Hooogle

http://haskell.org/hoogle

Neil Mitchell
Haskell Types 101

- `isURI :: String -> Bool`
- `(||) :: Bool -> Bool -> Bool`
- `or :: [Bool] -> Bool`
- `id :: a -> a`
- `Just :: a -> Maybe a`
- `map :: (a -> b) -> [a] -> [b]`
- `(+) :: Num a => a -> a -> a`
What does Hoogle do?

- Search for Haskell functions
  - By type
  - By name

- Demonstration…
What did Hoogle used to do?

- **Version 1**
  - Written in Javascript
  - Only exact matches

- **Version 2**
  - Written in Haskell
  - Partial matches
  - Only the Haskell 98 libraries
The bits inside Hoogle

- Generate a list of functions
- Search
  - By name
  - By type
- Decide on a ranking
- Display documentation
A list of functions

- Version 1 & 2
  - Borrow them from Zvon

- Version 3
  - Take HTML documentation by Haddock
  - Process it
  - Try and figure out the original data

- Dies on certain files…
Search by Type – v1

- Rename all free variables to a canonical form
  - [fred] -> bob → [a] -> b
- Match by string comparison
- No argument reordering
- No “close” standards
Search by Type – v2

- Use **unification**
  - Have argument reordering on top
  - Have missing arguments allowed
- Example: search \([c] \rightarrow [c]\)
- \(\text{map} :: (a \rightarrow b) \rightarrow [a] \rightarrow [b]\)
  - \(a = b = c\)
  - \((c \rightarrow c)\) is a missing argument
A problem...

- Search for: \( k \rightarrow [(k,v)] \rightarrow v \)
- lookup :: \( a \rightarrow [(a,b)] \rightarrow \text{Maybe} \ b \)
  - \( a = k \)
  - \( b = v = \text{Maybe} \ b \) [occurs check fails]
  - \( = \text{Maybe} \ (\text{Maybe} \ b) = \text{Maybe} \ (\text{Maybe}…) \)
- Also:
  - Ranking is very hard
Searching by Type – v3

- Convert type sig -> single steps
- Apply each step
  - Can fail => no match
  - Can part fail => bad marks
- map :: (a -> b) -> [a] -> [b]
  - #1{-} -> {#1.1,#2.1}
  - #2{[]} -> {#1.2,##.1}
  - ##{[]}
Matching to filter

- `filter :: (a -> Bool) -> [a] -> [a]`

- #1{->} (a->Bool){->} Yes

- #2{[]} [a]{[]} Yes

- ##{[]} [a]{[]} Yes

- `{#1.1,#2.1} {a, a} Yes

- `{#1.2,##.1} {Bool, a} Partial

- a in two different sets Partial
Bad marks

- Accumulate bad marks
  - Argument missing
  - Data too free/specific
  - 6 in total
- Matches two ways, so a multiset of items from a 12 item set
- Assign each item a score
- Sum all the items
How to assign scores

- Almost impossible to get right
- Highly subjective
- So don’t do it!
  - Have an example set – when searching for a, I expect result b above result c
  - Run a program, get a constraint program
  - Solve constraint program, get answer
  - Put back into Hoogle
What do people search for?

- 3300 searches (about in a month)
- 600 used the prewritten searches
- Lots of people search for “where”
What else do people search?

- hotmail.com
- google
- eastenders
- california public schools portable classes
- Nintendo Revlution
- Bondage
Conclusion

- A useful practical tool for working with Haskell
- Often just a fast way to lookup the documentation!
- Online at http://haskell.org/hoogle
- Open source, patches welcome!