Uniform Boilerplate and List Processing

Or: Scrap Your Scary Types

Neil Mitchell and Colin Runciman,
Haskell Workshop, 2007

Simple generics (Haskell ’98)
Hutton’s Razor++

```haskell
data Exp = Lit Int
    | Neg Exp
    | Add Exp Exp
    | Sub Exp Exp
    | Mul Exp Exp
    | Div Exp Exp
```

• What literals are in an expression?
• Change all Sub to Add/Neg?
Literals in an expression

literals (Lit i ) = [i]
literals (Neg x ) = literals x
literals (Add x y) = literals x ++ literals y
literals (Sub x y) = literals x ++ literals y
literals (Mul x y) = literals x ++ literals y
literals (Div x y) = literals x ++ literals y
Uniplate in action

• What literals are in an expression?

\[
\text{literals } x = [i \mid \text{Lit } i \leftarrow \text{universe } x]
\]

• Change all Sub to Add/Neg?

\[
\text{removeSub } = \text{transform } f \\
\text{where } f (\text{Sub } x y) = \text{Add } x (\text{Neg } y) \\
f x = x
\]

* Was called “Play” before Colin renamed it
Simplicity of Haskell ‘98

```haskell
class Uniplate a where
    uniplate :: a -> ([a], [a] -> a)

universe :: Uniplate a => a -> [a]
transform :: Uniplate a => (a -> a) -> a -> a
```

Compared to Scrap Your Boilerplate (SYB):

```haskell
class Data a where
    gfoldl :: (forall d b. Data d => c (d -> b) -> c b) -> (forall g. g -> c g) -> a -> c a
```
Applications (48 on Hackage)

• HLint – Haskell linting tool
• Reduceron – FPGA compiler
• Supero – Haskell optimiser
• Hoogle – Haskell search engine
• NSIS – Windows installer generator
• Scion – IDE backend
• Tamarin prover – Security theorem prover
• Codo notation – Comonad notation
• Yi – text editor
• …
Retrieving re-usable software components by polymorphic type

Colin Runciman and Ian Toyn, JFP, 1991

Let’s define a type-based search engine!

Mikael Rittri, Using Types as Search Keys in Function Libraries. FPCA 1989
... recent developments in so-called hypertext systems ...
**Hoogle (2003-), Хххоо! (2007-)**

- Web based, Haskell servers
- Name and type-based search
- Search 8,457 functions
  - vs 203 in 1991
- Many company-local copies
  - Instant reports if it goes down!
- Integrated in FP Complete IDE
  - People were paid to work on it
• What should match?
• In what order?
• Not too slow...

http://haskell.org/hoogle
\[(a \rightarrow a \rightarrow a) \rightarrow [a] \rightarrow a\]  

- \[(a \rightarrow a \rightarrow a) \rightarrow [a] \rightarrow [a]\]  
  - \[\text{scanl1, scanr1}\]

- Foldable \(t\) \(\Rightarrow\) \[(a \rightarrow a \rightarrow a) \rightarrow t\ a \rightarrow a\]  
  - \[\text{foldl1, foldr1}\]

- \[(a \rightarrow a \rightarrow \text{Ordering}) \rightarrow [a] \rightarrow a\]  
  - \[\text{minimumBy}\]

- \[(a \rightarrow b \rightarrow a) \rightarrow a \rightarrow [b] \rightarrow a\]  
  - \[\text{foldl1}\]

- \[(a \rightarrow b \rightarrow b) \rightarrow b \rightarrow [a] \rightarrow b\]  
  - \[\text{foldr}\]

- \[(a \rightarrow a \rightarrow \text{Bool}) \rightarrow [a] \rightarrow [a]\]  
  - \[\text{nubBy}\]
I would love to update this, but the log file is now 8.4Gb
~30 million searches since 2009
Funny Searches

• Colin Runciman
• :: Colin Runciman
• eastenders
• california public schools portable classes
• diem chuan truong dai hoc su pham ha noi 2008
• ebay consistency version
• videos pornos gratis
• Gia savores de BARILLOCHE
• name of Peanuts cartoon bird