Curry Crash Course

Sebastian Fischer

Bonn 2013

Functional-Logic Programming

functional

- algebraic datatypes
- higher-order functions
- laziness, monadic IO

Haskell with simpler type system (no type classes, GADTs, ...)

logic

- free variables for unknown values
- first-class nondeterminism and failure
- built-in search

Implementations

PAKCS
- based on Prolog
- many experimental features

KCC
- based on C
- fewer features

KiCS^2
- based on Haskell
- different search strategies

Mountains

Mountains

1. Enumerate all mountains of size 7!

2. How many mountains exist of size 17?

Arithmetic Expressions

$1+2$ $\quad 3*(5+4)$ $\quad 7/2-3$

$$[\![n]\!] = n \quad \text{for integer } n$$

$$[\![e_1 + e_2]\!] = [\![e_1]\!] + [\![e_2]\!]$$

$$[\![e_1 - e_2]\!] = [\![e_1]\!] - [\![e_2]\!]$$

$$[\![e_1 * e_2]\!] = [\![e_1]\!] * [\![e_2]\!]$$

$$[\![e_1 / e_2]\!] = [\![e_1]\!] / n \quad \text{partial}$$

$$\text{if } [\![e_2]\!] = n \neq 0$$

Replacing Subexpressions

$$(1+2)[3]_{\langle 1 \rangle} = 3+2$$

expression — replacement — position

$$((2+3)*7)[8/2]_{\langle 1,2 \rangle}$$

$$= ((2+8/2)*7)$$

Replacing Subexpressions

$$e[r]_{\langle\rangle} = r$$

$$(e_1 \circ e_2)[r]_{\langle 1,p\rangle} = e_1[r]_p \circ e_2$$

$$(e_1 \circ e_2)[r]_{\langle 2,p\rangle} = e_1 \circ e_2[r]_p$$

for $\circ \in \{+, -, *, /\}$

Selecting Literals

literal (1+2) → 1

literal (1+2) → 2

literal (e[n]$_p$) → n

for integer n

Simplification

$1+0 \to 1$ $\qquad$ $2+0 \to 2$

$$simplesS(e_1[e_2+0]_\rho) = e_1[e_2]_\rho$$

$42+0 \to 42$

Simplification

$$\begin{aligned} \text{moreComplex } e &= e + 0 \\ \text{moreComplex } e &= 0 + e \\ &\vdots \end{aligned}$$

$$\text{simplex } (e_1[\text{moreComplex } e_2]_p) \longrightarrow e_1[e_2]_p$$

Simplification

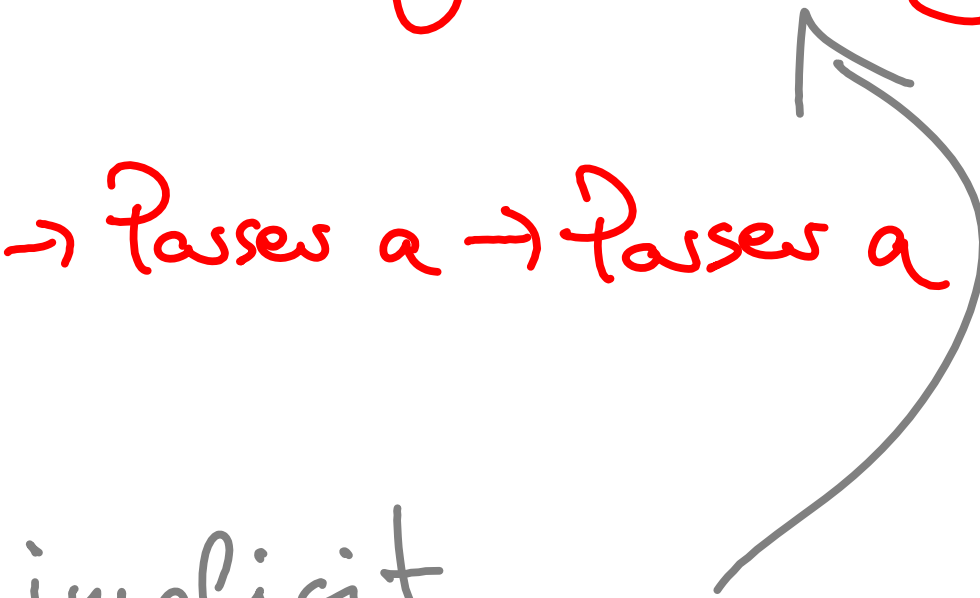
simplify e = e if simpler fails for e

simplify e = simplify (simpler e) otherwise

Passes

```
type Passes a = String -> (a, String)

(|||) :: Passes a -> Passes a -> Passes a
```



Passes / Evaluators

1. Enumerate all strings of length at most 3 for expressions that evaluate to 4!
2. How many strings of length at most 5 correspond to expressions that evaluate to 42?

Summary

Built-in failure simplifies programs that may fail. evaluator, division by zero

Built-in nondeterminism simplifies programs that perform search. simplification, passing

Programs can be run backwards to search arguments for results. passes + evaluator