Master Informatique

“Preuve mécanisée de l’algorithme de Tarjan”

Rapport
en vue de la validation de
l’UE Initiation à la recherche

2019-2020

Étudiants : Angela Ipseiz
Maxime Nicolas
Matthieu Olejniczak

Encadrant : Monsieur MERZ

Décharge de responsabilités

L'Université de Lorraine n'entend donner ni approbation ni improbation aux opinions émises dans ce rapport, ces opinions devant être considérées comme propres à leur auteur.

Remerciements

Nous tenons à remercier toutes les personnes qui ont contribué au bon déroulement de notre travail et qui nous ont aidé lors de la rédaction de ce rapport.

Tout d'abord, nous adressons nos remerciements à notre encadrant, Monsieur Stephan Merz, enseignant chercheur au LORIA, pour son accueil, le temps passé ensemble, son écoute, ses conseils et ses interventions qui nous ont permis de progresser dans notre démarche et notre compréhension sur les preuves mécaniques. Il fut d'une aide précieuse dans les moments les plus délicats.

Enfin, nous tenons à remercier toutes les personnes qui nous ont conseillé et relu lors de la rédaction de ce rapport : nos familles, nos amis et surtout notre professeur de communication Madame Marie-Laure Alves.

Sommaire

Introduction

« Nous dépendons de plus en plus des logiciels, et il est important qu'ils fonctionnent de mieux en mieux. »[(1)] déclare Leslie Lamport, pionnier de l'algorithme distribué et du concept de preuve algorithmique. En effet, les logiciels sont de plus en plus présents dans notre quotidien et pour certains, notre sécurité en dépend, comme les logiciels utilisés dans l'aviation ou le ferroviaire. De nos jours, un programme incertain, c'est-à-dire avec au moins une condition d'utilisation qui rend le résultat incorrect, peut engendrer d'énormes conséquences financières, humaines,.... C'est pourquoi, il est impératif de réaliser des **preuves d'algorithmes**, notamment pour ceux qui sont liés directement à la sécurité, comme celles qui ont été faites pour l'automatisation de la ligne de métro 14 à Paris, ou le respect de la vie privée que doivent garantir les entreprises du Cloud.

Différents types d'algorithmes peuvent nécessiter une preuve. En effet, il existe des algorithmes impliquant des **graphes** qui sont au coeur de nombreux problèmes actuels. Par exemple, la résolution d'un sudoku peut être transformée en problème 2-SAT qui détermine s'il existe une solution possible grâce à des graphes. Il est des graphes, des structures de données représentées par un ensemble de sommets et d'arêtes. Un graphe est **orienté** si ses arêtes ne sont parcourables que dans un seul sens dont la direction est représentée par une flèche. Le problème 2-SAT utilise la décomposition en **composantes fortement connexes**. On appelle composantes fortement connexes un sous-graphe maximal G' d'un graphe orienté G tel que, pour tout couple (u,v) de noeuds de G,' il existe un chemin de u à v. Ainsi, un graphe fortement connexe est un graphe formé d'une seule composante fortement connexe.

Pour les identifier, on peut utiliser différents algorithmes tels que l'algorithme de Kosaraju et l'**algorithme de Tarjan**, tous deux fondés sur un algorithme de parcours en profondeur en temps linéaire. La principale différence entre ces deux algorithmes est que celui de Tarjan identifie les composantes fortement connexes en parcourant une seule fois le graphe au lieu de deux pour celui de Kosaraju. Nous voulons ici nous intéresser à l'algorithme de Tarjan : Est-ce que l'algorithme de Tarjan nous donne l'ensemble des composantes fortement connexes quelque soit le graphe orienté sur lequel on l'applique ?

Pour répondre à cette problématique, nous regarderons dans un premier temps le bon fonctionnement de l'algorithme par le biais de l'outil TLA+. Ensuite, nous nous intéresserons à sa preuve. Enfin, nous étudierons les problèmes que nous avons rencontrés et comment nous les avons réglés.

I. Les outils utilisés

I.i. TLA+

- Description du langage TLA+

I.ii. PlusCal

- PlusCal permet de transcrire un algorithme en TLA+

I.iii. Model checker

- Permet de montrer la correction de l'algorithme sur des exemples concrets

I.iv. TLA+ proof system

- Outil qui permet de faire la preuve de l'algorithme
- Développé par les chercheurs du LORIA

II. Algorithme de Tarjan

II.i Description de l'algorithme

- Cherche toutes les composantes fortement connexes du graphe

II.ii Fonctionnement de l'algorithme

- Système de classification des nœuds
- Basé sur un parcours en profondeur d'abord
- Utilisation d'une pile

II.iii Correction de l'algorithme

- Utilisation du model checker sur des exemples concrets
- Résultats du model checker

III. Preuve de l’algorithme de Tarjan

III.i Invariants

- Un invariant ? Explication
- Comment trouver un invariant

III.ii Preuve des invariants

- Invariant de typage
- Invariant des sommets sur la pile
- Invariant des couleurs

Problèmes et solutions apportées

Transcription de l'algorithme en TLA+

- Problème de transcription
- Tla+ et la récursivité
- Solution : PlusCal

Preuve de l'algorithme

- difficultés pour les parties non déterministes
- difficultés pour les parties reposants sur le fonctionnement de la pile
- Renforcement des invariants

Conclusion

Il est facile de comprendre comment peut fonctionner un programme, mais il est difficile de prouver qu'il fonctionnera dans tous les cas. Au mieux, on peut s'informer sur les résultats qu'il peut donner en utilisant des outils comme un model checker. Mais ceci ne suffit pas. Comme nous l'avons vu, ces outils ont leurs limites. Par exemple, pour l'algorithme de Tarjan, nous ne pouvons avoir un retour que sur des petits graphes. Ainsi, il est très compliqué de savoir s'il fonctionne et finit sur un graphe avec plus d'une centaine de nœuds et il est impossible de tester tous les graphes possibles car il en existe une infinité. Cependant, même si cela permet d'avoir une idée sur la correction du programme, encore reste-t-il à la prouver. Pour se faire, nous avons vu que l'on peut utiliser des logiciels de preuve semi-automatique comme TLA+ proof system, une extension de TLA+. Bien qu'ils soient d'une grande aide pour avoir une idée de ce qu'il faut prouver, ils ne nous indiquent pas comment le faire.

Malgré ces outils, les preuves continuent de se faire de manière empirique. Par exemple, lorsque l'on veut prouver un invariant, on peut se rendre compte assez vite que l'on est bloqué, qu'il nous manque quelque chose pour avancer et donc on se retrouve obligé de lui rajouter de nouvelles conditions, de le rendre plus strict. Évidemment, ce que l'on ajoute doit être à nouveau prouvé. C'est donc ici que l'on fait un pas en arrière pour pouvoir continuer à avancer. Et des pas en arrière, nous en avons fait quelques uns lors de la preuve de l'algorithme de Tarjan, mais qui nous ont permis de faire beaucoup de pas en avant dans notre preuve. Nous avons donc pu faire une partie de la preuve et démontrer que l'algorithme de Tarjan, est correct pour les cas traités.

L'algorithme de Tarjan n'est qu'un cas d'étude. Il existe des algorithmes et des programmes bien plus compliqués à prouver. Cependant, certaines méthodes, comme celle que nous avons utilisée, peuvent et doivent leurs être appliquées. En effet, si nous développons un algorithme qui va être utilisé pour gérer le trafic ferroviaire, il est nécessaire de s'assurer qu'il fonctionne dans n'importe quelle situation. Fort heureusement, le domaine de la preuve est un domaine émergent sur lequel de grandes entreprises comme Amazon investissent de plus en plus, c'est pourquoi nous pouvons espérer un jour savoir pertinemment ce que l'on fait.

Annexes

Bibliographie

« Ça ne sert à rien de commencer à programmer si l'algorithme lui-même est mauvais » Leslie Lamport à propos de TLA+.
(1)https://www.lemonde.fr/sciences/article/2014/06/16/leslie-lamport-un-informaticien-dans-les-nuages_4439171_1650684.html

https://fr.wikipedia.org/wiki/Qualit%C3%A9_logicielle
https://fr.wikipedia.org/wiki/Graphe_orient%C3%A9

http://zanotti.univ-tln.fr/ALGO/I31/Complexite.html#def:successeur

Déclaration sur l’honneur contre le plagiat

Je soussigné(e),

Ipseiz Angela

Régulièrement inscrit à l’Université de Lorraine

N° de carte d’étudiant : 31608993

Année universitaire : 2019-2020

Niveau d’études : M

Parcours : Informatique

N° UE : 811

Certifie qu’il s’agit d’un travail original et que toutes les sources utilisées ont été indiquées dans leur totalité. Je certifie, de surcroît, que je n’ai ni recopié ni utilisé des idées ou des formulations tirées d’un ouvrage, article ou mémoire, en version imprimée ou électronique, sans mentionner précisément leur origine et que les citations intégrales sont signalées entre guillemets.

Conformément à la loi, le non-respect de ces dispositions me rend passible de poursuites devant la commission disciplinaire et les tribunaux de la République Française.

Fait à Vandœuvre-lès-Nancy, le……..

Signature :

Déclaration sur l'honneur contre le plagiat

Je soussigné(e),

Nicolas Maxime

Régulièrement inscrit à l'Université de Lorraine

N° de carte d'étudiant : 31509056

Année universitaire : 2019-2020

Niveau d'études : M

Parcours : Informatique

N° UE : 811

Certifie qu'il s'agit d'un travail original et que toutes les sources utilisées ont été indiquées dans leur totalité. Je certifie, de surcroît, que je n'ai ni recopié ni utilisé des idées ou des formulations tirées d'un ouvrage, article ou mémoire, en version imprimée ou électronique, sans mentionner précisément leur origine et que les citations intégrales sont signalées entre guillemets.

Conformément à la loi, le non-respect de ces dispositions me rend passible de poursuites devant la commission disciplinaire et les tribunaux de la République Française.

Fait à Vandoeuvre-lès-Nancy, le XX avril 2020

Signature :

Déclaration sur l'honneur contre le plagiat

Je soussigné(e),

Olejniczak Matthieu

Régulièrement inscrit à l'Université de Lorraine

N° de carte d'étudiant : 31506421

Année universitaire : 2019-2020

Niveau d'études : M

Parcours : Informatique

N° UE : 811

Certifie qu'il s'agit d'un travail original et que toutes les sources utilisées ont été indiquées dans leur totalité. Je certifie, de surcroît, que je n'ai ni recopié ni utilisé des idées ou des formulations tirées d'un ouvrage, article ou mémoire, en version imprimée ou électronique, sans mentionner précisément leur origine et que les citations intégrales sont signalées entre guillemets.

Conformément à la loi, le non-respect de ces dispositions me rend passible de poursuites devant la commission disciplinaire et les tribunaux de la République Française.

Fait à Vandœuvre-lès-Nancy, le……..

Signature :

(Quatrième de couverture)