Tytuł

(English title)

Mateusz Lewko

Praca licencjacka

Promotor: dr hab. Dariusz Biernacki

Uniwersytet Wrocławski
Wydział Matematyki i Informatyki
Instytut Informatyki

27 czerwca 2018

Streszczenie

TODO polish abstract

TODO english abstract

Spis treści

Rozdział 1.

Wprowadzenie

// Co zrobiłem, po co, dlaczego // co to let polymorphism, type class

1.1. Język ML

1. Dlaczego ML, jakie są inne języki ML 2. Bazowanie na *F#*

Rozdział 2.

Cechy języka *lang*

// Cechy z przykładami

2.0.1. Składnia

1. Opis, szczegóły składni, (przykłady: każda cecha języka i krótki przykład)

2.1. Cechy języka

1. Proste wyrażenia, rekurencja, let-polymorphism, rekordy, wzajemnie rekurencyjne funkcje na top levelu, klasy typów, proste moduły, wyrażanie na top levelu, efekty uboczne, inferencja typów, anotacje.

2.2. Klasy typów

1. Wprowadzenie czym są

2. Dlaczego? Jakie są alternatywy

3. Opis tego co zostało zaimplementowane, porównanie do innych języków, (Haskell, Rust, Scala)

2.3. Infrastruktura LLVM

1. Co to jest?

2. Dlaczego LLVM i jakie są inne opcje (C, asembler)?

3. Jak działa kompilowanie do LLVM?

4. Krótki opis high-ollvm

Rozdział 3.

Kompilator

3.1. Etapy kompilacji

1. Jakie są etapy (lexer → parser → untyped ast → typed ast bez zagnieżdżonych funkcji → generowanie kodu (ast high-ollvm) → wywoływanie funkcji z api llvma → llc → gcc i external)

2. Krótko o każdym etapie

3.2. Analiza leksykalna

1. Czego użyłem.

2. Analiza wcięć

3.3. Parsowanie

1. Czego użyłem, coś o Menhirze, dlaczego Menhir

2. Wyzwania (składnia bazująca na wcięciach)

3. Gramatyka

3.4. Inferencja typów

1. Po co? Jak działa u mnie

Rozdział 4.

Generowanie kodu

4.1. Częściowa aplikacja

4.1.1. Opis działania

1. Dlaczego jest to nietrywialne

2. Jakie miałem cele

3. Jak to działa u mnie

4. Przykład (wygenerowanego pseudo-kodu)

4.1.2. Porównanie z innymi implementacjami

1. Push/enter vs eval/apply

Porównanie z pracą "Making a fast curry: ..."

4.2. Zagnieżdżone funkcje

1. Co to są zagnieżdżone funkcje

2. Na czym polega trudność w ich implementacji

3. Jak zostały zaimplementowane: lambda lifting + closure conversion + wykorzystanie aplikacji częściowej

4.3. Rekordy

Implementacja, porównanie do rekordów w F#.

4.4. Let polimorfizm

1. Krótki opis, czym jest let-polimorfizm
2. Sposoby implementacji w różnych językach, zalety i wady
3. Sposób implementacji u mnie

4.5. Klasy typów

1. Czym są? Po co?
2. Sposoby implementowania, porównanie do pracy TODO
3. Jak zostały zaimplementowane, dlaczego tak