A SPECIFICATION LANGUAGE FOR AUTOMATED TESTING OF HASKELL IO PROGRAMS

Oliver Westphal, Janis Voigtländer

University of Duisburg-Essen

MOTIVATION

- **What:**
  - Automatically test correctness of weekly student submissions for Haskell programming tasks
- **How:**
  - pure programs -> QuickCheck properties
  - IO programs -> ???

ALGEBRAIC EFFECTS

```haskell
data IO' a = ...

instance Monad IO' where ...

putStrLn :: String -> IO' ()
getLine :: IO' String

run :: IO' () -> [String] -> Trace
```

"Beauty in the Beast", Swierstra and Altenkirch, 2007

IMPLEMENTING TASKS

```
validInputs :: Gen [String]
checkCorrectness :: Trace -> Bool

prog :: IO' () {- student solution -}

check = quickCheck $ forAll validInputs $
  \xs -> checkCorrectness (run prog xs)
```

Automatically generate `validInputs` and `checkCorrectness`
from a common specification of behavior

EXAMPLE SPECIFICATION

Read a natural number n from `stdin`*, then read n additional numbers and print the sum of those n numbers to* `stdout`.

$$[\,\triangleright\, n\,]^{\mathbb{N}}\left([\,\triangleright\, x\,]^{\mathbb{Z}} \underset{len(x_A)=n_C}{\triangle} \mathbf{E}\right)^{\rightarrow^{\mathbf{E}}}[\,sum(x_A)\triangleright\,]$$

DEMO

https://autotool.fmi.iw.uni-due.de/spec-demo