

Traduction d'OCaml vers une variante de Système F

Jonathan Protzenko
sous la direction de François Pottier

June 14, 2010

Plan

Introduction

Aperçu du problème

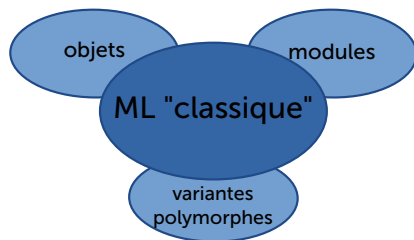
Plan

Introduction

Aperçu du problème

Pourquoi traduire?

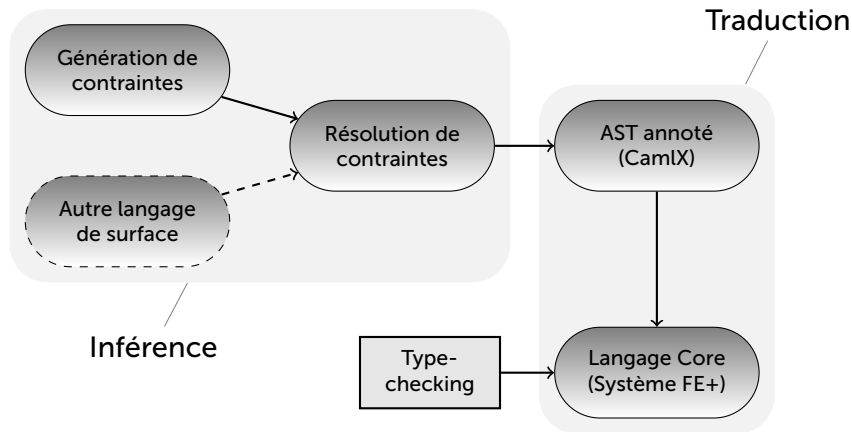
- On veut augmenter la confiance dans la chaîne de compilation
- "Well-typed programs can't go wrong" (Milner)
- Le système de types d'OCaml est trop complexe
 - Traduire le programme dans un langage de base
 - Vérifier le typage *a posteriori*



Objectifs à long terme

- Fournir un langage intermédiaire pour *effectuer des analyses* et compiler plus en avant: expressions *simples*, informations de type *riches*.
- Augmenter la confiance dans la chaîne de compilation: à défaut de prouver la correction du typeur, prouver la cohérence de ses résultats.
- Clarifier la sémantique du langage original: quelles sont les constructions qui s'expriment bien dans FE+?

Dans les grandes lignes...



Le processus se découpe en deux parties : génération/résolution de contraintes, et traductions jusqu'à Système FE+.

Plan

Introduction

Aperçu du problème

Trois grands axes de travail

- Récrire un système d'inférence par contraintes, et l'adapter pour donner un *AST annoté*.
- Élaborer un processus de traduction d'un fragment d'OCaml vers un langage minimaliste
- Concevoir le système de types qui permet de justifier le comportement d'OCaml

```

protzenk@sauternes:~... x  protzenk@sauternes:~... x  prot
fun (x: 0) -> fun (y: 1) -> x
in
let i: λ. [1 → 1] =
  s [1, 1, 0 → 1] k [1, 0 → 1] k [1, 0]
in
()
[DLet] Found a regular let
[DLet] Found a regular let
[DLet] Found a regular let
let s/70 =
  λ. λ (x/75: 1 → 0 → 2) ->
    λ (y/76: 1 → 0) ->
      λ (z/77: 1) ->
        (x/75) z/77 (y/76) z/77
in
let k/71 =
  λ. λ (x/73: 0) ->
    λ (y/74: 1) ->
      x/73
in
let i/72 =
  λ. (s/70·[1]·[1]·[0 → 1]) k/71·[1]·[0 → 1] k/71·[1]·[0]
in
()
  
```


Plan

Introduction

Plan

Introduction

Présentation de l'inférence par contraintes

- Nouvelle présentation d'un algorithme « classique »
- Séparation claire et élégante entre génération et résolution
-

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>

<++>