

TD déduction automatique en logique propositionnelle classique

David Delahaye

Faculté des Sciences
David.Delahaye@lirmm.fr

Master Informatique M2 2021-2022

Exercice (mise en forme clausale)

Nier et mettre en forme clausale les propositions suivantes

- ❶ $A \Rightarrow B \Rightarrow A$
- ❷ $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C$
- ❸ $A \wedge B \Rightarrow B$
- ❹ $B \Rightarrow A \vee B$
- ❺ $(A \vee B) \Rightarrow (A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow C$
- ❻ $A \Rightarrow \perp \Rightarrow \neg A$
- ❼ $\perp \Rightarrow A$

Exercice

Nier et mettre en forme clausale les propositions suivantes

- ❶ $\neg(A \Rightarrow B \Rightarrow A) \rightarrow \neg(\neg A \vee \neg B \vee A) \rightarrow \neg\neg A \wedge \neg\neg B \wedge \neg A \rightarrow A \wedge B \wedge \neg A$
- ❷ $\neg((A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C) \rightarrow (\neg A \vee \neg B \vee C) \wedge (\neg A \vee B) \wedge A \wedge \neg C$
- ❸ $\neg(A \wedge B \Rightarrow B) \rightarrow A \wedge B \wedge \neg B$
- ❹ $\neg(B \Rightarrow A \vee B) \rightarrow B \wedge \neg(A \vee B) \rightarrow B \wedge \neg A \wedge \neg B$
- ❺ $\neg((A \vee B) \Rightarrow (A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow C) \rightarrow (A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee C) \wedge \neg C$
- ❻ $\neg(A \Rightarrow \perp \Rightarrow \neg A) \rightarrow A \wedge \perp \wedge A$
- ❼ $\neg(\perp \Rightarrow A) \rightarrow \perp \wedge \neg A$

Exercice (DPLL)

Appliquer DPLL sur les propositions suivantes

- ❶ $A \Rightarrow B \Rightarrow A$
- ❷ $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C$
- ❸ $A \wedge B \Rightarrow B$
- ❹ $B \Rightarrow A \vee B$
- ❺ $(A \vee B) \Rightarrow (A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow C$
- ❻ $A \Rightarrow \perp \Rightarrow \neg A$
- ❼ $\perp \Rightarrow A$

Exercice

Appliquer DPLL sur les propositions suivantes

① $A \Rightarrow B \Rightarrow A$:

Forme clausale : $S = \{A, B, \neg A\}$

Même que précédemment, même résultat.

Exercice (résolution)

Appliquer la résolution sur les propositions suivantes

- ❶ $A \Rightarrow B \Rightarrow A$
- ❷ $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C$
- ❸ $A \wedge B \Rightarrow B$
- ❹ $B \Rightarrow A \vee B$
- ❺ $(A \vee B) \Rightarrow (A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow C$
- ❻ $A \Rightarrow \perp \Rightarrow \neg A$
- ❼ $\perp \Rightarrow A$

Exercice (tableaux)

Appliquer la méthode des tableaux sur les propositions suivantes

- ❶ $A \Rightarrow B \Rightarrow A$
- ❷ $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C$
- ❸ $A \wedge B \Rightarrow B$
- ❹ $B \Rightarrow A \vee B$
- ❺ $(A \vee B) \Rightarrow (A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow C$
- ❻ $A \Rightarrow \perp \Rightarrow \neg A$
- ❼ $\perp \Rightarrow A$

TP (à faire chez soi)

Un club très sélectif

- Un club privé a les règles suivantes :
 - ① Tout membre non écossais porte des chaussettes rouges ;
 - ② Tout membre porte un kilt ou ne porte pas de chaussettes rouges ;
 - ③ Les membres mariés ne sortent pas le dimanche ;
 - ④ Un membre sort le dimanche si et seulement s'il est écossais ;
 - ⑤ Tout membre qui porte un kilt est écossais et marié ;
 - ⑥ Tout membre écossais porte un kilt.
- Démontrer qu'en réalité, personne ne peut être accepté !
- Pour ce faire, on encodera le problème comme un problème de logique propositionnelle et on utilisera MiniSat pour le résoudre.
- MiniSat est utilisable en ligne (voir Moodle).
- MiniSat prend des formules sous forme CNF uniquement.
- Le format d'entrée est le format DIMACS (voir Moodle).

TP (à faire chez soi)

Une princesse ou un tigre ? (R. Smullyan)

- Un prisonnier doit choisir entre deux cellules dont l'une cache une princesse et l'autre un tigre.
- S'il choisit la princesse, il doit l'épouser, mais s'il tombe sur le tigre, il est dévoré.
- Toutes les combinaisons sont possibles : il peut y avoir deux tigres, deux princesses, ou un tigre et une princesse.
- La cellule ne peut pas être vide, et il ne peut pas y avoir deux tigres, ou deux princesses, ou un tigre et une princesse dans une même cellule.

TP (à faire chez soi)

Une princesse ou un tigre ? (R. Smullyan)

- Il y a plusieurs épreuves, on va s'intéresser à la première épreuve.
- Il y a deux affiches collées sur les portes :
 - ❶ Il y a une princesse dans cette cellule et un tigre dans l'autre.
 - ❷ Il y a une princesse dans une cellule et il y a un tigre dans une cellule.
- Le roi dit (et il dit toujours la vérité) : « Une des affiches dit la vérité, et l'autre ment. ».
- Quelle cellule doit choisir le prisonnier ?

TP (à faire chez soi)

Une princesse ou un tigre ? (R. Smullyan)

- Modéliser ce problème et sa solution en utilisant Coq, et démontrer (en faisant une preuve) que la solution proposée est correcte.
- Modéliser ce problème et sa solution, puis le donner à Limboole pour qu'il vérifie que la solution proposée est correcte.
- Limboole doit être téléchargé (voir Moodle).
- Limboole prend des formules quelconques (pas forcément sous forme CNF) en entrée (voir le format dans le README).
- Limboole utilise PicoSAT par défaut comme solveur SAT.

TP (à faire chez soi)

Implantation de DPLL

- Implanter DPLL dans le langage de votre choix ;
- Plusieurs étapes dans cette implantation :
 - ▶ Type de données pour les propositions (pas de « parser ») ;
 - ▶ Fonction de clausification ;
 - ▶ Fonction de substitution d'une variable par une valeur (\top/\perp) ;
 - ▶ Fonction pour les tautologies ;
 - ▶ Fonction pour la résolution unitaire ;
 - ▶ Fonction pour les clauses pures ;
 - ▶ Algorithme lui-même.