Implementation of static and dynamic semantics for a calculus with algebraic effects and handlers using PLT Redex

Maciej Buszka

Instytut Informatyki UWr

04.10.2019

Plan prezentacji

Cel pracy

- Rachunek oraz jego semantyka
  - Efekty algebraiczne
  - Pomocny system typów
- Implementacja modelu
  - Algorytmiczna inferencja typu
  - Ewaluator wyrażeń rachunku
    - Obrazowanie wykonania krok po kroku
    - Redukcja do wyniku końcowego

Formalizacja rachunku

- Składnia abstrakcyjna
- Semantyka statyczna (głównie systemy typów)
- Semantyka dynamiczna (definiuje wykonanie programu)
- Ujawnia interakcje między różnymi elementami rachunku
- Pozwala na udowodnienie właściwości

Formalizacja rachunku - zastosowania

- Rozbudowanie do języka programowania
- Podstawa do wnioskowania o programach
- Rozwijanie nowych rozwiązań językowych
- Języki o dobrych podstawach formalnych są:
  - Spójniejsze
  - Bardziej przewidywalne
  - Łatwiejsze w utrzymaniu i rozwoju

PLT Redex

- Biblioteka do języka `Racket`
- Modelowanie rachunku i jego semantyki
  - Wykonywalne relacje
  - Czytelny opis tych relacji
- Ułatwia testowanie semantyki
- Pozwala na iteracyjne zmiany

Efekty obliczeniowe

- Zjawisko powszechne i znane każdemu programiście
- Zrozumienie działania programu wymaga nielokalnego wnioskowania
- Wpływ programu na otoczenie (inne niż zwrócenie wyniku)
- Lub zależność programu od otoczenia (inna niż przekazane argumenty)
- Dwie ogólne kategorie:
  - Modyfikacja lub odczytanie stanu
  - Alternatywne ścieżki sterowania

Efekty obliczeniowe - podejścia konwencjonalne

- Program może mieć dowolne efekty, a system typów się nimi nie interesuje (C, C++, Java ..., OCaml, Scala)
  - Brak pewności co do zachowania funkcji
  - Niemożliwość wnioskowania równościowego
  - Niejawne zależności między programami
- Język jest czysty, a programy z efektami są modelowane za pomocą monad (Haskell, PureScript)
  - Osobna składnia dla programów z efektami
  - Łączenie różnych efektów jest skomplikowane (np transformatory monad)
  - Problem z dodaniem efektów *post-factum*

Efekty algebraiczne - możliwości

- Jednolita składnia dla programów z efektami i bez nich
- Kompozycja wielu różnych efektów obliczeniowych
- Modelowanie obu wymienionych rodzajów efektów
- Definicja własnych efektów

Efekty algebraiczne - działanie

- Podstawowe założenie: rozdzielenie operacji i ich znaczenia
- Zbiór operacji – wymagany interfejs
- Wyrażenia obsługujące – konkretna implementacja

Efekty algebraiczne - wyzwania

- Nowe, aktualnie rozwijane zagadnienie
- Dość skomplikowana semantyka, ważne niuanse
- Różne podejścia w literaturze:
  - Semantyka statyczna: row-types, capabilities, abstract effects
  - Semantyka dynamiczna: shallow/deep handlers, effect lifting
- Wiele rozbudowanych języków eksperymentalnych: Koka, Helium, Eff, Frank, Links

Rachunek

- Abstrakcyjne operacje
  - Nie muszą być zdefiniowane a priori
  - Przyjmują jeden argument, zwracają jedną wartość
- Wyrażenia obsługujące (*handler*)
  - Dostęp do wznowienia
  - Semantyka tzw. głębokiej obsługi
  - Obsługa wielu operacji naraz
  - Klauzula `return`
- Wyrażenia podnoszące (*lift*)
  - 'przeskoczenie' najbliższego wyrażenia obsługującego
- Wartości bazowe oraz operacje na nich

Model

- Składnia abstrakcyjna
- Relacja redukcji
- System typów
  - Odtwarza typ i efekt wyrażenia
  - Polimorfizm nie jest wspierany
  - Najogólniejszy typ wyrażenia
- Maszyna abstrakcyjna
  - Jawne środowisko
  - Stos i meta-stos

Podsumowanie

- Rachunek z efektami algebraicznymi
  - Operacje, wyrażenia obsługujące i podnoszące
  - Wyrażenia ogólnego zastosowania
- Implementacja modelu
  - Inferencja typu i efektu
  - Pełna redukcja wyrażeń
  - Wizualizacja krok po kroku
  - Front-end – *algeff*
  - Integracja ze środowiskiem `Racket`
- Rozszerzenie rachunku o polimorfizm
- Zbadanie własności rachunku za pomocą automatycznego generowania programów

Dziękuję za uwagę

Implementation of static and dynamic semantics for a calculus with algebraic effects and handlers using PLT Redex

Maciej Buszka

Instytut Informatyki UWr

04.10.2019