Source-Based Trace Exploration with Hat

Olaf Chitil
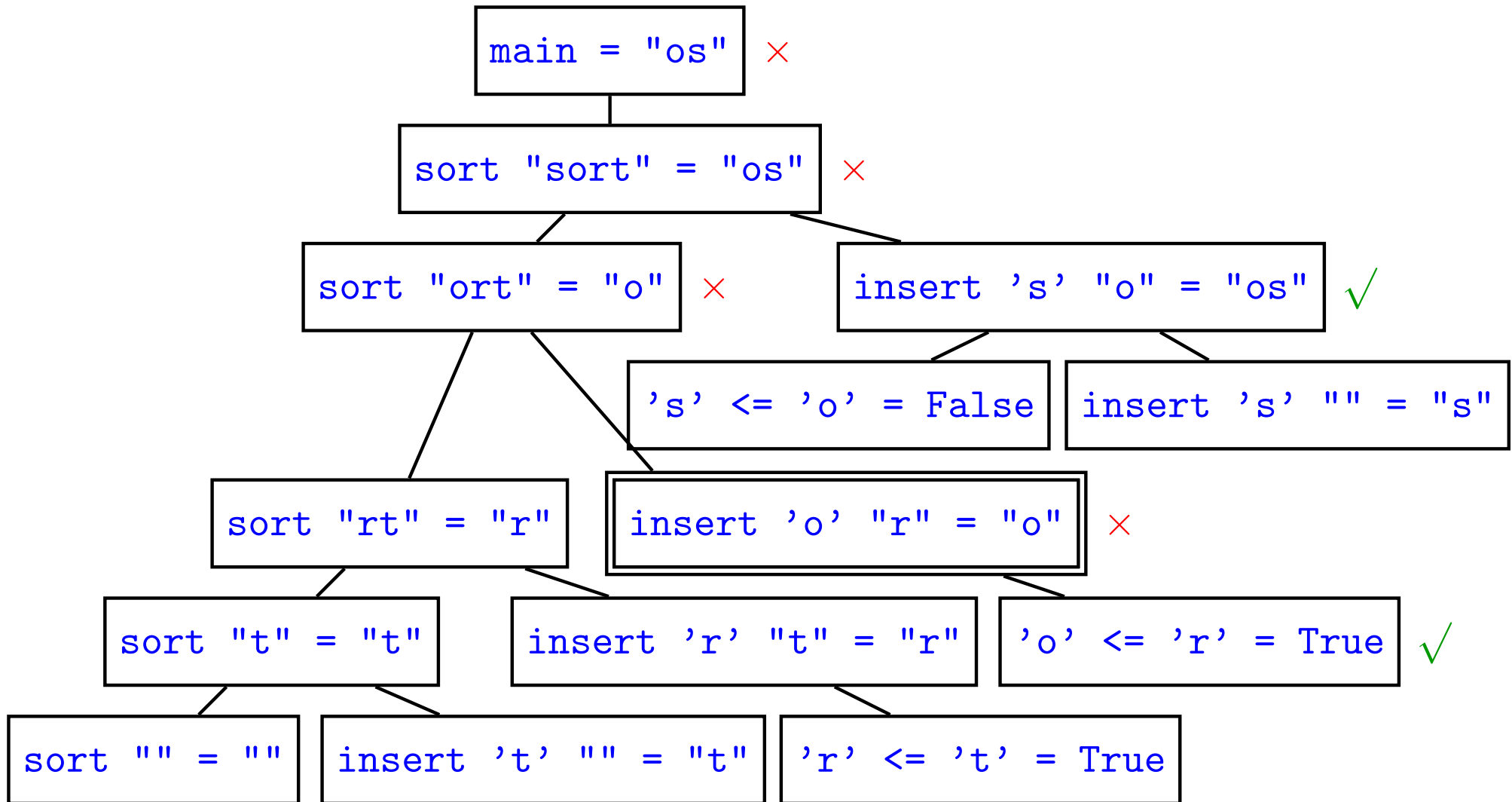
University of Kent
United Kingdom

The Haskell Tracer Hat

input   output

self-tracing computation	→ trace →	hat-observe
		hat-trail
		hat-detect
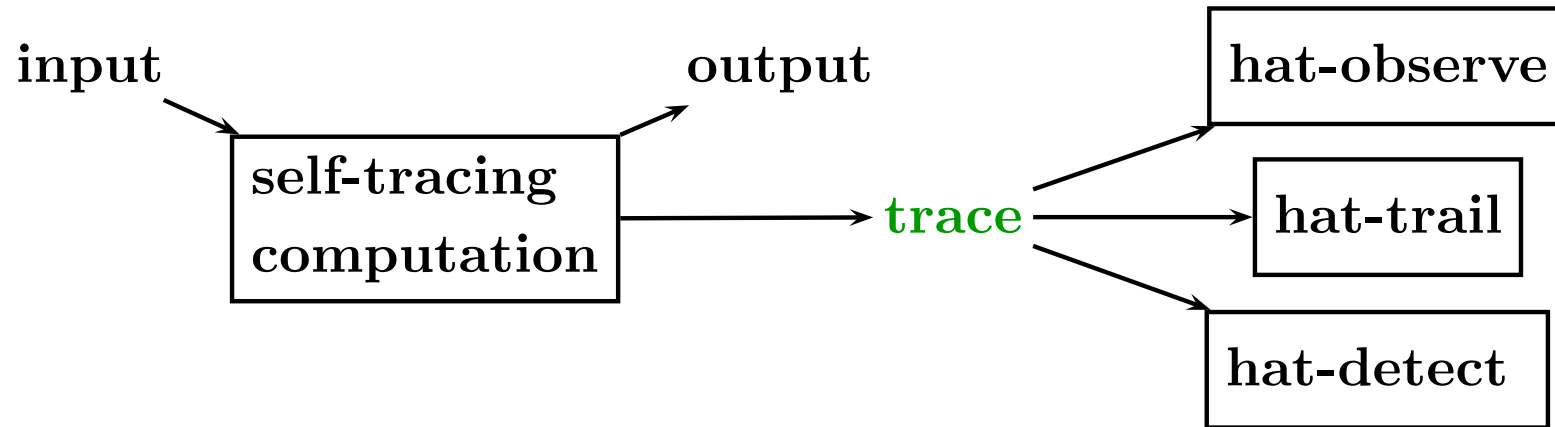
- Two phases: trace generation + trace viewing

- Trace liberates from time arrow of computation

Algorithmic Debugging

```
main = "os"    ✗
```

```
sort "sort" = "os"    ✗
```

```
sort "ort" = "o"    ✗
```

```
insert 's' "o" = "os"    ✓
```

```
's' <= 'o' = False
```

```
insert 's' "" = "s"
```

```
sort "rt" = "r"
```

```
insert 'o' "r" = "o"    ✗
```

```
sort "t" = "t"
```

```
insert 'r' "t" = "r"
```

```
'o' <= 'r' = True    ✓
```

```
sort "" = ""
```

```
insert 't' "" = "t"
```

```
'r' <= 't' = True
```

3

We need source-based viewing tools

- current tools show expressions + equations but no source program

- easy to "get lost" in current viewing tools

- programmer familiar with program

- program smaller than computation

- program will have to be modified

- navigation of hat-trail + hat-detect follows program structure

Summary

- **Orientation through Source**

- **Free Navigation**

- **Fault Slicing of Source**

- **Use Coverage of Source**

http://www.haskell.org/hat