CHALMERS UNIVERSITY OF TECHNOLOGY

Matthías Páll Gissurarson
pallm@chalmers.se
Department of Computer Science and Engineering

CSI: Haskell

– Fault-Localization in Lazy Languages using Runtime Tracing

Questions?

@tritlo

Problem and Background

- Data in **errors** is often *recently* evaluated.
- `divs 13` results in the empty list `[]`.
- `head` on the empty list raises an error:

```
divs: Prelude.head: empty list
CallStack (from HasCallStack):
error, called at libraries/…/List.hs:1643:3 in base:GHC.List
errorEmptyList, called at libraries/…/List.hs:82:11 in base:GHC.List
badHead, called at libraries/…/List.hs:78:28 in base:GHC.List
head, called at Div.hs:10:17 in main:Main
```

```haskell
module Main where
divs :: Int -> [Int]
divs n = go 2
  where go i | i == n = []
        go i = if d i
               then i:(go (i+1))
               else go (i+1)
        d i = n `mod` i == 0

smallestDiv n = head (divs n)

main :: IO ()
main = print (smallestDiv 13)
```

However, the fault originates in the

PRODUCER

but the error mentions only the

CONSUMER

Artifact available!

https://l.mpg.is/ghc-csi

Approach and Contributions

1. **Extend** Haskell Program Coverage (HPC) [1] in GHC and track *recently evaluated expressions*.
2. **Summarize** traces for readability.
3. **Add** trace summary to error messages.

Now the faulty **producer** is mentioned as well!

- This **improves** fault-localization heuristics in automatic program repair tools like PropR [2].
- Unlike HAT [3], no program transformation beyond HPC is needed, allowing **easy integration** into error messages, debuggers and IDEs.

```
Recently evaluated locations (from HPC):
 Div.hs:4:25-4:26 alternative branch taken
 Div.hs:4:16-4:21 guarded branch taken
 repeats (11 times):
  Div.hs:4:9-7:28 Main:divs>go
  Div.hs:7:21-7:28 alternative branch taken
  Div.hs:5:19-5:21 else branch taken
  Div.hs:8:9-8:28 Main:divs>d
  Div.hs:5:16-7:28 alternative branch taken
  Div.hs:4:16-4:21 guarded branch not taken
 Div.hs:4:9-7:28 Main:divs>go
 Div.hs:3:1-8:28 Main:divs
 Div.hs:10:1-10:29 Main:smallestDiv
 Div.hs:13:1-13:29 Main:main
```

References

[1] Andy Gill and Colin Runciman. 2007. Haskell Program Coverage. In *Proceedings of the ACM SIGPLAN Workshop on Haskell Workshop (Freiburg, Germany) (Haskell '07)*. Association for Computing Machinery, New York, NY, USA, 1–12.
[2] Matthías Páll Gissurarson, Leonhard Applis, Annibale Panichella, Arie van Deursen, and David Sands. 2022. PropR: Property-Based Automatic Program Repair. In The 44th IEEE/ACM International Conference on Software Engineering (ICSE). IEEE/ACM.
[3] Olaf Chitil, Colin Runciman, and Malcolm Wallace. 2002. Transforming Haskell for tracing. In *Symposium on Implementation and Application of Functional Languages*. Springer, 165–181.