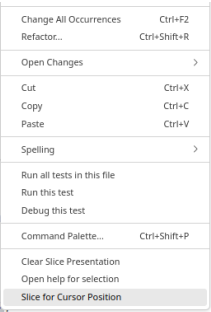


```
1 sum ← 0
2 product ← 1
3 w ← 7
4 N ← 10
5
6 for (i in 1:(N-1)) {
7   sum ← sum +
8   product ← p
9 }
10
11 cat("Sum:", sum,
12 cat("Product:", product, "\n")
13
```



A context menu is open over the code, showing various actions: Change All Occurrences (Ctrl+F2), Refactor... (Ctrl+Shift+R), Open Changes, Cut (Ctrl+X), Copy (Ctrl+C), Paste (Ctrl+V), Spelling, Run all tests in this file, Run this test, Debug this test, Command Palette... (Ctrl+Shift+P), Clear Slice Presentation, Open help for selection, and Slice for Cursor Position (highlighted).



```
1 sum ← 0
2 product ← 1
3 w ← 7
4 N ← 10
5
6 for (i in 1:(N-1)) {
7   sum ← sum + i + w
8   product ← product * i
9 }
10
11 cat("Sum:", sum, "\n")
12 cat("Product:", product, "\n")
13
```

flowR: A Program Slicer for the R Programming Language

deRSE '24 | Ulm University | Florian Sihler and Prof. Matthias Tichy | March 6, 2024

The R Programming Language

The R Programming Language

- R is mainly designed for statistical computing^[1]

[1] <https://cran.r-project.org/>

The R Programming Language

- R is mainly designed for statistical computing^[1]
 - Heavily used in research (e.g., social science)^[2]

[2] Trisovic et al., "A Large-Scale Study on Research Code Quality and Execution" (2022, Nature Publishing Group)

[1] <https://cran.r-project.org/>

The R Programming Language

- R is mainly designed for statistical computing^[1]
 - Heavily used in research (e.g., social science)^[2]
 - Ranks 6th on PYPL^[3]

[3] <https://pypl.github.io/> [archived]

[2] Trisovic et al., "A Large-Scale Study on Research Code Quality and Execution" (2022, Nature Publishing Group)

[1] <https://cran.r-project.org/>

The R Programming Language

- R is mainly designed for statistical computing^[1]
 - Heavily used in research (e.g., social science)^[2]
 - Ranks 6th on PYPL^[3]
 - Over 20 000 packages on CRAN^[1]

[3] <https://pypl.github.io/> [archived]

[2] Trisovic et al., "A Large-Scale Study on Research Code Quality and Execution" (2022, Nature Publishing Group)

[1] <https://cran.r-project.org/>

The R Programming Language

- R is mainly designed for statistical computing^[1]
 - Heavily used in research (e.g., social science)^[2]
 - Ranks 6th on PYPL^[3]
 - Over 20 000 packages on CRAN^[1]
- Most users are from non-computer-science domains

[3] <https://pypl.github.io/> [archived]

[2] Trisovic et al., "A Large-Scale Study on Research Code Quality and Execution" (2022, Nature Publishing Group)

[1] <https://cran.r-project.org/>

The R Programming Language

- R is mainly designed for statistical computing^[1]
 - Heavily used in research (e.g., social science)^[2]
 - Ranks 6th on PYPL^[3]
 - Over 20 000 packages on CRAN^[1]
- Most users are from non-computer-science domains
- Several problems in practice^[2, 4]

[4] Wonsil et al., “Reproducibility as a Service” (2023, Software: Practice and Experience)

[3] <https://pypl.github.io/> [archived]

[2] Trisovic et al., “A Large-Scale Study on Research Code Quality and Execution” (2022, Nature Publishing Group)

[1] <https://cran.r-project.org/>

The R Programming Language

- R is mainly designed for statistical computing^[1]
 - Heavily used in research (e.g., social science)^[2]
 - Ranks 6th on PYPL^[3]
 - Over 20 000 packages on CRAN^[1]
- Most users are from non-computer-science domains
- Several problems in practice^[2, 4]
 - Replication

[4] Wonsil et al., “Reproducibility as a Service” (2023, Software: Practice and Experience)

[3] <https://pypl.github.io/> [archived]

[2] Trisovic et al., “A Large-Scale Study on Research Code Quality and Execution” (2022, Nature Publishing Group)

[1] <https://cran.r-project.org/>

The R Programming Language

- R is mainly designed for statistical computing^[1]
 - Heavily used in research (e.g., social science)^[2]
 - Ranks 6th on PYPL^[3]
 - Over 20 000 packages on CRAN^[1]
- Most users are from non-computer-science domains
- Several problems in practice^[2, 4]
 - Replication
 - Program comprehension

[4] Wonsil et al., “Reproducibility as a Service” (2023, Software: Practice and Experience)

[3] <https://pypl.github.io/> [archived]

[2] Trisovic et al., “A Large-Scale Study on Research Code Quality and Execution” (2022, Nature Publishing Group)

[1] <https://cran.r-project.org/>

The R Programming Language

- R is mainly designed for statistical computing^[1]
 - Heavily used in research (e.g., social science)^[2]
 - Ranks 6th on PYPL^[3]
 - Over 20 000 packages on CRAN^[1]
- Most users are from non-computer-science domains
- Several problems in practice^[2, 4]
 - Replication
 - Program comprehension
 - Missing tool support

[4] Wonsil et al., “Reproducibility as a Service” (2023, Software: Practice and Experience)

[3] <https://pypl.github.io/> [archived]

[2] Trisovic et al., “A Large-Scale Study on Research Code Quality and Execution” (2022, Nature Publishing Group)

[1] <https://cran.r-project.org/>

The R Programming Language

- R is mainly designed for statistical computing^[1]
 - Heavily used in research (e.g., social science)^[2]
 - Ranks 6th on PYPL^[3]
 - Over 20 000 packages on CRAN^[1]
- Most users are from non-computer-science domains
- Several problems in practice^[2, 4]
 - Replication
 - Program comprehension
 - Missing tool support
- Demand for software engineering practices^[5]

[5] Thimbleby, “Improving Science That Uses Code” (2023, Oxford University Press)

[4] Wonsil et al., “Reproducibility as a Service” (2023, Software: Practice and Experience)

[3] <https://pypl.github.io/> [archived]

[2] Trisovic et al., “A Large-Scale Study on Research Code Quality and Execution” (2022, Nature Publishing Group)

[1] <https://cran.r-project.org/>

R Scripts

We analyzed 4 083 R-files^[7]

[7] Sihler et al., "On the Anatomy of Real-World R Code for Static Analysis" (2024, MSR)

R Scripts

```
## R script for data analysis
## Author: F. Sihler
## Date: 2024-01-01
## Description: This script performs a series of data analysis steps, including data loading, cleaning, and visualization.

# Load the data
data <- read.csv("data.csv")

# Clean the data
data <- na.omit(data)
data <- data[, c("Year", "Phenology")]

# Visualize the data
plot(data$Year, data$Phenology, main="Phenology vs Year", xlab="Year", ylab="Phenology")
```

We analyzed 4 083 R-files^[7]

[7] Sihler et al., "On the Anatomy of Real-World R Code for Static Analysis" (2024, MSR)

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

R Scripts

We analyzed 4 083 R-files^[7]



```
# set the data directory and load workspace
setwd("G:/Shared drives/Fenologija/Raksti/abeles/Zenodo/")
load("Apple_Pure_phenology_R_workspace_image.RData")

# -----
# script to recreate components included in of "Apple_Pure_phenology_R_workspace_image.RData"
# -----
# phenology data subset having at least 14 observations overlapping with meteorology data set
{
  dM <- d %>%
    filter(Year %in% (meteoH$e_obs$Year))
  dM <- dM %>%
    group_by(Variety) %>%
    summarize(N = n()) %>%
    arrange(N) %>%
    filter(N >= 14) %>% # 12 skirnes
    select(-N) %>%
    left_join(dM)
}

# -----
# FUNCTION definitions
# phenology model defined as function, see Kalvans et al, 2015, DDcos model for details
```

[7] Sihler et al., "On the Anatomy of Real-World R Code for Static Analysis" (2024, MSR)

[6] 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)

1. **Identify the problem.** The problem is that the company is not meeting its sales targets.

2. **Analyze the problem.** The company is not meeting its sales targets because it is not effectively marketing its products.

3. **Develop a solution.** The company should develop a new marketing strategy that focuses on reaching its target audience.

4. **Implement the solution.** The company should implement the new marketing strategy by launching a series of targeted advertising campaigns.

5. **Evaluate the results.** The company should evaluate the results of the new marketing strategy by tracking sales and customer feedback.

[illegible]

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)

R Scripts Fail to Replicate

```
## 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)
##
## This script is a part of the R package 'flowR' (https://github.com/fahsler/flowR)
##
## The script is designed to be run from the command line using the following command:
## Rscript Apple_Pure_phenology_R_workspace_image.RData
##
## The script will create a directory named 'Apple_Pure_phenology_R_workspace_image' in the current working directory.
## It will then download the data from the Zenodo repository and load it into R.
## The data is a subset of the full data set, containing only the data for the years 1959 to 2019.
## The data is then processed and saved as a new RData file.
##
## The script is designed to be run from the command line using the following command:
## Rscript Apple_Pure_phenology_R_workspace_image.RData
##
## The script will create a directory named 'Apple_Pure_phenology_R_workspace_image' in the current working directory.
## It will then download the data from the Zenodo repository and load it into R.
## The data is a subset of the full data set, containing only the data for the years 1959 to 2019.
## The data is then processed and saved as a new RData file.
##
## The script is designed to be run from the command line using the following command:
## Rscript Apple_Pure_phenology_R_workspace_image.RData
##
## The script will create a directory named 'Apple_Pure_phenology_R_workspace_image' in the current working directory.
## It will then download the data from the Zenodo repository and load it into R.
## The data is a subset of the full data set, containing only the data for the years 1959 to 2019.
## The data is then processed and saved as a new RData file.
```

```
# set the data directory and load workspace
setwd("G:/Shared drives/Fenologija/Raksti/abeles/Zenodo/")
load("Apple_Pure_phenology_R_workspace_image.RData")
```

Hardcoded Paths

```
# script to recreate components included in of "Apple_Pure_phenology_R_workspace_image.RData"
{
# -----
# phenology data subset having at least 14 observations overlapping with meteorology data set
{
  dM <- d %>%
    filter(Year %in% (meteoH$e_obs$Year))
  dM <- dM %>%
    group_by(Variety) %>%
    summarize(N = n()) %>%
    arrange(N) %>%
    filter(N >= 14) %>% # 12 skirnes
    select(-N) %>%
    left_join(dM)
}

# -----
# FUNCTION definitions
# phenology model defined as function, see Kalvans et al, 2015, DDcos model for details
```

[6] 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)

1. **Identify the problem.** The problem is that the company is not meeting its sales targets.

2. **Analyze the problem.** The problem is caused by a combination of factors, including a weak marketing strategy, poor timing of the product launch, and a lack of competitive pricing.

3. **Develop a solution.** The solution is to develop a new marketing strategy, improve the timing of the product launch, and implement competitive pricing.

4. **Implement the solution.** The solution is implemented by developing a new marketing strategy, improving the timing of the product launch, and implementing competitive pricing.

5. **Evaluate the solution.** The solution is evaluated by monitoring sales performance and customer feedback.

α β γ δ ϵ ζ η θ ι κ λ μ ν ξ \omicron π ρ σ τ υ ϕ χ ψ ω
 \sin \cos \tan \cot \sec \csc \sinh \cosh \tanh \coth \exp \ln \log \exp \ln \log
 $\frac{1}{2}$ $\frac{1}{3}$ $\frac{1}{4}$ $\frac{1}{5}$ $\frac{1}{6}$ $\frac{1}{7}$ $\frac{1}{8}$ $\frac{1}{9}$ $\frac{1}{10}$ $\frac{1}{11}$ $\frac{1}{12}$ $\frac{1}{13}$ $\frac{1}{14}$ $\frac{1}{15}$ $\frac{1}{16}$ $\frac{1}{17}$ $\frac{1}{18}$ $\frac{1}{19}$ $\frac{1}{20}$
 $\frac{1}{21}$ $\frac{1}{22}$ $\frac{1}{23}$ $\frac{1}{24}$ $\frac{1}{25}$ $\frac{1}{26}$ $\frac{1}{27}$ $\frac{1}{28}$ $\frac{1}{29}$ $\frac{1}{30}$ $\frac{1}{31}$ $\frac{1}{32}$ $\frac{1}{33}$ $\frac{1}{34}$ $\frac{1}{35}$ $\frac{1}{36}$ $\frac{1}{37}$ $\frac{1}{38}$ $\frac{1}{39}$ $\frac{1}{40}$
 $\frac{1}{41}$ $\frac{1}{42}$ $\frac{1}{43}$ $\frac{1}{44}$ $\frac{1}{45}$ $\frac{1}{46}$ $\frac{1}{47}$ $\frac{1}{48}$ $\frac{1}{49}$ $\frac{1}{50}$ $\frac{1}{51}$ $\frac{1}{52}$ $\frac{1}{53}$ $\frac{1}{54}$ $\frac{1}{55}$ $\frac{1}{56}$ $\frac{1}{57}$ $\frac{1}{58}$ $\frac{1}{59}$ $\frac{1}{60}$
 $\frac{1}{61}$ $\frac{1}{62}$ $\frac{1}{63}$ $\frac{1}{64}$ $\frac{1}{65}$ $\frac{1}{66}$ $\frac{1}{67}$ $\frac{1}{68}$ $\frac{1}{69}$ $\frac{1}{70}$ $\frac{1}{71}$ $\frac{1}{72}$ $\frac{1}{73}$ $\frac{1}{74}$ $\frac{1}{75}$ $\frac{1}{76}$ $\frac{1}{77}$ $\frac{1}{78}$ $\frac{1}{79}$ $\frac{1}{80}$
 $\frac{1}{81}$ $\frac{1}{82}$ $\frac{1}{83}$ $\frac{1}{84}$ $\frac{1}{85}$ $\frac{1}{86}$ $\frac{1}{87}$ $\frac{1}{88}$ $\frac{1}{89}$ $\frac{1}{90}$ $\frac{1}{91}$ $\frac{1}{92}$ $\frac{1}{93}$ $\frac{1}{94}$ $\frac{1}{95}$ $\frac{1}{96}$ $\frac{1}{97}$ $\frac{1}{98}$ $\frac{1}{99}$ $\frac{1}{100}$

1. **Identify the problem**
 2. **Define the problem**
 3. **Generate hypotheses**
 4. **Test the hypotheses**
 5. **Implement the solution**
 6. **Evaluate the solution**

1. **Introduction**

2. **Background**

3. **Methodology**

4. **Results**

5. **Discussion**

6. **Conclusion**

7. **References**

8. **Appendix**

9. **Figure 1**

10. **Figure 2**

11. **Figure 3**

12. **Figure 4**

13. **Figure 5**

14. **Figure 6**

15. **Figure 7**

16. **Figure 8**

17. **Figure 9**

18. **Figure 10**

19. **Figure 11**

20. **Figure 12**

21. **Figure 13**

22. **Figure 14**

23. **Figure 15**

24. **Figure 16**

25. **Figure 17**

26. **Figure 18**

27. **Figure 19**

28. **Figure 20**

29. **Figure 21**

30. **Figure 22**

31. **Figure 23**

32. **Figure 24**

33. **Figure 25**

34. **Figure 26**

35. **Figure 27**

36. **Figure 28**

37. **Figure 29**

38. **Figure 30**

39. **Figure 31**

40. **Figure 32**

41. **Figure 33**

42. **Figure 34**

43. **Figure 35**

44. **Figure 36**

45. **Figure 37**

46. **Figure 38**

47. **Figure 39**

48. **Figure 40**

49. **Figure 41**

50. **Figure 42**

51. **Figure 43**

52. **Figure 44**

53. **Figure 45**

54. **Figure 46**

55. **Figure 47**

56. **Figure 48**

57. **Figure 49**

58. **Figure 50**

59. **Figure 51**

60. **Figure 52**

61. **Figure 53**

62. **Figure 54**

63. **Figure 55**

64. **Figure 56**

65. **Figure 57**

66. **Figure 58**

67. **Figure 59**

68. **Figure 60**

69. **Figure 61**

70. **Figure 62**

71. **Figure 63**

72. **Figure 64**

73. **Figure 65**

74. **Figure 66**

75. **Figure 67**

76. **Figure 68**

77. **Figure 69**

78. **Figure 70**

79. **Figure 71**

80. **Figure 72**

81. **Figure 73**

82. **Figure 74**

83. **Figure 75**

84. **Figure 76**

85. **Figure 77**

86. **Figure 78**

87. **Figure 79**

88. **Figure 80**

89. **Figure 81**

90. **Figure 82**

91. **Figure 83**

92. **Figure 84**

93. **Figure 85**

94. **Figure 86**

95. **Figure 87**

96. **Figure 88**

97. **Figure 89**

98. **Figure 90**

99. **Figure 91**

100. **Figure 92**

101. **Figure 93**

102. **Figure 94**

103. **Figure 95**

104. **Figure 96**

105. **Figure 97**

106. **Figure 98**

107. **Figure 99**

108. **Figure 100**

109. **Figure 101**

110. **Figure 102**

111. **Figure 103**

112. **Figure 104**

113. **Figure 105**

114. **Figure 106**

115. **Figure 107**

116. **Figure 108**

117. **Figure 109**

118. **Figure 110**

119. **Figure 111**

120. **Figure 112**

121. **Figure 113**

122. **Figure 114**

123. **Figure 115**

124. **Figure 116**

125. **Figure 117**

126. **Figure 118**

127. **Figure 119**

128. **Figure 120**

129. **Figure 121**

130. **Figure 122**

131. **Figure 123**

132. **Figure 124**

133. **Figure 125**

134. **Figure 126**

135. **Figure 127**

136. **Figure 128**

137. **Figure 129**

138. **Figure 130**

139. **Figure 131**

140. **Figure 132**

141. **Figure 133**

142. **Figure 134**

143. **Figure 135**

144. **Figure 136**

145. **Figure 137**

146. **Figure 138**

147. **Figure 139**

148. **Figure 140**

149. **Figure 141**

150. **Figure 142**

151. **Figure 143**

152. **Figure 144**

153. **Figure 145**

154. **Figure 146**

155. **Figure 147**

156. **Figure 148**

157. **Figure 149**

158. **Figure 150**

159. **Figure 151**

160. **Figure 152**

161. **Figure 153**

162. **Figure 154**

163. **Figure 155**

164. **Figure 156**

165. **Figure 157**

166. **Figure 158**

167. **Figure 159**

168. **Figure 160**

169. **Figure 161**

170. **Figure 162**

171. **Figure 163**

172. **Figure 164**

173. **Figure 165**

174. **Figure 166**

175. **Figure 167**

176. **Figure 168**

177. **Figure 169**

178. **Figure 170**

179. **Figure 171**

180. **Figure 172**

181. **Figure 173**

182. **Figure 174**

183. **Figure 175**

184. **Figure 176**

185. **Figure 177**

186. **Figure 178**

187. **Figure 179**

188. **Figure 180**

189. **Figure 181**

190. **Figure 182**

191. **Figure 183**

192. **Figure 184**

193. **Figure 185**

194. **Figure 186**

195. **Figure 187**

196. **Figure 188**

197. **Figure 189**

198. **Figure 190**

199. **Figure 191**

200. **Figure 192**

201. **Figure 193**

202. **Figure 194**

203. **Figure 195**

204. **Figure 196**

205. **Figure 197**

206. **Figure 198**

207. **Figure 199**

208. **Figure 200**

209. **Figure 201**

210. **Figure 202**

211. **Figure 203**

212. **Figure 204**

213. **Figure 205**

214. **Figure 206**

215. **Figure 207**

216. **Figure 208**

217. **Figure 209**

218. **Figure 210**

219. **Figure 211**

220. **Figure 212**

221. **Figure 213**

222. **Figure 214**

223. **Figure 215**

224. **Figure 216**

225. **Figure 217**

226. **Figure 218**

227. **Figure 219**

228. **Figure 220**

229. **Figure**

```
# ----- Hardcoded Paths -----
# script to recreate components included in of "Apple_Pure_phenology_R_workspace_image.RData"
{
# -----
# phenology data subset having at least 14 observations overlapping with meteorology data set
```

```
# loweri - lower bound of model parameter range (Tb, Dd)
# upperi - upper bound of model parameter range (Tb, Dd)
```

```
print("DDSinOpt")
a ← GenSA::GenSA(par = pari,
                 fn = DDSin_stat,
                 lower = loweri,
                 upper = upperi,
                 stats = c("RMSE"),
                 sadaliiums = sadaliiums.)
```

F. Sihler (Ulm University) *flowR* – Problems 4.3

R Scripts Fail to Replicate

The diagram illustrates the problem of replicating R scripts. It features a vertical column of R code snippets on the left, with two callout boxes on the right. The top callout box, titled "Hardcoded Paths", points to a snippet of code that sets a directory path and loads an RData file. The bottom callout box, titled "(Pseudo-)Randomness without Seed", points to a snippet of code that generates random numbers without a seed, which is a common cause of non-replicable results.

```
# set the data directory and load workspace
setwd("G:/Shared drives/Fenologija/Raksti/abeles/Zenodo/")
load("Apple_Pure_phenology_R_workspace_image.RData")

# ----- Hardcoded Paths -----
# scrip to recreate components included in of "Apple_Pure_phenology_R_workspace_image.RData"
{
# -----
# phenology data subset having at least 14 observations cwerlapping wiht meteorology data set
# loweri - lower bound of model parameter range (Tb, DD)
# upperi - upper bound of model parameter range (Tb, DD)

# if pari == NA a random set is generated
if (any(is.na(pari))) {
  pari = runif(2) * (upperi - loweri) + loweri
}

# test if the phenology and meteorology data overlap in time
if (any(dd$Year %in% meteod$Year)) {
  print(Sys.time())
}

print("DDsinOpt")
a <- GenSA::GenSA(par = pari,
  fn = DDsin_stat,
  lower = loweri,
  upper = upperi,
  stats = c("RMSE"),
  sadalijums = sadalijums,
```

[6] 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)

R Scripts Fail to Replicate

74 % even fail to complete!^[2]



[2] Trisovic et al., "A Large-Scale Study on Research Code Quality and Execution" (2022, Nature Publishing Group)

[6] 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)

R Scripts Do Too Much



[6] 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)

R Scripts Do Too Much

```
## Load packages
library(tidyverse)
library(lme4)
library(MuMIn)
library(ggplot2)
library(cowplot)

## Load data
data <- read_csv("data/apple_phenology.csv")

## Data cleaning
data <- data %>%
  filter(!is.na(year)) %>%
  group_by(year) %>%
  mutate(
    day_of_year = dayofyear(date),
    day_of_year_lag1 = lag(day_of_year, 1),
    day_of_year_lag2 = lag(day_of_year, 2)
  ) %>%
  ungroup()

## Model fitting
model <- glmer(
  day_of_year ~ (1 | year) + (1 | orchard),
  data = data,
  REML = TRUE
)

## Model selection
AICc <- AICcmodexp(model)
AICc <- AICc[order(AICc), ]

## Figure 1: AICc values
ggplot(AICc, aes(x = rank(AICc), y = AICc)) +
  geom_line() +
  geom_point() +
  theme_minimal()

## Figure 2: AICc values
ggplot(AICc, aes(x = rank(AICc), y = AICc)) +
  geom_line() +
  geom_point() +
  theme_minimal()

## Figure 3: AICc values
ggplot(AICc, aes(x = rank(AICc), y = AICc)) +
  geom_line() +
  geom_point() +
  theme_minimal()
```

Model

Model

Figure

Figure

Figure

- Several analyses in one script

R Scripts Do Too Much

```
## Load packages
library(tidyverse)
library(lme4)
library(MuMIn)
library(ggplot2)
library(cowplot)

## Load data
data <- read_csv("data/apple_phenology.csv")

## Data cleaning
data <- data %>%
  filter(!is.na(year)) %>%
  group_by(year) %>%
  mutate(
    day_of_year = dayofyear(date),
    day_of_year_lag1 = lag(day_of_year, 1),
    day_of_year_lag2 = lag(day_of_year, 2)
  ) %>%
  ungroup()

## Model fitting
mod1 <- lmer(day_of_year ~ 1, data = data)
mod2 <- lmer(day_of_year ~ year, data = data)
mod3 <- lmer(day_of_year ~ day_of_year_lag1, data = data)
mod4 <- lmer(day_of_year ~ day_of_year_lag2, data = data)

## Model selection
AICc <- AICcmodexp(mod1, mod2, mod3, mod4)
AICc <- AICc[order(AICc)]

## Figure 1: AICc values
ggplot(AICc, aes(x = model, y = AICc)) +
  geom_bar() +
  theme_minimal()

## Figure 2: AICc values
ggplot(AICc, aes(x = model, y = AICc)) +
  geom_bar() +
  theme_minimal()

## Figure 3: AICc values
ggplot(AICc, aes(x = model, y = AICc)) +
  geom_bar() +
  theme_minimal()

## Figure 4: AICc values
ggplot(AICc, aes(x = model, y = AICc)) +
  geom_bar() +
  theme_minimal()
```

Model

Model

Figure

Figure

Figure

- Several analyses in one script
- Hard to comprehend

R Scripts Do Too Much



Model

Model

Figure

Figure

Figure

- Several analyses in one script
- Hard to comprehend
- Hard to extract/re-use parts

[6] 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)

R Scripts Are Hard to Analyze

R Scripts Are Hard to Analyze

```
[8] mod.proj_wc <- ecospat.ESM.Projection(ESM.modeling.output=mod,  
    new.env=eval(parse(text = paste("env_",period,"_wc",sep=""))))
```

[8] Ma et al., *Predicting range shifts of pikas (Mammalia, Ochotonidae) in China under scenarios incorporating land-use change, climate change, and dispersal limitations* (2021, Zenodo) [L. 135f]

R Scripts Are Hard to Analyze

```
[8] mod.proj_wc <- ecospat.ESM.Projection(ESM.modeling.output=mod,  
    new.env=eval(parse(text = paste("env_",period,"_wc",sep=""))))
```

- ▶ String-based code evaluation

R Scripts Are Hard to Analyze

```
[8] mod.proj_wc <- ecospat.ESM.Projection(ESM.modeling.output=mod,  
    new.env=eval(parse(text = paste("env_",period,"_wc",sep=""))))
```

► String-based code evaluation

```
[9] pull.cat <- function(x) {  
  bins <- 5  
  increments <- (range(x)[2] - range(x)[1])/(bins - 1)  
  to_return <- seq(range(x)[1], range(x)[2], increments)  
  return(to_return)  
}  
  
up.cat <- function(new_bins) {  
  up_bins = new_bins  
  body(pull.cat)[[2]] <- substitute(bins <- up_bins)  
}
```

[9] Robertson, *Social hierarchy reveals thermoregulatory trade-offs in response to repeated stressors* (2020, Zenodo) [L. 68ff]

[8] Ma et al., *Predicting range shifts of pikas (Mammalia, Ochotonidae) in China under scenarios incorporating land-use change, climate change, and dispersal limitations* (2021, Zenodo) [L. 135f]

R Scripts Are Hard to Analyze

```
[8] mod.proj_wc ← ecospat.ESM.Projection(ESM.modeling.output=mod,  
    new.env=eval(parse(text = paste("env_",period,"_wc",sep=""))))
```

► String-based code evaluation

```
[9] pull.cat ← function(x) {  
  bins ← 5  
  increments ← (range(x)[2] - range(x)[1])/(bins - 1)  
  to_return ← seq(range(x)[1], range(x)[2], increments)  
  return(to_return)  
}  
  
up.cat ← function(new_bins) {  
  up_bins = new_bins  
  body(pull.cat)[[2]] ← substitute(bins ← up_bins)  
}
```

► Self-modifying code

[9] Robertson, *Social hierarchy reveals thermoregulatory trade-offs in response to repeated stressors* (2020, Zenodo) [L. 68ff]

[8] Ma et al., *Predicting range shifts of pikas (Mammalia, Ochotonidae) in China under scenarios incorporating land-use change, climate change, and dispersal limitations* (2021, Zenodo) [L. 135f]

R Scripts Are Hard to Analyze

```
[8] mod.proj_wc <- ecospat.ESM.Projection(ESM.modeling.output=mod,  
    new.env=eval(parse(text = paste("env_",period,"_wc",sep=""))))
```

► String-based code evaluation

```
[9] pull.cat <- function(x) {  
  bins <- up_bins # (e.g., 6)  
  increments <- (range(x)[2] - range(x)[1])/(bins - 1)  
  to_return <- seq(range(x)[1], range(x)[2], increments)  
  return(to_return)  
}  
  
up.cat <- function(new_bins) {  
  up_bins = new_bins  
  body(pull.cat)[[2]] <- substitute(bins <- up_bins)  
}
```

► Self-modifying code

[9] Robertson, *Social hierarchy reveals thermoregulatory trade-offs in response to repeated stressors* (2020, Zenodo) [L. 68ff]

[8] Ma et al., *Predicting range shifts of pikas (Mammalia, Ochotonidae) in China under scenarios incorporating land-use change, climate change, and dispersal limitations* (2021, Zenodo) [L. 135f]

R Misses Sophisticated Analysis Tools

R Misses Sophisticated Analysis Tools

- RStudio IDE posit.co

R Misses Sophisticated Analysis Tools

- RStudio IDE posit.co
 - Syntax-highlighting and auto-completion
 - Refactorings (rename, extract functions and variables)

R Misses Sophisticated Analysis Tools

- RStudio IDE posit.co
 - Syntax-highlighting and auto-completion
 - Refactorings (rename, extract functions and variables)
- R language server github.com/REditorSupport

R Misses Sophisticated Analysis Tools

- RStudio IDE posit.co
 - Syntax-highlighting and auto-completion
 - Refactorings (rename, extract functions and variables)
- R language server github.com/REditorSupport
 - Syntax-highlighting and auto-completion
 - Reference tracing & Refactorings (rename)

R Misses Sophisticated Analysis Tools

- RStudio IDE posit.co
 - Syntax-highlighting and auto-completion
 - Refactorings (rename, extract functions and variables)
- R language server github.com/REditorSupport
 - Syntax-highlighting and auto-completion
 - Reference tracing & Refactorings (rename)
- {lintr} github.com/r-lib/lintr

R Misses Sophisticated Analysis Tools

- RStudio IDE posit.co
 - Syntax-highlighting and auto-completion
 - Refactorings (rename, extract functions and variables)
- R language server github.com/REditorSupport
 - Syntax-highlighting and auto-completion
 - Reference tracing & Refactorings (rename)
- {lintr} github.com/r-lib/lintr
 - Style & syntax errors
 - Potential semantic errors

R Misses Sophisticated Analysis Tools

- RStudio IDE posit.co
 - Syntax-highlighting and auto-completion
 - Refactorings (rename, extract functions and variables)
- R language server github.com/REditorSupport
 - Syntax-highlighting and auto-completion
 - Reference tracing & Refactorings (rename)
- {lintr} github.com/r-lib/lintr
 - Style & syntax errors
 - Potential semantic errors
- {CodeDepends} github.com/duncantl/CodeDepends

R Misses Sophisticated Analysis Tools

- RStudio IDE posit.co
 - Syntax-highlighting and auto-completion
 - Refactorings (rename, extract functions and variables)
- R language server github.com/REditorSupport
 - Syntax-highlighting and auto-completion
 - Reference tracing & Refactorings (rename)
- {lintr} github.com/r-lib/lintr
 - Style & syntax errors
 - Potential semantic errors
- {CodeDepends} github.com/duncantl/CodeDepends
 - Dependency analysis
 - Creation of call-graphs

R Misses Sophisticated Analysis Tools

- RStudio IDE posit.co
 - Syntax-highlighting and auto-completion
 - Refactorings (rename, extract functions and variables) ← Often wrong (simple heuristics)
- R language server github.com/REditorSupport
 - Syntax-highlighting and auto-completion
 - Reference tracing & Refactorings (rename)
- {lintr} github.com/r-lib/lintr
 - Style & syntax errors
 - Potential semantic errors
- {CodeDepends} github.com/duncantl/CodeDepends
 - Dependency analysis
 - Creation of call-graphs

R Misses Sophisticated Analysis Tools

- RStudio IDE posit.co
 - Syntax-highlighting and auto-completion
 - Refactorings (rename, extract functions and variables) ← Often wrong (simple heuristics)
- R language server github.com/REditorSupport
 - Syntax-highlighting and auto-completion
 - Reference tracing & Refactorings (rename) ← Often wrong (XPath-Expressions)
- {lintr} github.com/r-lib/lintr
 - Style & syntax errors
 - Potential semantic errors
- {CodeDepends} github.com/duncantl/CodeDepends
 - Dependency analysis
 - Creation of call-graphs

R Misses Sophisticated Analysis Tools

- RStudio IDE posit.co
 - Syntax-highlighting and auto-completion
 - Refactorings (rename, extract functions and variables) ← Often wrong (simple heuristics)
- R language server github.com/REditorSupport
 - Syntax-highlighting and auto-completion
 - Reference tracing & Refactorings (rename) ← Often wrong (XPath-Expressions)
- {lintr} github.com/r-lib/lintr
 - Style & syntax errors
 - Potential semantic errors
 - ↑
XPath-Expressions, packages
- {CodeDepends} github.com/duncantl/CodeDepends
 - Dependency analysis
 - Creation of call-graphs

R Misses Sophisticated Analysis Tools

- RStudio IDE posit.co
 - Syntax-highlighting and auto-completion
 - Refactorings (rename, extract functions and variables) ← Often wrong (simple heuristics)
- R language server github.com/REditorSupport
 - Syntax-highlighting and auto-completion
 - Reference tracing & Refactorings (rename) ← Often wrong (XPath-Expressions)
- {lintr} github.com/r-lib/lintr
 - Style & syntax errors
 - Potential semantic errors
 - ↑
XPath-Expressions, packages
- {CodeDepends} github.com/duncantl/CodeDepends
 - Dependency analysis ← Only top scope
 - Creation of call-graphs

R Scripts ...

R Scripts ...

1. fail to replicate

R Scripts ...

1. fail to replicate
2. do too much

R Scripts ...

1. fail to replicate
2. do too much
3. are hard to analyze

R Scripts ...

1. fail to replicate
2. do too much
3. are hard to analyze
4. are not well supported by tools

R Scripts ...

1. fail to replicate
2. do too much
3. are hard to analyze
4. are not well supported by tools

Better Software, Better Research

R Scripts ...

1. fail to replicate
2. **do too much**
3. are hard to analyze
4. are not well supported by tools

Better Software, Better Research

The Goal of flowR



[6] 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 Goal of flowR



- Interested in a single figure

[6] 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 Goal of flowR



The diagram illustrates the proposed system architecture. It features a central 'Proposed System' block, which is connected to 'User' and 'System' blocks. The 'Proposed System' block is further divided into 'User' and 'System' sub-blocks. The 'User' sub-block contains 'User' and 'System' components. The 'System' sub-block contains 'User' and 'System' components. The 'Proposed System' block is also connected to 'User' and 'System' blocks. The 'Proposed System' block is further divided into 'User' and 'System' sub-blocks. The 'User' sub-block contains 'User' and 'System' components. The 'System' sub-block contains 'User' and 'System' components. The 'Proposed System' block is also connected to 'User' and 'System' blocks. The 'Proposed System' block is further divided into 'User' and 'System' sub-blocks. The 'User' sub-block contains 'User' and 'System' components. The 'System' sub-block contains 'User' and 'System' components.

- Interested in a single figure

[6] 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 Goal of flowR

[illegible]

- Interested in a single figure
- $\approx 70\%$ reduction

[6] 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 Architecture



[10] Sihler, "Constructing a static program slicer for R programs" (2023, Ulm University)

[11] Weiser, "Program Slicing" (1984, IEEE Transactions on Software Engineering)

The Architecture

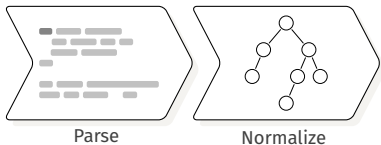


Parse

[10] Sihler, "Constructing a static program slicer for R programs" (2023, Ulm University)

[11] Weiser, "Program Slicing" (1984, IEEE Transactions on Software Engineering)

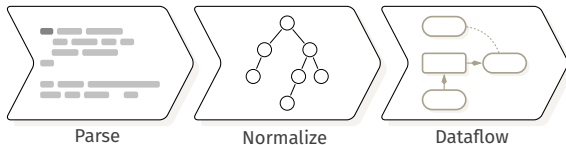
The Architecture



[10] Sihler, "Constructing a static program slicer for R programs" (2023, Ulm University)

[11] Weiser, "Program Slicing" (1984, IEEE Transactions on Software Engineering)

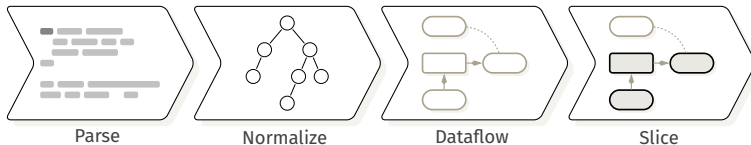
The Architecture



[10] Sihler, "Constructing a static program slicer for R programs" (2023, Ulm University)

[11] Weiser, "Program Slicing" (1984, IEEE Transactions on Software Engineering)

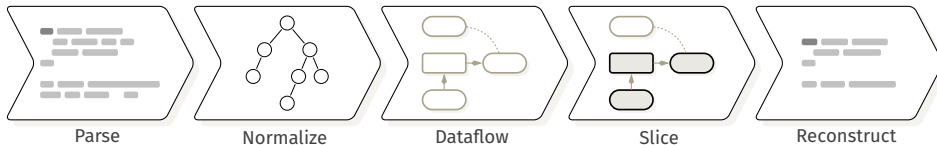
The Architecture



[10] Sihler, "Constructing a static program slicer for R programs" (2023, Ulm University)

[11] Weiser, "Program Slicing" (1984, IEEE Transactions on Software Engineering)

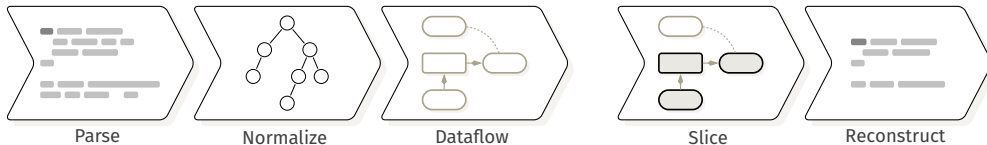
The Architecture



[10] Sihler, "Constructing a static program slicer for R programs" (2023, Ulm University)

[11] Weiser, "Program Slicing" (1984, IEEE Transactions on Software Engineering)

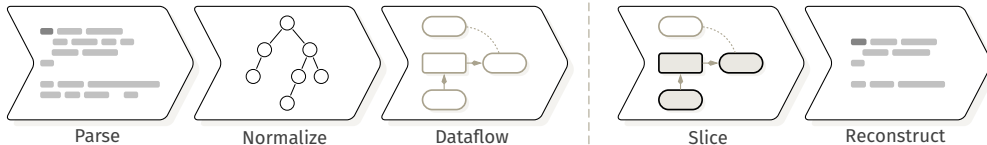
The Architecture



[10] Sihler, "Constructing a static program slicer for R programs" (2023, Ulm University)

[11] Weiser, "Program Slicing" (1984, IEEE Transactions on Software Engineering)

The Architecture




[10] Sihler, "Constructing a static program slicer for R programs" (2023, Ulm University)

[11] Weiser, "Program Slicing" (1984, IEEE Transactions on Software Engineering)

Visual Studio Code Integration


Visual Studio Code Integration

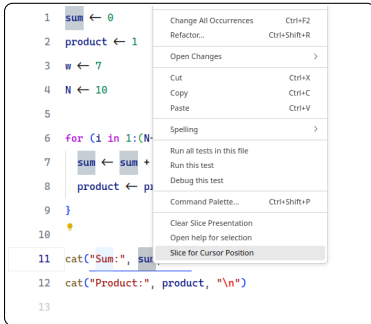
Rudimentary VSCode integration at:

 github.com/Code-Inspect/vscode-flowr

Visual Studio Code Integration

Rudimentary VSCode integration at:

 github.com/Code-Inspect/vscode-flowr



The screenshot shows a Visual Studio Code editor window with a code file. The code is as follows:


```
1 sum ← 0
2 product ← 1
3 w ← 7
4 N ← 10
5
6 for (i in 1:(N-1))
7   sum ← sum + w
8   product ← product * w
9 }
10
11 cat("Sum:", sum)
12 cat("Product:", product, "\n")
13
```

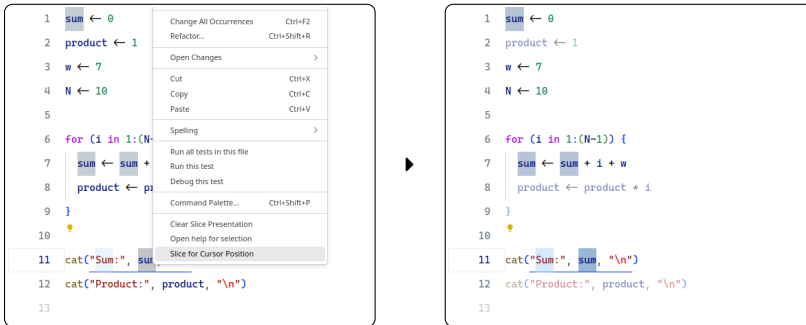
A context menu is open over the selection `sum` on line 11. The menu items are:

- Change All Occurrences Ctrl+F2
- Refactor... Ctrl+Shift+R
- Open Changes >
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Spelling >
- Run all tests in this file
- Run this test
- Debug this test
- Command Palette... Ctrl+Shift+P
- Clear Slice Presentation
- Open help for selection
- Slice for Cursor Position





Visual Studio Code Integration

Rudimentary VSCode integration at:

 github.com/Code-Inspect/vscode-flowr







Using flowR

-  github.com/Code-Inspect/flowr
-  github.com/Code-Inspect/vscode-flowr
-  hub.docker.com/r/eagleoutice/flowr
-  npmjs.com/package/@eagleoutice/flowr





Using flowR

Server

```
{  
  "type": "request-file-analysis",  
  "id": "1",  
  "filetoken": "123",  
  "content": "x ← 1; ⊥x*y"  
}  
  
{  
  "type": "request-slice",  
  "id": "2",  
  "filetoken": "123",  
  "criterion": ["1@x"]  
}
```

-  github.com/Code-Inspect/flowr
-  github.com/Code-Inspect/vscode-flowr
-  hub.docker.com/r/eagleoutice/flowr
-  npmjs.com/package/@eagleoutice/flowr

Using flowR

-  github.com/Code-Inspect/flowr
-  github.com/Code-Inspect/vscode-flowr
-  hub.docker.com/r/eagleoutice/flowr
-  npmjs.com/package/@eagleoutice/flowr

Server





```
{
  "type": "request-file-analysis",
  "id": "1",
  "filetoken": "123",
  "content": "x ← 1; x * y"
}

{
  "type": "request-slice",
  "id": "2",
  "filetoken": "123",
  "criterion": ["1@x"]
}
```

Library

```
const s = new SteppingSlicer({
  shell, tokenMap,
  request:
    requestFromInput("x ← 1; x * y"),
  criterion: ['1@x'],
})
const slice =
  await s.allRemainingSteps()
```

Using flowR

-  github.com/Code-Inspect/flowr
-  github.com/Code-Inspect/vscode-flowr
-  hub.docker.com/r/eagleoutice/flowr
-  npmjs.com/package/@eagleoutice/flowr

Server

```
{
  "type": "request-file-analysis",
  "id": "1",
  "filetoken": "123",
  "content": "x ← 1; x * y"
}

{
  "type": "request-slice",
  "id": "2",
  "filetoken": "123",
  "criterion": ["1@x"]
}
```

Library





```
const s = new SteppingSlicer({
  shell, tokenMap,
  request:
    requestFromInput("x ← 1; x * y"),
  criterion: ['1@x'],
})
const slice =
  await s.allRemainingSteps()
```

REPL

```
R> :parse "x ← 1; x * y"
exprlist
└─ expr
  │ └─ expr
  │ │ └─ SYMBOL "x" (1:1)
  [...]

R> :dataflow* "x ← 1; x * y"
https://mermaid.live/edit#base64:eyJj...
```

Using flowR

-  github.com/Code-Inspect/flowr
-  github.com/Code-Inspect/vscode-flowr
-  hub.docker.com/r/eagleoutice/flowr
-  npmjs.com/package/@eagleoutice/flowr

Server

```
{
  "type": "request-file-analysis",
  "id": "1",
  "filetoken": "123",
  "content": "x ← 1; x * y"
}

{
  "type": "request-slice",
  "id": "2",
  "filetoken": "123",
  "criterion": ["1@x"]
}
```

Library

```
const s = new SteppingSlicer({
  shell, tokenMap,
  request:
    requestFromInput("x ← 1; x * y"),
  criterion: ["1@x"],
})
const slice =
  await s.allRemainingSteps()
```

REPL

```
R> :parse "x ← 1; x * y"
exprlist
└─ expr
  │ └─ expr
  │ │ └─ SYMBOL "x" (1:1)
  [...]

R> :dataflow* "x ← 1; x * y"
https://mermaid.live/edit#base64:eyJj...
```

```
docker run -it --rm eagleoutice/flowr
```


Appendix

The R Code Static Analysis Landscape

The R Code Static Analysis Landscape

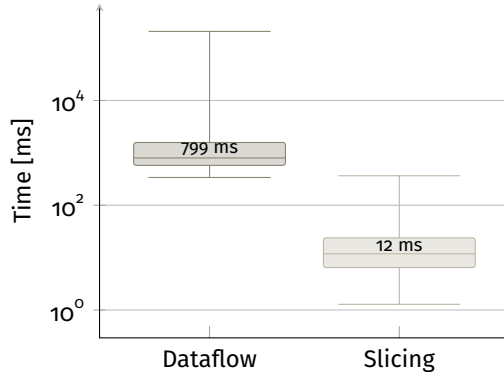
	goal	method	impl. lang.	op. assignments	func. assignments	value trace (a.i., ...)	controlflow	non-std. eval.	special operators	function calls	libraries	quotation	reflection	side effects	static scope	dynamic scope	type inference	pointer analysis	external files	pre-processors	hooks	FFI
[12] {CodeDepends}	static analysis	AST visitor	R	●	○		○	○	○	○			○	●								
[13] {codetools}	static analysis	AST visitor	R	●	○	○	○	○	○		○			●								
[14] {checkglobals}	missing libs.	AST visitor	R, C	●	○		○	○	○	○	○			○							○	
[15] {rstatic}	static analysis	AST visitor	R	●		○	●		○					●								
[16] {CodeAnalysis}	static analysis	AST visitor	R	●	○	○	●	○	○	○	○	○		●		○		○				
[17] {RTypeInference}	type inference	AST visitor	R	●		○	●		○					●		●						
[18] {pkgstats}	package insight	ctags & gtags	R, C++	○					○	○				○								
[19] {globals}	distributed env.	AST visitor	R	●		○	○	○	○			○		●								
[20] {Rclean}	debug/refactor	PDG traversal	R	●	○		○	○	○	○	○			●								
[21] {lintr}	linting	XPath, visitor	R	●	○		○		○	○				●								
[22] {Similar}	plagiarism	PDG, visitor	R, C++	○			○		○					●								
[23] {rco}	optimization	AST visitor	R	●		●	○							●								
[24] {cyclocomp}	code complexity	AST visitor	R				○							●								
[25] {flow}	visualize, debug	AST visitor	R, C	○			●		○		○	○						○				
[26] {PaRe}	code review	Regex	R	○			○		○													
[27] {dfgraph}	static analysis	AST visitor	R	○			○		○													
[28] {rflowgraph}	call graph	AST visitor	R						○	○												
[29] {languageserver}	editor support	XPath, visitor	R, C	●	○				○	○				●								
[30] RStudio	editor support	AST visitor	Java, C++, TS, ...	●	○				○	○				●								
[31] ROSA	optimization	visitor	C++, R	●			○			○		○		●		●	○					
[32] Random	abstract-int.	trace & visitor	R	○	○	○	○			○				○								
[4] RaaS	reproducibility	AST visitor	Python, R	●	○			○	○	○	○			●								
[33] GNU R	execute R	bytecode	C, Fortran, R	●	○	●	○	○	●	○		○		●								
[34] FastR	execute R	AST visitor	Java, C, R, ...	●		○			○	○	○			●								
[35] R̃	execute R	SSA, bytecode	C++, R, C, ...	●	●	○	●	○	●		○			●		○		●				
[36] renjin	execute R	SSA, CFG	R, Java, C, ...	●	○	○	●		○	○			○	●				●				
[37] pgR	execute R	bytecode	C, R, ...	●	○	●	○		○	○		○		●								
[38] MRO	execute R	bytecode	R, C, Fortran, ...	●	○	○	○							●								
[39] R2C	transpile C	CFG, bytecode	C/C++, Fortran, R, ...	●		●	○	○		○			○	●								

Performance Measurements

- We generated *every* possible variable of interest

Performance Measurements

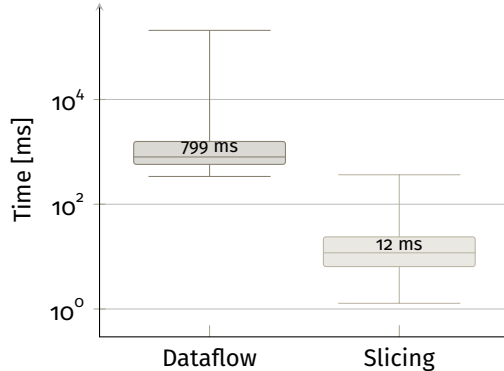
- We generated every possible variable of interest



99th percentile

Performance Measurements

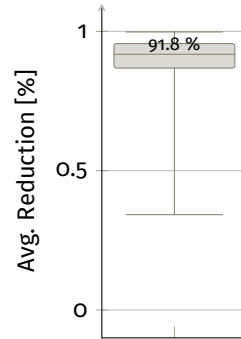
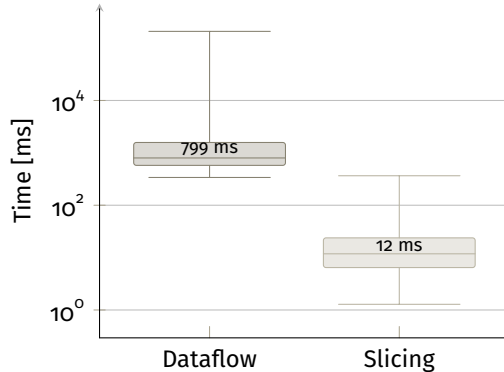
- We generated *every possible* variable of interest
- Dataflow results can be cached



99th percentile

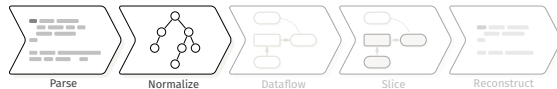
Performance Measurements

- We generated every possible variable of interest
- Dataflow results can be cached

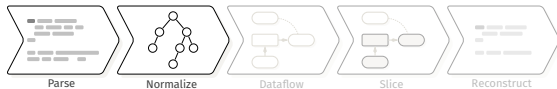


99th percentile

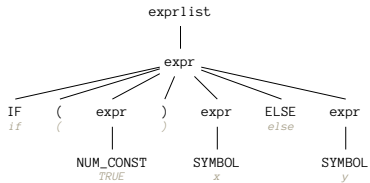
Parse & Normalize



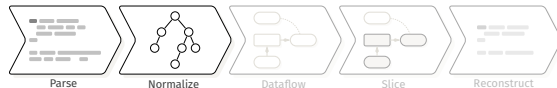
Parse & Normalize



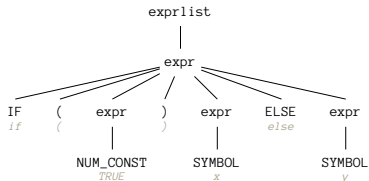
```
parse(text="if(TRUE) x else y")
```



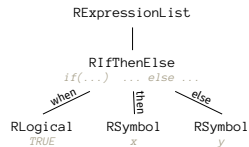
Parse & Normalize



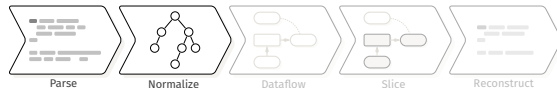
```
parse(text="if(TRUE) x else y")
```



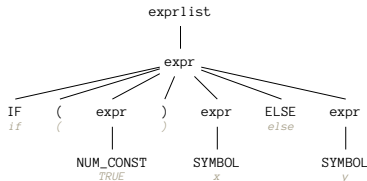
normalized



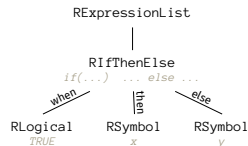
Parse & Normalize



```
parse(text="if(TRUE) x else y")
```

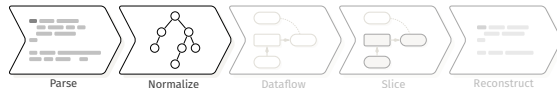


normalized

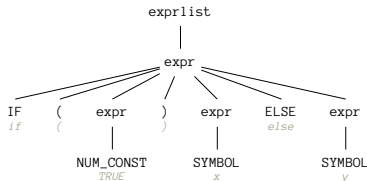


- Normalizing constants, namespacing, operators, ...

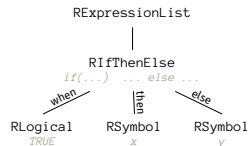
Parse & Normalize



```
parse(text="if(TRUE) x else y")
```



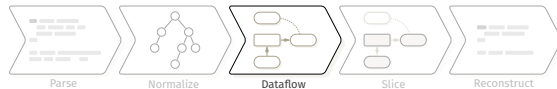
normalized →



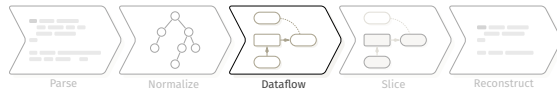
- Normalizing constants, namespacing, operators, ...
- We use the “R language definition”^[40] as a basis

[40] R Core Team, *R Language Definition* (2023)

Dataflow



Dataflow

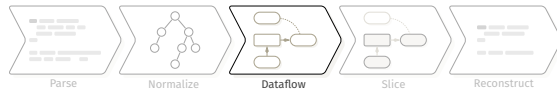


$x_0 \leftarrow 21$

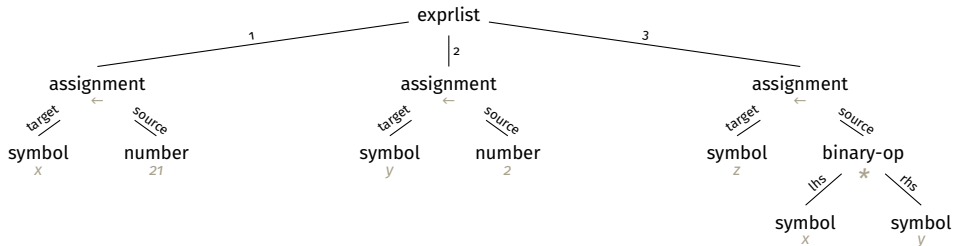
$y_0 \leftarrow 2$

$z_0 \leftarrow x_1 * y_1$

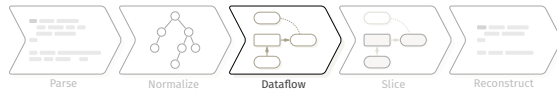
Dataflow



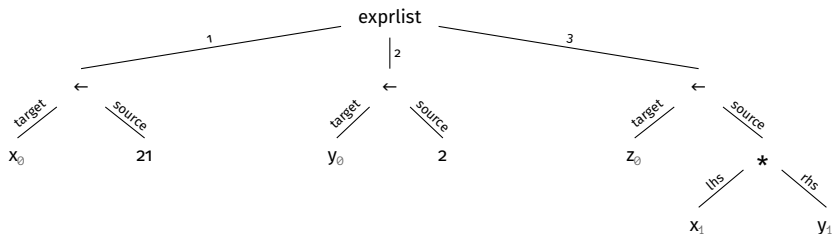
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



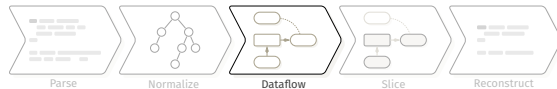
Dataflow



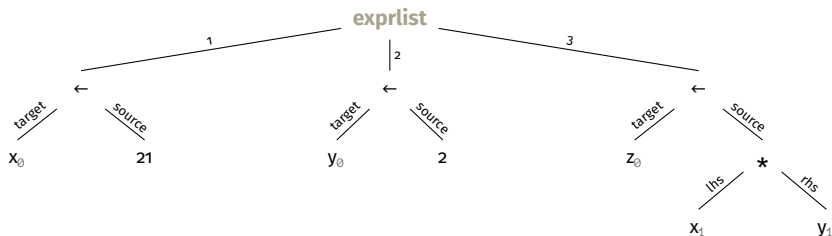
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



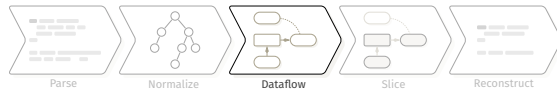
Dataflow



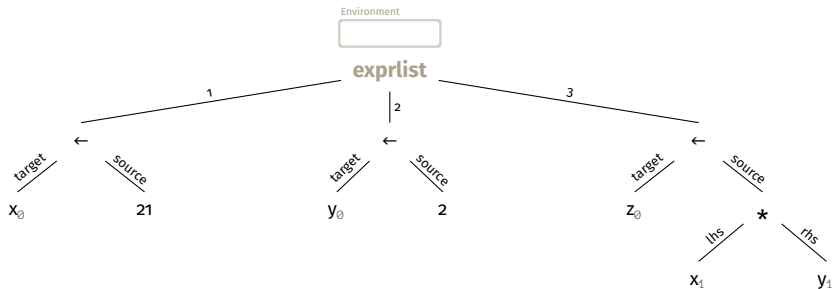
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



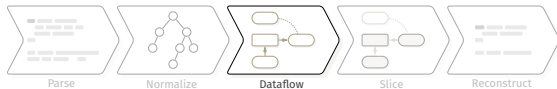
Dataflow



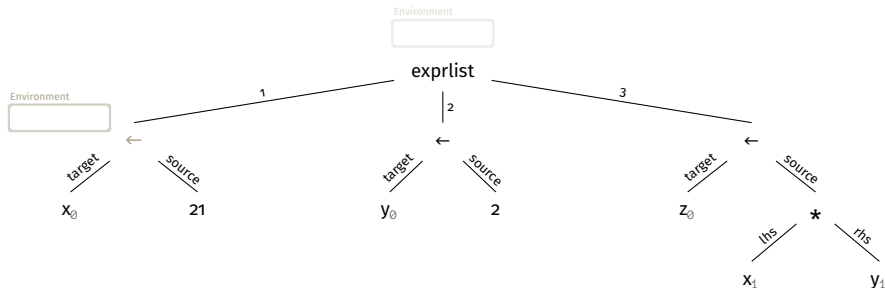
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



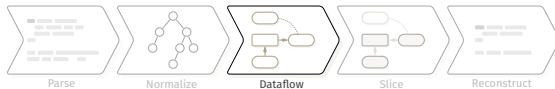
Dataflow



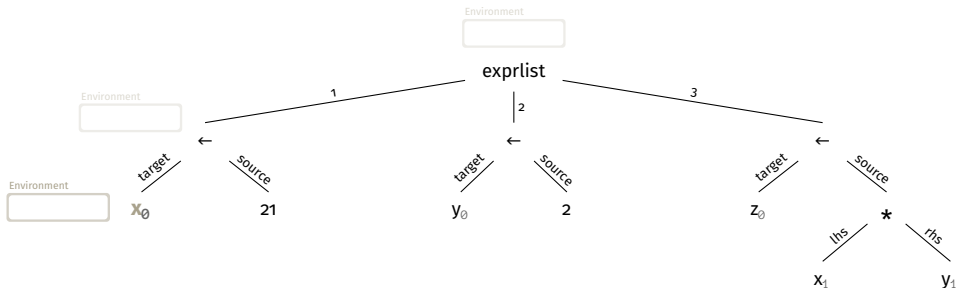
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



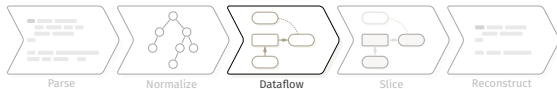
Dataflow



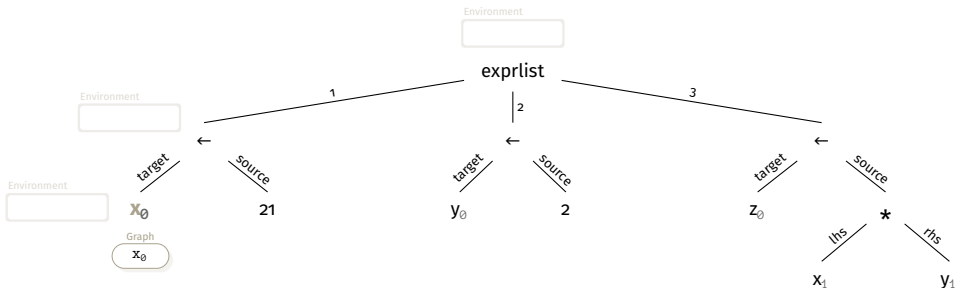
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



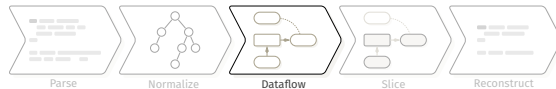
Dataflow



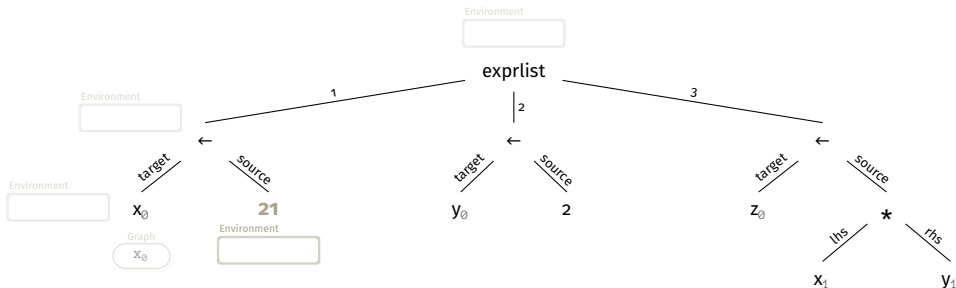
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



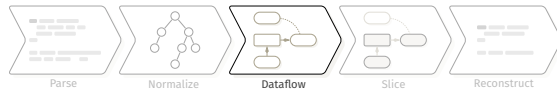
Dataflow



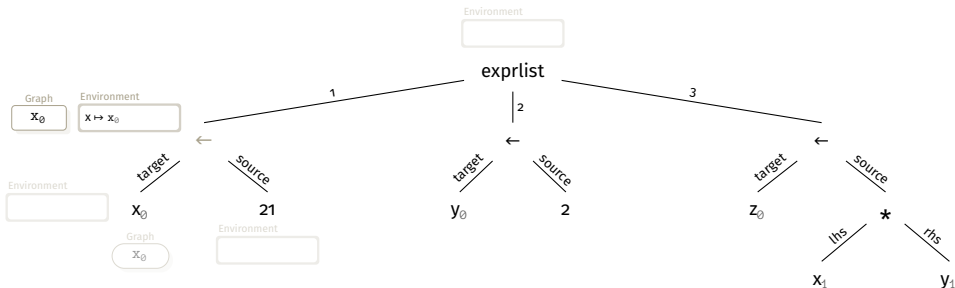
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



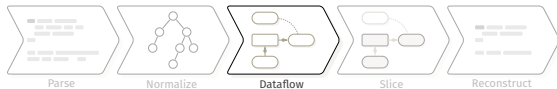
Dataflow



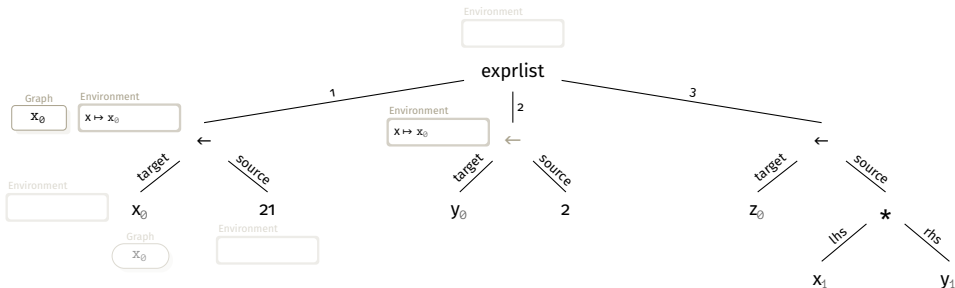
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



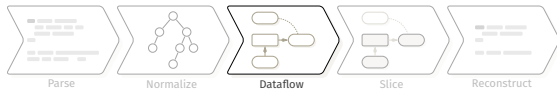
Dataflow



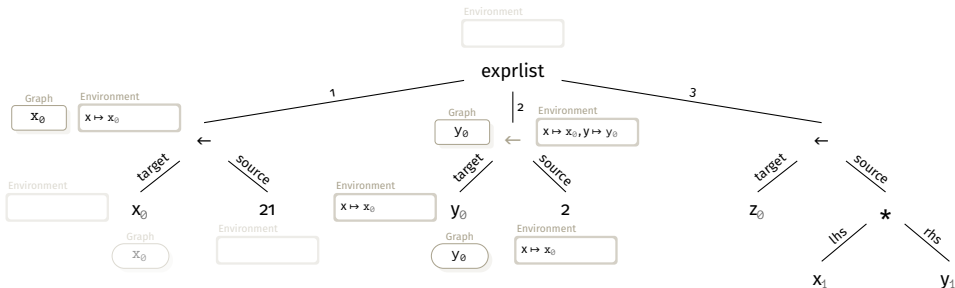
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



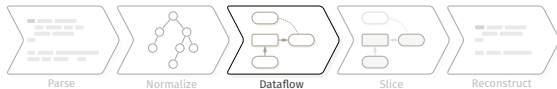
Dataflow



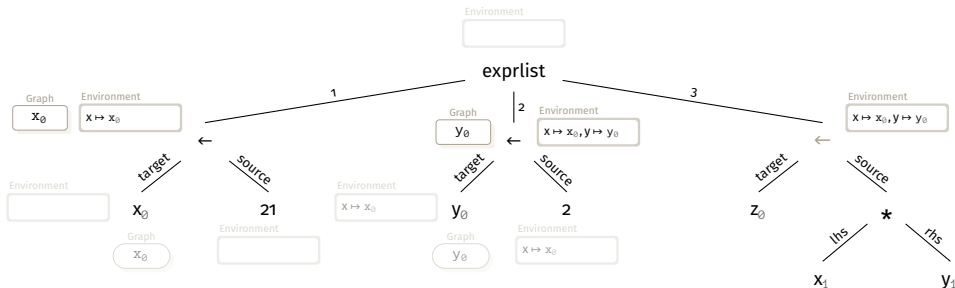
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



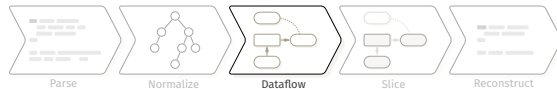
Dataflow



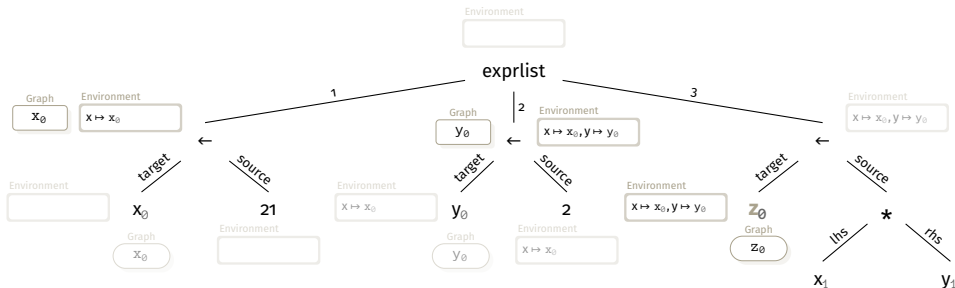
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



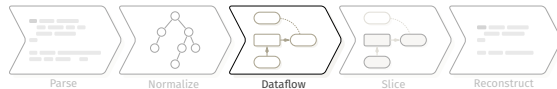
Dataflow



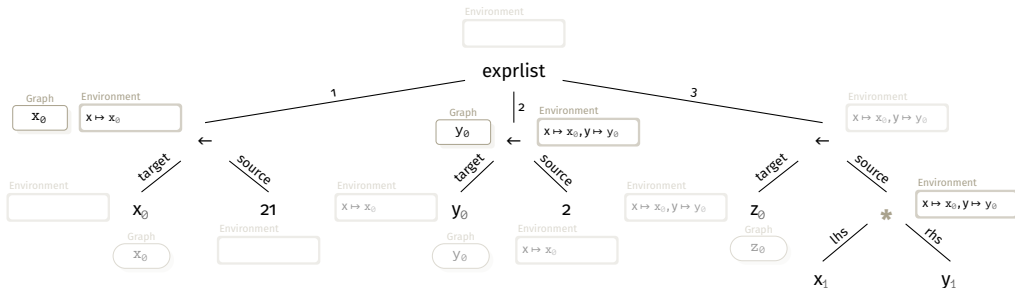
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



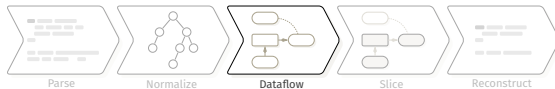
Dataflow



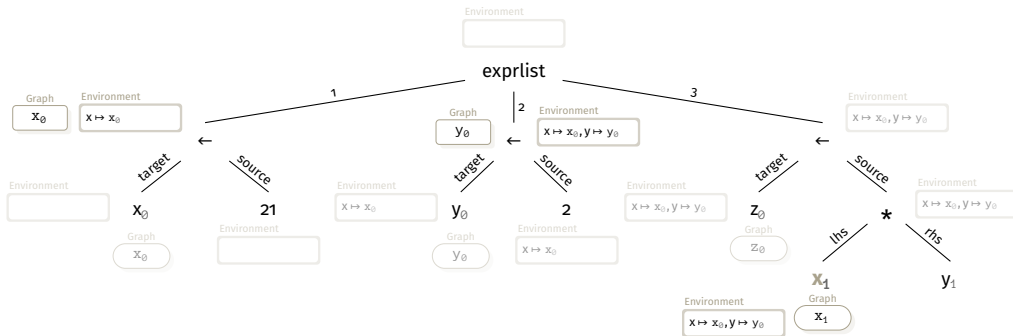
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



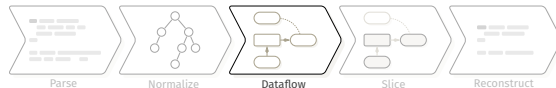
Dataflow



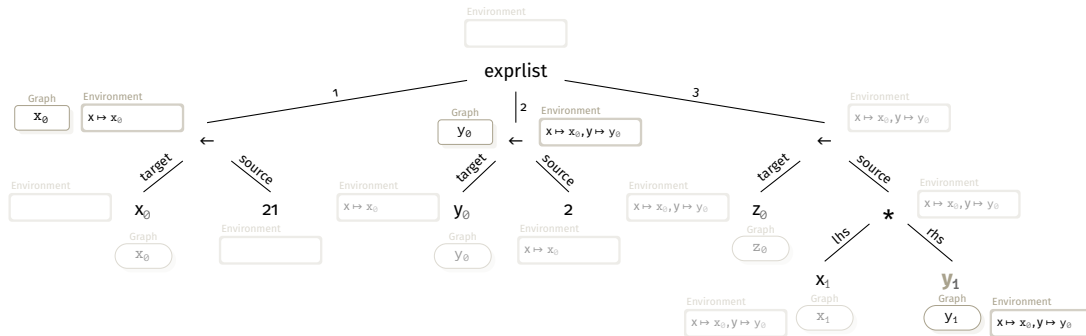
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



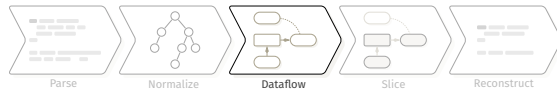
Dataflow



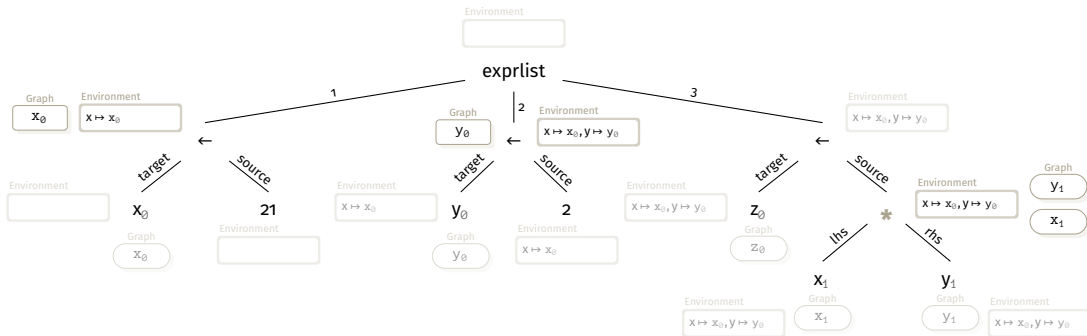
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



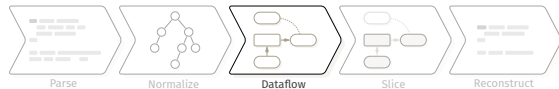
Dataflow



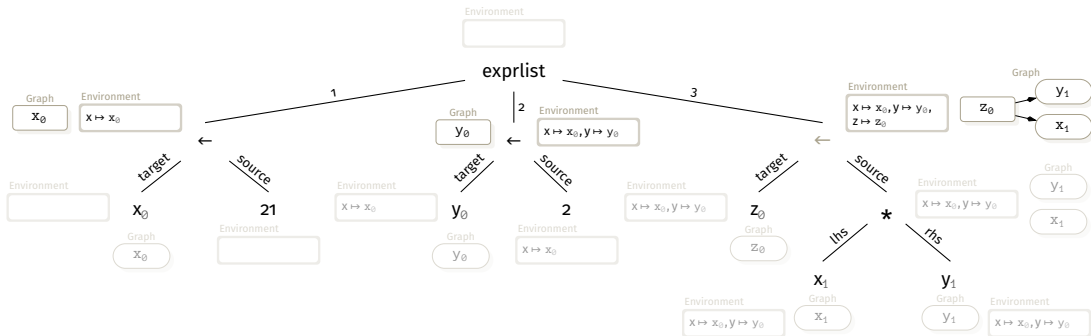
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



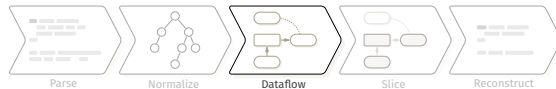
Dataflow



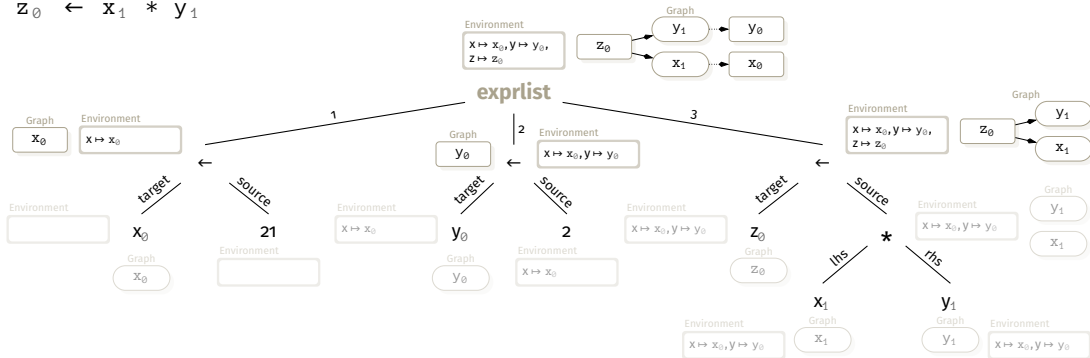
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



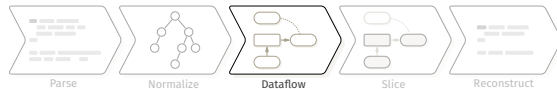
Dataflow



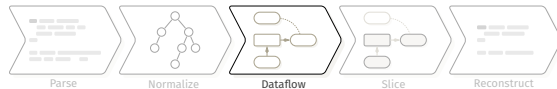
$x_0 \leftarrow 21$
 $y_0 \leftarrow 2$
 $z_0 \leftarrow x_1 * y_1$



Resulting Dataflow



Resulting Dataflow

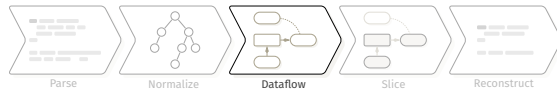


```
a ← 3
a ← x * m

if(m > 3) {
  a ← 5
}

b ← a + c
```

Resulting Dataflow



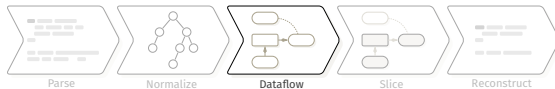
$a_0 \leftarrow 3$

$a_1 \leftarrow x_0 * m_0$

```
if( $m_1 > 3$ ) {  
   $a_2 \leftarrow 5$   
}
```

$b_0 \leftarrow a_3 + c_0$

Resulting Dataflow



a_0

```
>  $a_0 \leftarrow 3$ 
```

```
 $a_1 \leftarrow x_0 * m_0$ 
```

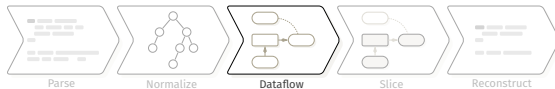
```
if( $m_1 > 3$ ) {
```

```
   $a_2 \leftarrow 5$ 
```

```
}
```

```
 $b_0 \leftarrow a_3 + c_0$ 
```

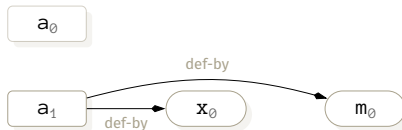
Resulting Dataflow



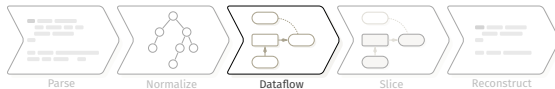
```
a0 ← 3  
> a1 ← x0 * m0
```

```
if(m1 > 3) {  
  a2 ← 5  
}
```

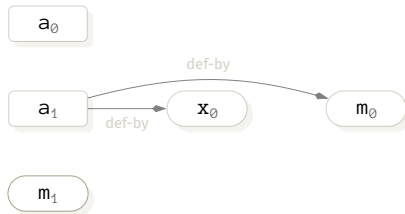
```
b0 ← a3 + c0
```



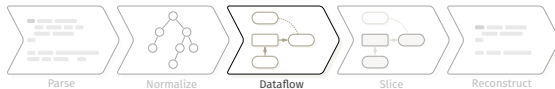
Resulting Dataflow



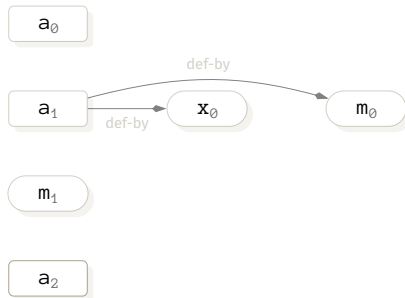
```
a0 ← 3  
a1 ← x0 * m0  
  
> if(m1 > 3) {  
  a2 ← 5  
}  
  
b0 ← a3 + c0
```



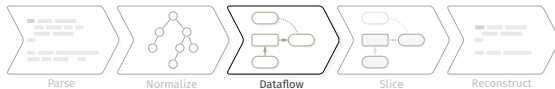
Resulting Dataflow



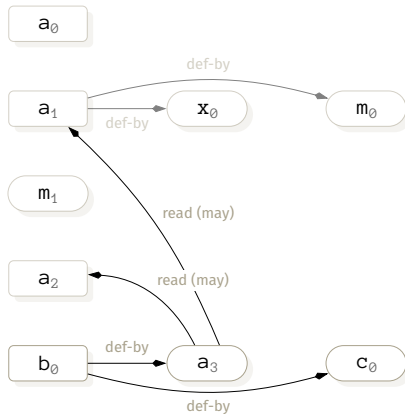
```
a0 ← 3  
a1 ← x0 * m0  
  
if(m1 > 3) {  
>   a2 ← 5  
}  
  
b0 ← a3 + c0
```



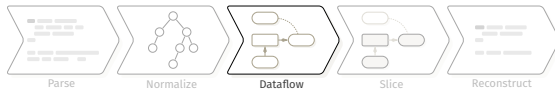
Resulting Dataflow



```
a0 ← 3  
a1 ← x0 * m0  
  
if(m1 > 3) {  
  a2 ← 5  
}  
  
> b0 ← a3 + c0
```



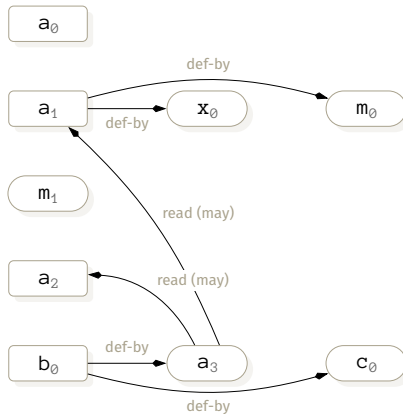
Resulting Dataflow



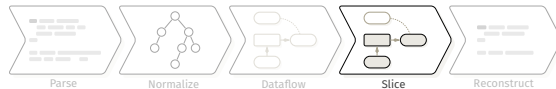
```
a0 ← 3  
a1 ← x0 * m0
```

```
if(m1 > 3) {  
  a2 ← 5  
}
```

```
b0 ← a3 + c0
```



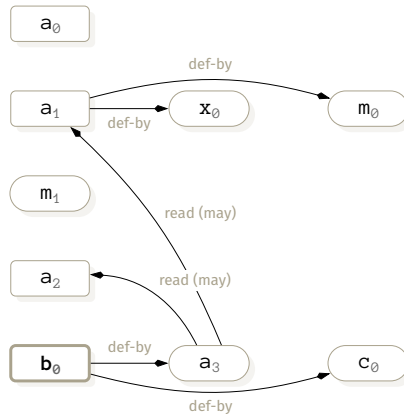
Slicing, I



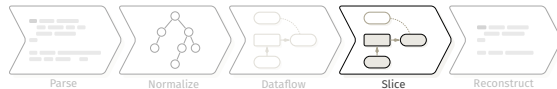
```
a0 ← 3  
a1 ← x0 * m0
```

```
if(m1 > 3) {  
  a2 ← 5  
}
```

```
b0 ← a3 + c0
```



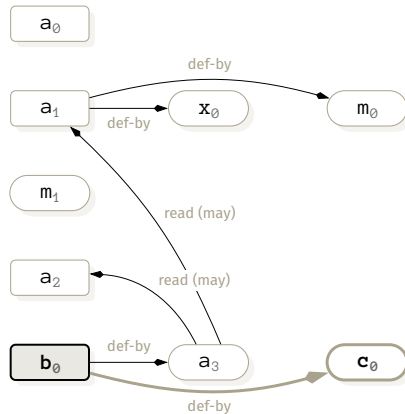
Slicing, I



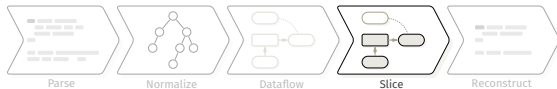
```
a0 ← 3  
a1 ← x0 * m0
```

```
if(m1 > 3) {  
    a2 ← 5  
}
```

```
b0 ← a3 + c0
```



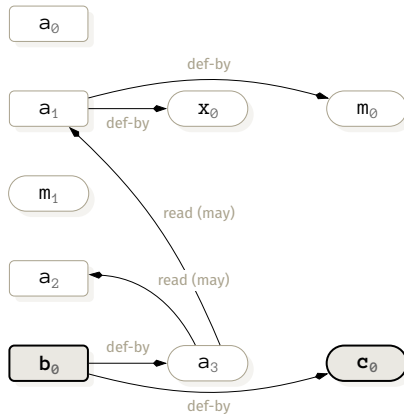
Slicing, I



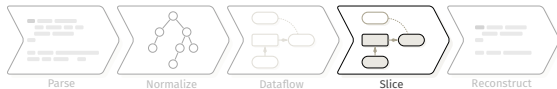
```
a0 ← 3  
a1 ← x0 * m0
```

```
if(m1 > 3) {  
  a2 ← 5  
}
```

```
b0 ← a3 + c0
```



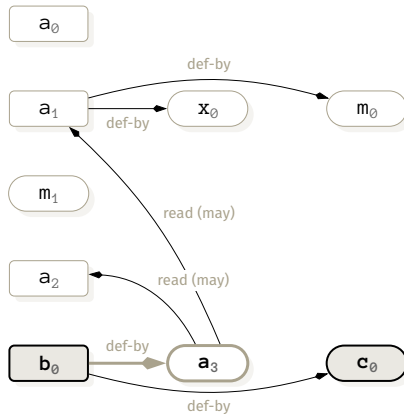
Slicing, I



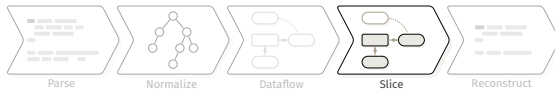
```
a0 ← 3  
a1 ← x0 * m0
```

```
if(m1 > 3) {  
  a2 ← 5  
}
```

```
b0 ← a3 + c0
```



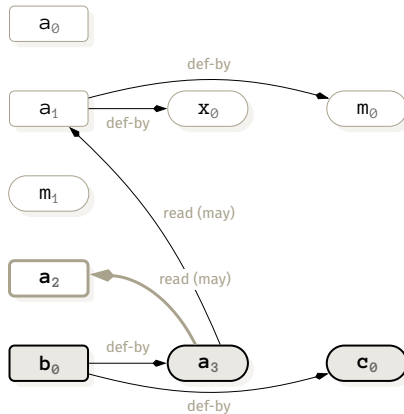
Slicing, I



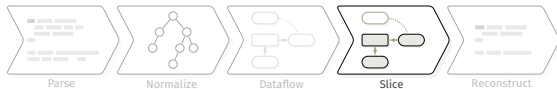
```
a0 ← 3  
a1 ← x0 * m0
```

```
if(m1 > 3) {  
    a2 ← 5  
}
```

```
b0 ← a3 + c0
```



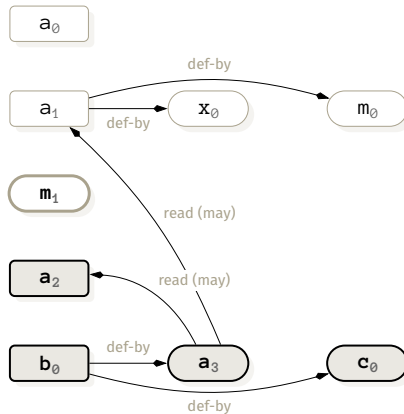
Slicing, I



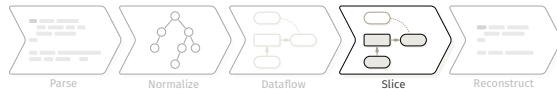
```
a0 ← 3  
a1 ← x0 * m0
```

```
if(m1 > 3) {  
  a2 ← 5  
}
```

```
b0 ← a3 + c0
```



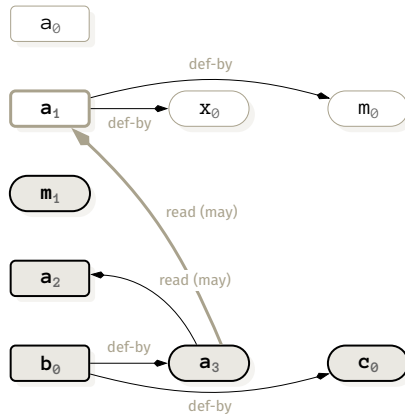
Slicing, I



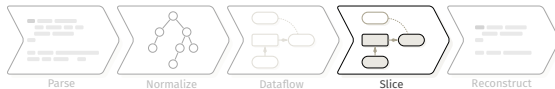
```
a0 ← 3  
a1 ← x0 * m0
```

```
if(m1 > 3) {  
  a2 ← 5  
}
```

```
b0 ← a3 + c0
```



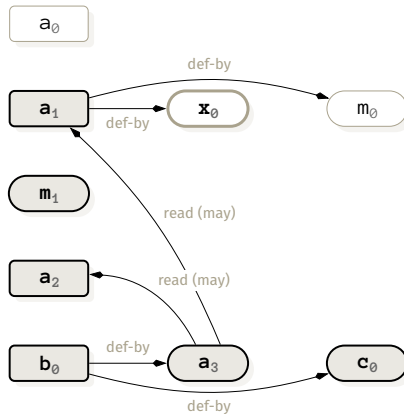
Slicing, I



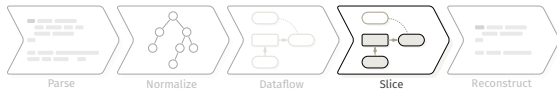
```
a0 ← 3  
a1 ← x0 * m0
```

```
if(m1 > 3) {  
    a2 ← 5  
}
```

```
b0 ← a3 + c0
```



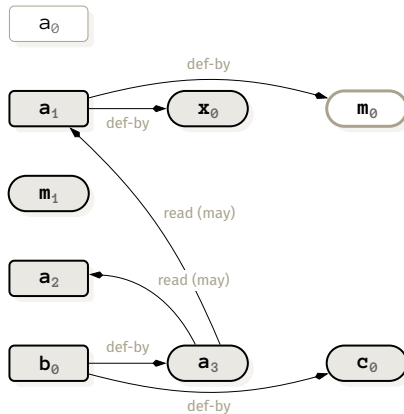
Slicing, I



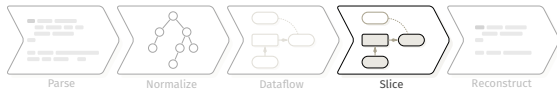
```
a0 ← 3  
a1 ← x0 * m0
```

```
if(m1 > 3) {  
  a2 ← 5  
}
```

```
b0 ← a3 + c0
```



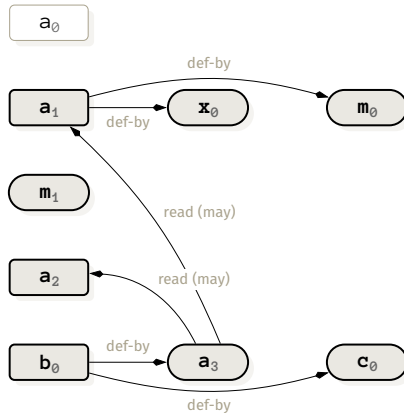
Slicing, I



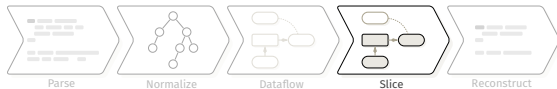
```
a0 ← 3  
a1 ← x0 * m0
```

```
if(m1 > 3) {  
  a2 ← 5  
}
```

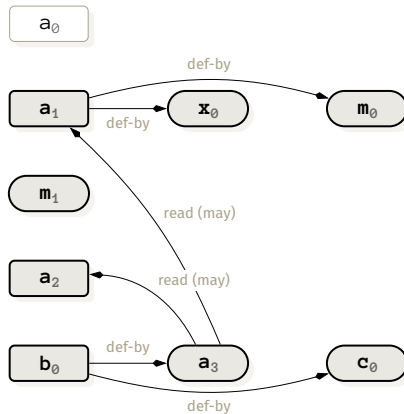
```
b0 ← a3 + c0
```



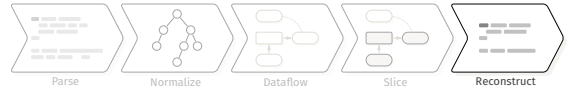
Slicing, I



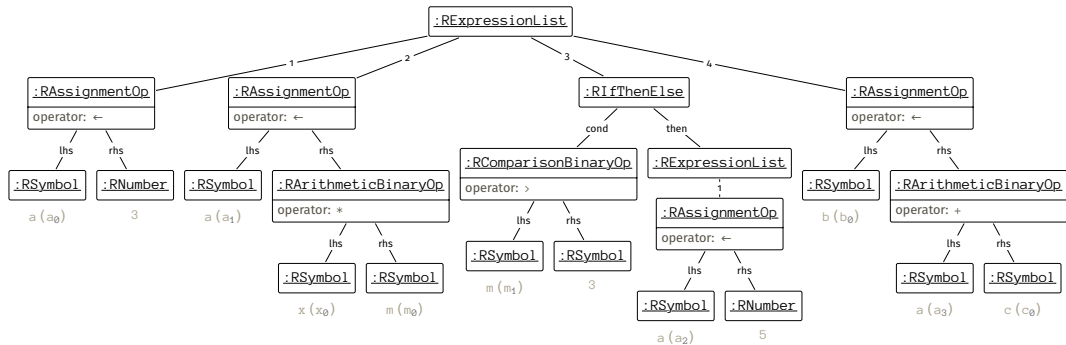
```
a0 ← 3  
a1 ← x0 * m0  
  
if(m1 > 3) {  
  a2 ← 5  
}  
  
b0 ← a3 + c0
```



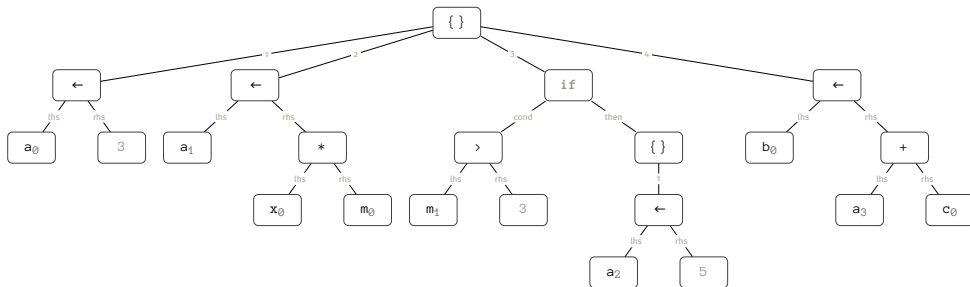
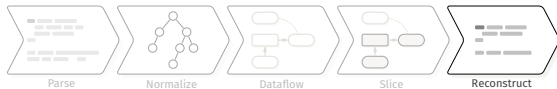
Slicing, II



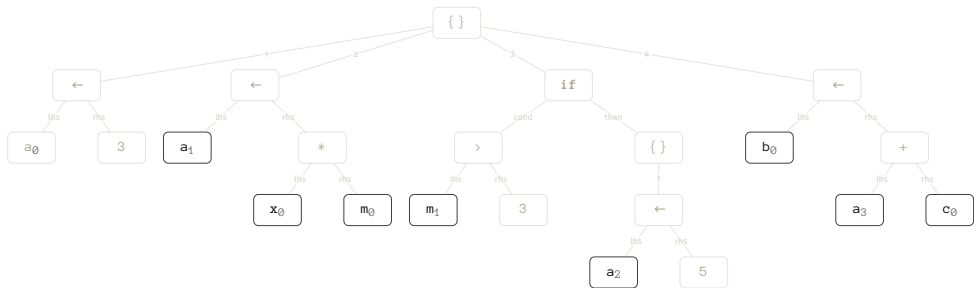
Slicing, II



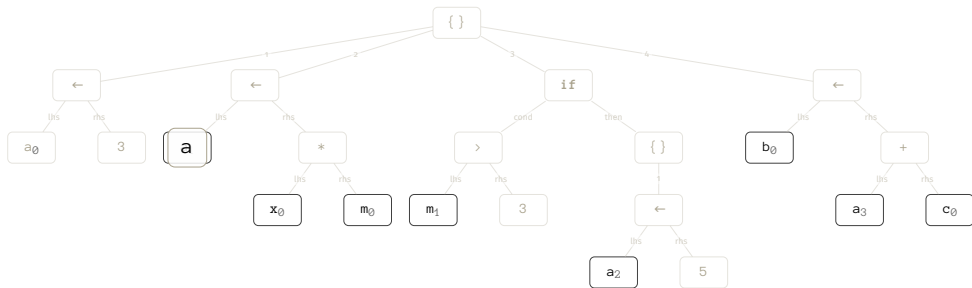
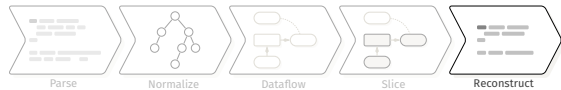
Slicing, II



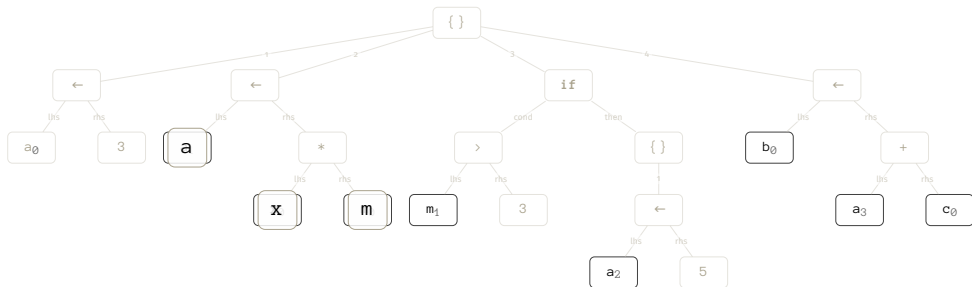
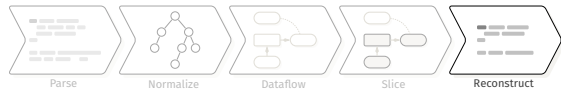
Slicing, II



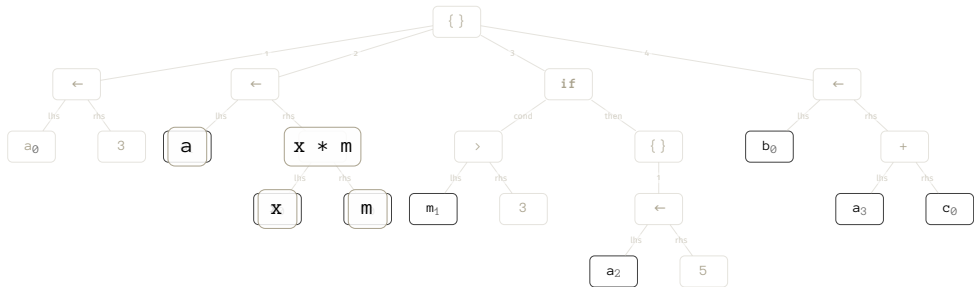
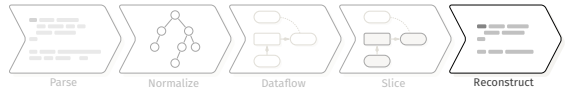
Slicing, II



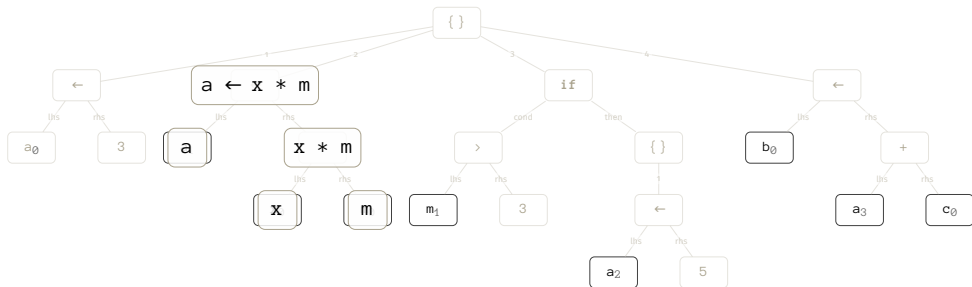
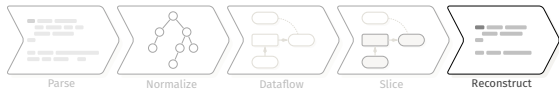
Slicing, II



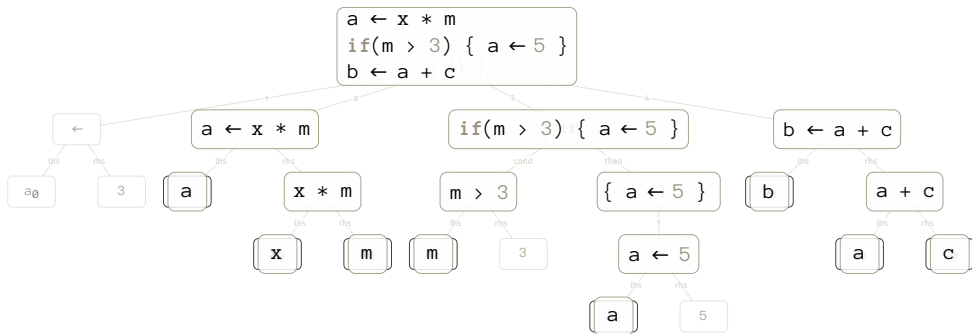
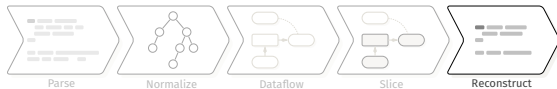
Slicing, II



Slicing, II



Slicing, II



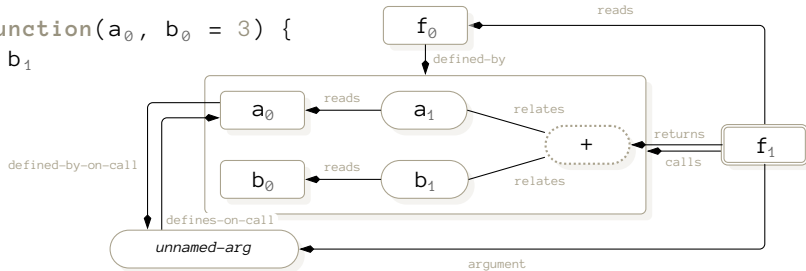
There Is More...

There Is More...

```
f0 ← function(a0, b0 = 3) {  
  a1 + b1  
}  
f1(39)
```


There Is More...

```
f0 ← function(a0, b0 = 3) {  
  a1 + b1  
}  
f1(39)
```



Definition-Retrieval

```
paste(  
  "(*|descendant-or-self::exprlist/*)[self::FUNCTION or self::OP-LAMBDA]/  
    following-sibling::SYMBOL_FORMALS[text() = '{token_quote}' and @line1 <= {  
      row}]",  
  "(*|descendant-or-self::exprlist/*)[LEFT_ASSIGN[preceding-sibling::expr[count  
    (*)=1]/SYMBOL[text() = '{token_quote}' and @line1 <= {row}] and following-  
    sibling::expr[@start > {start} or @end < {end}]]]",  
  "(*|descendant-or-self::exprlist/*)[RIGHT_ASSIGN[following-sibling::expr[count  
    (*)=1]/SYMBOL[text() = '{token_quote}' and @line1 <= {row}] and preceding-  
    sibling::expr[@start > {start} or @end < {end}]]]",  
  "(*|descendant-or-self::exprlist/*)[EQ_ASSIGN[preceding-sibling::expr[count(*)=  
    1]/SYMBOL[text() = '{token_quote}' and @line1 <= {row}] and following-  
    sibling::expr[@start > {start} or @end < {end}]]]",  
  "forcond/SYMBOL[text() = '{token_quote}' and @line1 <= {row}]",  
  sep = "|")
```

References I

- [1] *The Comprehensive R Archive Network — cran.r-project.org.* 2024
- [2] Ana Trisovic et al. “A Large-Scale Study on Research Code Quality and Execution”. 2022
- [3] *PYPL — PopularitY of Programming Language index.* 2024
- [4] Joseph Wonsil et al. “Reproducibility as a Service”. 2023
- [5] Harold Thimbleby. “Improving Science That Uses Code”. 2023
- [6] 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
- [7] Florian Sihler et al. “On the Anatomy of Real-World R Code for Static Analysis”. 2024
- [8] Liang Ma et al. *Predicting range shifts of pikas (Mammalia, Ochotonidae) in China under scenarios incorporating land-use change, climate change, and dispersal limitations.* Aug. 2021
- [9] Joshua Robertson. *Social hierarchy reveals thermoregulatory trade-offs in response to repeated stressors.* Oct. 2020
- [10] Florian Sihler. “Constructing a static program slicer for R programs”. 2023

References II

- [11] Mark Weiser. “Program Slicing”. July 1984
- [12] Duncan Lang et al. *CodeDepends. Analysis of R Code for Reproducible Research and Code Comprehension.* 2018
- [13] Luke Tierney. *codetools: Code Analysis Tools for R.* 2023
- [14] Joris Chau. *checkglobals: Static Analysis of R-Code Dependencies.* 2023
- [15] Nick Ulle and Duncan Temple Lang. *rstatic: Low-level Static Analysis Tools for R Code.* 2019
- [16] Duncan Lang et al. *CodeAnalysis. Tools for static analysis of R code.* 2023
- [17] Nick Ulle and Duncan Temple Lang. *RTypeInference: Infer Types of Inputs and Outputs for R Expressions.* 2021
- [18] Mark Padgham. *pkgstats.* 2021
- [19] Henrik Bengtsson. *globals: Identify Global Objects in R Expressions.* 2022
- [20] Matthew Lau. *Rclean: A Tool for Writing Cleaner, More Transparent Code.* 2022
- [21] Jim Hester et al. *lintr: A ‘Linter’ for R Code.* 2023
- [22] Maciej Bartoszek and Marek Gagolewski. *SimilaR: R Source Code Similarity Evaluation.* 2020

References III

- [23] Juan Cruz Rodriguez. *rco: The R Code Optimizer*. 2021
- [24] Gabor Csardi. *cyclocomp: Cyclomatic Complexity of R Code*. 2023
- [25] Antoine Fabri. *flow: View and Browse Code Using Flow Diagrams*. 2023
- [26] Maarten van Kessel. *PaRe: A Way to Perform Code Review or QA on Other Packages*. 2023
- [27] Dan Kary. *dfgraph: Visualize R Code with Data Flow Graphs*.
- [28] Evan Patterson. *The algebra and machine representation of statistical models*. 2020
- [29] Randy Lai. *languageserver: Language Server Protocol*. 2023
- [30] Posit team. *RStudio: Integrated Development Environment for R*. 2023
- [31] Rathijit Sen et al. *ROSA: R Optimizations with Static Analysis*. 2017
- [32] Gianluca Amato and Francesca Scozzari. “Random: R-Based Analyzer for Numerical Domains”. 2012
- [33] R Core Team. *R: A Language and Environment for Statistical Computing*. 2023
- [34] Tomas Kalibera et al. “A fast abstract syntax tree interpreter for R”. 2014
- [35] Olivier Flückiger et al. “Sampling optimized code for type feedback”. 2020

References IV

- [36] Alexander Bertram. “Renjin: A new r interpreter built on the jvm”. 2013
- [37] Radford M Neal. “Speed Improvements in pqR: Current Status and Future Plans”. 2014
- [38] *Microsoft R Open Source*. 2019
- [39] John Garvin. *RCC: A compiler for the R language for statistical computing*. 2004
- [40] R Core Team. *R Language Definition*. 2023