

Monads reference card

Jean-Philippe Bernardy

December 20, 2011

	monad component	DSL application
Syntax	$m :: \star \rightarrow \star$	Expressions parameterized on return type
	$return :: a \rightarrow m a$	constant expression
	$(\gg=) :: m a \rightarrow (a \rightarrow m b) \rightarrow m b$	bind an a returned by the lhs into the rhs

	name	law	a DSL aspect
Laws	left identity	$return a \gg= (\lambda x. m x) \equiv m a$	inlining/factorizing a constant
	right identity	$m \gg= (\lambda x. return x) \equiv m$	removal/introduction of useless return
	associativity	$(m \gg= f) \gg= g \equiv m \gg= (\lambda x. f x \gg= g)$	extension/shrinking of scope

“do”

$$\begin{array}{l|l}
 \text{do } x \leftarrow \alpha & \alpha \gg= \lambda x. \\
 y \leftarrow \beta & \beta \gg= \lambda y. \\
 \gamma & \gamma
 \end{array}$$

- parentheses are not needed
- x may appear in γ

Comprehensions

$$\begin{array}{l|l}
 [& \gamma \\
 | & x \leftarrow \alpha \\
 , & y \leftarrow \beta \\
] & \text{return } \gamma
 \end{array}
 \begin{array}{l}
 \alpha \gg= \lambda x. \\
 \beta \gg= \lambda y.
 \end{array}$$

- $\gg=$ can be used to “flatten” levels of the monad.
- $join :: m(m a) \rightarrow m a$
- $join xs = xs \gg= id$