Towards an atomic $\lambda\mu$-calculus

YR-ICALP 2015

Fanny He
f.he@bath.ac.uk

5 July 2015

Sharing, laziness and atomicity

The $\lambda\mu$-calculus: classical logic and continuations

An atomic $\Lambda\mu S$-calculus

Sharing subexpressions

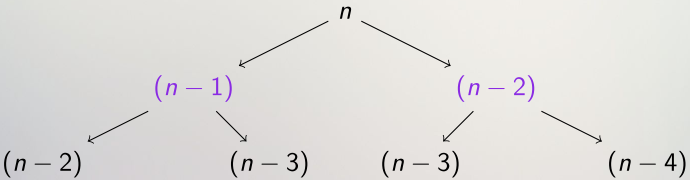
- ```
  fibonacci n | (n == 0) = 0
              | (n == 1) = 1
              | (n > 1)  = fibonacci (n-1) + fibonacci (n-2)
  ```

- ```
  fibonacci2 n = fib 1 0 n
    where
    fib n1 n2 n | (n == 0) = n2
                | (n == 1) = n1
                | (n > 1)  = fib (n1+n2) n1 (n-1)
  ```

Sharing subexpressions

Exponential:

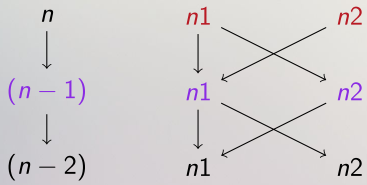
```
fibonacci n | (n == 0) = 0
            | (n == 1) = 1
            | (n > 1)  = fibonacci (n-1) + fibonacci (n-2)
```



Sharing subexpressions

Linear:

```
fibonacci2 n = fib 1 0 n
  where
  fib n1 n2 n | (n == 0) = n2
              | (n == 1) = n1
              | (n > 1)  = fib (n1+n2) n1 (n-1)
```



Lazy evaluation

```
fib = 0:1:zipWith (+) fib (tail fib)
fibo n = fib !! n
```

- `fib`$= [0, 1, 1, \ldots]$
- `tail fib` $= [1, 1, 2, \ldots]$
- `0:1:zipWith (+) fib (tail fib)`
  $= [0, 1, 0+1, 1+1, 1+2, \ldots]$
  $= [0, 1, 1, 2, 3, \ldots]$

A λ-calculus with atomicity

The λ-calculus
[Church]

$$\Lambda : \quad t, u ::= x \mid \lambda x.t \mid (t)u$$

- $(\lambda x.t)(\lambda y.u) \rightarrow_\beta t\{(\lambda y.u)/x\}$

The atomic λ-calculus
[Gundersen, Heijltjes, Parigot]

$$\begin{array}{lcl} \Lambda_a : t, u & ::= & x \mid \lambda x.t \mid (t)u \mid u[c] \\ [c] & ::= & [x_1, \ldots, x_p \leftarrow t] \mid \\ & & [\vec{x_p} \twoheadleftarrow \lambda y.\langle \vec{t_p} \rangle [c_1] \ldots [c_r]] \end{array}$$

- Independent duplication of λy and u
- Naturally retrieves sharing and laziness

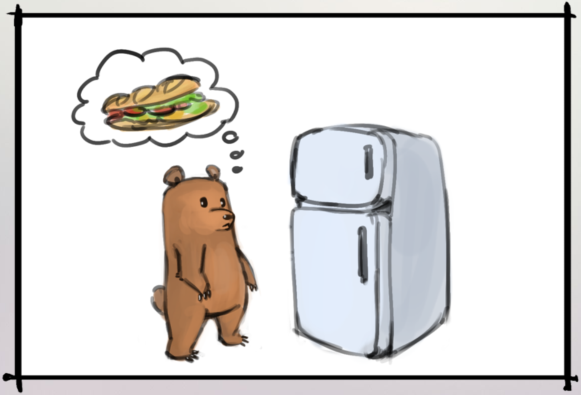
Extend these properties to other calculi?

Continuations: the sandwich approach

1. In front of the refrigerator, thinking about a sandwich,
2. Stick a continuation in your pocket,
3. Use ingredients and make a sandwich (sitting on the counter),
4. Invoke the continuation in your pocket,
5. Back to 1, but there is a sandwich on the counter, and all ingredients are gone: eat the sandwich.

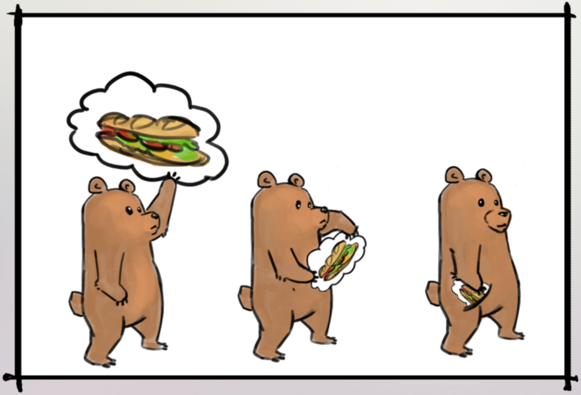
Continuations: the sandwich approach

1. In front of the refrigerator, thinking about a sandwich,
2. Stick a continuation in your pocket,
3. Use ingredients and make a sandwich (sitting on the counter),
4. Invoke the continuation in your pocket,
5. Back to 1, but there is a sandwich on the counter, and all ingredients are gone: eat the sandwich.



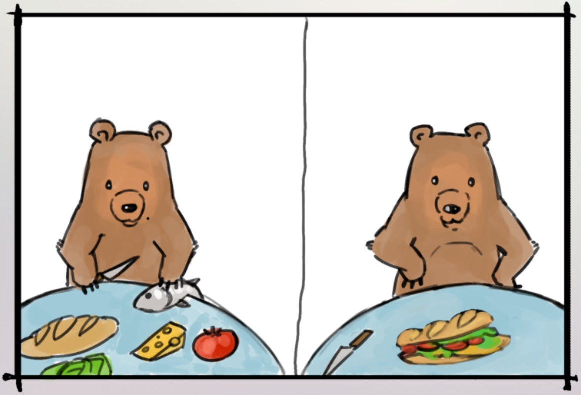
Continuations: the sandwich approach

1. In front of the refrigerator, thinking about a sandwich,
2. Stick a continuation in your pocket,
3. Use ingredients and make a sandwich (sitting on the counter),
4. Invoke the continuation in your pocket,
5. Back to 1, but there is a sandwich on the counter, and all ingredients are gone: eat the sandwich.



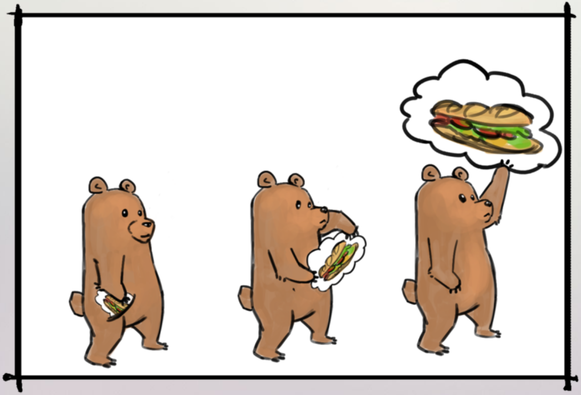
Continuations: the sandwich approach

1. In front of the refrigerator, thinking about a sandwich,
2. Stick a continuation in your pocket,
3. Use ingredients and make a sandwich (sitting on the counter),
4. Invoke the continuation in your pocket,
5. Back to 1, but there is a sandwich on the counter, and all ingredients are gone: eat the sandwich.



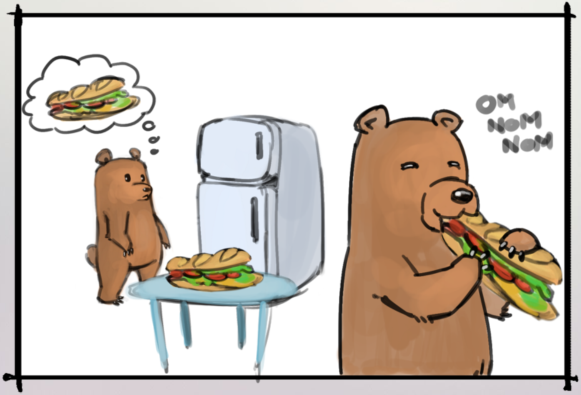
Continuations: the sandwich approach

1. In front of the refrigerator, thinking about a sandwich,
2. Stick a continuation in your pocket,
3. Use ingredients and make a sandwich (sitting on the counter),
4. Invoke the continuation in your pocket,
5. Back to 1, but there is a sandwich on the counter, and all ingredients are gone: eat the sandwich.



Continuations: the sandwich approach

1. In front of the refrigerator, thinking about a sandwich,
2. Stick a continuation in your pocket,
3. Use ingredients and make a sandwich (sitting on the counter),
4. Invoke the continuation in your pocket,
5. Back to 1, but there is a sandwich on the counter, and all ingredients are gone: eat the sandwich.



$\lambda\mu$-calculi [Parigot, Saurin]

- $\lambda \cong_{CH}$ Intuitionistic Logic
- $\lambda\mu \cong_{CH}$ Classical Logic ($A \vee \neg A$)
- Classical operators $\leftrightarrow$ Continuations [Griffin]

The $\Lambda\mu S$-calculus:

$$\begin{aligned} \textit{Streams } S, T &::= \alpha \mid t \circ S \\ \textit{Terms } t, u &::= x \mid \lambda x.t \mid (t)u \mid (t)S \mid \mu\alpha.t \end{aligned}$$

$$(\mu\alpha.t)u \longrightarrow_{\mu} \mu\alpha.\ t\{u \circ \alpha/\alpha\}$$

Explicit sharings and atomicity in $\lambda\mu$-calculi?

A $\Lambda\mu S$-calculus with explicit sharings

$$
\begin{aligned}
&\textit{Closures } [\phi],[\psi] ::= \ [x_1,\dots,x_p \leftarrow t] \mid [\gamma_1,\dots,\gamma_p \leftarrow S]\\
&\textit{Streams } S, T ::= \ \alpha \mid t \circ S \mid S[\phi]\\
&\textit{Terms } t, u ::= \ x \mid \lambda x.t \mid (t)u \mid (t)S \mid \mu\alpha.t \mid u[\phi]
\end{aligned}
$$

The atomic $\Lambda\mu S$-calculus

$$
\begin{aligned}
&\textit{Closures } [\phi],[\psi] ::= \ [\vec{x_q} \leftarrow t] \mid [\vec{\gamma_q} \leftarrow S] \mid [\vec{x_q} \twoheadleftarrow \lambda y.t^q] \mid [\vec{x_q} \twoheadleftarrow \mu\beta.t^q]\\
&\textit{Streams } S,T ::= \ \alpha \mid t \circ S \mid S[\phi]\\
&\textit{Terms } t,u ::= \ x \mid \lambda x.t \mid (t)u \mid (t)S \mid \mu\alpha.t \mid u[\phi]\\
&\lambda\textit{-tuples } t^p ::= \langle\, t_1,\ldots,t_p \,\rangle \mid t^p[\phi]
\end{aligned}
$$

Future work

- Check the properties of this calculus:
  - Termination in a typed setting ✓
  - Type preservation under reduction
  - Confluence
  - Preserving termination w.r.t. λ
- Extend to calculi with more general effects