Cut to the Core

Automated Feature Extraction in R Using Program Slicing

FOSD '24 | Ulm University | ***Ruben Dunkel***, Florian Sihler, Thomas Thüm and Matthias Tichy | April, 2024

SP Software Engineering
Programming Languages

universität ulm

Real-World R Code

[1] Drudze et al., *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019"* (2021, Zenodo)

Real-World R Code

- Very long and complex
- Partial data availability (reproducibility problem)
- Take long to run

[1] Drudze et al., *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019"* (2021, Zenodo)

Real-World R Code

1 } Load

2 } Model

3 } Model

4 } Figure

5 } Figure

6 } Figure

- Very long and complex
- Partial data availability (reproducibility problem)
- Take long to run

[1] Drudze et al., *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019"* (2021, Zenodo)

Real-World R Code

1 } Load

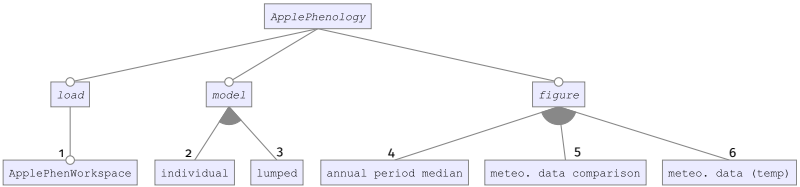
2 } Model

3 } Model

4 } Figure

5 } Figure

6 } Figure



[1] Drudze et al., *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019"* (2021, Zenodo)

The Plan

The Plan

```
sum  ← 0
prod ← 1
n    ← 10

for (i in 1:(n-1)) {
   prod ← prod * i
   sum  ← sum + 2
}

cat("Sum:", sum, "\n")
cat("Product:", prod, "\n")
```

The Plan

```
sum  ← 0
prod ← 1
n    ← 10

for (i in 1:(n-1)) {
   prod ← prod * i
   sum  ← sum + 2
}

cat("Sum:", sum, "\n")
cat("Product:", prod, "\n")
```

The Plan

```
sum  ← 0
prod ← 1
n    ← 10

for (i in 1:(n-1)) {
   prod ← prod * i
   sum  ← sum + 2
}

cat("Sum:", sum, "\n")
cat("Product:", prod, "\n")
```

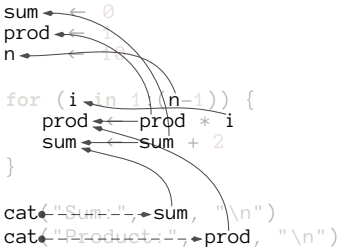
The Plan

```
sum  ← 0
prod ← 1
n    ← 10

for (i in 1:(n-1)) {
   prod ← prod * i
   sum  ← sum + 2
}

cat("Sum:", sum, "\n")
cat("Product:", prod, "\n")
```

(simplified dataflow)

Dataflow-Analysis

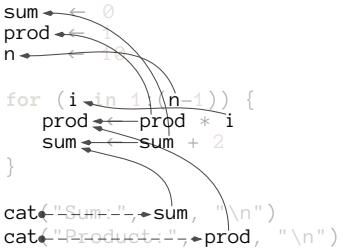
The Plan



(simplified dataflow)

Dataflow-Analysis

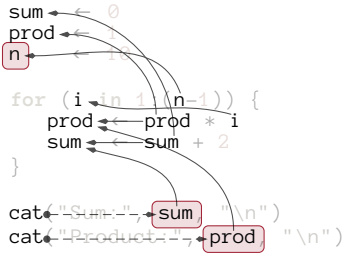
The Plan



(simplified dataflow)

Dataflow-Analysis → **Collect in- and outputs**

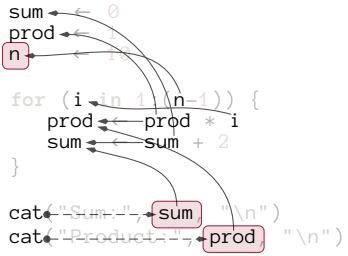
The Plan



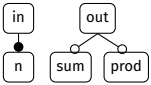
(simplified dataflow)

Dataflow-Analysis → **Collect in- and outputs**

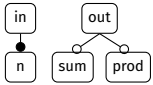
The Plan

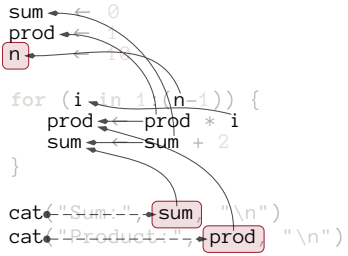


(simplified dataflow)



The Plan

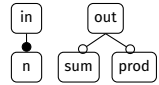


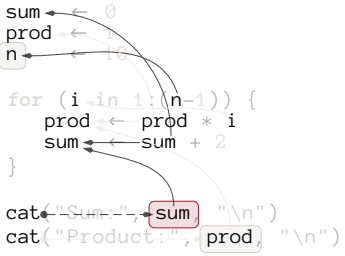


(simplified dataflow)

Dataflow-Analysis → **Collect in- and outputs** → Slicing with *flowR*

The Plan

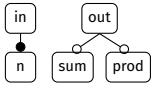




(simplified dataflow)

Dataflow-Analysis → **Collect in- and outputs** → Slicing with *flowR*

The Plan

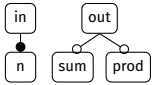


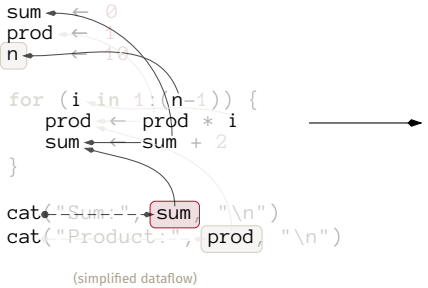

```
sum  ← 0
prod ← 1
n    ← 10

for (i in 1:(n-1)) {
   prod ← prod * i
   sum  ← sum + 2
}

cat("Sum:", sum, "\n")
cat("Product:", prod, "\n")
```

(simplified dataflow)

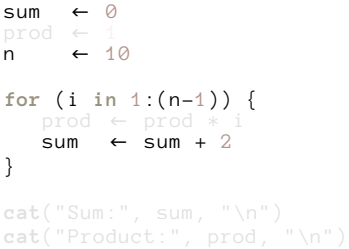
Dataflow-Analysis → **Collect in- and outputs** → Slicing with *flowR*

The Plan



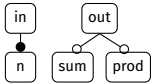


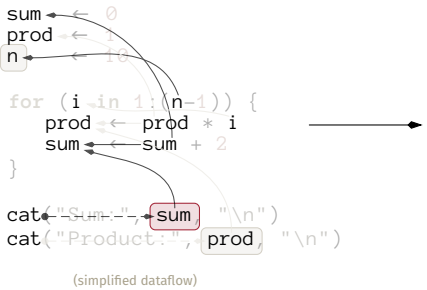
(simplified dataflow)



Dataflow-Analysis → **Collect in- and outputs** → Slicing with *flowR* → **Slices to Proposed Features**

The Plan



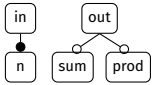


(simplified dataflow)

```
sum  ← 0                          (sum)
prod ← 1                          (prod)
n    ← 10                         (core)

for (i in 1:(n-1)) {              (core)
    prod ← prod * i               (prod)
    sum  ← sum + 2                (sum)
}                                 (core)

cat("Sum:", sum, "\n")            (sum)
cat("Product:", prod, "\n")       (prod)
```

Dataflow-Analysis → **Collect in- and outputs** → Slicing with *flowR* → **Slices to Proposed Features**

The Plan



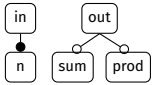

```
sum ← 0
prod ← 1
n ← 10

for (i in 1:(n-1)) {
    prod ← prod * i
    sum ← sum + 2
}

cat("Sum:", sum, "\n")
cat("Product:", prod, "\n")
```

(simplified dataflow)

```
sum  ← 0                    (sum)
prod ← 1                    (prod)
n    ← 10                   (core)

for (i in 1:(n-1)) {        (core)
    prod ← prod * i         (prod)
    sum  ← sum + 2          (sum)
}                           (core)

cat("Sum:", sum, "\n")      (sum)
cat("Product:", prod, "\n") (prod)
```

Dataflow-Analysis → **Collect in- and outputs** → Slicing with *flowR* → **Slices to Proposed Features** → **Allow Manual Adjustments**

The Plan




```
sum ← 0
prod ← 1
n ← 10

for (i in 1:(n-1)) {
    prod ← prod * i
    sum ← sum + 2
}

cat("Sum:", sum, "\n")
cat("Product:", prod, "\n")
```

(simplified dataflow)

```
sum  ← 0          (sum)
prod ← 1          (prod)
n    ← 10         (core)

for (i in 1:(n-1)) {   (core)
    prod ← prod * i    (prod)
    sum  ← sum + 2     (sum)
}                      (core)

cat("Sum:", sum, "\n")      (sum)
cat("Product:", prod, "\n") (prod)
```

Dataflow-Analysis → **Collect in- and outputs** → Slicing with *flowR* → Slices to Proposed Features → Allow Manual Adjustments → Add Preprocessor Directives

Open Questions

1. Who already worked with R?
2. What is your experience with code-extraction?
3. What is good/important related work?

Open Questions

1. Who already worked with R?
2. What is your experience with code-extraction?
3. What is good/important related work?

ruben.dunkel@uni-ulm.de

References

[1] Inese Drudze et al. *Apple phenology data set and R script, related to publication "Full flowering phenology of apple tree (Malus domestica) in Pūre orchard, Latvia from 1959 to 2019".* June 2021