 \vdash_R \varphi_2$ alors $\varphi_1 \models \varphi_2$.

Exercice

Pourquoi la proposition ci-dessus permet-elle de conclure le Point 1. ?

Preuve de la correction du système R de résolution

Proposition

Si $\varphi_1 \vdash_R \varphi_2$ alors $\varphi_1 \models \varphi_2$.

Comme $\varphi_1 \vdash_R \varphi_2$ signifie qu'on a appliqué une succession des règles du système de résolution, il suffit de vérifier que les deux règles satisfont :

Lemme

- Correction de la règle de coupure : $\{C \vee \ell, C' \vee \neg\ell\} \models C \vee C'$
- Correction de la règle de factorisation : $\{C \vee \ell \vee \ell\} \models C \vee \ell$

Exercice

Montrer le Lemme ci-dessous.

Système de résolution et validité d'une formule

Pour prouver que φ est valide, on part $\neg\varphi$ et on montre que cela conduit à une absurdité, c-à-d. on montre que $\neg\varphi \vdash_R \bot$, ainsi en vertu de la correction de la résolution, on sait que $\neg\varphi \models \bot$, c-à-d. $\text{mod}(\neg\varphi) \subseteq \text{mod}(\bot) = \emptyset$.
Or si $\neg\varphi$ est insatisfaisable, alors φ est valide.

On dit la résolution est une méthode réfutationnelle.

Système de résolution et validité d'une formule

Pour prouver que φ est valide, on part $\neg\varphi$ et on montre que cela conduit à une absurdité, c-à-d. on montre que $\neg\varphi \vdash_R \bot$, ainsi en vertu de la correction de la résolution, on sait que $\neg\varphi \models \bot$, c-à-d. $\quad \text{mod}(\neg\varphi) \subseteq \quad \text{mod}(\bot) = \emptyset$.
Or si $\neg\varphi$ est insatisfaisable, alors φ est valide.

On dit la résolution est une méthode réfutationnelle.

Une méthode pour montrer $\models \varphi$:

1. Mettre $\neg\varphi$ sous forme clausale, c-à-d. calculer des clauses $C_1, \ldots, C_n$ telles que $\neg\varphi \equiv C_1 \wedge \ldots \wedge C_n$.
2. Saturer l'ensemble $\mathcal{C} = \{C_1, \ldots, C_n\}$ en rajoutant les clauses que l'on peut déduire à partir des deux règles (coupure et factorisation) du système formel de résolution.

Système de résolution et validité d'une formule

Pour prouver que φ est valide, on part $\neg\varphi$ et on montre que cela conduit à une absurdité, c-à-d. on montre que $\neg\varphi \vdash_R \bot$, ainsi en vertu de la correction de la résolution, on sait que $\neg\varphi \models \bot$, c-à-d. $\text{mod}(\neg\varphi) \subseteq \text{mod}(\bot) = \emptyset$.
Or si $\neg\varphi$ est insatisfaisable, alors φ est valide.

On dit la résolution est une méthode réfutationnelle.

Une méthode pour montrer $\models \varphi$:

1. Mettre $\neg\varphi$ sous forme clausale, c-à-d. calculer des clauses $C_1, \ldots, C_n$ telles que $\neg\varphi \equiv C_1 \wedge \ldots \wedge C_n$.
2. Saturer l'ensemble $\mathcal{C} = \{C_1, \ldots, C_n\}$ en rajoutant les clauses que l'on peut déduire à partir des deux règles (coupure et factorisation) du système formel de résolution.

On arrête le procédé de saturation dans les cas suivants.

- soit quand on vient d'y ajouter la clause vide $\bot$, et on en alors déduit que la formule $\neg\varphi$ est insatisfaisable, et donc que φ est valide ;
- soit quand l'application des règles ne modifie plus l'ensemble obtenu. Par complétude la méthode de résolution, le fait de ne pas avoir pu dériver $\bot$ indique que la formule $\neg\varphi$ est satisfaisable et donc que φ n'est pas valide.