LIF11 Logique - TD3
Correction
CNF et Résolution

Exercice 1:

On appelle *littéral* une formule réduite à une variable p (littéral positif) ou la négation d'une variable $\neg p$ (littéral négatif). Soit $L = \neg p$ un littéral négatif. Alors on assimilera $\neg L$ au littéral positif p.

Une clause est une formule de la forme $L_1 \vee \ldots \vee L_n$ où $L_1, \ldots, L_n$ sont des littéraux. Si $n = 0$, alors par convention la clause est la formule $\bot$. Une formule en forme normale conjonctive (également appelées FNC ou CNF) est une formule de la forme $C_1 \wedge \ldots \wedge C_m$ où $C_1, \ldots, C_m$ sont des clauses. Si $m = 0$, alors par convention la formule est $\top$.

Étant donnée une formule A, il est toujours possible de trouver une CNF A' telle que A est satisfiable si et seulement si A' est satisfiable (on dit alors que A et A' sont équi-satisfiables).

Une telle formule peut être obtenue par la transformation de Tseitin. Cette transformation s'appuie sur la fonction $tseitin(A)$ qui renvoie une paire (L, A'') où L est un littéral et A'' est une CNF. $tseitin(A)$ est inductivement définie comme suit :

- $tseitin(p) = (p, \top)$
- $tseitin(\top) = (q, q)$ avec q une variable fraîche [1].
- $tseitin(\bot) = (q, \neg q)$ avec q une variable fraîche.
- Si $tseitin(A) = (L, A'')$, alors $tseitin(\neg A) = (\neg L, A'')$.
- Si $tseitin(A) = (L_A, A'')$, si $tseitin(B) = (L_B, B'')$ et si q est une variable fraîche, alors :
  - $tseitin(A \vee B) = (q, A'' \wedge B'' \wedge (\neg L_A \vee q) \wedge (\neg L_B \vee q) \wedge (\neg q \vee L_A \vee L_B)\ )$
  - $tseitin(A \wedge B) = (q, A'' \wedge B'' \wedge (\neg L_A \vee \neg L_B \vee q) \wedge (\neg q \vee L_A) \wedge (\neg q \vee L_B)\ )$
  - $tseitin(A \Rightarrow B) = (q, A'' \wedge B'' \wedge (L_A \vee q) \wedge (\neg L_B \vee q) \wedge (\neg q \vee \neg L_A \vee L_B)\ )$
  - $tseitin(A \Leftrightarrow B) = (q, A'' \wedge B'' \wedge (\neg q \vee \neg L_A \vee L_B) \wedge (\neg q \vee L_A \vee \neg L_B) \wedge (q \vee L_A \vee L_B) \wedge (q \vee \neg L_A \vee \neg L_B)\ )$

Si $(L_A, A'') = tseitin(A)$, alors $A' = A'' \wedge L_A$ est satisfiable si et seulement si A est satisfiable.

Utiliser la transformation de Tseitin pour obtenir des CNF équi-satisfiables à chacune des formules suivantes :

- $\neg p$
  Correction: $tseitin(\neg p)$:
  - $tseitin(p) = (p, \top)$

  $\rightsquigarrow tseitin(\neg p) = (\neg p, \top)$

  Le résultat de la transformation est $\neg p \wedge \top$, i.e. $\neg p$. Remarque : dans la suite on utilise systématiquement l'équivalence remarquable $A \wedge \top \equiv A$.
- $p \wedge r$
  Correction: $tseitin(p \wedge r)$:
  - $tseitin(p) = (p, \top)$
  - $tseitin(r) = (r, \top)$

  $\rightsquigarrow tseitin(p \wedge r) = (\ q_1\ ,\ (\neg p \vee \neg r \vee q_1) \wedge (\neg q_1 \vee p) \wedge (\neg q_1 \vee r)\ )$

  Le résultat de la transformation est $q_1 \wedge (\neg p \vee \neg r \vee q_1) \wedge (\neg q_1 \vee p) \wedge (\neg q_1 \vee r)$.
- $p \Leftrightarrow (p \wedge r)$
  Correction: $tseitin(p \Leftrightarrow (p \wedge r))$:
  - $tseitin(p) = (p, \top)$
  - $tseitin(p \wedge r)$:
    - $tseitin(p) = (p, \top)$

1. c'est à dire une nouvelle variable, jamais rencontrée jusqu'ici. Comme l'ensemble des variables est infini, on peut toujours trouver une variable fraîche.

    - $tseitin(r) = (r, \top)$

    $\rightsquigarrow tseitin(p \wedge r) = (\ q_1\ ,\ (\neg p \vee \neg r \vee q_1) \wedge (\neg q_1 \vee p) \wedge (\neg q_1 \vee r)\ )$

  $\rightsquigarrow tseitin(p \Leftrightarrow (p \wedge r)) = (\ q_2\ ,\ (\neg p \vee \neg r \vee q_1) \wedge (\neg q_1 \vee p) \wedge (\neg q_1 \vee r) \wedge (\neg q_2 \vee \neg p \vee q_1) \wedge (\neg q_2 \vee p \vee \neg q_1) \wedge (q_2 \vee p \vee q_1) \wedge (q_2 \vee \neg p \vee \neg q_1)\ )$

  Le résultat de la transformation est $q_2 \wedge (\neg p \vee \neg r \vee q_1) \wedge (\neg q_1 \vee p) \wedge (\neg q_1 \vee r) \wedge (\neg q_2 \vee \neg p \vee q_1) \wedge (\neg q_2 \vee p \vee \neg q_1) \wedge (q_2 \vee p \vee q_1) \wedge (q_2 \vee \neg p \vee \neg q_1)$

- $(p \wedge r) \vee (\neg p \vee \neg r)$

  Correction: $tseitin((p \wedge r) \vee (\neg p \vee \neg r))$:
  - $tseitin(p \wedge r)$:
    - $tseitin(p) = (p, \top)$
    - $tseitin(r) = (r, \top)$

    $\rightsquigarrow tseitin(p \wedge r) = (\ q_1\ ,\ (\neg p \vee \neg r \vee q_1) \wedge (\neg q_1 \vee p) \wedge (\neg q_1 \vee r)\ )$
  - $tseitin(\neg p \vee \neg r)$:
    - $tseitin(\neg p))$:
      - $tseitin(p) = (p, \top)$

      $\rightsquigarrow tseitin(\neg p) = (\neg p, \top)$
    - $tseitin(\neg r))$:
      - $tseitin(r) = (r, \top)$

      $\rightsquigarrow tseitin(\neg r) = (\neg r, \top)$

    $\rightsquigarrow tseitin(\neg p \vee \neg r) = (q_2\ ,\ (p \vee q_2) \wedge (r \vee q_2) \wedge (\neg q_2 \vee \neg p \vee \neg r)\ )$

  $\rightsquigarrow tseitin((p \wedge r) \vee (\neg p \vee \neg r)) = (q_3\ ,\ (\neg p \vee \neg r \vee q_1) \wedge (\neg q_1 \vee p) \wedge (\neg q_1 \vee r) \wedge (p \vee q_2) \wedge (r \vee q_2) \wedge (\neg q_2 \vee \neg p \vee \neg r) \wedge (\neg q_1 \vee q_3) \wedge (\neg q_2 \vee q_3) \wedge (\neg q_3 \vee q_1 \vee q_2)\ )$

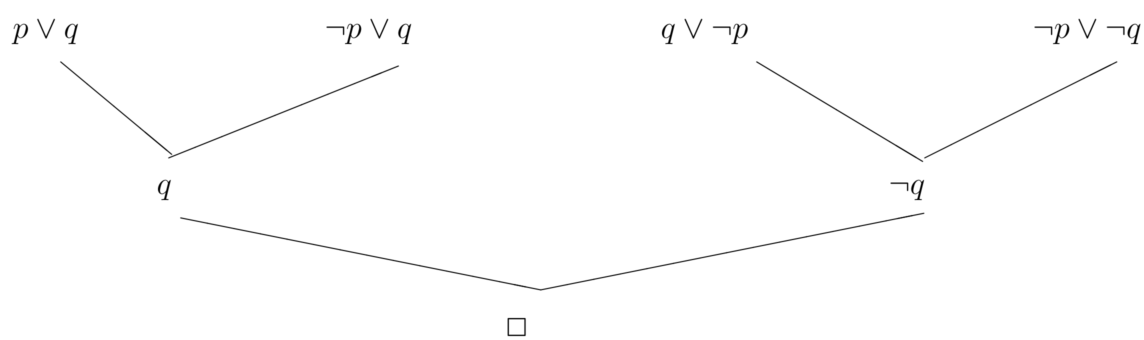
  Le résultat de la transformation est $q_3 \wedge (\neg p \vee \neg r \vee q_1) \wedge (\neg q_1 \vee p) \wedge (\neg q_1 \vee r) \wedge (p \vee q_2) \wedge (r \vee q_2) \wedge (\neg q_2 \vee \neg p \vee \neg r) \wedge (\neg q_1 \vee q_3) \wedge (\neg q_2 \vee q_3) \wedge (\neg q_3 \vee q_1 \vee q_2)$.

Exercice 2:

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 3:

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 démonter 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.

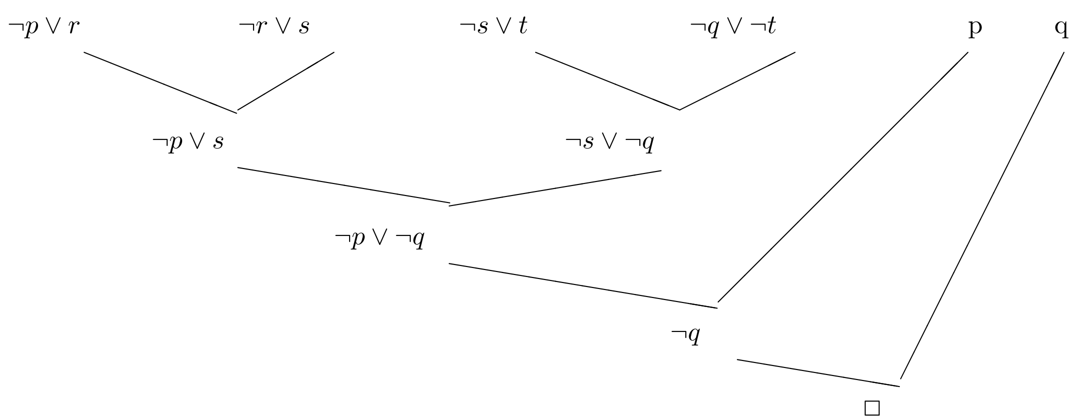
   Correction: $A = ((p \Rightarrow r) \wedge (r \Rightarrow s) \wedge (s \Rightarrow t) \wedge (q \Rightarrow \neg t)) \Rightarrow (p \Rightarrow \neg q)$

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 trouvée en 1. pour écrire un séquent correspondant au problème et montrer que ce séquent est correct en utilisant le système $\mathcal{G}$.

   Correction:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{r \Rightarrow s, s \Rightarrow t, q \Rightarrow \neg t, p, q \vdash p}\,(Axiome) \qquad \dfrac{\dfrac{}{s \Rightarrow t, q \Rightarrow \neg t, p, q, r \vdash r}\,(Axiome) \qquad \dfrac{(1)}{s \Rightarrow t, q \Rightarrow \neg t, p, q, r, s \vdash}}{r \Rightarrow s, s \Rightarrow t, q \Rightarrow \neg t, p, q, r \vdash}\,(\Rightarrow_G)}{p \Rightarrow r, r \Rightarrow s, s \Rightarrow t, q \Rightarrow \neg t, p, q \vdash}\,(\Rightarrow_G)}{p \Rightarrow r, r \Rightarrow s, s \Rightarrow t, q \Rightarrow \neg t, p \vdash \neg q}\,(\neg_D)}{p \Rightarrow r, r \Rightarrow s, s \Rightarrow t, q \Rightarrow \neg t \vdash p \Rightarrow \neg q}\,(\Rightarrow_D)$$

$$(1):\quad \dfrac{\dfrac{}{q \Rightarrow \neg t, p, q, r, s \vdash s}(\mathit{Axiome}) \qquad \dfrac{\dfrac{(\mathit{Axiome})}{p, q, r, s, t \vdash q} \qquad \dfrac{\dfrac{}{p, q, r, s, t \vdash t}(\mathit{Axiome})}{p, q, r, s, t, \neg t \vdash}(\neg_G)}{q \Rightarrow \neg t, p, q, r, s, t \vdash}}{s \Rightarrow t, q \Rightarrow \neg t, p, q, r, s \vdash}$$

Exercice 4:

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 deE_{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 a 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 $\square \notin \mathcal{R}^{\uparrow}(E)$.