

Utilisation d'une machine à pile pour la génération d'arbres de preuve en déduction naturelle et leurs traductions en Français

Arthur Correnson

20 mai 2020

Résumé

La déduction naturelle offre un système confortable pour assister la construction de preuves par ordinateur. Si la vérification de la validité des preuves en déduction naturelle est facilement réalisable à l'aide d'un programme, ces dernières restent relativement illisibles. Nous explorons ici l'idée d'exploiter un même programme pour générer de manière interactive un terme de preuve (vérifié ou non) ou une traduction de cette preuve en Français. L'approche repose sur l'utilisation d'une machine à pile capable d'exécuter un langage intermédiaire de construction de preuves. Par souci de simplicité, on se placera dans le cadre de la logique propositionnelle.

1 Introduction

Vérifier la correction d'une preuve est une tâche qui se délègue assez volontiers à un ordinateur. Elle est rendue possible par l'exploitation de systèmes formelles comme la déduction naturelle qui fixent un cadre suffisamment stricte pour mener à bien une preuve tout en rendant possible sa vérification automatique. Toutefois, si une preuve exprimée en déduction naturelle est facilement manipulable par un programme, elle l'est bien moins pour un être humain. Par ailleurs, construire une preuve en interaction avec la machine est un problème bien plus complexe que d'en vérifier une déjà construite et complète.

En effet, le développement d'une preuve mathématique est un procédé incrémental par nature. On essaye, on corrige à de multiples reprises avant d'enfin pouvoir écrire CQFD en bas d'une page noircie d'encre. Un programme lui, a besoin de la preuve complète avant de pouvoir vérifier sa validité. Aussi la construction de preuves en interaction avec la machine n'est pas un sujet facile. Des systèmes tels que Coq réalisent se défi en proposant un système de tactiques qui permet non pas de générer une preuve vérifiée mais un programme permettant de la construire. Dans ce document, nous présentons Oratio, un outil simple pour la construction et la vérification de preuves qui exploite cette idée de programme de construction pour générer indistinctement tout type d'*objet preuve*, en particulier des textes d'explication en langue Française.

2 Structure du programme

Oratio est un programme conçu et structuré rigoureusement. Il est composé de trois pièces centrales. En premier lieu un noyau encodant les règles d'inférence en déduction naturelle pour la logique propositionnelle. Celui-ci est gardé minimaliste (75 lignes d'OCaml) et donc très facile à vérifier à la main. En second lieu un moteur d'exécution exposant à la fois un langage de construction de preuve et une machine à pile pour évaluer ce langage. C'est ce moteur à la fois simple et modulaire qui est au cœur de notre approche. Enfin, Oratio met à disposition un jeu de tactiques pour l'élaboration de preuves en interaction. Chaque tactique traduisant un contexte de preuve en une série d'instructions pour le moteur de construction. Selon le modèle d'évaluation choisie, un programme généré de manière interactive pourra être utilisé et ré-utilisé pour construire et vérifier un arbre de preuve, un lambda-terme, le texte en français d'une preuve ou tout autre *objet preuve* pourvu qu'un module OCaml approprié soit fournis.

3 Modèle d'évaluation

3.1 Vérification

3.2 Traduction(s)

4 Discussion