

Équipe kHORN

1. Créer un dépôt git et m'inviter dessus.
2. Lire attentivement les définitions ci-dessous.

Définition 1. Une clause de Horn est une clause avec au plus un littéral positif, c'est-à-dire de la forme $\neg p_1 \vee \dots \vee \neg p_n \vee q$, où p_i sont des variables, et q est soit une variable, soit \perp .

Une formule est en forme normale de Horn (Horn-NF) si c'est une conjonction de clauses de Horn.

Les clauses ayant au moins \perp comme littéral sont dites *clauses de but*. Les autres clauses sont appelées *clauses d'implication*.

Une clause d'implication $\neg p_1 \vee \dots \vee \neg p_n \vee q$ peut être vue comme la formule $p_1 \wedge \dots \wedge p_n \implies q$. Ainsi, une clause de but peut être vue comme la formule $\top \implies q$.

3. Écrire le pseudo-code d'un algorithme polynomial testant la satisfiabilité d'une formule en forme Horn-NF.
4. Écrire le pseudo-code d'un algorithme polynomial cherchant le nombre minimal de variables à mettre à \top pour satisfaire toutes les clauses.
5. Implémenter l'algorithme proposé dans les questions précédentes dans un solveur. Ce solveur prendra en entrée un fichier écrit dans le format DIMACS, où toutes les clauses sont des clauses de Horn. La sortie du solveur est SAT et un modèle si la formule est satisfiable, sinon la sortie du solveur est UNSAT et la trace utilisée pour générer la clause conflictuelle.