

TP n°3

Mise en forme Normale Conjonctive

DIMACS DIMACS est un format standard de description de formule sous forme normale conjonctive. Voici un exemple de fichier au format DIMACS

```
p cnf 3 2
1 -3 0
2 3 -1 0
```

la premiere ligne indique, apres les mots clef « p cnf » le nombre de variables intervenant dans la formule (ici 3), ainsi que le nombre de clauses disjonctives (ici 2). Chaque ligne suivante represente une clause disjonctive : les variables apparaissant dans la clause sont indiquees, avec un signe - si elles apparaissent sous forme de negation. Chaque ligne se termine par un 0. Le fichier ci-dessus correspond donc a la formule suivante :

$$(x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee \neg x_1)$$

Introduction Dans ce TP on se propose de realiser des methodes qui permettront de mettre une formule en forme normale conjonctive. On voudra ensuite pouvoir engendrer un fichier en format DIMACS.

Nous allons proceder en plusieurs etapes. Il s'agira d'abord de construire des methodes qui mettent les objets de type `Formule` en forme normale conjonctive. Cela fait, il s'agira de programmer une methode qui prendra en argument des objets de type `Formule` et qui genere des fichiers texte en format DIMACS. Ces methodes seront ecrites dans la classe `Formule`.

Mise en forme normale d'un objet de type `Formule`

Exercice 1 Dans la classe `Formule`, programmez une methode `boolean isNnf()` qui dit si une formule est en forme normale de negation.

Exercice 2 Programmez une methode `Formule appliqueReglesNnf()` qui retourne une formule equivalente apres lui avoir applique une ou plusieurs des regles de mise en forme normale de negation du cours.

Exercice 3 Programmez une methode `Formule miseEnNnf()` qui retourne une formule equivalente apres lui avoir applique tant que possible les regles de mise en forme normale de negation.

Exercice 4 Programmez une methode `boolean isClauseDisjonctive()` qui dit si la formule est une clause disjonctive. Programmez une methode `boolean isCnf()` qui dit si la formule est en forme normale de conjonction.

Exercice 5 Programmez une methode `Formule appliqueReglesCnf()` qui suppose que la formule est deja en forme normale de negation, et qui retourne une formule equivalente apres lui avoir applique une ou plusieurs des regles 4.6, 4.7, 4.15 et 4.16 de mise en forme normale de conjonction.

Exercice 6 Programmez une methode `Formule miseEnCnf()` qui suppose que la formule est deja en forme normale de negation, et qui retourne une formule equivalente apres lui avoir applique tant que possible les regles de mise en forme normale de conjonction.

Fonctions annexes

Exercice 7 Programmez `Formule[] unionTableau (Formule[] t, Formule[] s)` qui va prendre en argument deux tableaux de formules et retourne l'union de ces deux tableaux.

Exercice 8 Programmez une methode `Formule[] tableauClause()` qui, quand la formule est en forme normale conjonctive, retourne un tableaux des clauses disjonctive qui y apparaissent.

Exercice 9 Programmez une methode `Formule[] tableauLitteraux()` qui, quand la formule est une clause disjonctive, retourne le tableaux des litteraux qui apparaissent dans cette clause.

Exercice 10 Programmez une methode `String ligneCnf ()` qui, quand la formule est une clause disjonctive, retourne la ligne en format DIMACS qui contient les litteraux de la formule et se termine par 0.

Exercice 11 Programmez une methode `String toDIMACS ()` qui utilise l'ensemble des methodes que vous avez deja programmees. Il s'agira de construire une cha ne qui correspond aux format DIMACS pour une forme normale conjonctive de la formule.

Pour finalement ecrire la cha ne dans un fichier, on utilisera la classe `Ecrivain` fournie. Pour ecrire sur un fichier, il faut creer un objet de la classe `Ecrivain`, qu'on peut voir comme un magnetophone. Le fichier est au debut vide (si le fichier existait deja il est e cr ). Pour des raisons d'efficacite, le magnetophone garde le texte a ecrire dans un tampon. A un instant donne, la t te d'ecriture se trouve apres la derniere ligne du tampon. Un appel a la methode `writeln(String s)` ecrivit sur le tampon la cha ne `s` et passe a la ligne suivante. Pour ecrire le contenu du tampon dans le fichier, on appelle la methode `flush()`.

Les plus avertis pourront egalement utiliser la classe `FileWriter`

Exercice 12 Programmez une methode `main` qui cree un objet de type `Formule` et ecrivit un fichier en format DIMACS qui represente une forme normale de conjonction de la formule.