A core language for optimised compilation of algebraic effect handlers

Axel Faes
KULeuven

Algebraic effect handlers

Exception handlers on steroids

BUT

Implementations have runtime penalty

```
effect Decide : unit -> bool;;

let choose_all = handler
    | #Decide () k -> k true @ k false
    | val x -> [x];;

with choose_all handle  (if #Decide () then 10 else 20)
(* Output: [10; 20] *)
```

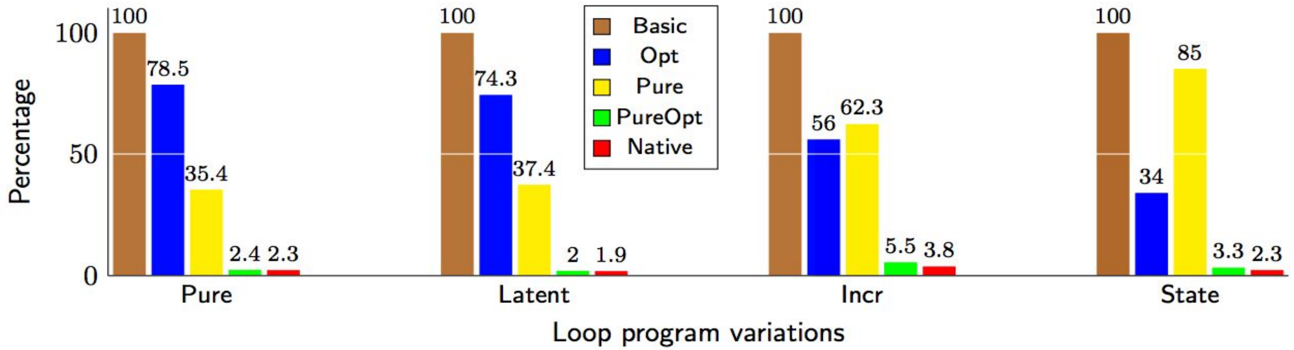
Optimization

Term rewrite rules

- remove handlers / apply effects
- expose more optimisations

Purity aware compilation

- check if computation is pure



Problem

Add optimisations for edge cases

But source-to-source transformations are error prone.

Ensuring transformations do not break typability is time consuming.

New core language

Explicitly typed calculus with row-based effects

Based on Links

Ongoing work

Implementation

Metatheory

Other solutions

https://theaxec.github.io