Motivation

Lazy evaluation would be nice to have in Isabelle

- Computing with codatatypes
  
  \[
  \text{codatatype } 'a \text{ stream } = \text{SCons } 'a (\text{'} 'a \text{ stream})
  \]

- Data-driven programming
  
  \[
  \text{to_list :: ('} 'a, 'b) \text{ rbt } \Rightarrow (\text{'a } \ast \text{'b}) \text{ list}
  \]
Motivation

Lazy evaluation would be nice to have in Isabelle

- Computing with codatatypes
  
  codatatype 'a stream = SCons 'a ('a stream)

- Data-driven programming
  
  to_list :: ('a, 'b) rbt ⇒ ('a * 'b) list

Requirements

- Work with the existing code generator and all target languages
- Transparent to definitions and proofs
Motivation

Lazy evaluation would be nice to have in Isabelle

- Computing with codatatypes
  
  codatatype 'a stream = SCons 'a ('a stream)

- Data-driven programming
  
  to_list :: ('a, 'b) rbt ⇒ ('a * 'b) list

Requirements

- Work with the existing code generator and all target languages
- Transparent to definitions and proofs

HOL-Library.Code_Lazy
Suspension ADT lazy

\[\text{delay :: (unit} \Rightarrow 'a)} \Rightarrow 'a \text{ lazy}\]
\[\text{force :: 'a lazy} \Rightarrow 'a\]

Wadler et al. 1998: “even with difficulty”
Suspension ADT lazy

delay :: (unit ⇒ 'a) ⇒ 'a lazy
force :: 'a lazy ⇒ 'a

Types

datatype 'a list
    = Nil
    | Cons 'a ('a list)
datatype 'a list
    = Lazy_list (('a lazy_list) lazy)
    and 'a lazy_list
    = Nil_lazy
    | Cons_lazy 'a ('a list)

Wadler et al. 1998: “even with difficulty”
Suspension ADT lazy

delay :: (unit ⇒ 'a) ⇒ 'a lazy
force :: 'a lazy ⇒ 'a

Types

datatype 'a list
  = Nil
  | Cons 'a ('a list)

datatype 'a list
  = Lazy_list (('a lazy_list) lazy)
  and 'a lazy_list
  = Nil_lazy
  | Cons_lazy 'a ('a list)

Functions

app xs ys = case xs of
  Nil ⇒ ys
  | Cons x xs' ⇒ Cons x (app xs' ys)

app xs ys = case force (unpack xs) of
  Nil_lazy ⇒ ys
  | Cons_lazy x xs' ⇒ Lazy_list (delay (λ_ ⇒ Cons_lazy x (app xs' ys)))

unpack (Lazy_list susp) = susp

Wadler et al. 1998: “even with difficulty”
Demo
Available in Isabelle2018-RC1

HOL-Library.Code_Lazy
Available in Isabelle2018-RC1

HOL-Library.Code_Lazy

Pattern-matching elimination independently usable:

- case_of_simps
- Code_Target_Nat