

Inférence par contrainte pour les GADTs

Contexte

Le typeur actuel de OCaml est, par certains côtés, compliqué à maintenir (structures de données efficaces mais impératives, modifications à la volée de structures, ...)

L'inférence de type dans OCaml actuellement n'est pas basée sur la résolution de contraintes. Une approche par résolution de contraintes permet une plus grande flexibilité et des meilleures possibilités de maintenance.

Mais cela nécessite de faire rentrer des parties avancées du typeur qui n'ont pas été présentées sous formes de contraintes. Il faut aussi transférer des algorithmes et des structures de données.

La bibliothèque Inferno, développée par François Pottier, nous donne un framework pour construire notre propre typeur par génération de contraintes <http://gallium.inria.fr/~fpottier/publis/fpottier-elaboration.pdf> Une des spécificités de cette bibliothèque est de combiner la génération de contraintes et l'élaboration vers des termes explicitement typés à partir de la solution.

On s'est appuyé sur une formalisation de l'inférence de types par François Pottier et Didier Rémy <http://crystal.inria.fr/attapl/emlti-long.pdf>

On essaye durant ma thèse d'étendre Inferno avec une gestion des GADTs, en rajoutant de nouvelles contraintes, de nouvelles structures de données, des nouveaux mécanismes d'inférence, ...

Travail réalisé

Nous nous étions réunis en mai 2022 durant ma deuxième année.

Depuis, j'ai avancé mes travaux pendant environ une année, puis ai entamé la rédaction jusqu'à aujourd'hui.

En mai 2022, nous avons rajouté le support pour les variables rigides dans Inferno, et nous comptons rajouter les GADTs puis envisager d'intégrer d'autres fonctionnalités d'OCaml.

Durant l'été, nous avons rajouté une notion d'égalité de type dans Système F, ce qui était nécessaire puisqu'il nous fallait une façon d'élaborer des programmes qui introduisent des égalités de types. La plus grosse partie du travail a consisté à augmenter le typechecker de Système F en rajoutant les hypothèses d'égalités.

En septembre/octobre, nous avons travaillé sur la formalisation de l'implémentation, notamment sur une présentation de la comparaison de types Système F avec union-find et la rédaction d'un article pour les JFLA (que nous n'avons pas soumis finalement).

Nous avons continué à travailler jusqu'à Noël sur la formalisation de deux sémantiques pour une nouvelle contrainte d'inférence et leur correspondance.

Par la suite, entre janvier et mai, nous avons travaillé sur l'implémentation : nous avons changé de façon de gérer les variables rigides, en ne les traitant plus comme des variables mais comme de la structure abstraite. Puis nous avons implémenté un mécanisme de portées, et enfin, adapté l'élaboration des schémas de types à notre système. À la fin de cette période, notre implémentation semblait gérer une version minimale des GADTs, avec ambivalence à la OCaml.

Depuis mai, nous travaillons sur le manuscrit. En rédigeant, j'ai effectué quelques modifications dans la formalisation et réfléchi à des façons de présenter une partie du travail que je n'avais pas encore complètement formalisé, comme la gestion des multi-équations dans le solveur ou la sémantique d'une contrainte let que l'on a implémenté. J'ai commencé par rédiger 4 chapitres importants :

- le chapitre 2 qui revient sur le fonctionnement du solveur décrit dans The essence of ML type inference “contributions”
- le chapitre 6 sur le traitement et les sémantiques d'une nouvelle contrainte d'introduction d'hypothèse d'égalité
- le chapitre 7 sur l'extension du solveur pour la résolution de cette contrainte
- le chapitre 8 qui développe plus en détail sur la façon dont on gère les variables rigides

Ces 4 chapitres, dont les 3 derniers sont des contributions, sont dans un état “brouillon complet”. Je m'attaque en ce moment au reste des chapitres, qui sont principalement des explications de ce qui existait déjà, mis à part un chapitre dédié à des explications sur l'implémentation.

Enseignement

J'ai donné des TP de programmation à Paris Diderot en 2022-2023 pour des étudiants en L2 (cours de C) et L3 (programmation fonctionnelle). Comme durant mes autres expériences d'enseignement, j'ai bien apprécié donner ces TP et aider à les préparer.

Plans pour la suite

D'ici Noël, je compte finir mon manuscrit, afin de préparer une soutenance début 2024. L'année scolaire prochaine, je compte suivre une prépa agreg et tenter les concours d'enseignement capes/agreg en 2025.