

# LIF11 Logique - TD3

## Correction

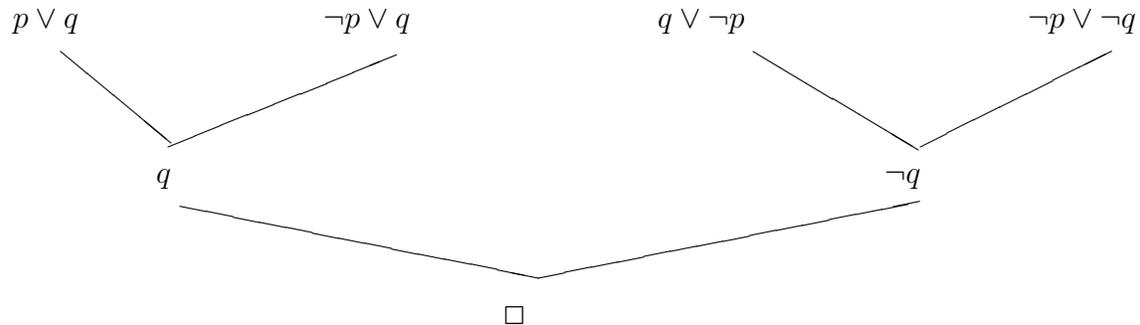
### Résolution

**Exercice 1:**

Montrer par résolution que l'ensemble de clauses suivant est contradictoire :

$$\{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$$

**Correction:**



**Exercice 2:**

Un meurtre a été commis au laboratoire, le corps se trouve dans la salle de conférences. On dispose des informations suivantes :

- La secrétaire déclare qu'elle a vu l'ingénieur dans le couloir qui donne sur la salle de conférences
- Le coup de feu a été tiré dans la salle de conférences, on l'a donc entendu de toutes les pièces voisines
- L'ingénieur affirme n'avoir rien entendu

On souhaite démontrer que si la secrétaire dit vrai, alors l'ingénieur ment. Pour cela on utilise les variables propositionnelles suivantes :

- $p$  est vraie si la secrétaire dit la vérité
- $q$  est vraie si l'ingénieur dit la vérité
- $r$  est vraie si l'ingénieur était dans le couloir qui donne sur la salle de conférences
- $s$  est vraie si l'ingénieur était dans une pièce voisine de la salle de conférences
- $t$  est vraie si l'ingénieur a entendu le coup de feu

1. Traduire les informations précédentes en 4 formules en utilisant les variables ci-dessus. Traduire également la conclusion à démontrer en formule.

**Correction:**

- La secrétaire déclare qu'elle a vu l'ingénieur dans le couloir qui donne sur la salle de conférences :  $p \Rightarrow r$
- Si l'ingénieur était dans le couloir qui donne sur la salle de conférences, alors il était dans une pièce voisine de la salle de conférences :  $r \Rightarrow s$
- Le coup de feu a été tiré dans la salle de conférences, on l'a donc entendu de toutes les pièces voisines :  $s \Rightarrow t$
- L'ingénieur affirme n'avoir rien entendu :  $q \Rightarrow \neg t$

– Si la secrétaire dit la vérité, alors l'ingénieur ment :  $p \Rightarrow \neg q$

2. Regrouper ensuite ces formules en une seule formule  $A$  résumant le problème.

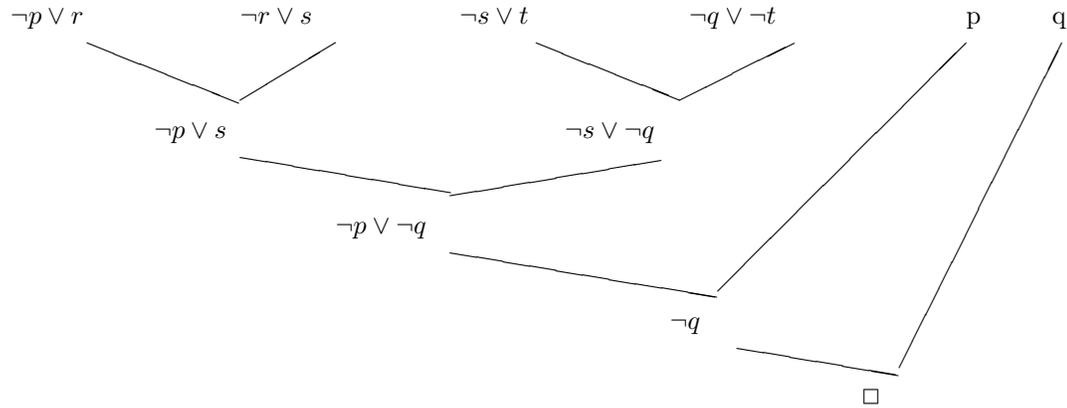
**Correction:**  $A = ((p \Rightarrow r) \wedge (r \Rightarrow s) \wedge (s \Rightarrow t) \wedge (q \Rightarrow \neg t)) \Rightarrow (p \Rightarrow \neg q)$

3. Transformer la négation  $\neg A$  de cette formule en forme conjonctive.

**Correction:**  $\neg A \equiv (\neg p \vee r) \wedge (\neg r \vee s) \wedge (\neg s \vee t) \wedge (\neg q \vee \neg t) \wedge p \wedge q$

4. Montrer, grâce à la méthode de résolution, que  $\neg A$  est insatisfiable.

**Correction:**



5. Que peut-on en déduire à propos de  $A$  et du problème de départ ?

**Correction:** On peut en déduire que  $A$  est valide, donc que si la secrétaire dit la vérité, alors l'ingénieur ment.

6. (facultatif) Utiliser les formules trouvées en 1. pour écrire un séquent correspondant au problème et montrer que ce séquent est correct en utilisant le système  $\mathcal{G}$ .

**Correction:**

$$\begin{array}{c}
 \frac{}{r \Rightarrow s, s \Rightarrow t, q \Rightarrow \neg t, p, q \vdash p} (Ax) \quad \frac{}{s \Rightarrow t, q \Rightarrow \neg t, p, q, r \vdash r} (Ax) \quad \frac{(1)}{s \Rightarrow t, q \Rightarrow \neg t, p, q, r, s \vdash} (\Rightarrow_G)}{\frac{}{r \Rightarrow s, s \Rightarrow t, q \Rightarrow \neg t, p, q, r \vdash} (\Rightarrow_G)}{\frac{\frac{p \Rightarrow r, r \Rightarrow s, s \Rightarrow t, q \Rightarrow \neg t, p, q \vdash} (\neg_D)}{p \Rightarrow r, r \Rightarrow s, s \Rightarrow t, q \Rightarrow \neg t, p \vdash \neg q} (\Rightarrow_D)}{p \Rightarrow r, r \Rightarrow s, s \Rightarrow t, q \Rightarrow \neg t \vdash p \Rightarrow \neg q}} \\
 (1) : \frac{\frac{}{q \Rightarrow \neg t, p, q, r, s \vdash s} (Ax) \quad \frac{\frac{(Ax)}{p, q, r, s, t \vdash q} \quad \frac{}{p, q, r, s, t \vdash t} (Ax)}{p, q, r, s, t \vdash} (\neg_G)}{q \Rightarrow \neg t, p, q, r, s, t \vdash}}{s \Rightarrow t, q \Rightarrow \neg t, p, q, r, s \vdash}
 \end{array}$$

### Exercice 3:

On considère la fonction  $resolv(A, B)$  qui, étant données deux clauses  $A$  et  $B$  donne l'ensemble des clauses qui sont des résolvantes de  $A$  et  $B$ . On considère également la fonction suivante, où  $E$  est un ensemble de clauses :

$$\mathcal{R}(E) = \{C \mid C \in resolv(A, B), A \in E \text{ et } B \in E\}$$

On considère la suite  $(E_i)_{i \in \mathbb{N}}$  telle que  $E_0 = E$  et  $E_{i+1} = E_i \cup \mathcal{R}(E_i)$ .

1. Montrer que pour tout  $m \leq n$ ,  $E_m \subseteq E_n$   
**Correction:** Par définition de  $E_{i+1}$ ,  $E_{i+1} \supseteq E_i$ . On obtient ensuite le résultat par transitivité de  $\subseteq$ .
2. Montrer que la suite admet un point fixe  $\mathcal{R}^\uparrow(E)$ , c'est-à-dire qu'il existe un nombre  $m$  tel que pour tout  $n \geq m$ ,  $E_m = E_n$ . Indice : considérer l'ensemble des clauses possibles que l'on peut construire avec les littéraux qui apparaissent dans  $E$ .

**Correction:** L'opération de calcul de résolvante n'introduit pas de nouveau littéral. Donc toute clause fabriquée lors du calcul de  $\mathcal{R}^\uparrow(E)$  ne contient que des littéraux qui étaient déjà présent dans  $E$ . Si on assimile une clause à l'ensemble des littéraux qu'elle contient, toute clause  $C$  dans un  $E_i$  est un sous-ensemble de  $lit(E)$  où  $lit(E)$  est l'ensemble des littéraux qui apparaissent dans une clause de  $E$ . On en déduit que pour tout  $i$ ,  $|E_i| \leq 2^{|lit(E)|}$ . Comme  $E_i \subseteq E_{i+1}$  on en déduit qu'à partir d'un certain rang  $m$ , les membres de la suite ne grossissent plus, c'est à dire qu'on a atteint le point fixe.

3. Un arbre de résolution partiel pour  $E$  est un arbre binaire dont les noeuds sont des clauses et vérifiant les conditions suivantes :
  - un noeud interne est une résolvante (des racines) de ses fils
  - les feuilles sont des éléments de  $E$

Montrer que pour tout  $i \in \mathbb{N}$ , toute clause  $C \in E_i$  est la racine d'un arbre de résolution partiel pour  $E$ .

**Correction:** On le montre par récurrence sur  $i$ . C'est vrai pour  $E_0$ , chaque élément de  $E = E_0$  étant la racine d'un ARP (arbre de résolution partiel) se réduisant à une feuille. Supposons le résultat vrai pour  $E_i$ . Soit  $C \in E_{i+1}$ . Si  $C \in E_i$  alors c'est la racine d'un ARP pour  $E$ . Sinon, par définition de  $E_{i+1}$ ,  $C \in resolv(A, B)$  pour  $A, B \in E_i$ . Soit  $\mathcal{A}_A$  et  $\mathcal{A}_B$  les ARP pour  $E$  ayant pour racine  $A$  et  $B$  respectivement. Comme  $C$  est une résolvante et  $A$  et  $B$ , l'arbre ayant pour racine  $C$  et  $\mathcal{A}_A$  et  $\mathcal{A}_B$  comme fils de  $C$  est bien un ARP pour  $E$ .

4. Montrer que pour tout arbre de résolution partiel  $\mathcal{A}$  pour  $E$ , la racine de  $\mathcal{A}$  est un élément de  $\mathcal{R}^\uparrow(E)$ . On fera cette démonstration par induction sur la structure de ces arbres, en remarquant pour le cas des feuilles que  $E_0 = E \subseteq \mathcal{R}^\uparrow(E)$ .

**Correction:** Si l'arbre se réduit à une feuille  $C$ , par définition d'ARP,  $C \in E \subseteq \mathcal{R}^\uparrow(E)$ . Sinon l'arbre a une racine  $C$  et les racines des fils sont deux clauses  $C_1$  et  $C_2$ , avec  $C \in resolv(C_1, C_2)$ . Par induction  $C_1$  et  $C_2$  sont des éléments de  $\mathcal{R}^\uparrow(E)$ . Par définition de  $\mathcal{R}$  et  $\mathcal{R}^\uparrow$ ,  $C \in \mathcal{R}(\mathcal{R}^\uparrow(E)) = \mathcal{R}^\uparrow(E)$ .

5. Montrer à l'aide de 3. et 4. que  $E$  admet un arbre de résolution si et seulement si  $\mathcal{R}^\uparrow(E)$  contient la clause vide  $\square$ .

**Correction:** Si  $\square \in \mathcal{R}^\uparrow(E)$ , alors  $\square \in E_m$  pour un certain  $m$ . Par 3. On en déduit que c'est la racine d'un ARP pour  $E$ . Or un ARP dont la racine est la clause vide est un arbre de résolution pour  $E$ .

Si  $E$  admet un arbre de résolution alors par 4. sa racine  $\square$  est dans  $\mathcal{R}^\uparrow(E)$ .

6. Donner un algorithme de test de satisfiabilité basé sur le calcul de  $\mathcal{R}^\uparrow(E)$  et montrer la correction de cet algorithme en utilisant 5.

**Correction:**

**entrée :**  $E$  : ensemble de clauses

**sortie :** vrai si l'ensemble de clauses est satisfiable, faux sinon

$E' \leftarrow E$

$E'' \leftarrow \emptyset$

tant que  $E' \neq E''$  :

$E'' \leftarrow E'$

$E' \leftarrow E' \cup \mathcal{R}(E')$

fin tant que

renvoyer  $\square \notin E'$

Dans l'algorithme ci-dessus, la boucle calcule itérativement les  $E_i$  jusqu'à obtention du point fixe. On a donc  $E' = \mathcal{R}^\uparrow(E)$  à la sortie de la boucle.  $E$  est satisfiable si et seulement si il n'admet pas d'arbre de résolution. D'après 5, c'est le cas si et seulement si  $\square \notin \mathcal{R}^\uparrow(E)$ .