

1 Introduction et méthode

Dans ce rapport, nous allons rapporter les performances de différentes versions des solveurs *SAT4J*, *MiniSAT* et *CryptoMiniSAT*. Pour ce faire, nous utiliserons la base de donnée de formules booléennes trouvable à cet url : <https://github.com/diverse-project/samplingfm/tree/master/Benchmarks>. Plus précisément, nous utiliserons uniquement les fichiers .cnf qui sont dans le répertoire "Benchmarks" de cette base de données.

Les solveurs testés sont :

- SAT4J - version 2.3.1 - API java - paramètres par défaut
- SAT4J - version 2.0.0 - JAR exécutable - paramètres par défaut
- SAT4J - version 2.3.1 - JAR exécutable - paramètres par défaut
- _____
- MiniSAT - version 1.14.0 - binaire - paramètres par défaut
- MiniSAT - version 2.2.0 - binaire - -rnd-freq=0 (Défaut)
- MiniSAT - version 2.2.0 - binaire - -rnd-freq=0.5
- MiniSAT - version 2.2.0 - binaire - -rnd-freq=0.9
- _____
- CryptoMiniSAT - version 2.4.0 - binaire - paramètres par défaut
- CryptoMiniSAT - version 3.1.0 - binaire - paramètres par défaut
- CryptoMiniSAT - version 4.5.3 - binaire - paramètres par défaut
- CryptoMiniSAT - version 5.6.8 - binaire - -freq=0 (Défaut)
- CryptoMiniSAT - version 5.6.8 - binaire - -freq=0.5
- CryptoMiniSAT - version 5.6.8 - binaire - -freq=0.9

Afin de comparer ces solveurs, nous récupérerons leur résultat et leur temps d'exécution sur toutes les formules du dossier "Benchmarks", avec un timeout à 10 minutes. Afin d'obtenir des résultats les moins bruités possible, nous effectuons ces calculs plusieurs fois et utilisons la moyenne.

Les tests sont lancés depuis un projet XText. Pour reproduire les données présentées dans ce rapport, il suffit d'exécuter le fichier "org.xtext.example.msat.tests/src/org/xtext/example/msat/theos/Mein.xtend" avec JUnit. Les résultats seront stockés dans les fichiers "result_i.csv".

2 Résultats

Les données brutes peuvent être trouvées dans le dossier "results", avec "results_i.csv" les données de chaque exécution de la base de données.

2.1 Aspect fonctionnel

Tous les solveurs ont renvoyé la même réponse pour toutes les formules présentes dans "Benchmarks".

2.2 Efficacité

Afin de comparer l'efficacité, nous ne parlerons ici que du temps d'exécution, sans compter d'autres facteurs tels que l'espace utilisé.

3 Questions

Given a benchmark (a set of SAT formulae), we aim to know: What is the “best” solver? Are there functional bugs in some solvers? An immediate approach is to control whether the solvers return the same result for the same formula (we hope so!). You can use the benchmark or generate random formulae. Are there performance deviations of some solvers? Significant deviations may suggest performance-related bugs in some solvers. Is it the case? Are there harder SAT formulae? Can we reduce a benchmark and only consider a subset? (open question) Numerous formulae are used, but only a few lead to performance variations. Can’t we only use a subse

Write a report that addresses the questions above: your answers should be supported by data (visualizations, statistics, tables, etc.). You should also include a technical description of your experiments (what solvers you use, what formulae, the measurements conditions, etc.), explain how to reproduce your results, and point out the data you have relied on. Push the report and all material in a subfolder (with a unique name) in the “reports” folder: repo.