# Cours 6

## Calcul propositionnel : déduction par coupure

# Système de déduction

Un système formel est constitué

- d'une syntaxe
  - un alphabet $A$
  - une procédé de formation des formules : $\mathcal{F} \subseteq A^*$
- d'une sémantique :

$$\Sigma \models F$$

- d'un système de déduction
  - un ensemble d'axiomes
  - un ensemble fini de règles de déduction

$$\Sigma \vdash F$$

Vocabulaire : le système de déduction

- est dit *correct* si $\Sigma \vdash F$ implique $\Sigma \models F$,
- est dit *complet* si $\Sigma \models F$ implique $\Sigma \vdash F$.

# La déduction par coupure

On se limite à un sous ensemble des formules : *les clauses*.

## Définition

Une clause est une disjonction de littéraux.

Il s'agit d'un sous ensemble suffisamment représentatif :

## Théorème

*Toute formule est sémantiquement équivalente à une conjonction de clauses.*

## Exemple

- $p \vee \neg q \vee r$

# Vocabulaire et notations

Nous noterons les clauses sous la forme d'un ensemble de littéraux en distinguant les variables *négatives* (apparaissant avec une négation) des variables *positives* (apparaissant sans négation).

$$C = (\Gamma, \Delta)$$

$\Gamma$ : variables propositionnelles négatives
$\Delta$ : variables propositionnelles positives

## Exemples

- $p \vee \neg q \vee r$ sera noté $(\{q\}, \{p, r\})$.
- $p \vee \neg q \vee \neg q \vee p$ sera noté $(\{q\}, \{p\})$.

# Vocabulaire et notations

$$C = (\Gamma, \Delta)$$

Cas particuliers :

- $\Delta = \emptyset$ : clause négative
- $\Gamma = \emptyset$ : clause positive
- $\Delta = \Gamma = \emptyset$ : clause vide, notée $\square$

Remarque : une clause $(\{a_1, \ldots, a_n\}, \{b_1, \ldots, b_m\})$ est équivalente à

$$(a_1 \wedge \cdots \wedge a_n) \Rightarrow (b_1 \vee \cdots \vee b_m)$$

# Règle de coupure

## Définition

Soit $C_1 = (\Gamma_1, \Delta_1)$ et $C_2 = (\Gamma_2, \Delta_2)$, et $p \in \Delta_1 \cap \Gamma_2$.
$C = (\Gamma, \Delta)$ se *déduit par coupure sur p* si

$$\Gamma = \Gamma_1 \cup (\Gamma_2 \setminus \{p\}) \text{ et } \Delta = \Delta_2 \cup (\Delta_1 \setminus \{p\})$$

On note :

$$\frac{C_1 \qquad C_2}{C}$$

# Preuve par coupure

## Définition (Preuve par coupure)

Soit $S$ un ensemble de clauses. L'ensemble des clauses $C$ *prouvables par coupure à partir de $S$* (noté $S \vdash C$) est définie inductivement par

- pour toute clause $C$

$$\text{si } C \in S \text{ alors } S \vdash C$$

- pour toutes clauses $C_1$, $C_2$ et $C$

$$\text{si } S \vdash C_1,\ S \vdash C_2 \text{ et } \frac{C_1 \qquad C_2}{C} \text{ alors } S \vdash C$$

Réfutation par coupure de $S$ : $S \vdash \square$

# Exemple

$\square$ est prouvable par coupure à partir de

$$S = \{p \vee \neg q,\ \neg p \vee \neg r,\ q \vee p,\ \neg p \vee q \vee r,\ \neg p \vee \neg q \vee r\}$$

d'après l'arbre de preuve suivant

$$\dfrac{\dfrac{p \vee \neg q \qquad \dfrac{\dfrac{\neg p \vee \neg r \quad \neg p \vee q \vee r}{\neg p \vee q} \qquad q \vee p}{q}}{p} \qquad \dfrac{\dfrac{\neg p \vee q \vee r \quad \neg p \vee \neg q \vee r}{\neg p \vee r} \qquad \neg p \vee \neg r}{\neg p}}{\square}$$

# Correction

## Lemme

*Si* $\dfrac{C_1 \qquad C_2}{C}$ *alors* $\{C_1, C_2\} \models C$.

## Théorème

*Si* $S \vdash C$*, alors* $S \models C$.

## Corollaire

*Si* $S \vdash \square$ *alors* $S$ *n'est pas satisfiable.*

# Complétude

## Lemme

*Soit $S$ un ensemble de clauses non satisfiables qui ne contient pas $\square$. Alors il existe $p$, $C_1$ et $C_2$ tels que $p \in \Gamma_1 \cap \Delta_2$.*

## Définition (Résolvant)

Avec les notations précédentes et $S_p$ le sous-ensemble des clauses de $S$ contenant $p$, on appelle *résolvant* de $S_p$ (noté $Res(S_p)$) l'ensemble des clauses obtenues à partir de deux clauses de $S_p$ par coupure sur $p$.

Remarque : Si $S$ est non satisfiable et ne contient pas $\square$, alors il existe $p$ tel que $Res(S_p) \neq \emptyset$.

# Complétude

**Lemme**

*$S$ satisfiable ssi $(S \setminus S_p) \cup Res(S_p)$ satisfiable.*

**Lemme**

*Si $S$ est fini et $S \models C$, alors $S \vdash C$.*

**Théorème**

*Si $S \models C$, alors $S \vdash C$.*