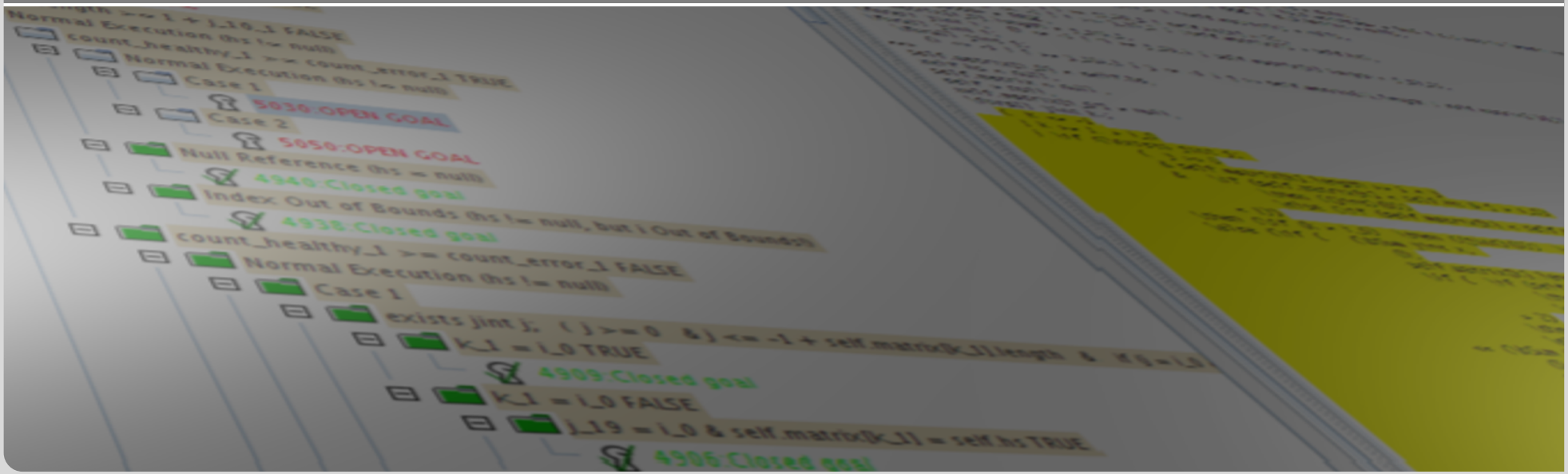
Karlsruhe Institute of Technology

Seamless Interactive Program Verification

Sarah Grebing, Jonas Klamroth, Mattias Ulbrich | 13. July 2019

Institute for Theoretical Informatics, KIT



Motivation

Increased effectiveness of verification systems

Motivation

More complex verification problems are tractable

Increased effectiveness of verification systems

Motivation

User interaction now on more complex proof states

More complex verification problems are tractable

Increased effectiveness of verification systems

Motivation

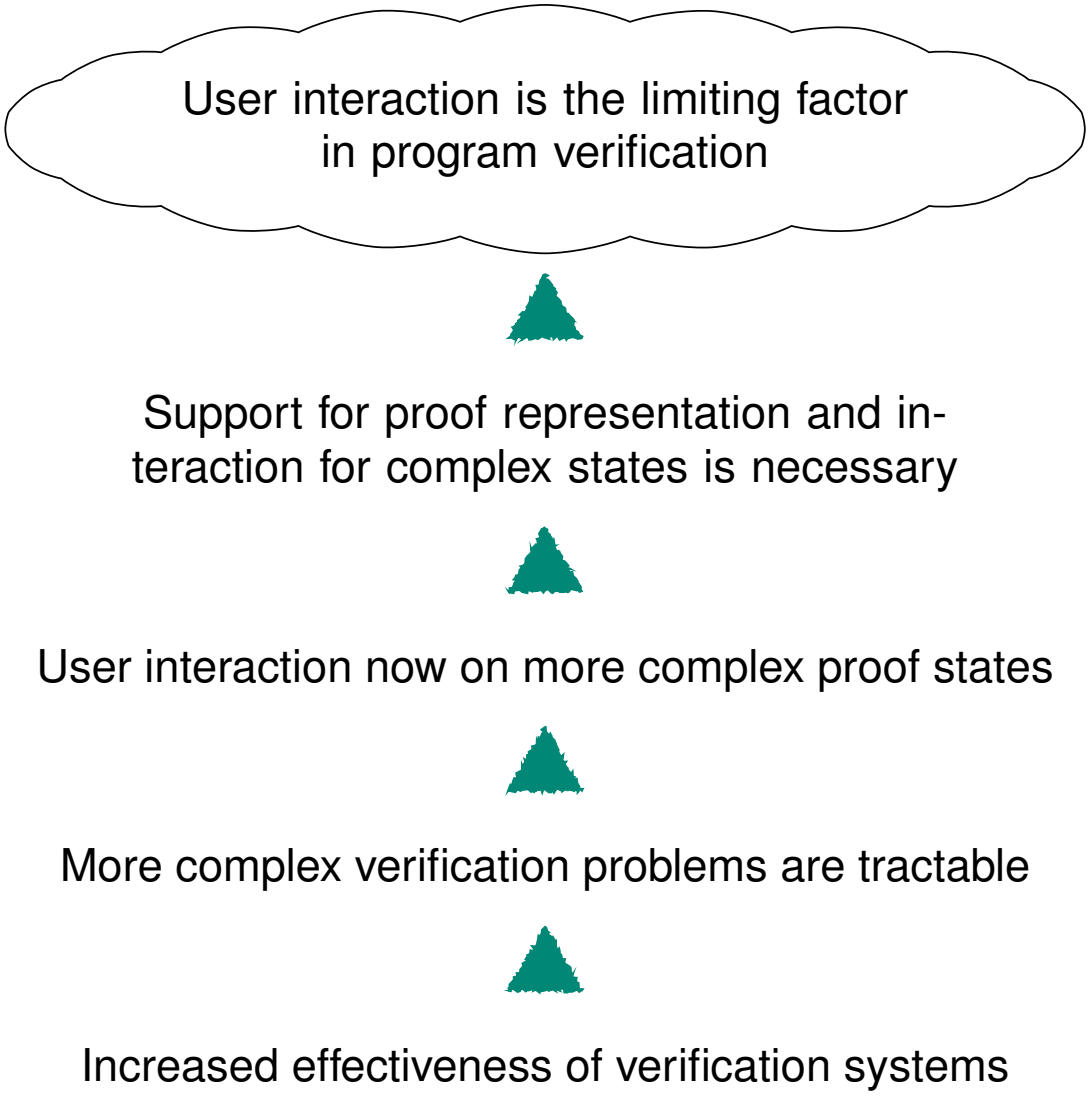
Support for proof representation and interaction for complex states is necessary

User interaction now on more complex proof states

More complex verification problems are tractable

Increased effectiveness of verification systems

Motivation

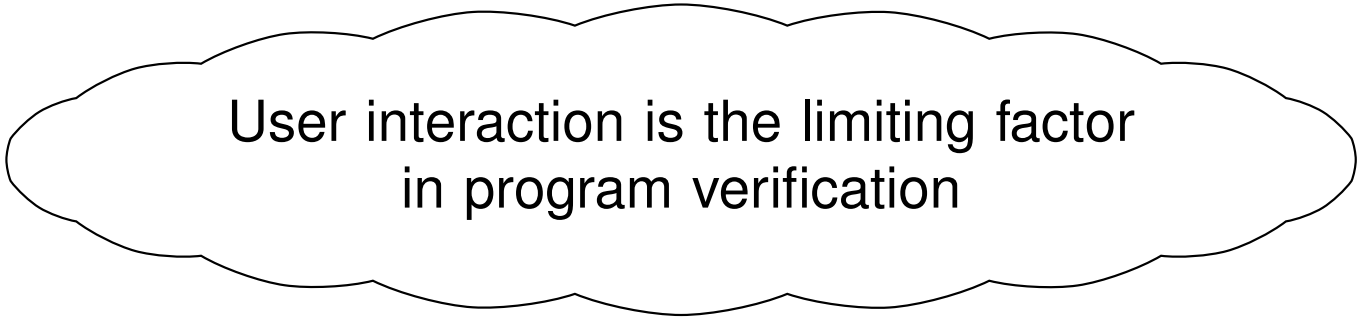

User interaction is the limiting factor
in program verification
Support for proof representation and in-
teraction for complex states is necessary
User interaction now on more complex proof states
More complex verification problems are tractable
Increased effectiveness of verification systems

Karlsruhe Institute of Technology

Working Hypothesis:

User interaction is the limiting factor
in program verification

Working Hypothesis:



Contributions

- user studies on interaction in program verification
- a new interaction concept
- prototypical implementation

Qualitative, Explorative User Studies

Karlsruhe Institute of Technology

Study Topics

- proof step granularity
- time-consuming actions
- feedback mechanisms
- proof comprehension

Qualitative, Explorative User Studies

Study Topics

- proof step granularity
- time-consuming actions
- feedback mechanisms
- proof comprehension

Methods

- focus group discussions: KeY and Isabelle/HOL
- semi-structured interviews with practical tasks: KeY
- qualitative content analysis and sequence models

Result of the User Studies

Understanding the proof state is crucial
and a central challenge for users

Key Observations of the User Studies

Key Observation (Context-switch)

Difficult: Switching between elements in proof and program

Key Observations of the User Studies

Key Observation (Context-switch)

Difficult: Switching between elements in proof and program

$\Longrightarrow$ Support access to and combination of information from different domains

Key Observations of the User Studies

Key Observation (Context-switch)

Difficult: Switching between elements in proof and program

$\Longrightarrow$ Support access to and combination of information from different domains

Key Observation (Orientation)

To gain orientation in the proof process typical patterns included:

- abstraction from and focusing on details
- (re)construct relation between proof state and annotated program

Key Observations of the User Studies

Key Observation (Context-switch)

Difficult: Switching between elements in proof and program

$\Longrightarrow$ Support access to and combination of information from different domains

Key Observation (Orientation)

To gain orientation in the proof process typical patterns included:

- abstraction from and focusing on details
- (re)construct relation between proof state and annotated program

$\Longrightarrow$ Provide a global overview and a stepwise focusing on details

Key Observations of the User Studies

Key Observation (Proof Process)

Every participant has an individual proof process.
Typical process patterns include

- using prover feedback for specification refinement
- completing specification before first verification attempt
- proof construction: single rule applications, strategies, and mixed

Key Observations of the User Studies

Key Observation (Proof Process)

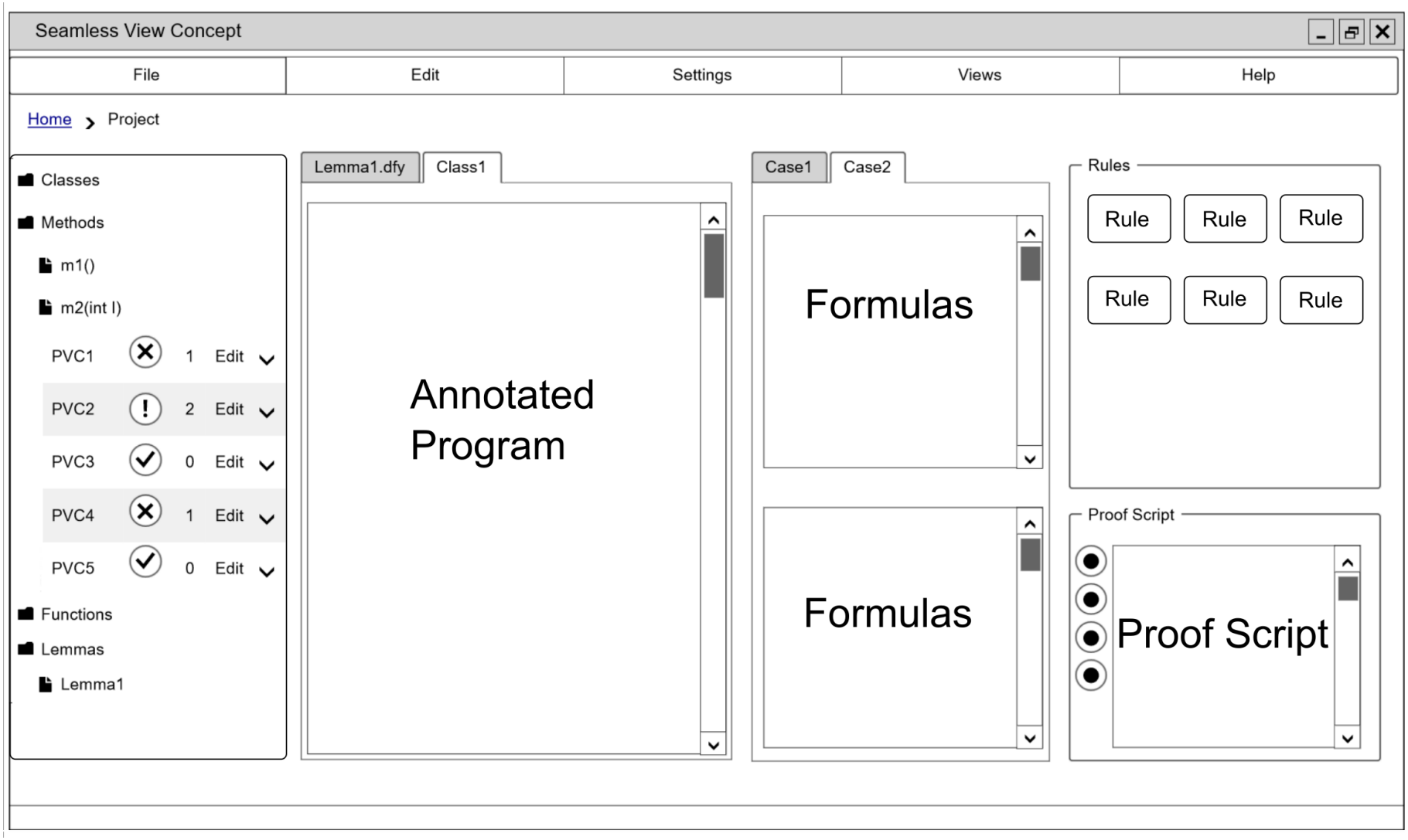
Every participant has an individual proof process.
Typical process patterns include

- using prover feedback for specification refinement
- completing specification before first verification attempt
- proof construction: single rule applications, strategies, and mixed

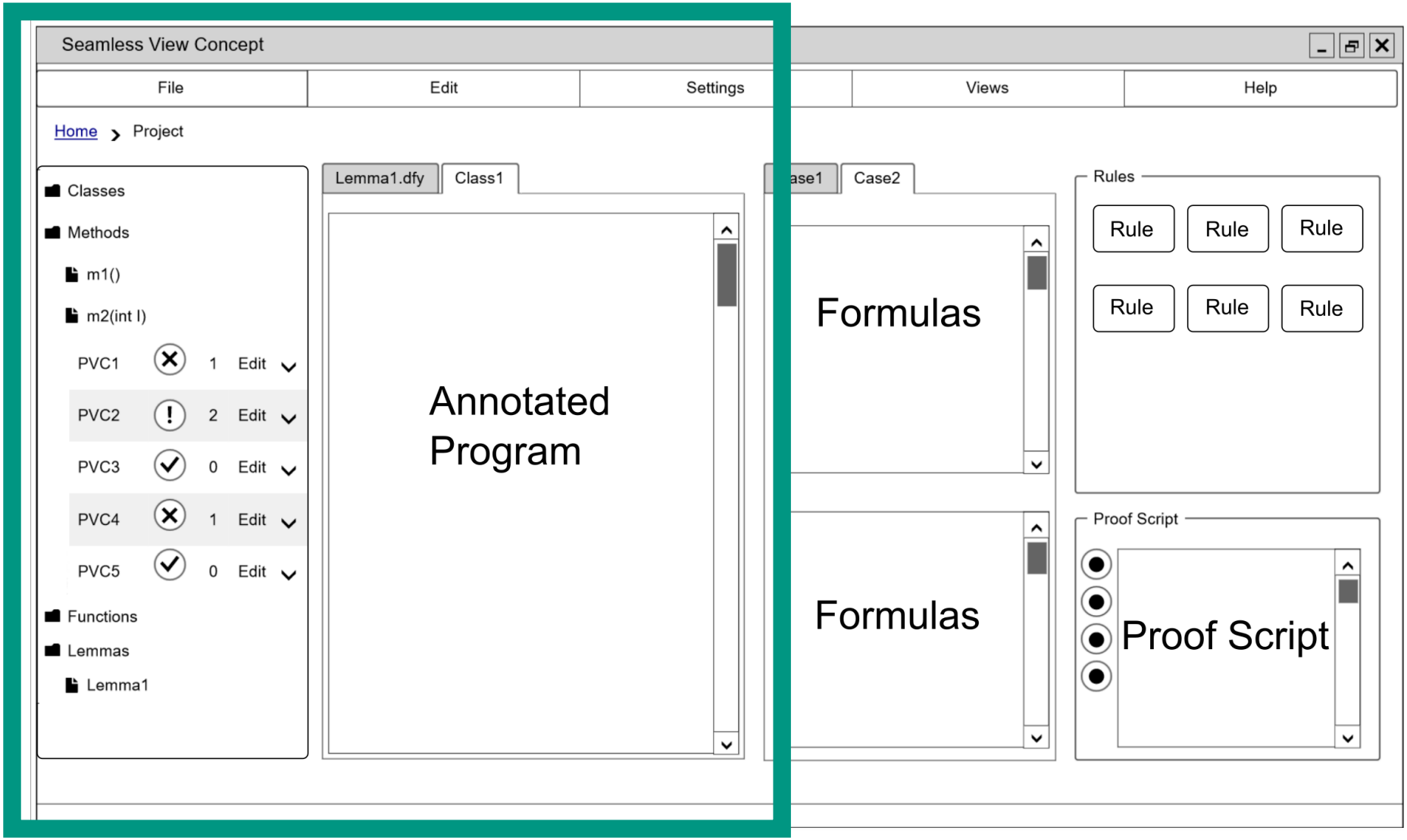
$\Longrightarrow$ Allow many degrees of freedom and support different interaction preferences

Interaction Concept

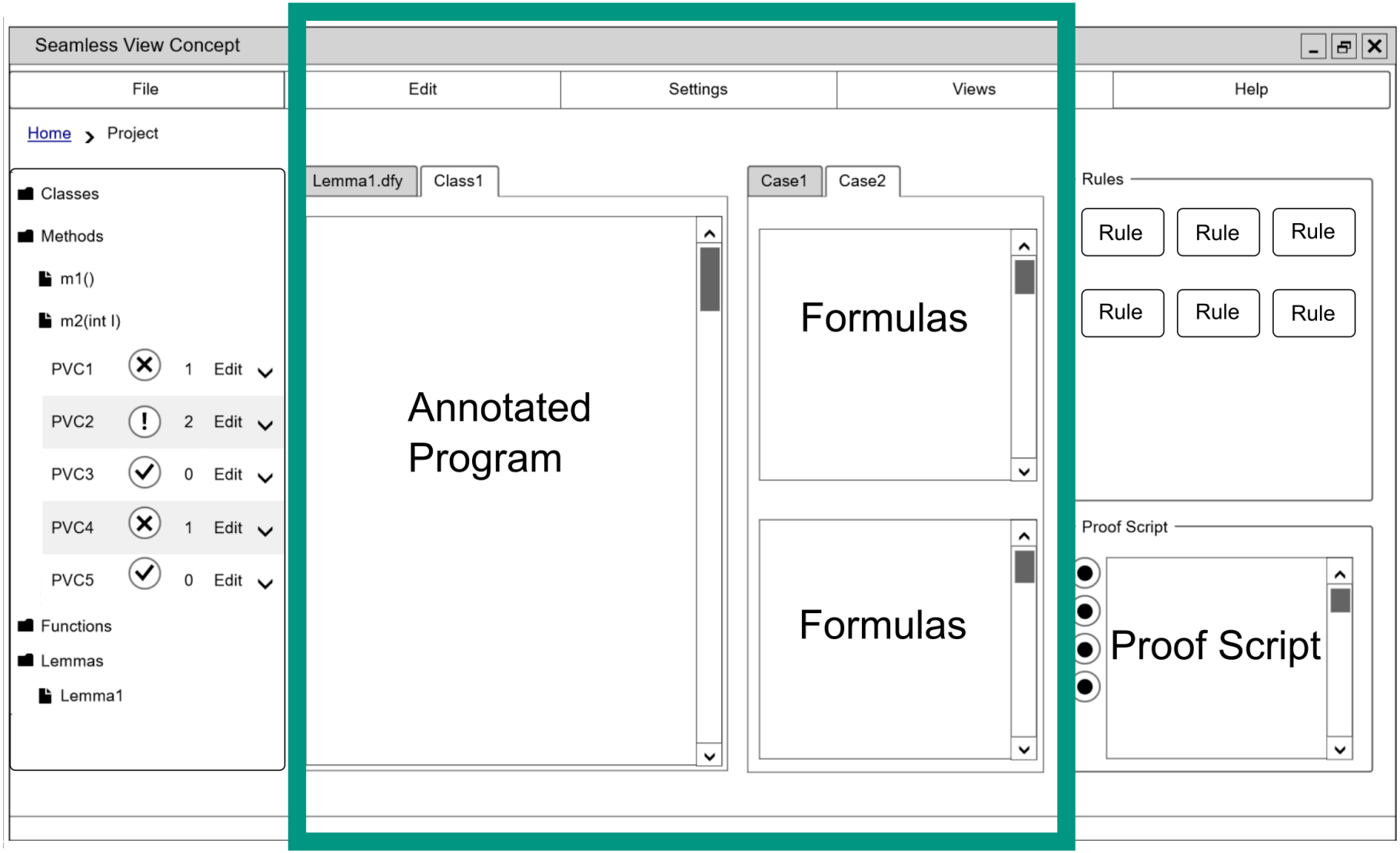
“Seamless Interactive Program Verification”



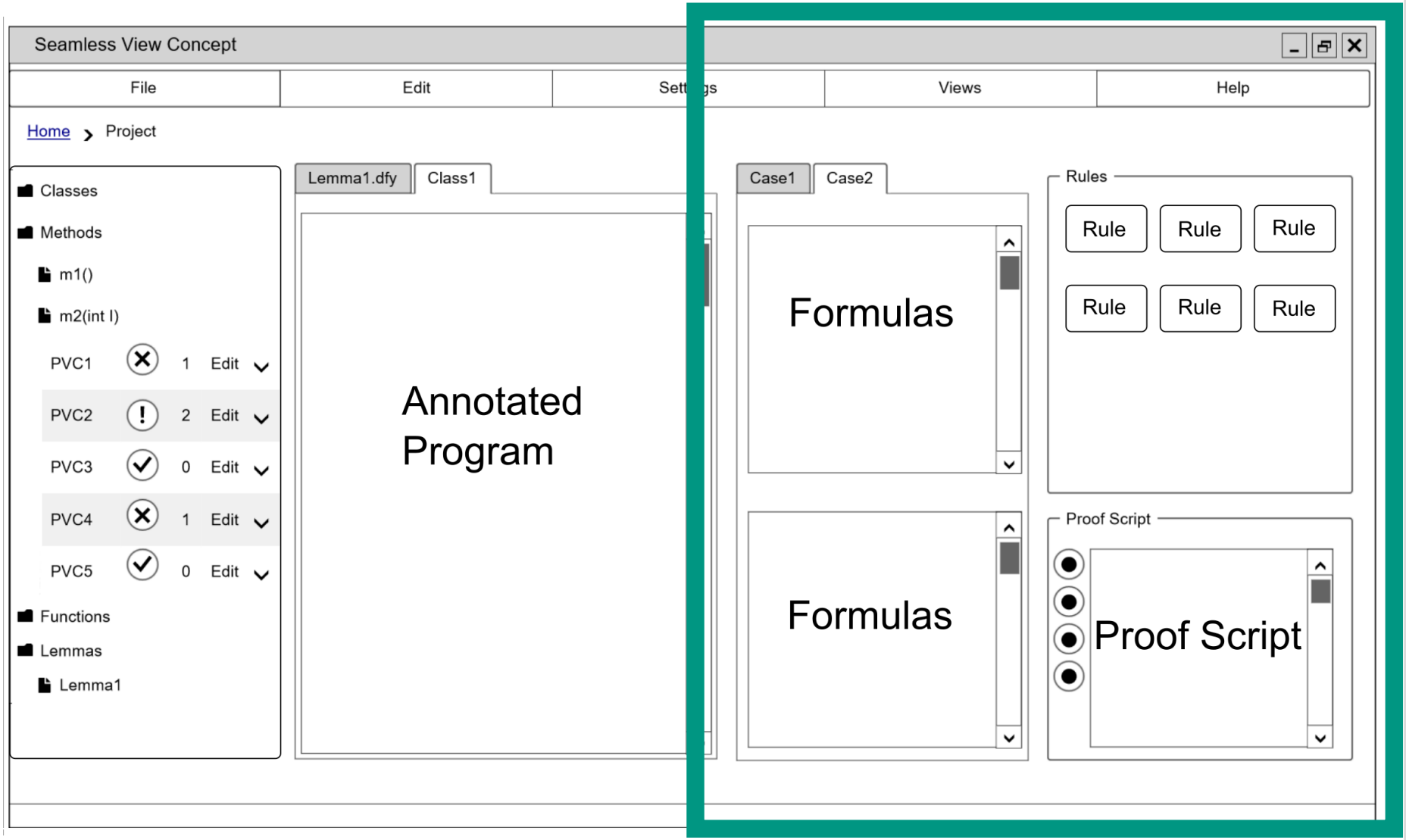
“Seamless Interactive Program Verification”



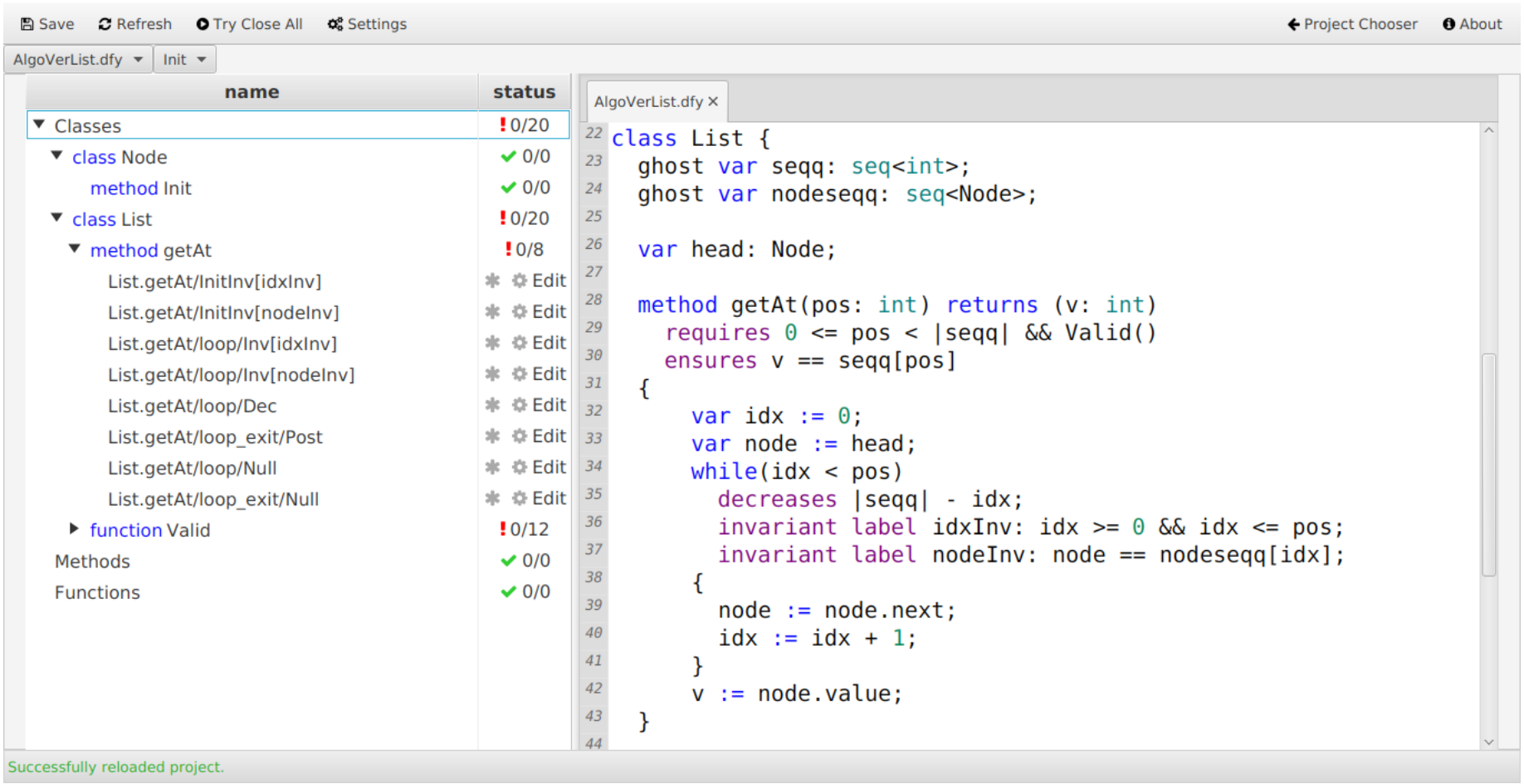
“Seamless Interactive Program Verification”



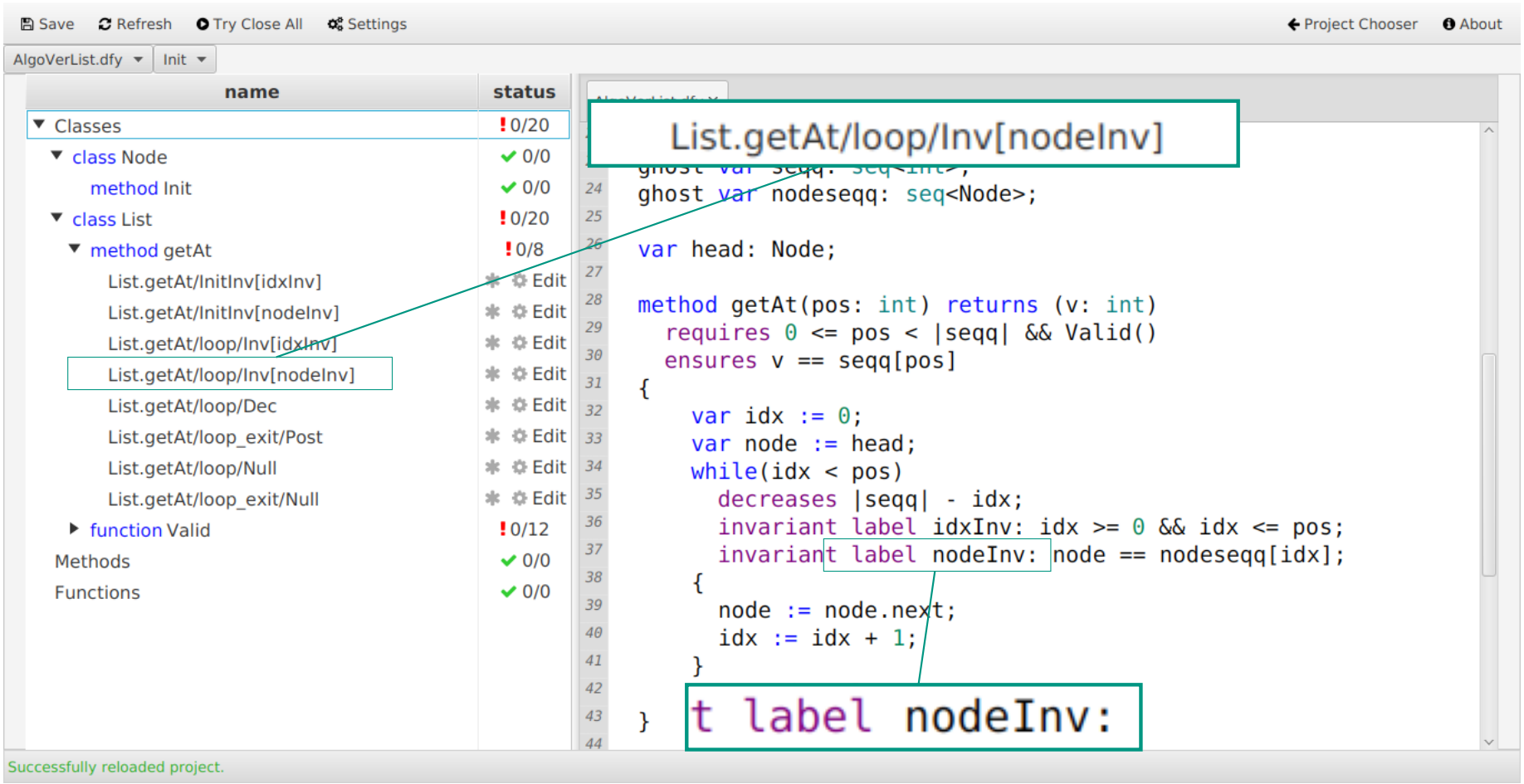
“Seamless Interactive Program Verification”



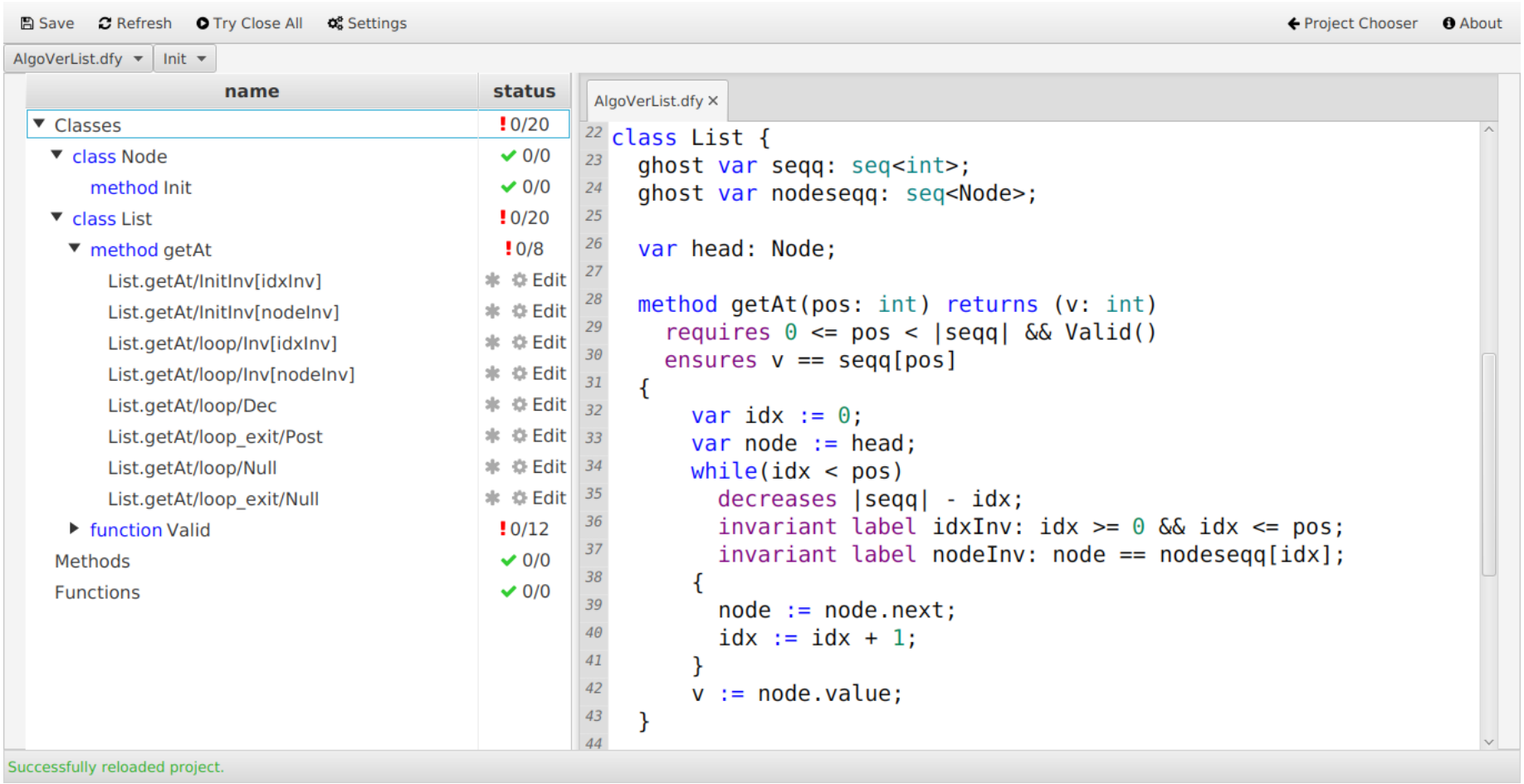
Exemplary Walkthrough



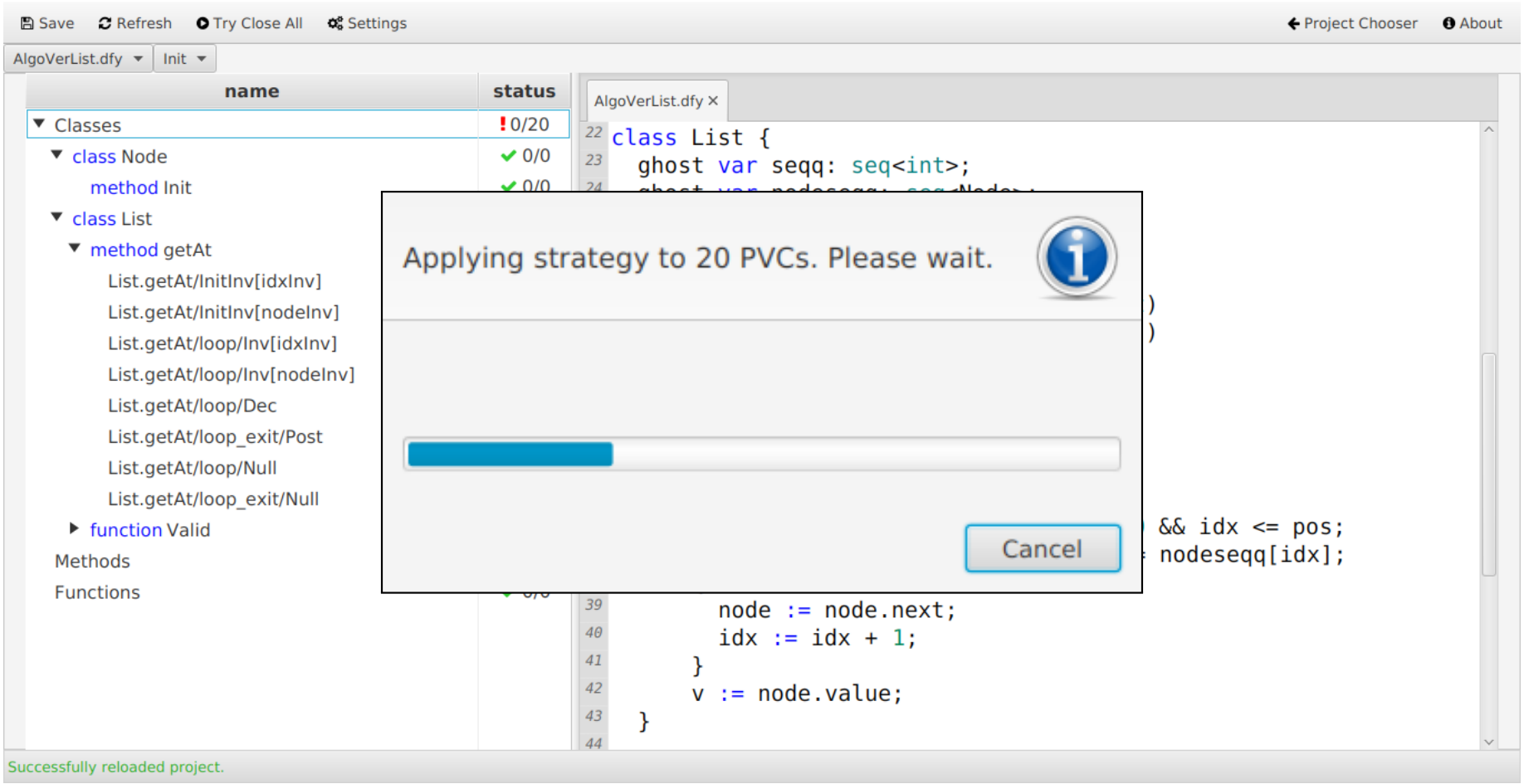
Exemplary Walkthrough



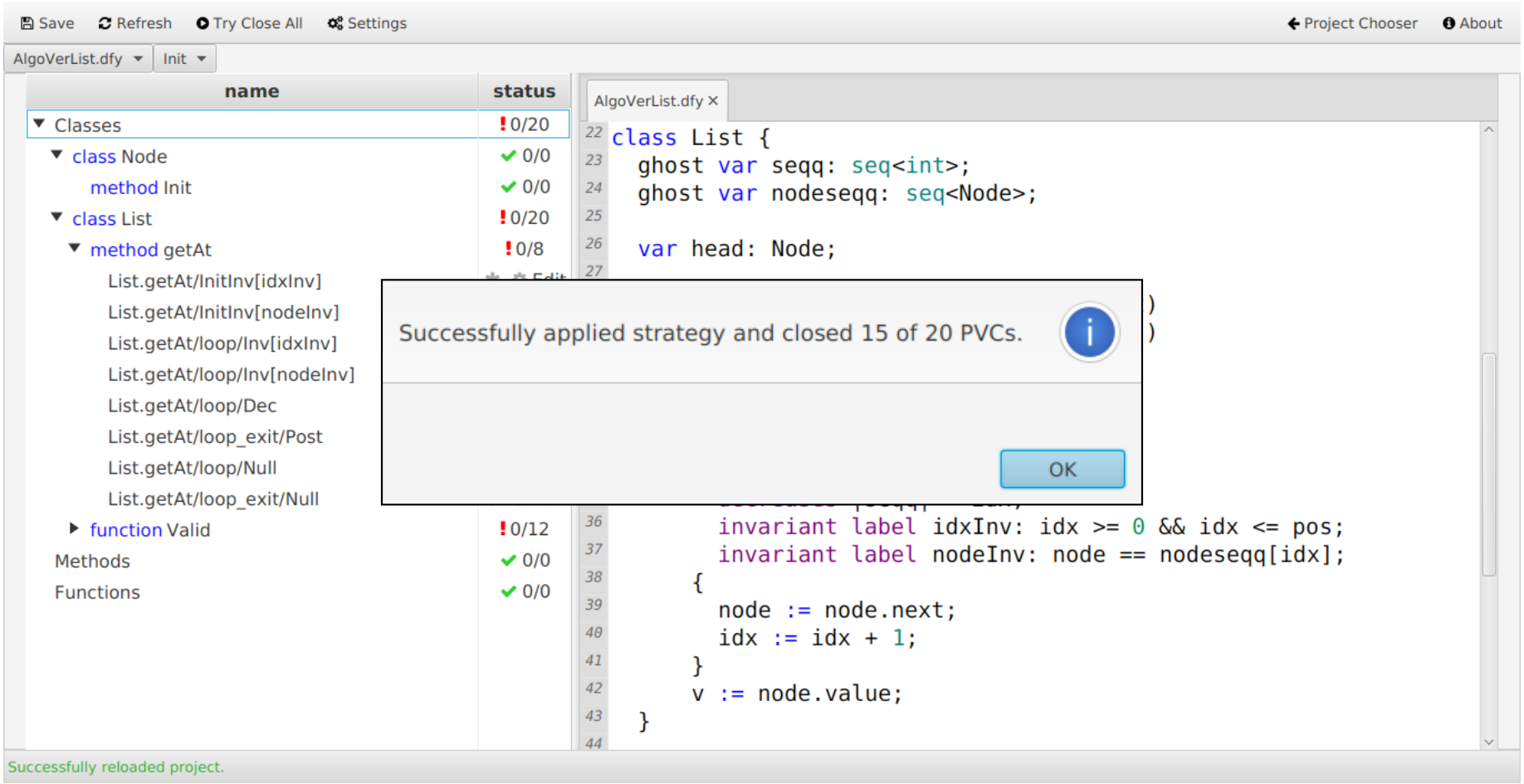
Exemplary Walkthrough



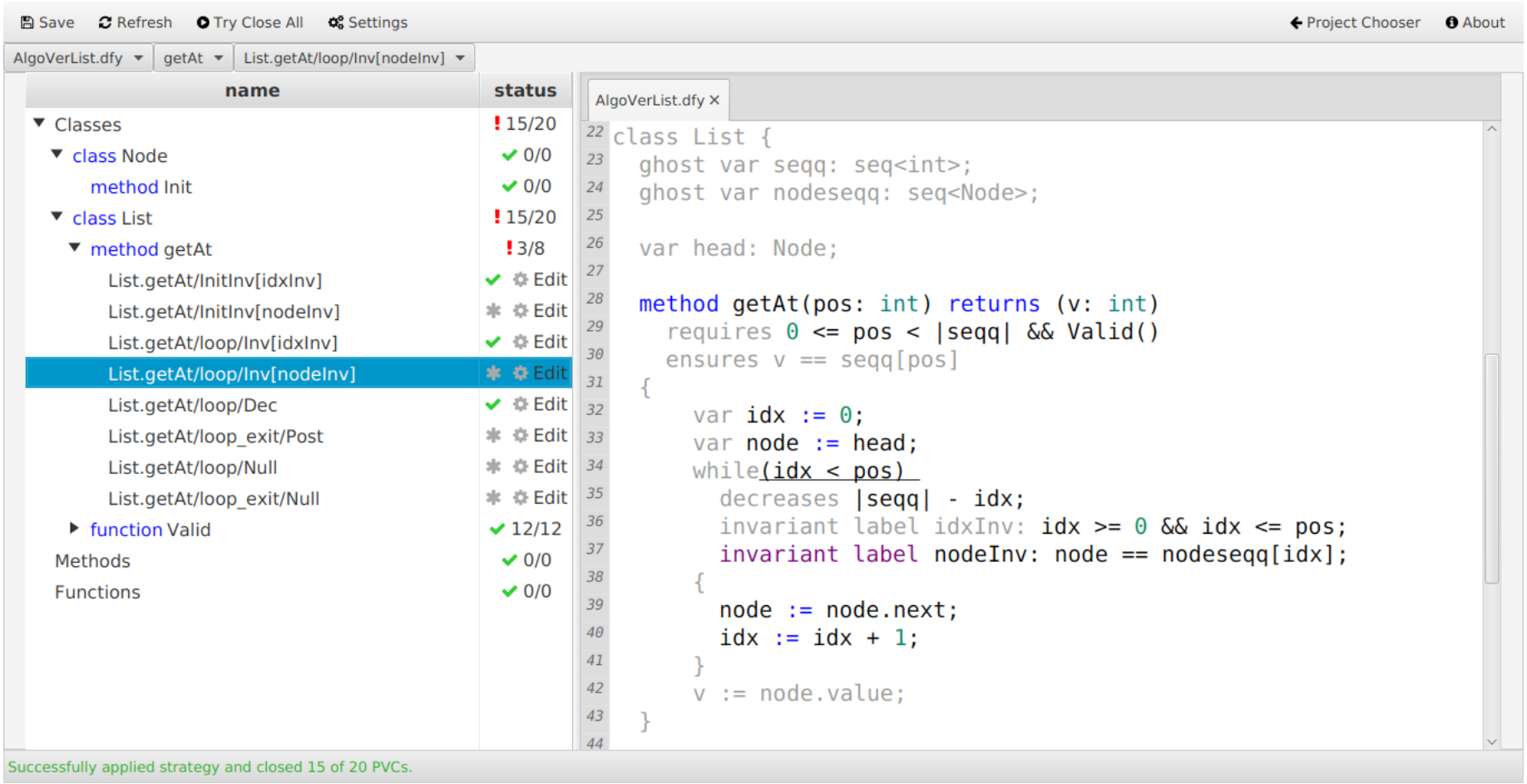
Exemplary Walkthrough



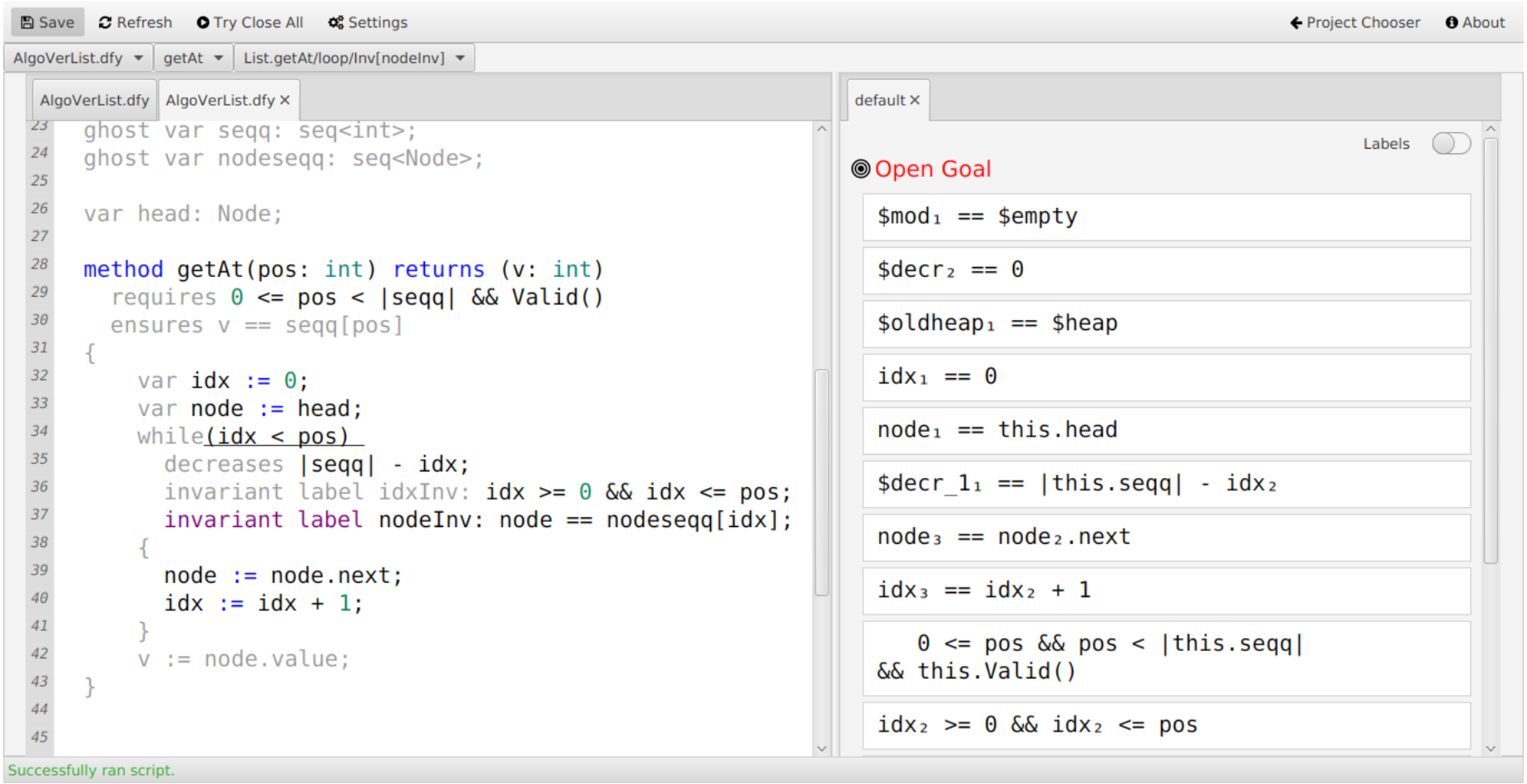
Exemplary Walkthrough



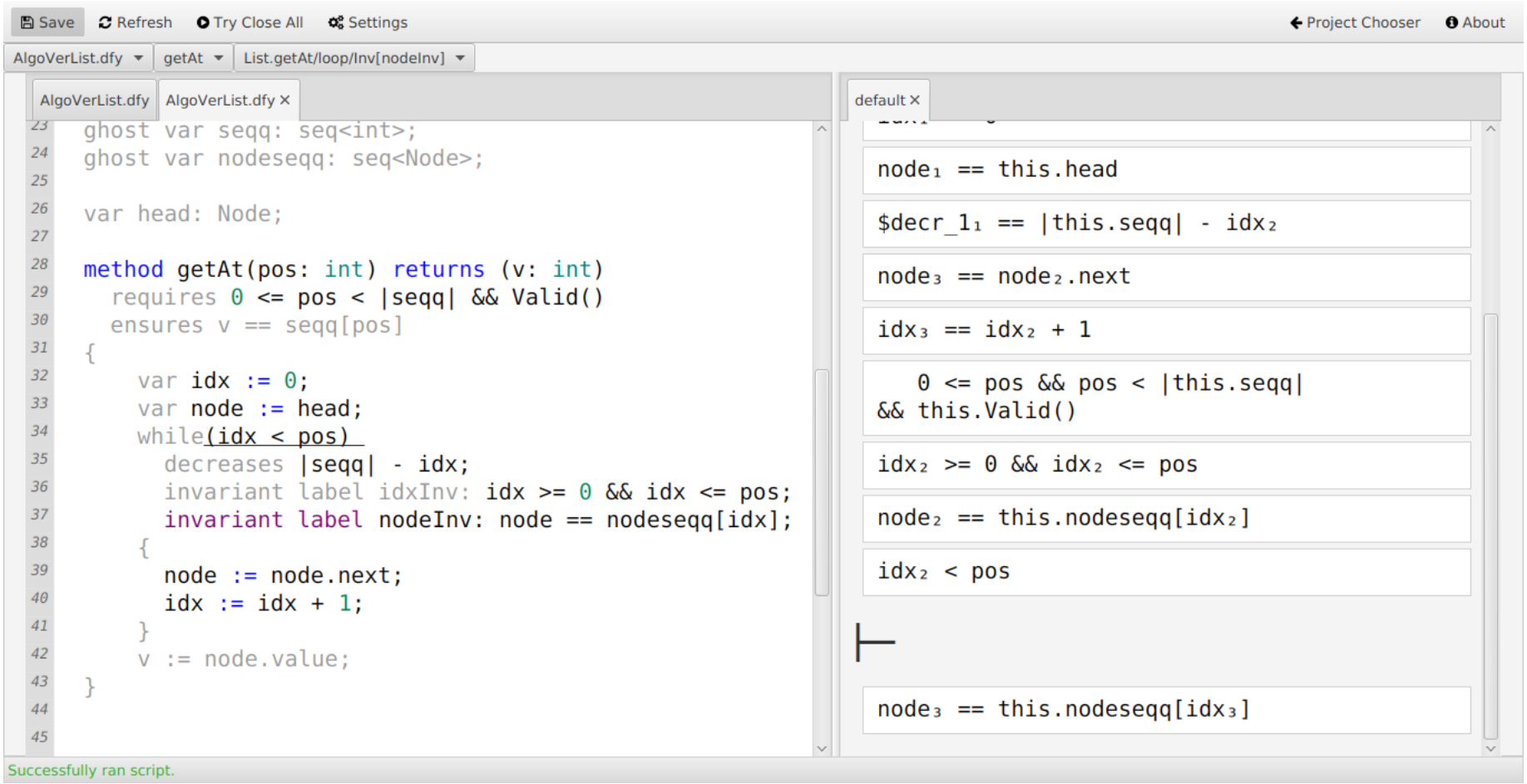
Exemplary Walkthrough



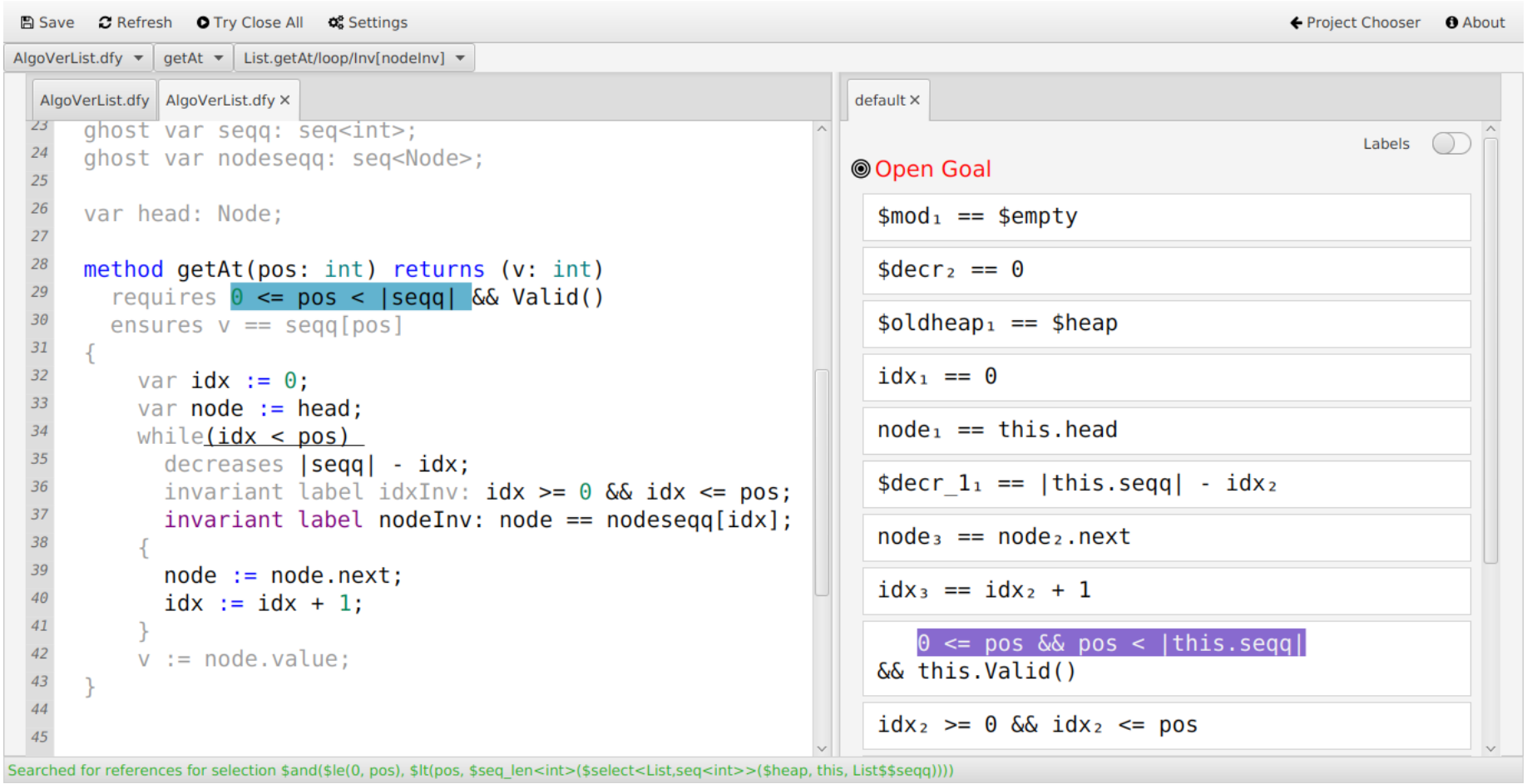
Exemplary Walkthrough



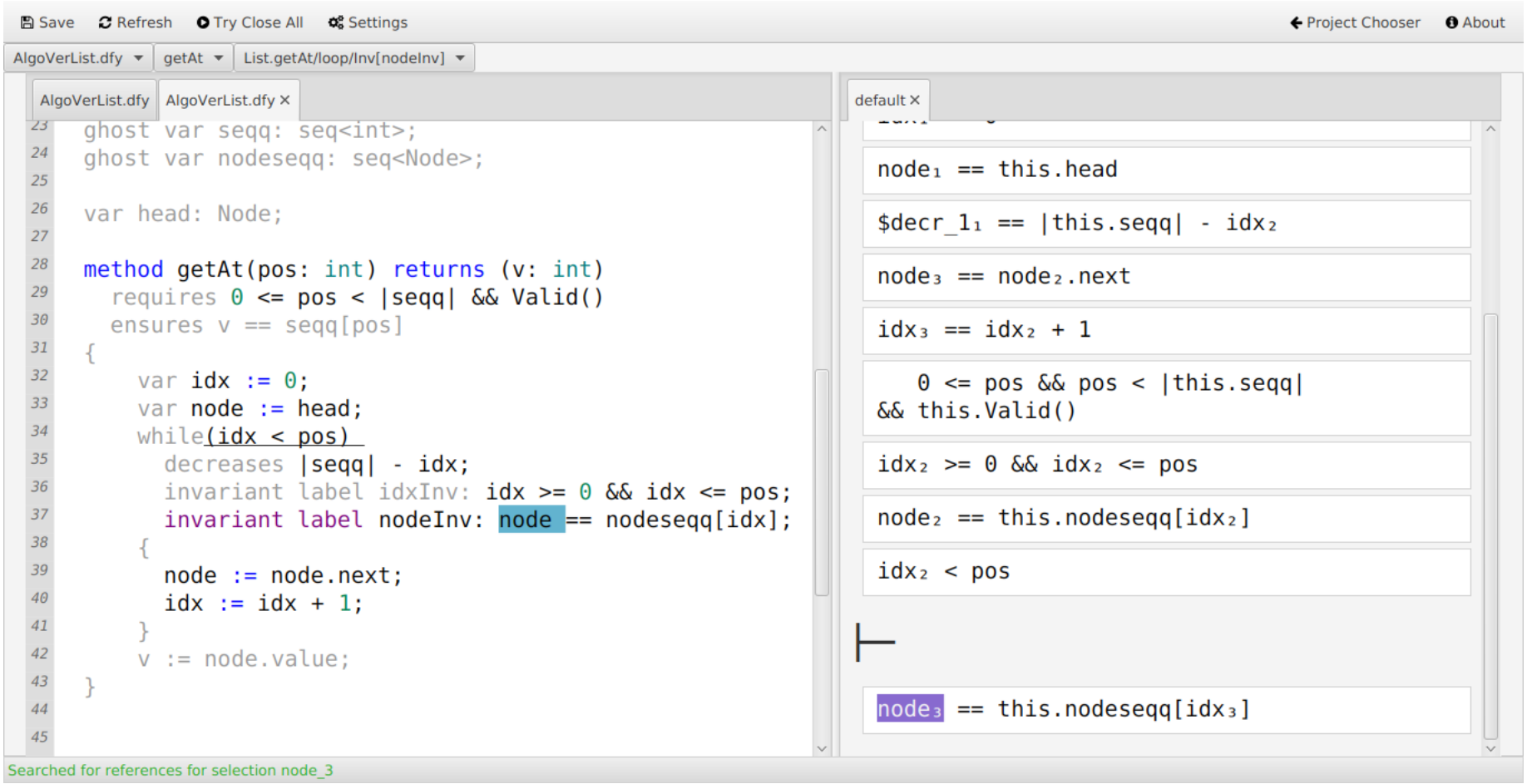
Exemplary Walkthrough



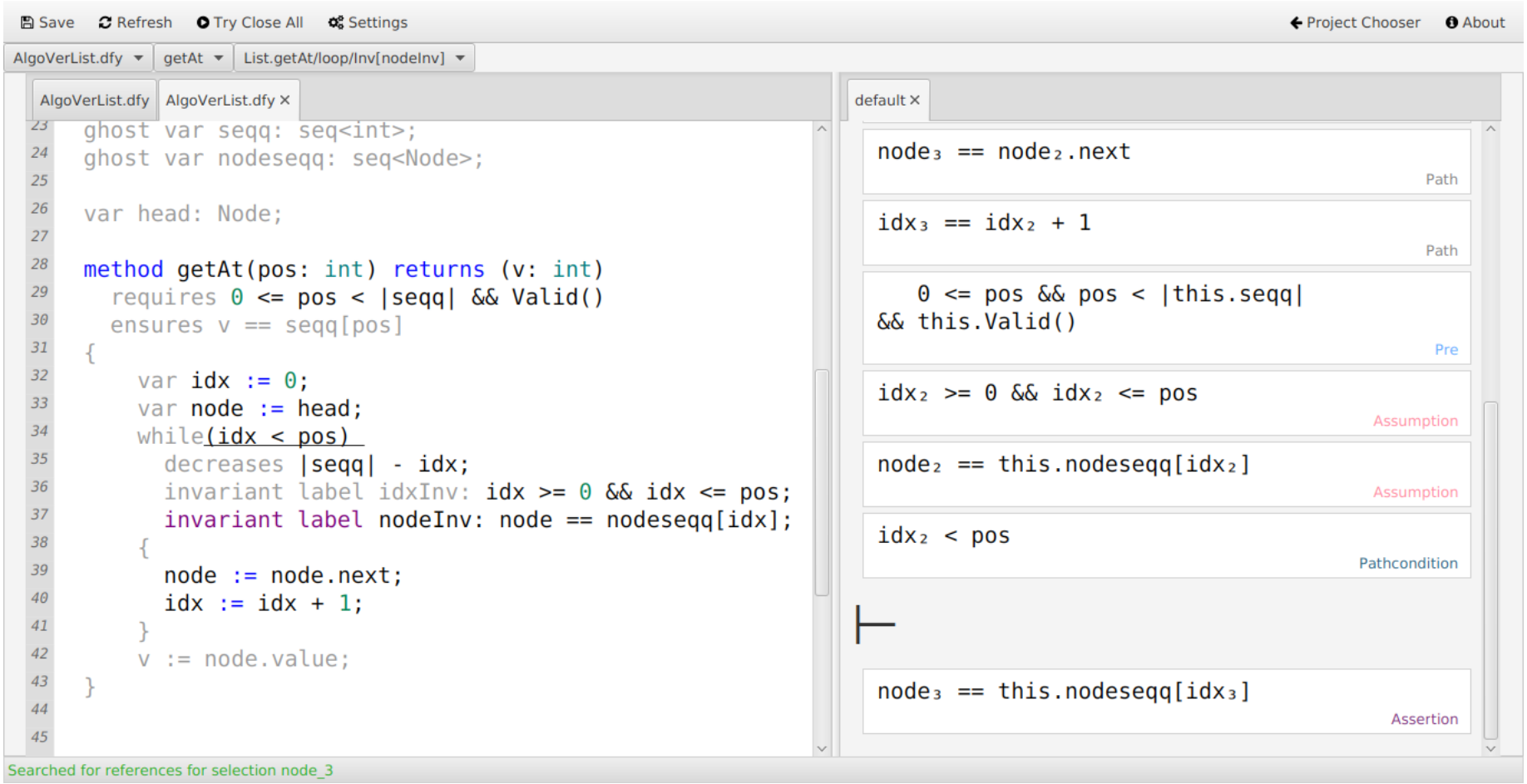
Exemplary Walkthrough



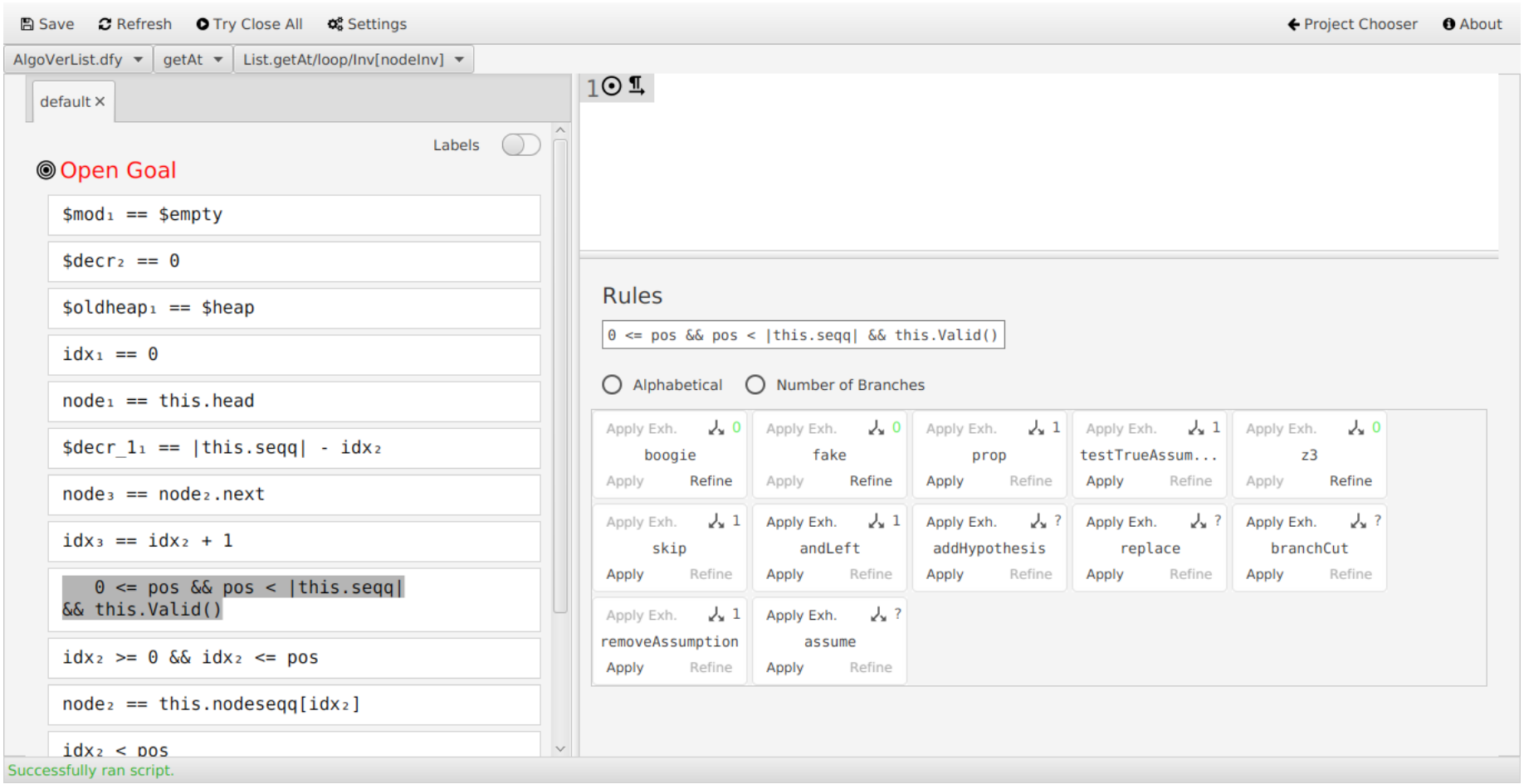
Exemplary Walkthrough



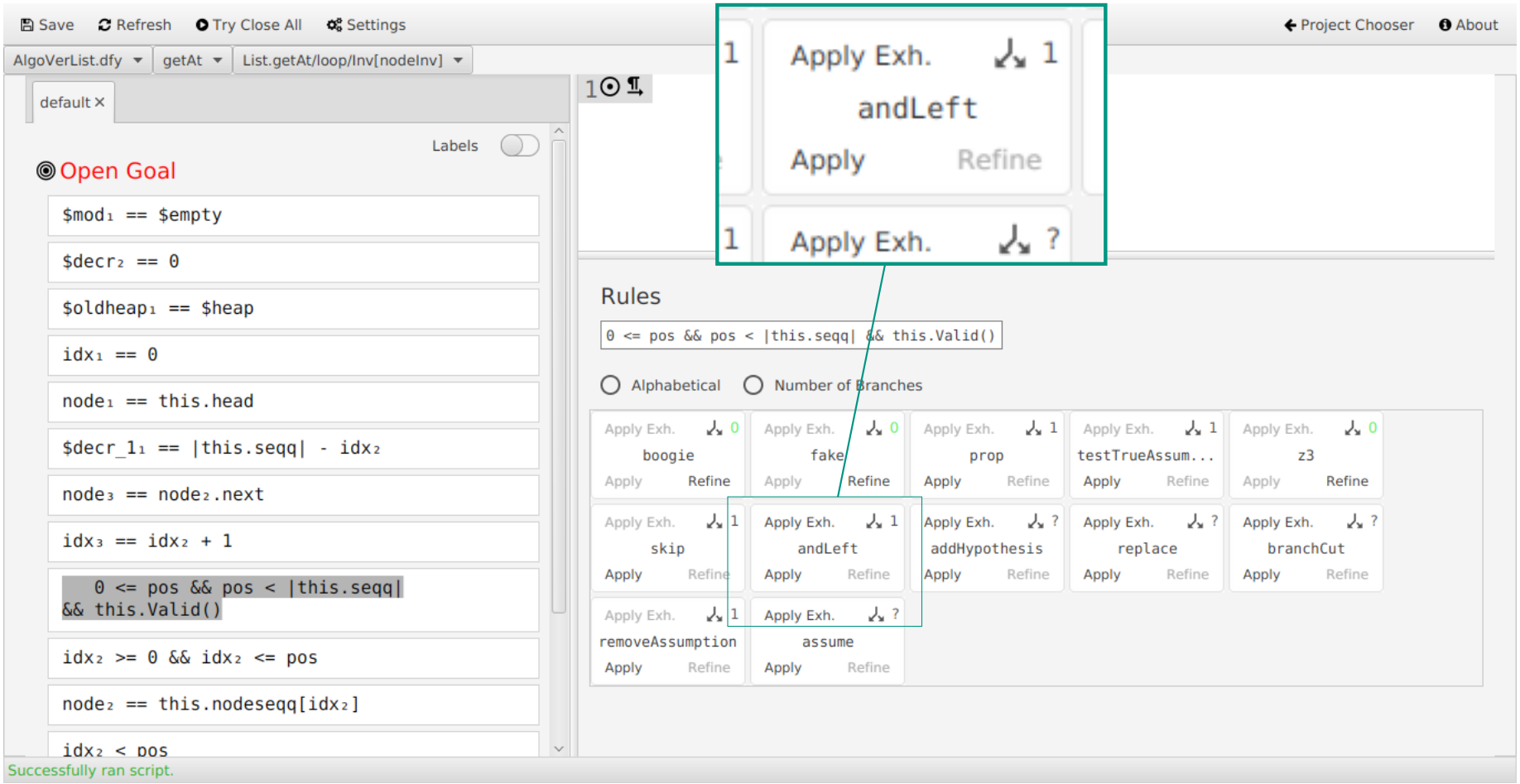
Exemplary Walkthrough



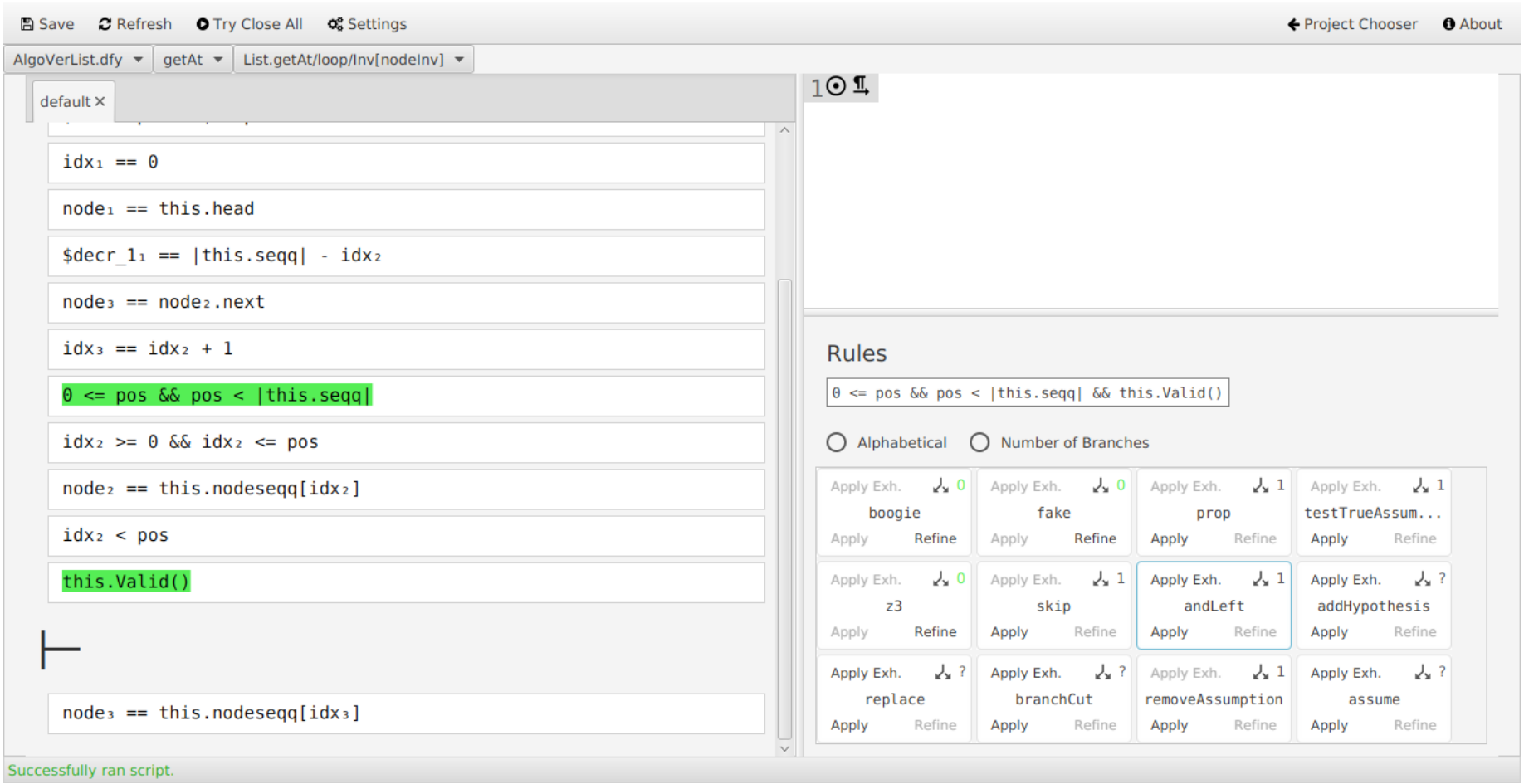
Exemplary Walkthrough



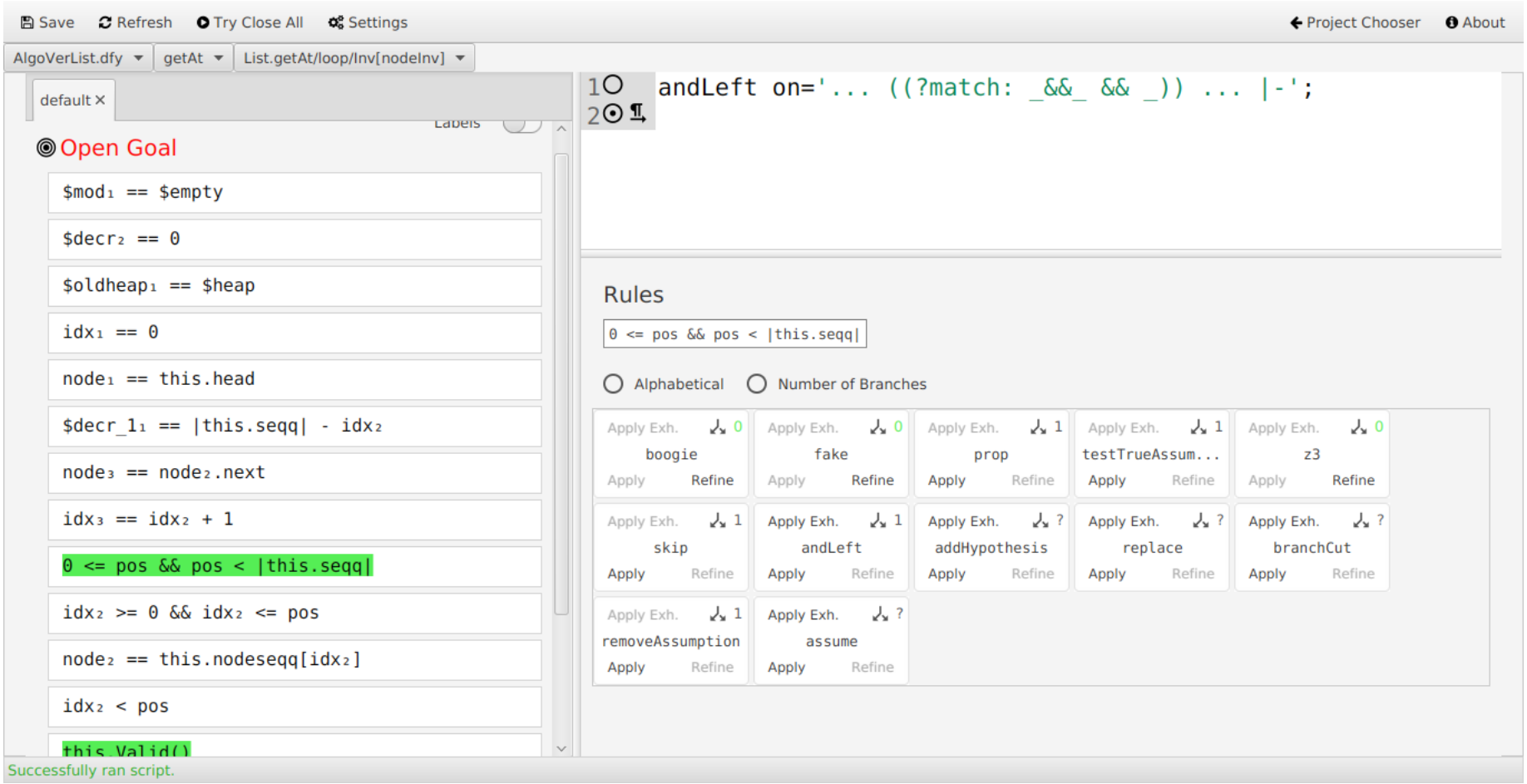
Exemplary Walkthrough



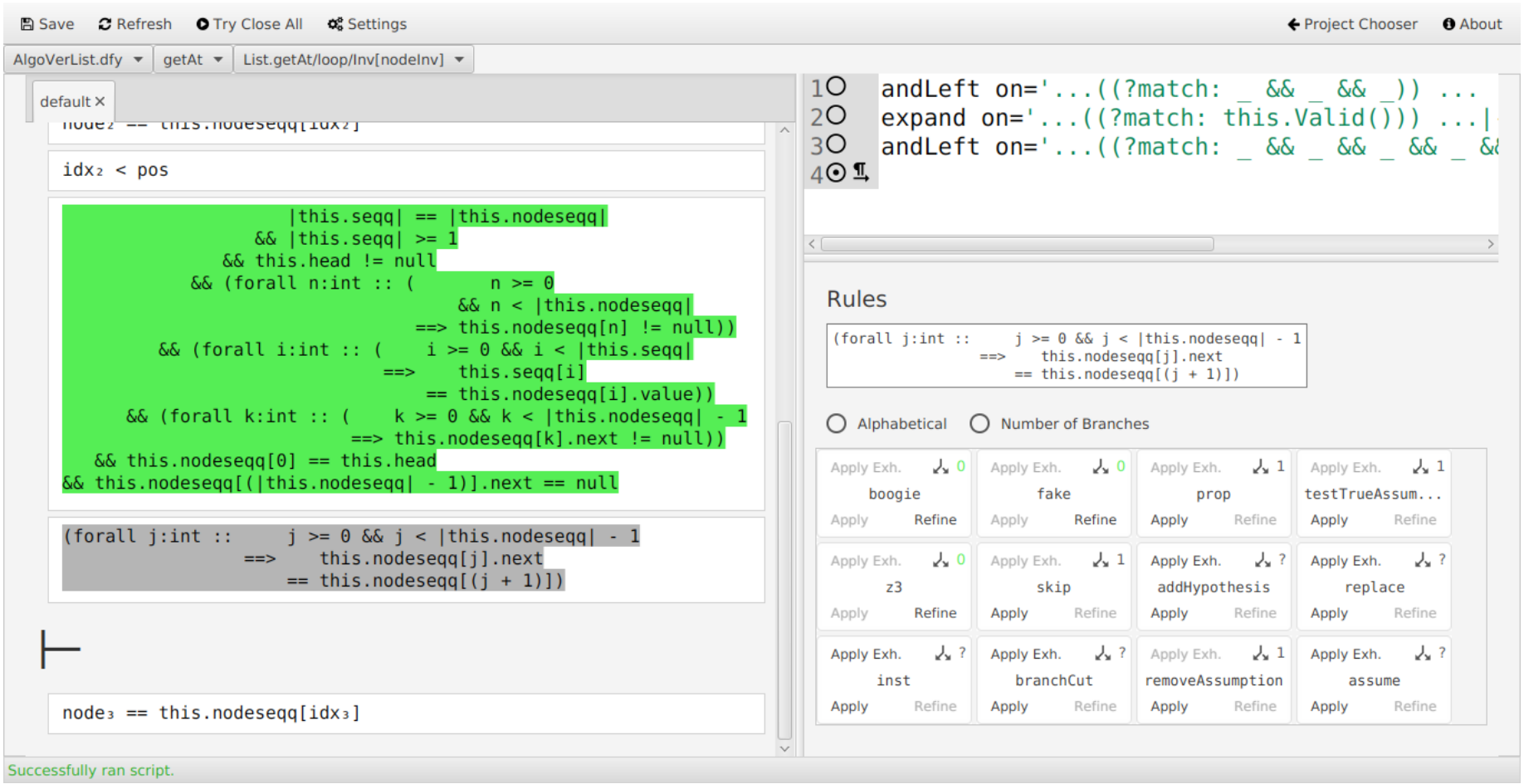
Exemplary Walkthrough



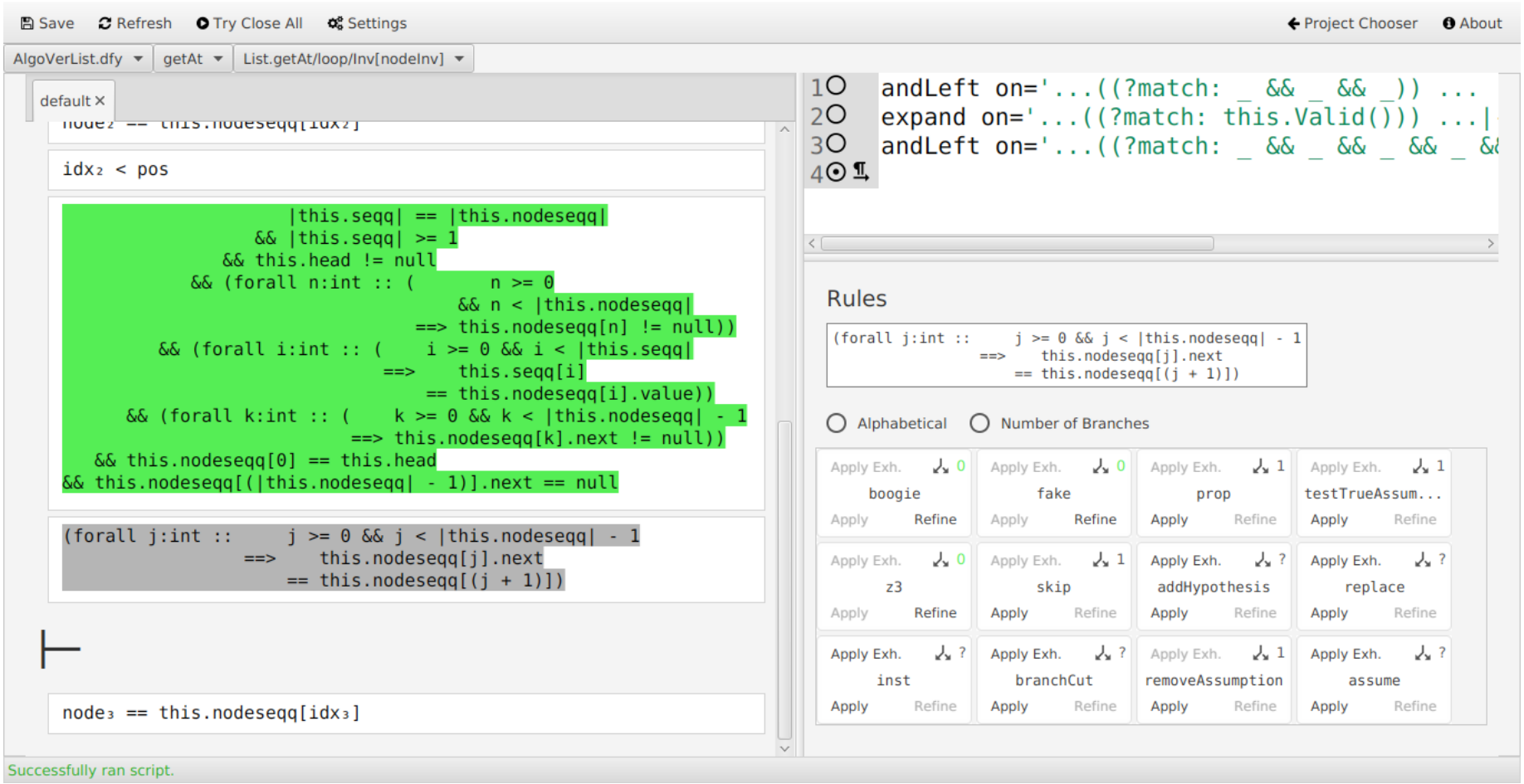
Exemplary Walkthrough



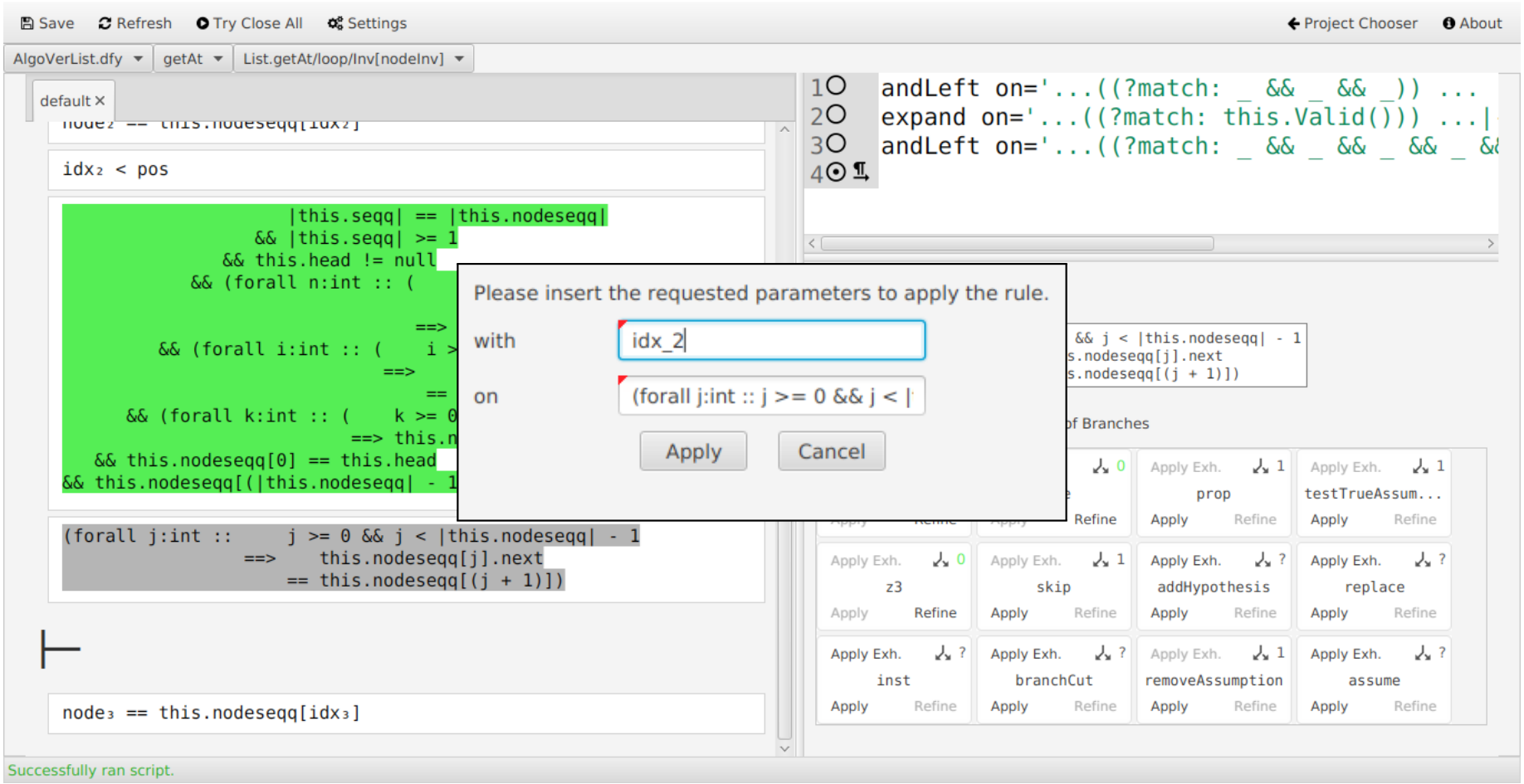
Exemplary Walkthrough



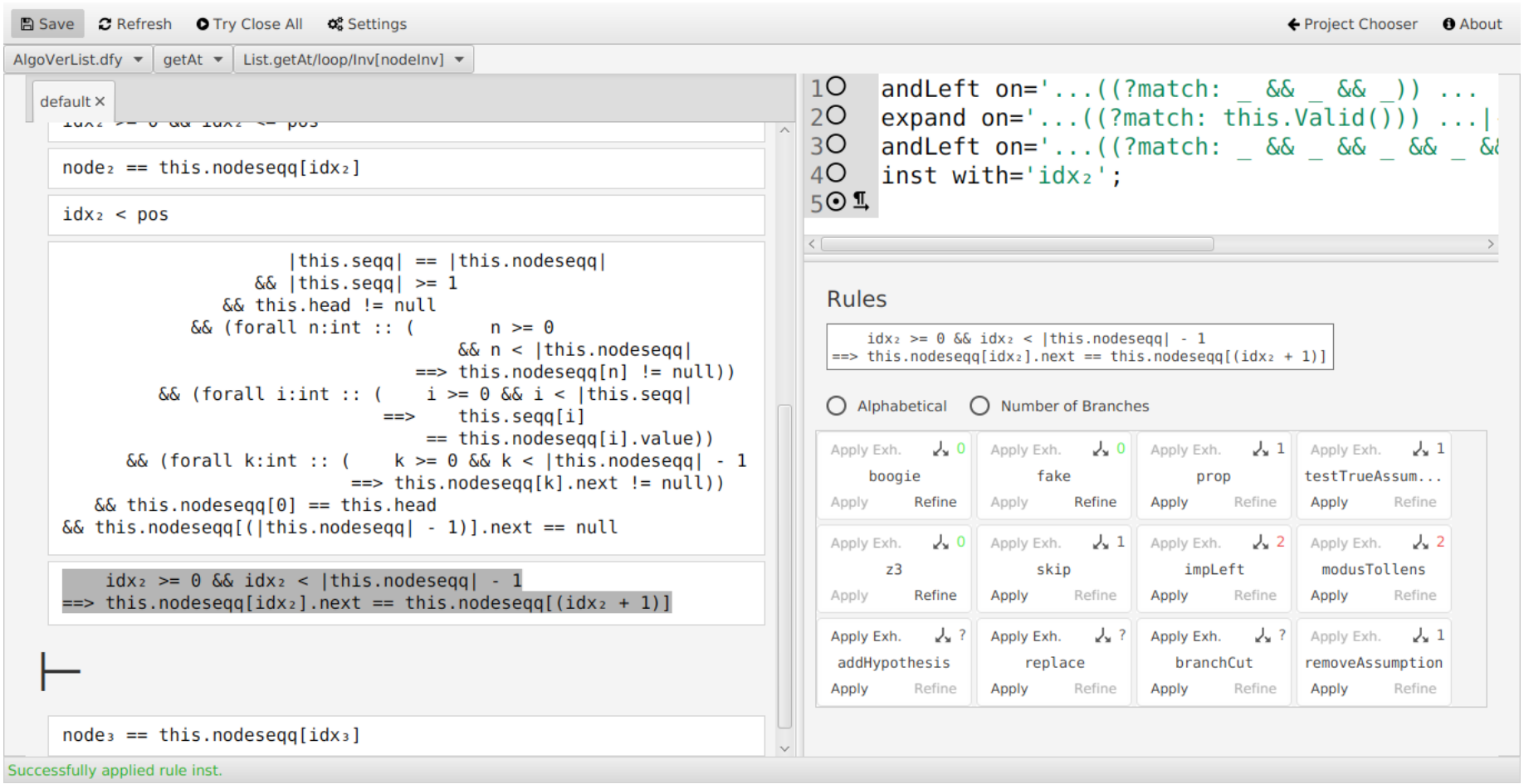
Exemplary Walkthrough



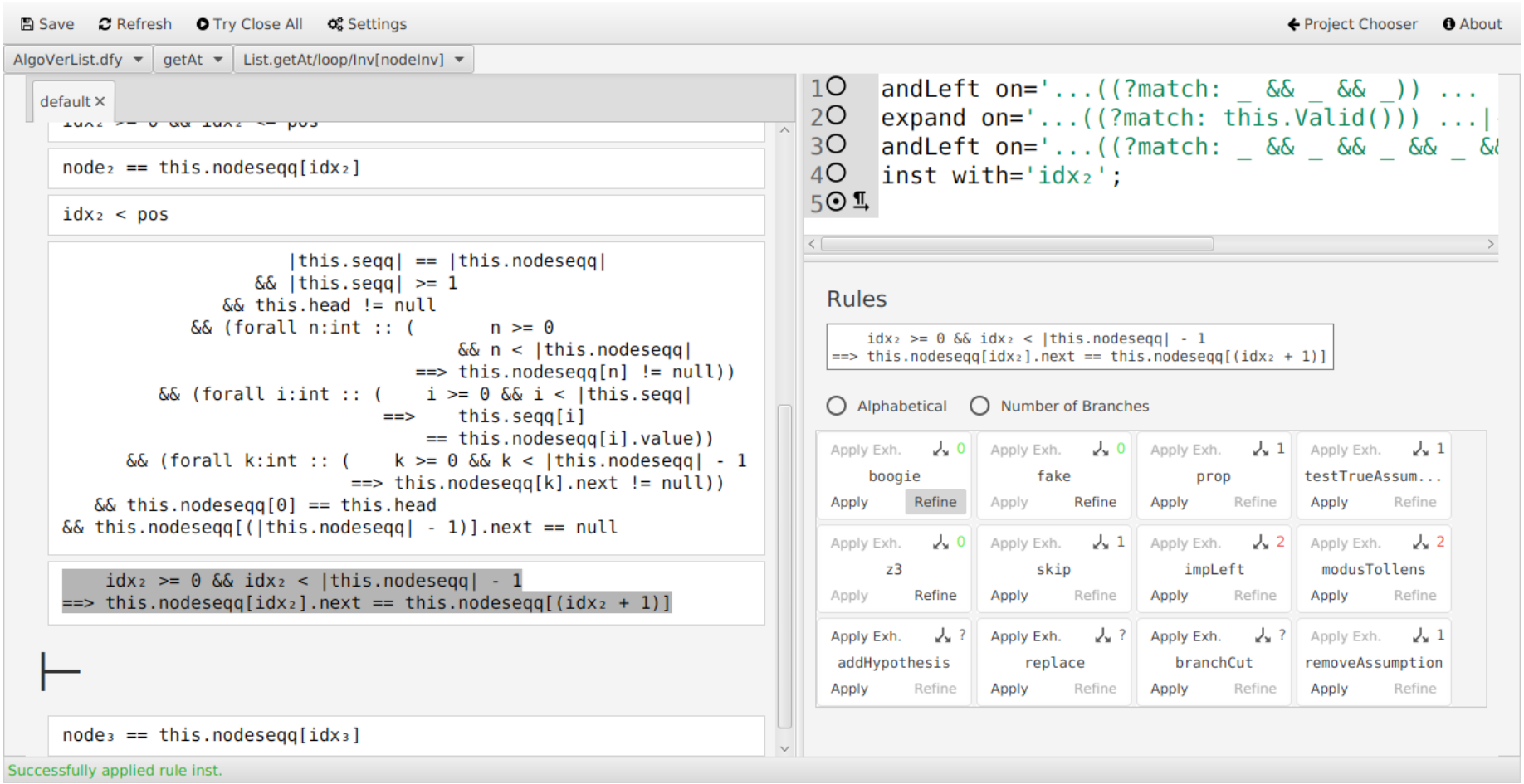
Exemplary Walkthrough



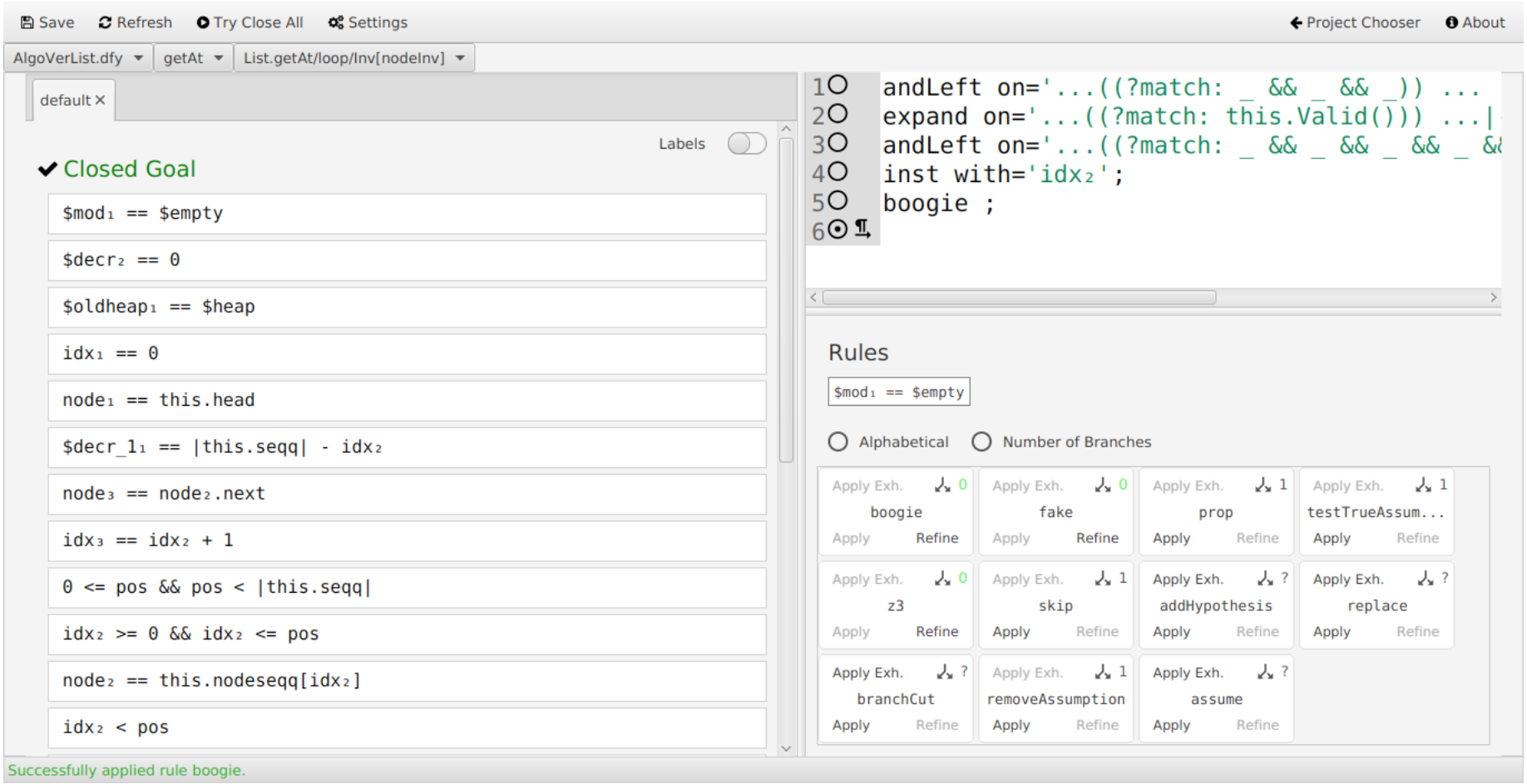
Exemplary Walkthrough



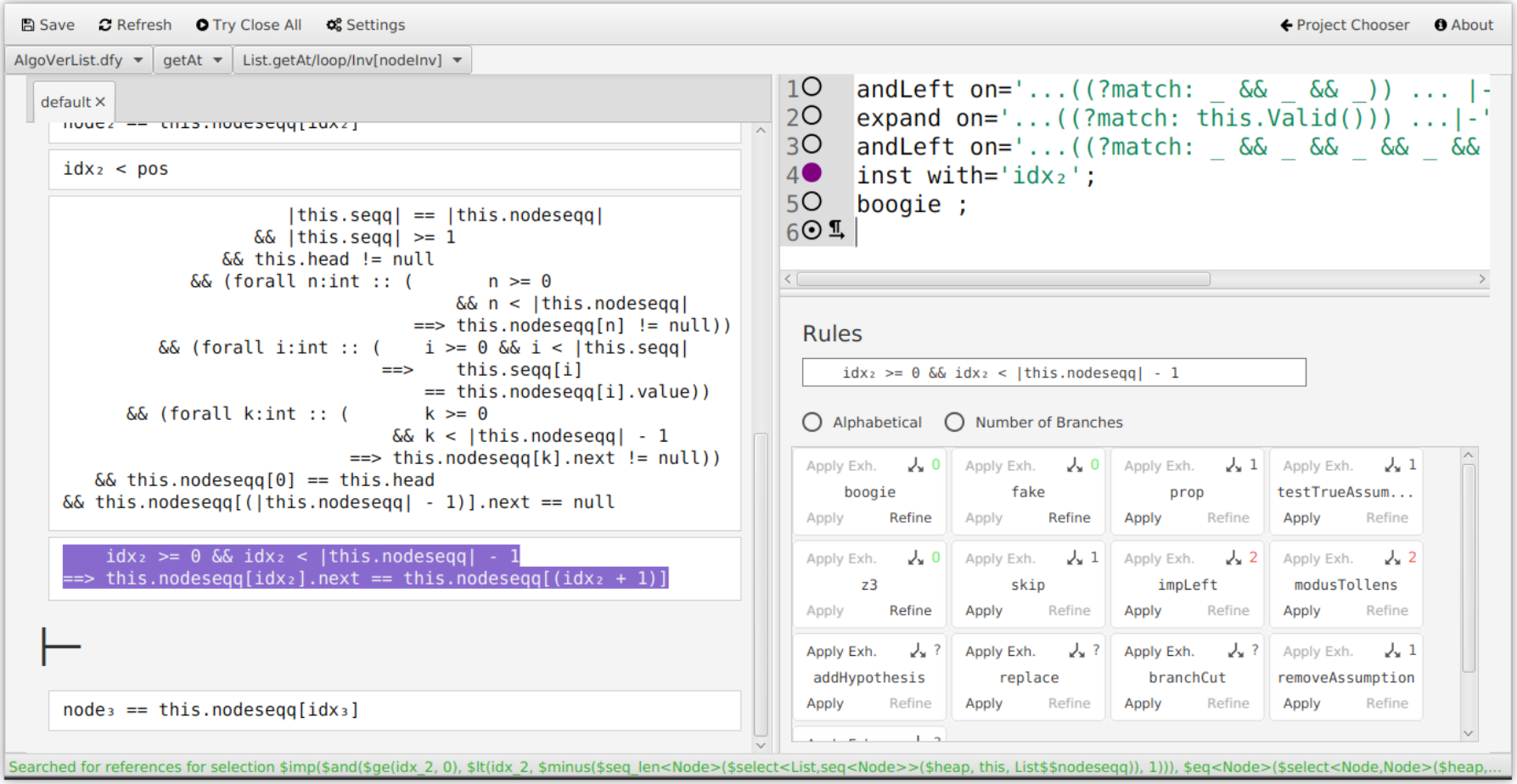
Exemplary Walkthrough



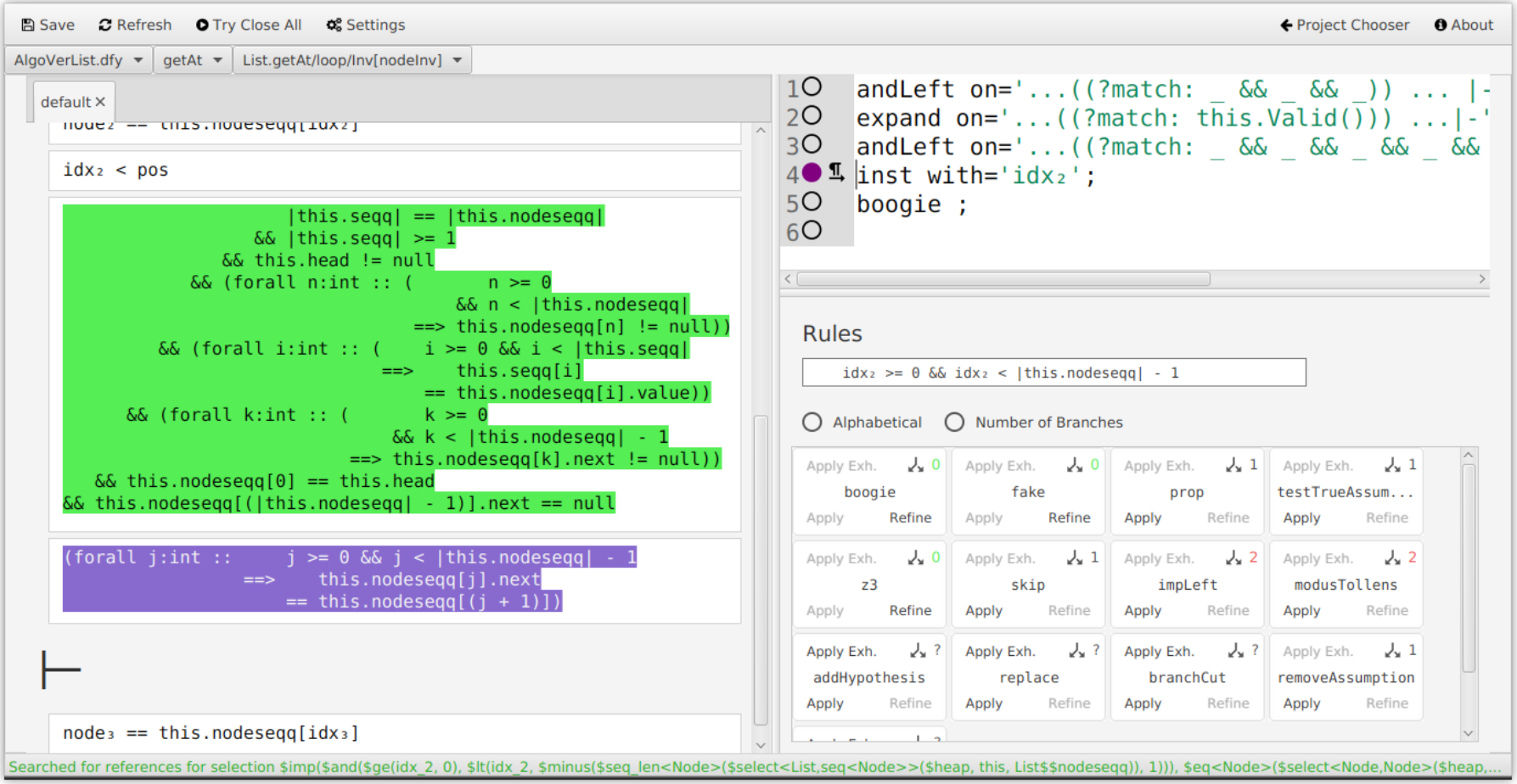
Exemplary Walkthrough



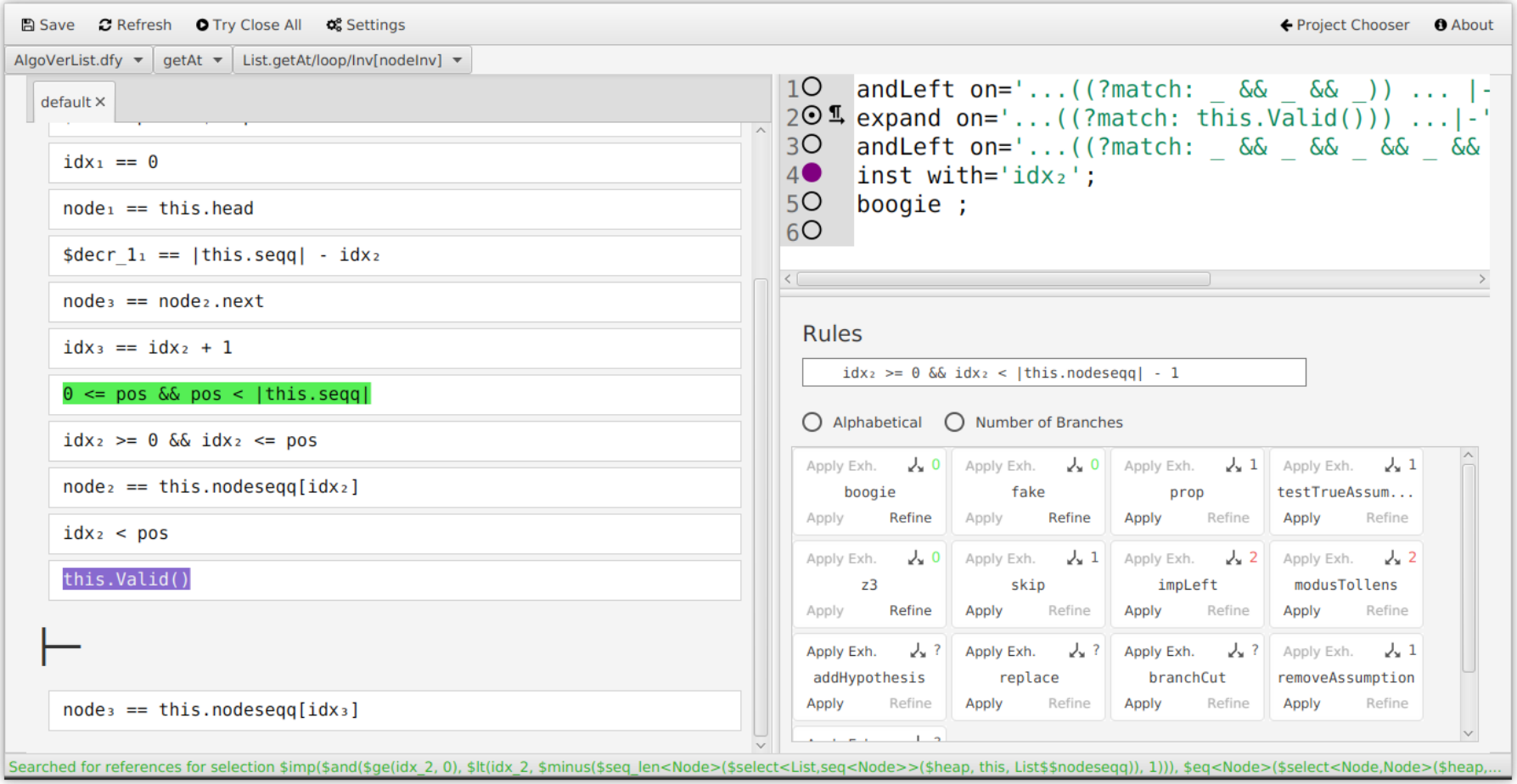
Relation in the Proof



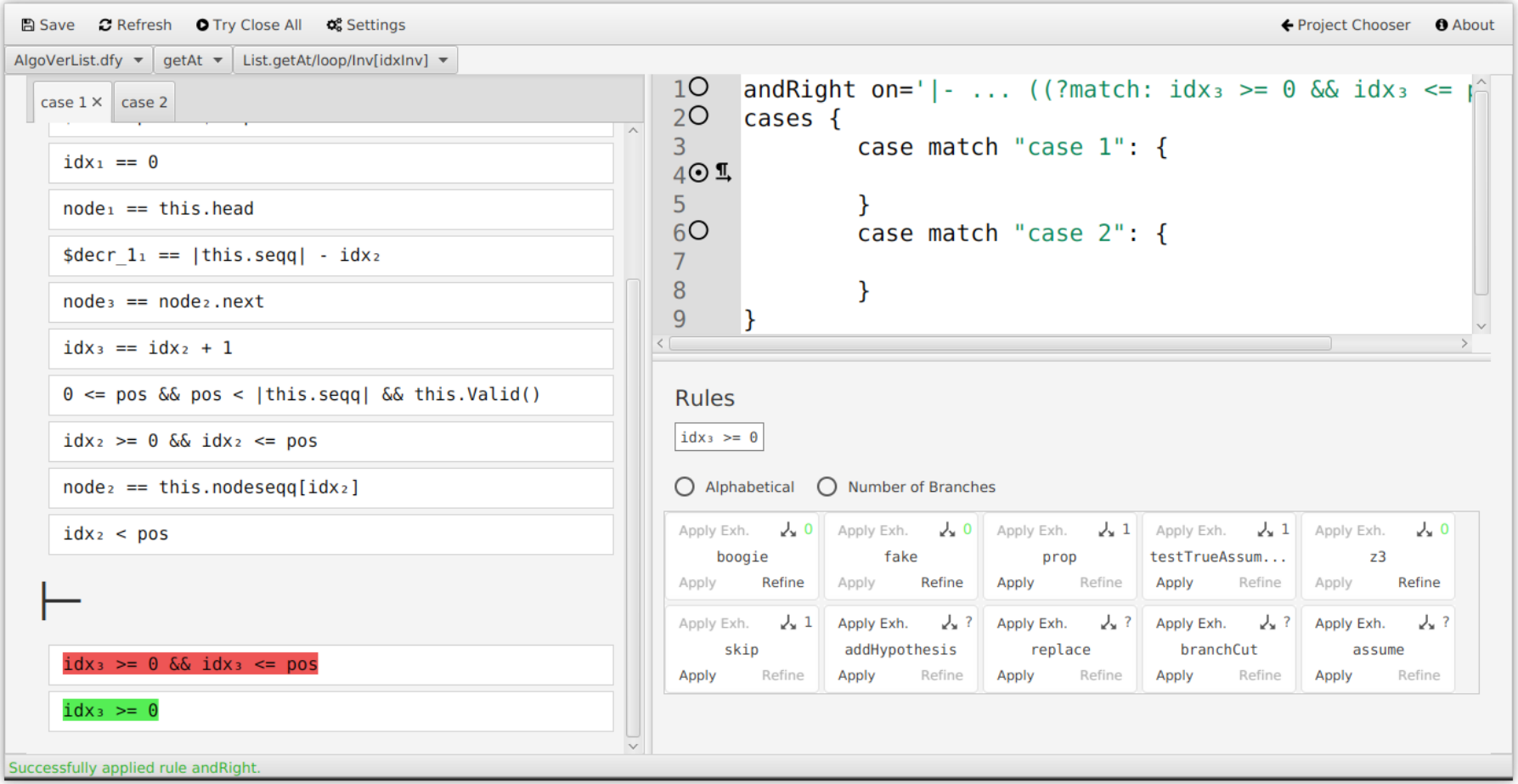
Relation in the Proof



Relation in the Proof

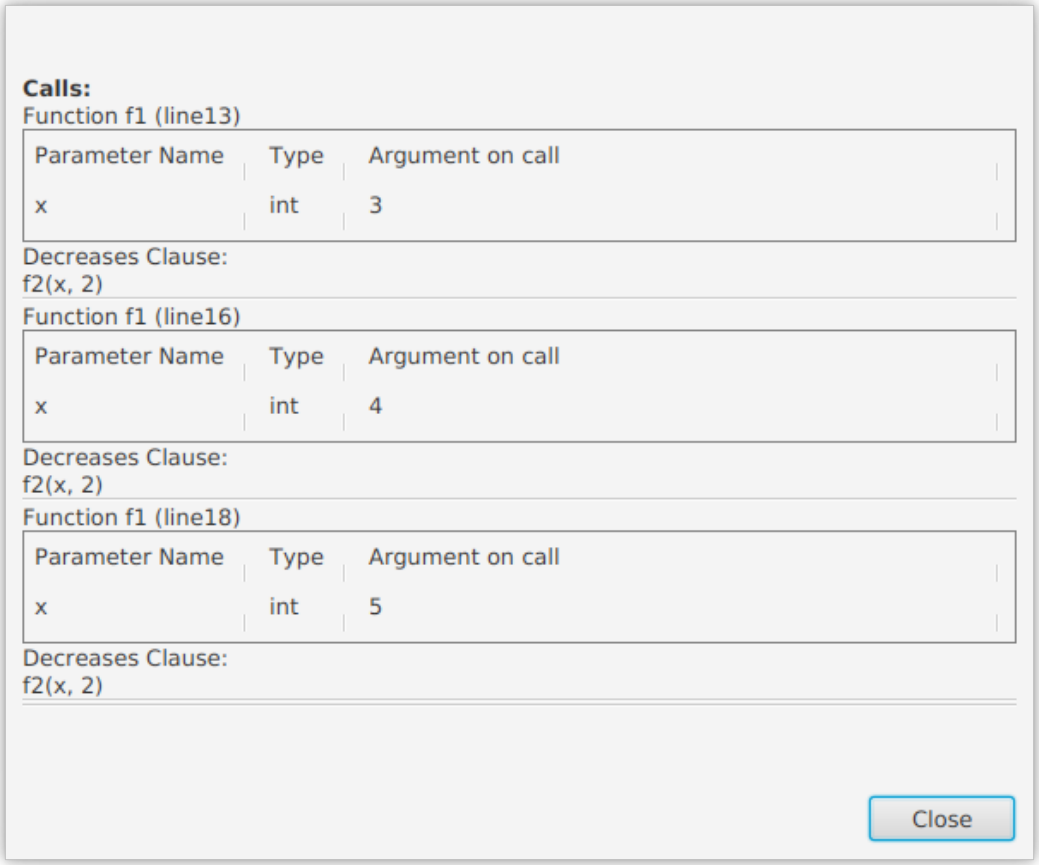


Branching in a Proof



Further Features and WIP

- simple lemmas can be encoded using Dafny syntax allowing to add rewrite rules
- call and callsite information is available



Related Work

Provide different views

e.g., KIV, Why3, KeY, KeYmaeraX, Coq, Isabelle,...

Combination of automation and interaction

Why3 (hierachical structuring of proof task and usage of different solvers)

Interaction styles

- auto-active verification: VCC, Dafny, OpenJML
- direct manipulation: KeY, KeYmaeraX, KIV
- text-based: Coq, Isabelle
- combination direct manipulation + text-based: KeYmaeraX

Verification IDEs

e.g., Why3, DafnyIDE

Summary and Conclusion

Our Contributions

- qualitative user studies on interaction in interactive verification systems
- a new user interaction concept
  - combination of different interaction styles
  - application of usability principles
    (e.g., visual clarity, substituitivity, anticipation, consistency, ...)
  - integration of supporting features
- an implementation of the concept in the research prototype DIVE
  (`https://github.com/mattulbrich/dive`)

Summary and Conclusion

Conclusion

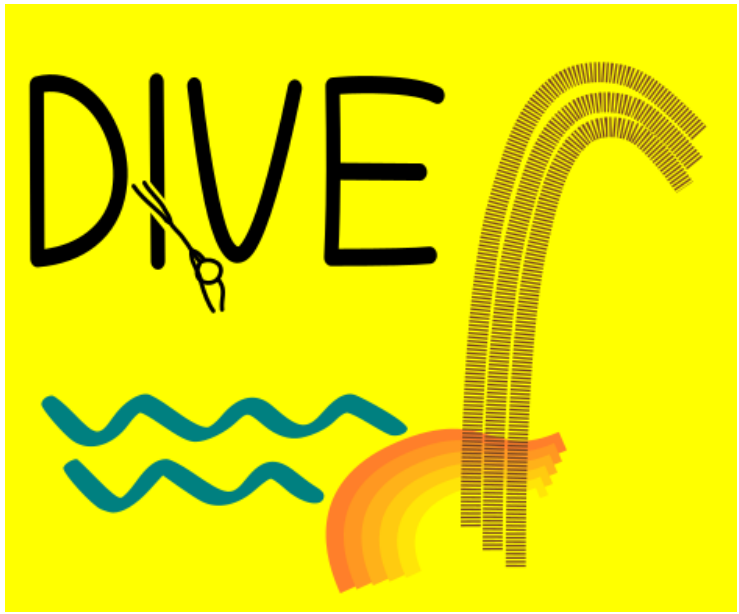
- proof state comprehension is one of the central issues
- there is not one single interaction strategy
- interaction styles can be combined for an improved user support

Future Work

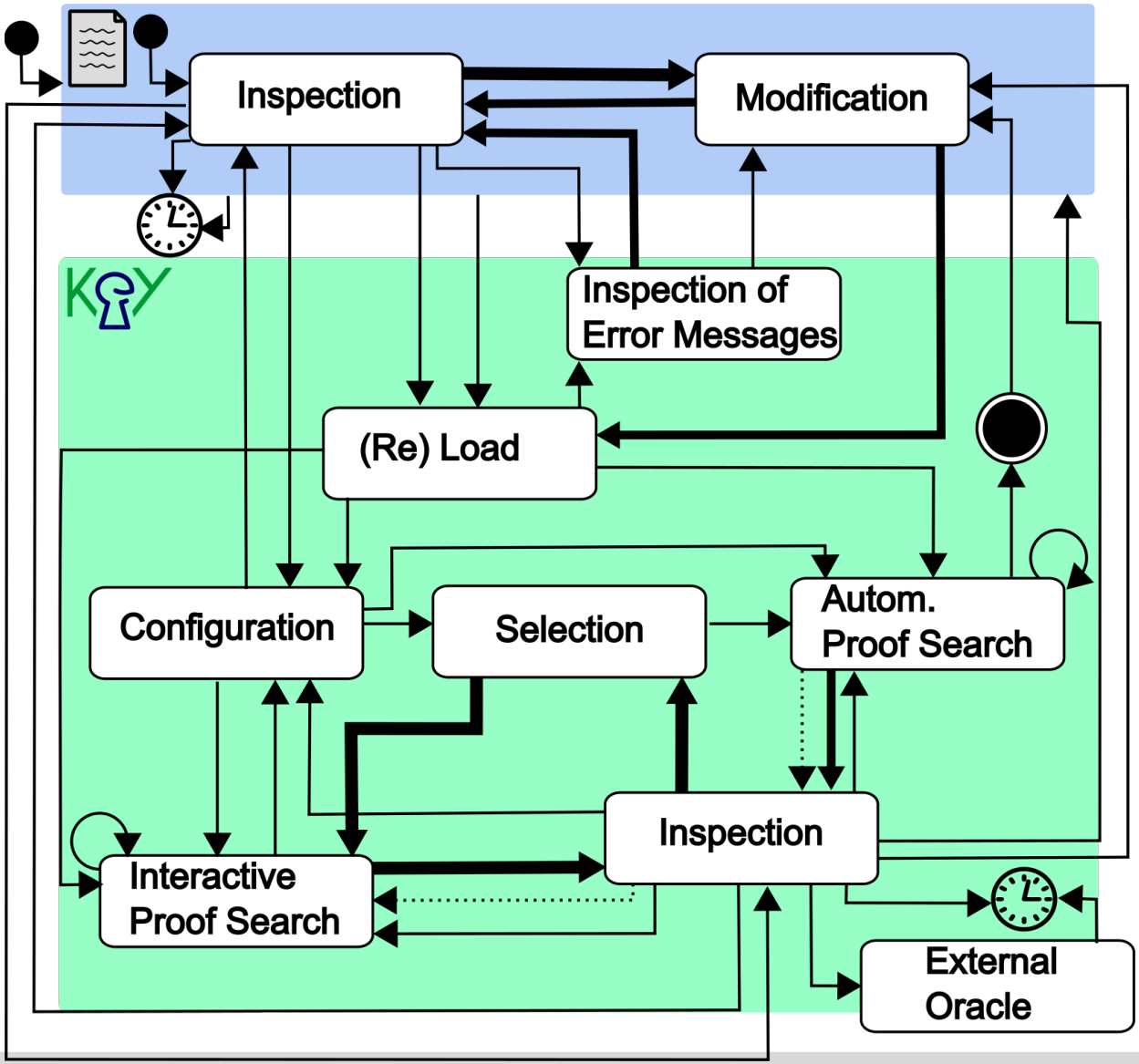
- encode user interactions back to annotations
- include counter examples in the DIVE user interface
- larger case studies and quantitative user studies

Thank you for your attention!

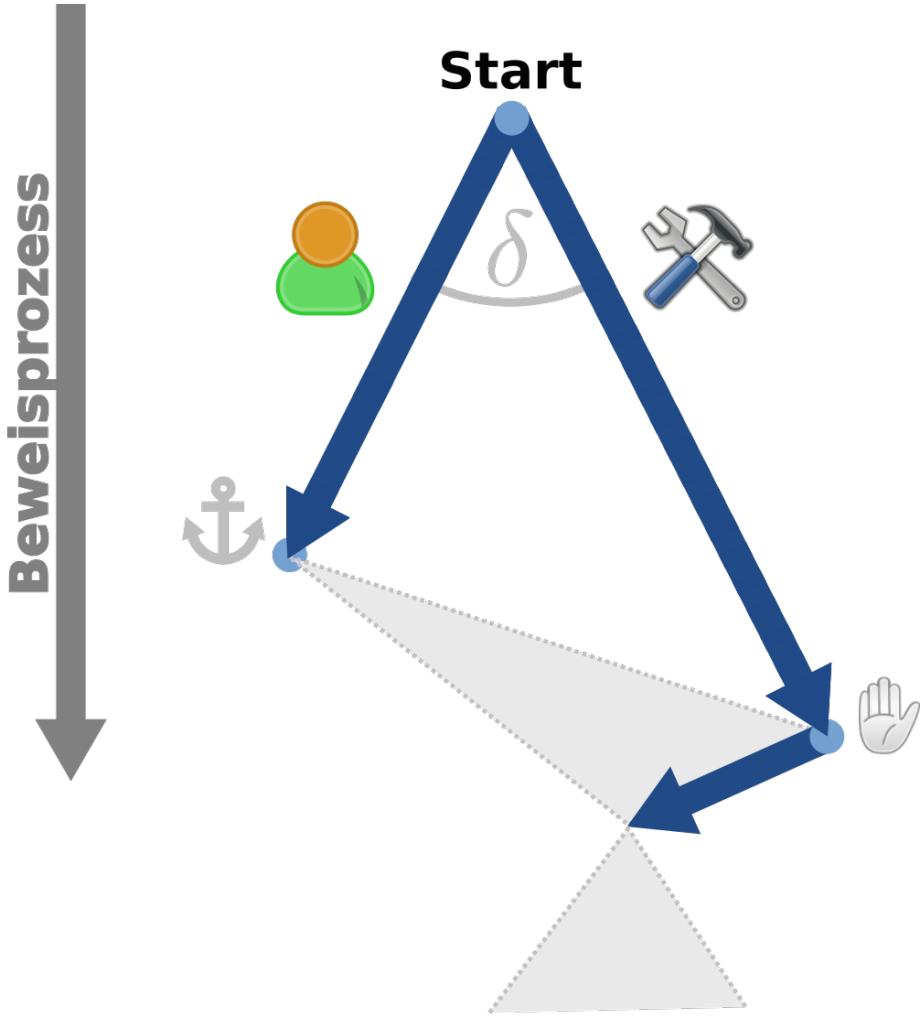
https://github.com/mattulbrich/dive



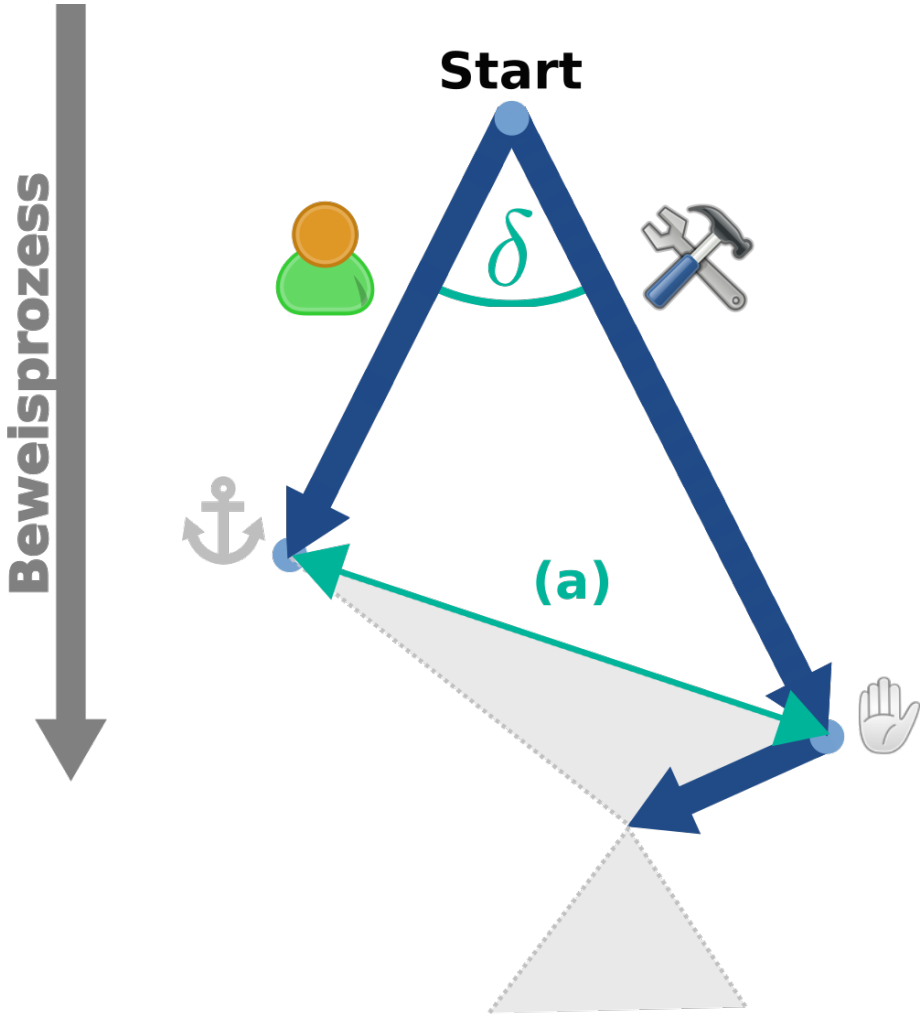
Consolidated Sequence Model



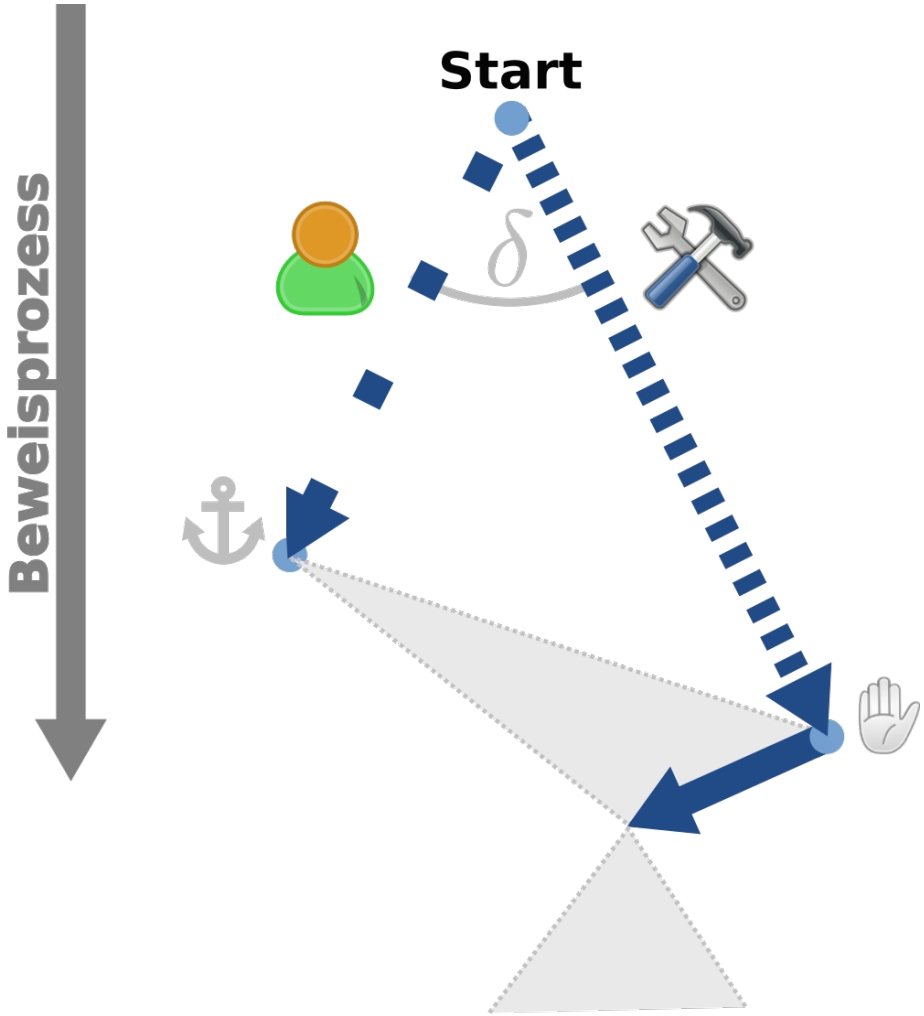
Model of Process



Model of Process



Model of Process



Model of Process