

Rapport projet logique - équipe COMBO

Juguet Léo, Hilaire Baptiste

Mai 2023

1 Sujet et réponses aux questions

L'objectif de ce projet est de résoudre le problème de satisfiabilité d'une formule dans certaines théories qui sont passées en paramètres. Afin que l'algorithme ait un sens, ces théories doivent contenir un symbole de fonction d'égalité noté '=', ainsi que la sorte des booléens, et l'on doit pouvoir implémenter un solveur dans celles-ci. De plus, chaque symbole de fonction ou de prédicat doit apparaître dans une seule théorie pour que l'algorithme soit bien défini et déterministe.

Nous avons choisi ce projet car son aspect général semblait intéressant, et il permet de mieux comprendre comment sont formées les théories pour pouvoir être utilisées en pratique.

Question 3 : on suppose que les symboles de fonctions présents dans la formule apparaissent dans des théories passées en paramètres, sinon, on renvoie UNSAT. f est présent dans une théorie T_f , d'arité (s_1, int, int) , x_1 est de sorte s_1 . Pour cet exemple, on renverra UNSAT, car le solveur de la théorie des entiers par exemple, voit que $f(x_1, 0) \geq x_3$, où x_3 est de sorte int , mais aussi que $f(x_1, 0) \leq x_3 - 1$ ce qui mène à une contradiction.

2 Choix d'implémentations

Pour les choix d'implémentations, nous avons donc besoin d'une structure pour les théories qui permette de stocker les sortes de celle-ci, un solveur adapté, les symboles de fonctions et aussi les variables pour l'étape de séparation. Nous devons aussi définir des fonctions sur cette structure, nous avons donc choisi d'implémenter un module *Entry* qui implémente le type des theories. Les sortes sont implémentées par des chaînes de caractère comme "string", ou "int", et l'on dispose d'une fonction `get kind of` valeur out of sorte qui depuis la sorte retrouve le type cherché. Le type des formules est défini inductivement. `Func` implémente les symboles de fonction, prenant en entrée le dit symbole ainsi

qu'un tableau représentant l'arité de la fonction. Une input du problème, de type entry, et donc un couple contenant une formule et un tableau de théories. On utilise un tableau car il va être pratique de manipuler la théorie avec des entiers représentant leur indice dans ce tableau.

2.1 Purification

Pour l'étape de purification, on utilise la modalité suivante : les variables rajoutées (z dans le sujet) sont précédées d'un underscore suivi d'un entier pour pouvoir les distinguer. La purification se fait ensuite comme le sujet la décrit. On utilise un tableau "separate and" qui à une case th va stocker la formule qui se résout dans la théorie située à l'indice th du tableau *theories* de l'entrée. On définit une fonction auxiliaire qui parcourt les théories. Dans celle-ci, si l'on croise un symbole de fonction : si le symbole de fonction est celui de la théorie en cours d'étude, on se rappelle récursivement, sinon, on crée un nouveau symbole de fonction égal dans la théorie, et on se rappelle récursivement. Les autres cas sont directs.

2.2 Le vrai solveur

Pour finalement résoudre le problème, on prend le tableau produit separate and, et on applique le solveur dans chaque case de celui-ci, en propageant les égalités. Pour cela, les solver renvoient un résultat qui est soit UNSAT, soit SAT et une liste de formules pour les variables : ces formules expriment soit les valeurs des variables pour connaître les égalités possibles, soit expriment les autres informations que l'on peut avoir sur elles, comme une plage de valeur : ex : $1 \leq x \leq 9$. On propage, à partir de ces résultats, les égalités entre les variables. Dès qu'une égalité est ajoutée, il faut reprendre depuis la dernière clause modifiée et appliquer de nouveau le solveur en prenant davantage d'égalités en compte.

3 Différents problèmes que l'on a rencontré

Beaucoup des choix d'implémentations se sont fait après avoir rencontré des problèmes : le résultat d'un solveur par exemple. Au départ celui-ci renvoyait une liste de valeurs pour les variables, mais cela posait un problème, deux variables pouvaient être mises à la même valeur "par hasard" et pas par vraie implication de la clause. Cette égalité "parasite" pouvait alors rendre une autre close UNSAT. ON avait aussi commencé par implémenter les théories dans des listes, ce qui rendait leur énumération difficile. Enfin, une autre difficulté était de trouver quelles étaient les contraintes sur les théories et les formules que nous allions manipuler.