École de technologie supérieure

LOG792 Projet de fin d'étude
Proposition de projet

Formalisation de systèmes de types à l'aide d'Isabelle/HOL

Auteur :
Martin Desharnais

Superviseur :
Dr. David Labbé

16 septembre 2014

Table des matières

1 Problématique et contexte

Ce projet s'intéresse à l'étude des systèmes de types, dans le contexte des langages de programmation, dont voici une définition [Pie02] :

> Un système de types est une méthode syntaxique tractable pour prouver l'absence de certains comportements des programmes par la classification des phrases selon le genre de valeurs qu'elles calculent.

L'objectif est donc de garantir, sans l'exécuter, qu'un programme est exempt de certaines erreurs telles qu'une faute typographique dans un nom de variable, l'appel d'une fonction non supportée dans un certain contexte ou encore une tentative de diviser un nombre par une chaine de caractères. De tels exemples peuvent sembler simplistes, mais sont très fréquents et peuvent avoir des conséquences désastreuses : une incohérence informatique entre les systèmes d'unités métrique et anglo-saxon a provoqué la destruction du Mars Climate Orbiter en 1999. Bien sûr, tous les défauts ne peuvent pas être décelés par un système de types. Cependant, il en existe un très grand nombre, de divers niveaux d'expressivité et de complexité, qui permettent de détecter un éventail varié d'erreurs.

Lors du développement d'un nouveau système de type, un ensemble de preuves est réalisé afin de démontrer que celui-ci respecte ses objectifs. L'étude de ces systèmes ainsi que des preuves qui les accompagnent est le sujet du présent projet.

2 Objectifs du projet

Les objectifs de ce projet sont quadruples : 1) s'initier à la formalisation avec l'assistant de preuve Isabelle/HOL ; 2) s'initier à la théorie des types ; 3) valider les preuves manuelles existantes et 4) clarifier les cas limites de ces dernières.

2.1 S'initier à la formalisation Isabelle/HOL

Il existe deux grandes catégories d'outils pour effectuer une formalisation : les prouveurs automatiques et les assistants de preuve interactifs. Quel que soit l'outil utilisé, il faut définir un contexte de travail et une équation que l'on veut prouver. La différence est que, dans le premier cas, l'outil tente de trouver une preuve entièrement automatiquement alors que, dans le second cas, il faut travailler interactivement avec l'outil pour prouver le théorème.

Isabelle/HOL est un assistant de preuve interactif utilisant la logique d'ordre supérieure. Il permet de spécifier des formules mathématiques, algorithmes et objets dans un langage déclaratif, fonctionnel et typé. Il est alors possible de spécifier des propriétés sur l'interaction entre les divers éléments. Une fois le système

formalisé, il est possible d'en extraire du code exécutable correspondant aux spécifications. Ce projet sera l'occasion de s'initier à la définition d'un système formel et aux preuves interactives à l'aide d'Isabelle/HOL.

2.2 S'initier à la théorie des types

Plusieurs des langages de programmation dominant actuellement ont une étape de validation des types appliquée à la compilation. La majorité des programmeurs sont donc familiarisés avec le concept. Malheureusement, les systèmes de types présents dans ces langages sont généralement simples, imposent nombre de contraintes à leurs utilisateurs et n'offrent qu'un nombre limité de garanties en retour. Certaines de ces limitations ont été mitigées dans de nouvelles versions du langage (e.g. l'ajout des types et fonctions génériques en Java).

Cependant, des options plus expressives et plus puissantes sont bien connues, ou bien actuellement en développement par les acteurs du milieu. Ce projet sera l'occasion de consolider les acquis fondamentaux et d'en apprendre plus sur les concepts plus avancés de la théorie des types

2.3 Valider les preuves manuelles existantes

La théorie des types étant un sujet de recherche très actif depuis plusieurs dizaines d'années, un grand nombre de publications décrivent les caractéristiques de différents systèmes de type. Cependant, une preuve manuelle étant validée par des êtres humains, il est toujours possible que des erreurs s'y soient glissées. La formalisation de celles-ci à l'aide d'un assistant de preuve permet de valider, sous réserve que l'assistant de preuve soit correct, qu'aucune erreur logique ne soit présente. S'il s'avérait qu'une erreur est découverte dans le cadre de ce projet, l'information serait transmise à l'auteur initial afin de l'informer de la situation.

2.4 Clarifier les cas limites des preuves manuelles

Les propriétés énoncées et prouvées manuellement semblent souvent évidentes dès lors qu'elles sont appliquées à un exemple concret. Cette méthode de visualisation a toutefois ses limites puisque certaines constructions plus complexes peuvent entrainer des résultats inattendus. La formalisation de ces propriétés à l'aide d'un assistant de preuve oblige son auteur à considérer la liste exhaustive des constructions du langage et permet ainsi d'acquérir une meilleure compréhension de la propriété et des cas limites. Cette technique fut utilisée avec succès afin d'enseigner l'ingénierie logicielle à des étudiants de premier cycle [PEF08, Pie09, Nip12].

3 Méthodologie

L'ouvrage de référence de ce projet est le livre *Types and Programming Languages* de Benjamin C. Pierce. Ce livre est composé de six sections : les systèmes non typés, les types simples, le sous-typage, les types récursifs, le polymorphisme et les systèmes d'ordre supérieur. Chaque section est composée de plusieurs chapitres décrivant un système de type bonifiant un système décrit précédemment en lui adjoignant un concept supplémentaire. Les figures 1 et 2 présentent les chapitres des deux premières sections sur lesquelles se concentrera ce projet — ceux en gras sont ceux qui seront formalisés — et la figure 3 présente les dépendances entre ceux-ci.

Ces chapitres furent choisis, car ils décrivent des langages et leurs théorèmes au lieu d'en expliquer la théorie ou d'en présenter une implémentation. De plus, ils culminent au lambda-calcul simplement typé, qui a la propriété de pouvoir représenter la majorité des langages de programmation existants [1].

§ 3	**Expressions arithmétiques non typées**
§ 4	Une implémentation en ML des expressions arithmétiques
§ 5	**Le lambda-calcul non typé**
§ 6	Représentation non nommée des termes
§ 7	Une implémentation en ML du lambda-calcul

Figure 1: Section I du livre de référence — Les systèmes non typés

§ 8	**Expressions arithmétiques typées**
§ 9	**Le lambda-calcul simplement typé**
§ 10	Une implémentation en ML des types simples
§ 11	Extensions simples
§ 12	Normalisation
§ 13	Références
§ 14	Exceptions

Figure 2: Section II du livre de référence — Les types simples

Le projet formalisera séquentiellement les différents chapitres en se basant sur le travail des chapitres précédents. Pour cette raison, le premier chapitre sera le

1. Ceci n'est pas tout à fait exact puisqu'il faudrait y ajouter la fonctionnalité de pouvoir communiquer avec l'environnement d'exécution afin de pouvoir faire des opérations comme lire ou écrire dans un fichier, effectuer de la communication interprocessus, etc.

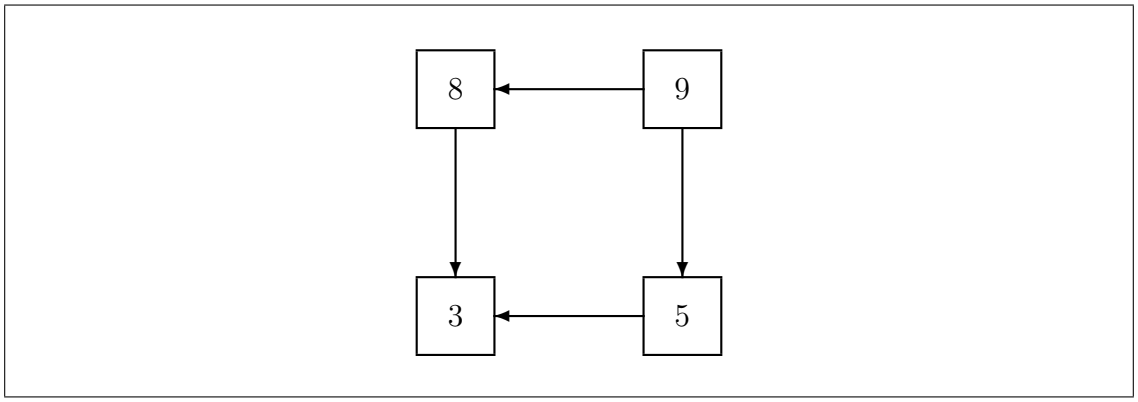

Figure 3: Dépendances entre les chapitres

plus long à formaliser : tout devant être fait à partir de zéro. Pour les chapitres suivants, la première étape sera de copier la formalisation des dépendances et de la modifier pour inclure les nouveaux concepts introduits.

Chacune des formalisations se fera en quatre étapes :

1. Lecture attentive du chapitre et compréhension générale des concepts énoncés ;
2. Définition dans Isabelle/HOL des structures nécessaires à la formalisation ;
3. Preuve des différents lemmes et théorèmes ainsi que, optionnellement, des exercices ;
4. Simplification des définitions et preuves.

L'objectif principal du projet étant de formaliser le chapitre 9 et ce genre de formalisation étant un projet ambitieux, un certain nombre d'actions pourraient être entreprises si le respect des échéanciers s'avère menacé :

1. Ne pas prendre en compte les exercices proposés ;
2. Définir certains lemmes comme des axiomes au lieu de les prouver ;
3. Sauter les chapitres 5 ou 8 afin de passer directement au chapitre 9.

4 Livrables et planification

4.1 Description des artéfacts

Fiche de renseignement Formulaire fournissant le titre du projet, un cours résumé ainsi que les noms des étudiants impliqués et du professeur superviseur.

Proposition de projet Document présentant la problématique du projet, les objectifs, la méthodologie, les livrables, le plan de travail, les risques ainsi que les techniques et outils utilisés.

Rapport d'étape Document présentant une version étoffée et mise à jour de la proposition de projet, ainsi qu'une version partielle du rapport technique.

Diapositives de la présentation orale Diapositives utilisées pour la présentation orale finale du projet.

Rapport final Document présentant la problématique du projet, les objectifs, la méthodologie employée, les hypothèses, les résultats, l'analyse des résultats, les conclusions, les recommandations et les références.

Théories Isabelle/HOL Fichiers sources utilisés par l'assistant de preuve Isabelle/HOL pour sauvegarder les définitions et théorèmes formalisés au cours de ce projet.

4.2 Planification

Le plan de projet se trouve à l'annexe A.

5 Risques

Un certain nombre de risques sont identifiés dans le tableau 1.

Risque	**Impact**	**Probabilité**	**Mitigation / atténuation**
Manque d'expérience avec la théorie des types	Monopolise du temps pour apprendre la théorie.	Faible	Étudier attentivement l'ouvrage de référence.
Manque d'expérience avec l'assistant de preuve Isabelle/HOL	Monopolise du temps pour apprendre le fonctionnement de l'outil.	Forte	Étudier attentivement et se référer au besoin à la documentation de l'outil.
Manque d'expérience en formalisation	Monopolise du temps pour apprendre la méthodologie.	Forte	S'informer auprès des chercheurs expérimentés de la chaire de recherche.
Objectifs trop ambitieux	Ne pas réaliser toutes les formalisations prévues.	Moyenne	Un certain nombre de mitigations sont décrites à la page 4 de la section 3.

TABLE 1 – Risques et mitigations

6 Techniques et outils

Isabelle Système générique pour l'implémentation de formalismes logiques.

Isabelle/HOL Spécialisation d'Isabelle pour la logique d'ordre supérieur (*Higher Order Logic* en anglais).

Isabelle/Isar Langage structuré permettant d'écrire des preuves plus lisibles.

Isabelle/jEdit Environement de développement intégré pour Isabelle basé sur l'éditeur jEdit.

Sledgehammer Outil appliquant des trouveurs automatiques de théorèmes ainsi que des solveurs de « satisfaisabilité modulo théories ».

7 Références

[Bla14] Jasmin C. Blanchette. *Hammering Away : A User's Guide to Sledgehammer for Isabele/HOL*, 2014.

[Nip12] Tobias Nipkow. Teaching semantics with a proof assistant : No more lsd trip proofs. In *Verification, Model Checking, and Abstract Interpretation (VMCAI 2012)*, 2012.

[Nip14] Tobias Nipkow. *Programming and Proving in Isabelle/HOL*, 2014.

[NPW14] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL : A Proof Assistant for Higher-Order Logic*, 2014.

[PEF08] Rex Page, Carl Eastlund, and Matthias Felleisen. Functional programming and theorem proving for undergraduates : A progress report. In *Functional and Declarative Programming in Education (FDPE08)*, 2008.

[Pie02] Benjamin C. Pierce. *Types and Programming Languages.* MIT Press, 2002.

[Pie09] Benjamin C. Pierce. Lambda, the ultimate TA : Using a proof assistant to teach programming language foundations, September 2009. Keynote address at *International Conference on Functional Programming (ICFP).*

8 Annexe A : Plan de travail

#	**Commence**	**Termine**	**Efforts estimés (heures)**	**Jalon**	**Artéfacts**
1		2014-09-02	1	Remise de la fiche de renseignement	Fiche de renseignements
1.1		2014-09-01	1	Rencontre — professeur superviseur	
2	2014-09-02	2014-09-19	5	Remise de la proposition de projet	Proposition de projet
2.1		à déterminer	1	Rencontre — professeur superviseur	
2.2.1	2014-09-19		2	Étude du chapitre 3	
2.2.2			10	Définitions du chapitre 3	
2.2.3			15	Preuves du chapitre 3	
2.2.4		2014-10-10	2	Simplification du chapitre 3	Théorie Isabelle/HOL
3	2014-09-19	2014-10-24	10	Remise du rapport d'étape	Rapport d'étape
3.1		à déterminer	1	Rencontre — professeur superviseur	
3.2.1	2014-10-10		2	Étude du chapitre 5	
3.2.2			5	Définitions du chapitre 5	
3.2.3			10	Preuves du chapitre 5	
3.2.4		2014-10-24	3	Simplification du chapitre 5	Théorie Isabelle/HOL
3.3.1	2014-10-24		2	Étude du chapitre 8	
3.3.2			5	Définitions du chapitre 8	
3.3.3			10	Preuves du chapitre 8	
3.3.4		2014-11-07	3	Simplification du chapitre 8	Théorie Isabelle/HOL
3.4.1	2014-11-07		2	Étude du chapitre 9	
3.4.2			5	Définitions du chapitre 9	
3.4.3			15	Preuves du chapitre 9	
3.4.4		2014-11-21	3	Simplification du chapitre 9	Théorie Isabelle/HOL
5	2014-11-07	2014-12	10	Présentation	Diapositives de la présentation orale
6	2014-12-01	2014-12	30	Remise du travail	Rapport final
6.1		à déterminer	1	Rencontre — professeur superviseur	