Encodage d'entiers en booléens dans des formules de logiques pour la vérification de propriété

Mots-clefs : Encodage d'entier; Formules Logiques; Sémantiques de Circuits Intégrés; Solveurs SAT; Solveurs SMT; Analyse de Performance; `OCaml`

Contexte

La fabrication d'un circuit électronique est composé de nombreuses étapes. Vous avez probablement vu en cours d'architecture la partie numérique du flot de conception, où la fonctionnalité du circuit est décrite essentiellement avec des 1 et 0. La description logique (sous forme de portes ou de programme dans des langages comme VHDL) est ensuite transformée en une implémentation utilisant des transistors, en ajoutant des notions qui ne sont pas visibles au niveau logique comme le fait que différentes parties du circuit peuvent être alimentées par des tensions différentes pour économiser l'énergie. Le passage du niveau logique au niveau transistor est essentiellement automatisé, mais certaines parties peuvent être réalisées par l'utilisateur, et donc sujettes à erreurs. Il est ainsi important d'être capable de vérifier l'absence d'erreur sur une description de circuit au niveau transistor. L'équipe CASH du laboratoire LIP travaille avec la start-up Aniah sur le sujet.

L'approche globale de vérification de circuit est une combinaison de méthodes rapides, mais qui peuvent lever un grand nombre de fausses alarmes (autrement dit, lever un avertissement là où il n'y a pas d'erreur), et de méthodes plus lentes et plus précises pour éliminer les fausses alarmes. L'équipe CASH s'intéresse surtout à cette seconde catégorie de méthodes, en modélisant le fonctionnement du circuit avec des formules de logique, et en vérifiant si ces formules de logiques sont satisfaisables avec des solveurs existants (notamment le solveur Z3, développé par Microsoft). La vérification de satisfaisabilité de formule est un problème NP-complet, mais les solveurs se comportent bien en pratique sur des formules de taille raisonnables. À l'heure actuelle, nous disposons d'un compilateur permettant de générer une formule de logique à partir d'une description de circuit, et nous pouvons l'utiliser pour prouver des propriétés. Pour l'instant, les formules de logique que nous utilisons mélangent des valeurs numériques et des booléens, et nous utilisons le solveur Z3 pour en vérifier la satisfaisabilité.

1 Objectifs du projet

L'objectif de ce projet est d'expérimenter un encodage purement booléen de ces formules. De la même manière que les entiers (bornés) peuvent être représentés en machine sous forme d'un nombre fini de bits, nous pouvons transformer les valeurs entières en un ensemble de booléens dans la formule. Nous souhaitons donc au moins :

- Expérimenter les encodages classiques (encodage compact avec $\log(n)$ bits pour représenter un entier dans l'intervalle $[0, n-1]$, encodage 1-parmi-N)
- Expérimenter des encodages non-standard, par exemple utiliser i bits à 1 puis $n-i$ bits à 0 pour représenter le nombre $i \in [0, n]$, similaire à l'encodage 1-parmi-N mais qui permet d'encoder très naturellement la relation $\leq$

- Comparer les performances de solveurs sur la formule obtenue (analyse de données sur les résultats pour différents cas de tests)

Bien sûr, ce travail sera accompagné d'une bibliographie, notamment sur les différents encodages possibles, et leurs interactions avec les solveurs de formules logiques (SAT et SMT notamment).

Profil attendu

Le projet peut être réalisé seul(e) ou en binôme.

L'étudiant(e) devra être intéressé(e) par l'algorithmique et la représentation des données, et avoir un intérêt pour la logique. Notre compilateur est implémenté dans le langage OCaml, donc une connaissance de ce langage serait un plus, mais il est aussi possible de réaliser les expérimentations en dehors de l'outil en utilisant n'importe quel autre langage.

Le projet peut prendre plusieurs orientations selon les goûts de l'étudiant(e) : on peut l'axer sur la théorie, comprendre et éventuellement modifier la sémantique sur laquelle nous nous basons pour la compilation, prouver formellement qu'une optimisation est correcte ; ou bien au contraire l'axer sur la pratique, l'implémentation et l'analyse des résultats expérimentaux.

Une poursuite du travail en stage de master, voire en thèse, est possible si l'étudiant(e) est motivé(e) pour le faire.

Encadrement

- Matthieu Moy, maître de conférences UCBL/LIP, `https://matthieu-moy.fr/`,
- Bruno Ferres, post-doctorant au LIP, `https://perso.ens-lyon.fr/bruno.ferres/`.