

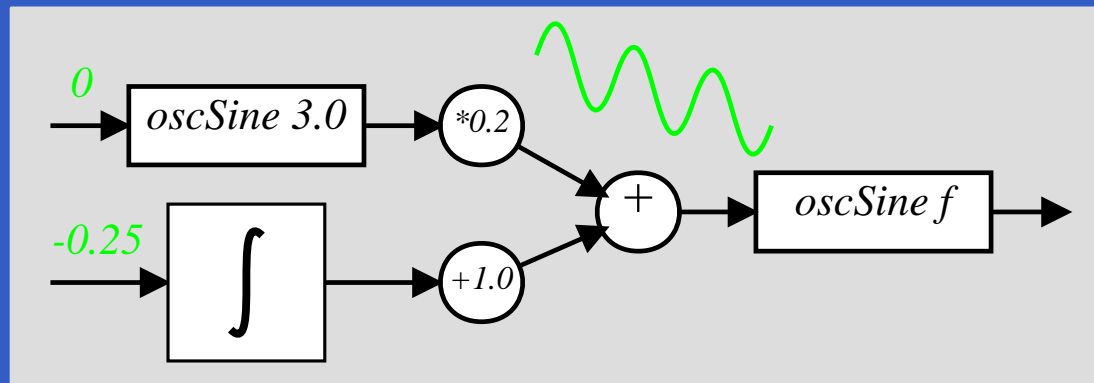
Slicing Functional Programs: A Suspicion

*10th CREST Open Workshop on Program
Analysis and Slicing*

Henrik Nilsson

School of Computer Science
The University of Nottingham, UK

Slicing Functional Programs?



sciFi :: SF () Sample

sciFi = **proc** () → do

und ← arr (*0.2) <<< oscSine 3.0 — 0

swp ← arr (+1.0) <<< integral — -0.25

audio ← oscSine 440 — *und* + *swp*

returnA — *audio*

Behind the Scenes

A signal function (SF) is roughly represented as:

$$\text{data } SF \ a \ b = SF \ (DTime \rightarrow a \rightarrow (b, SF \ a \ b))$$

Behind the Scenes

A signal function (SF) is roughly represented as:

$$\text{data } SF \ a \ b = SF \ (DTime \rightarrow a \rightarrow (b, SF \ a \ b))$$

- Top level loop repeatedly apply the system SF to input, computing output at that point in time and **next** system SF.

Behind the Scenes

A signal function (SF) is roughly represented as:

$$\text{data } SF \ a \ b = SF \ (DTime \rightarrow a \rightarrow (b, SF \ a \ b))$$

- Top level loop repeatedly apply the system SF to input, computing output at that point in time and **next** system SF.
- Output depends on the **entire SF computed** at the previous step.

Behind the Scenes

A signal function (SF) is roughly represented as:

$$\text{data } SF \ a \ b = SF \ (DTime \rightarrow a \rightarrow (b, SF \ a \ b))$$

- Top level loop repeatedly apply the system SF to input, computing output at that point in time and **next** system SF.
- Output depends on the **entire SF computed** at the previous step.

Arguably a somewhat flamboyant (even frivolous?) style of programming!

A Suspicion



A Suspicion



Style matters!

Higher-order Abstractions

Higher-order abstractions are a staple of modern functional programming:

- Monads
- Arrows
- Applicative functors
- Continuation Passing Style (CPS)
- ...

Higher-order Abstractions

Higher-order abstractions are a staple of modern functional programming:

- Monads
- Arrows
- Applicative functors
- Continuation Passing Style (CPS)
- ...

Flamboyance to be expected!

-
-
-

Accounting for Style

Lost cause?

Accounting for Style

Lost cause?

Maybe not: these abstractions:

- are principled
- satisfy useful mathematical laws
- capture simple ideas

Ought to be possible to exploit!

Accounting for Style

Lost cause?

Maybe not: these abstractions:

- are principled
- satisfy useful mathematical laws
- capture simple ideas

Ought to be possible to exploit!

Perhaps we need **Stylish Slicing**: extensible slicing tools capable of slicing **with** style!