

USTHB

Faculté d'Electronique et d'Informatique

Département d'Informatique

Master 1 Systèmes Informatiques Intelligents

Représentation des Connaissances et Raisonnement 1

Année Universitaire : 2020-2021

| |
|--|
| <p style="text-align: center;">TP N° 1 : Inférence logique basée sur un solveur SAT</p> |
|--|

Etape 1:

- Créer un répertoire UBCSAT par exemple,
- Copier les fichiers: ubcsat (qui représente le solveur) et les deux fichiers CNF.

Etape 2:

Afin d'exécuter le solveur SAT, il faut activer l'invite commande, en sélectionnant la touche windows, Accessoires, Invite de commande.

L'exécution du solveur se fait comme suit:

C:\UBCSAT> ubcsat -alg saps -i test.cnf -solve

La base de connaissances doit être sous la forme CNF:

Exemple 1 :

Soit la base test1.cnf suivante :

| | | | | |
|----|-----|----|---|---|
| p | cnf | 5 | 9 | |
| 2 | -3 | 0 | | $a \vee \neg c$ |
| -3 | 0 | | | $\neg c$ |
| 1 | -2 | -3 | 4 | $a \vee \neg b \vee \neg c \vee d$ |
| -1 | -4 | 0 | | $\neg a \vee \neg d$ |
| -1 | -2 | 3 | 5 | $\neg a \vee \neg b \vee \neg c \vee e$ |
| 2 | -5 | 0 | | $b \vee \neg e$ |
| -3 | 4 | -5 | 0 | $\neg c \vee d \vee \neg e$ |
| 1 | 2 | 5 | 0 | $a \vee b \vee e$ |
| -3 | 5 | 0 | | $\neg c \vee e$ |

La première ligne est définie par: p cnf nombre_variables nombre_clauses

Le nombre de clauses doit être exact.

L'exemple représente une base ayant 5 variables et 11 clauses. Chaque clause se termine par 0. -1 représente l'instance $\neg a$

5 représente l'instance e.

La base test1.cnf est satisfiable. Le solveur fournit diverses modèles : $\neg a \vee b \vee \neg c \vee d \vee e$; $\neg a \vee b \vee \neg c \vee d \vee \neg e$,...

Exemple 2 :

Soit la base test2.cnf suivante :

```

P cnf    5 11
2  -3  0
-3  0
1 -2 -3 4 0
-1 -4 0
2 -4 0
1 3 0
-1 -2 3 5 0
2 -5 0
-3 4 -5 0
1 2 5 0
3 5 0
-5 0

```

La base test2.cnf n'est pas satisfiable.

Etape 3:

- Traduire la base de connaissances relative aux connaissances zoologiques (céphalopodes) sous forme CNF, puis tester la satisfiabilité de cette base.
Remarque : Pour la mise sous forme CNF, il faudrait transformer l'implication en une disjonction : $(a \supset b) \equiv (\neg a \vee b)$.
- Télécharger des fichiers Benchmarks sous forme CNF afin de tester leur satisfiabilité en utilisant le solveur ubcsat.

Etape 4:

Ecrire un algorithme pour simuler l'inférence d'une base de connaissances.

Soit BC une base de connaissances et soit ϕ une formule. Pour tester si BC infère ϕ , on utilisera le raisonnement par l'absurde. Ce qui revient à tester si $BC \cup \{\neg \phi\}$ infère \perp .

Algorithme de raisonnement par l'absurde

Input :

BC sous forme CNF

Un littéral ϕ

début

$(BC \vdash \phi) \equiv BC \cup \neg \phi \vdash \perp$

Insérer le littéral dans la base

Appel SAT($BC \cup \neg \phi \vdash \perp$)

si $((BC \cup \neg \phi)$ est non satisfiable)

alors

BC $\vdash \phi$

sinon

BC non $\vdash \phi$

finsi

fin

Liens utiles :

Forme Normale Conjonctive

<http://people.sc.fsu.edu/~jburkardt/data/cnf/cnf.html>

Solveur minisat

<https://www.dwheeler.com/essays/minisat-user-guide.html>

Solveur ubcsat

<http://ubcsat.dtopkins.com/downloads>

Fichiers Benchmarks

<https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html>