

REFINITY to Model and Prove Program Transformation Rules

APLAS 2020 · December 2nd, 2020

Dominic Steinhöfel

✉ steinhoefel@cs.tu-darmstadt.de

🐦 [@DomSteinhoefel](https://twitter.com/DomSteinhoefel)

🌐 dominic-steinhoefel.de

Software Engineering Group · CS Department · TU Darmstadt

A photograph of a protest scene. In the foreground, a person's arm is raised, holding a cardboard sign. The sign has the word 'STOP' in large black letters, followed by a red exclamation mark, and the words 'COME NOW' in red letters below it. The background is a blurred crowd of people and a building with a flag. The text 'DEMO TIME!' is overlaid in white, bold, sans-serif font across the center of the image.

DEMO TIME!

Photo by Markus Spiske on Unsplash

CASE STUDIES

A person with a backpack is standing on a rocky mountain ridge, looking out over a vast mountain range. The sky is filled with dramatic, dark clouds. The overall tone is moody and adventurous.

Photo by Wojciech Then on Unsplash

Refactoring

```
try {  
  </>  
} catch ( ... ) {  
  </>  
}
```



```
if ( </> ) {  
  </>  
} else ( ... ) {  
  </>  
}
```

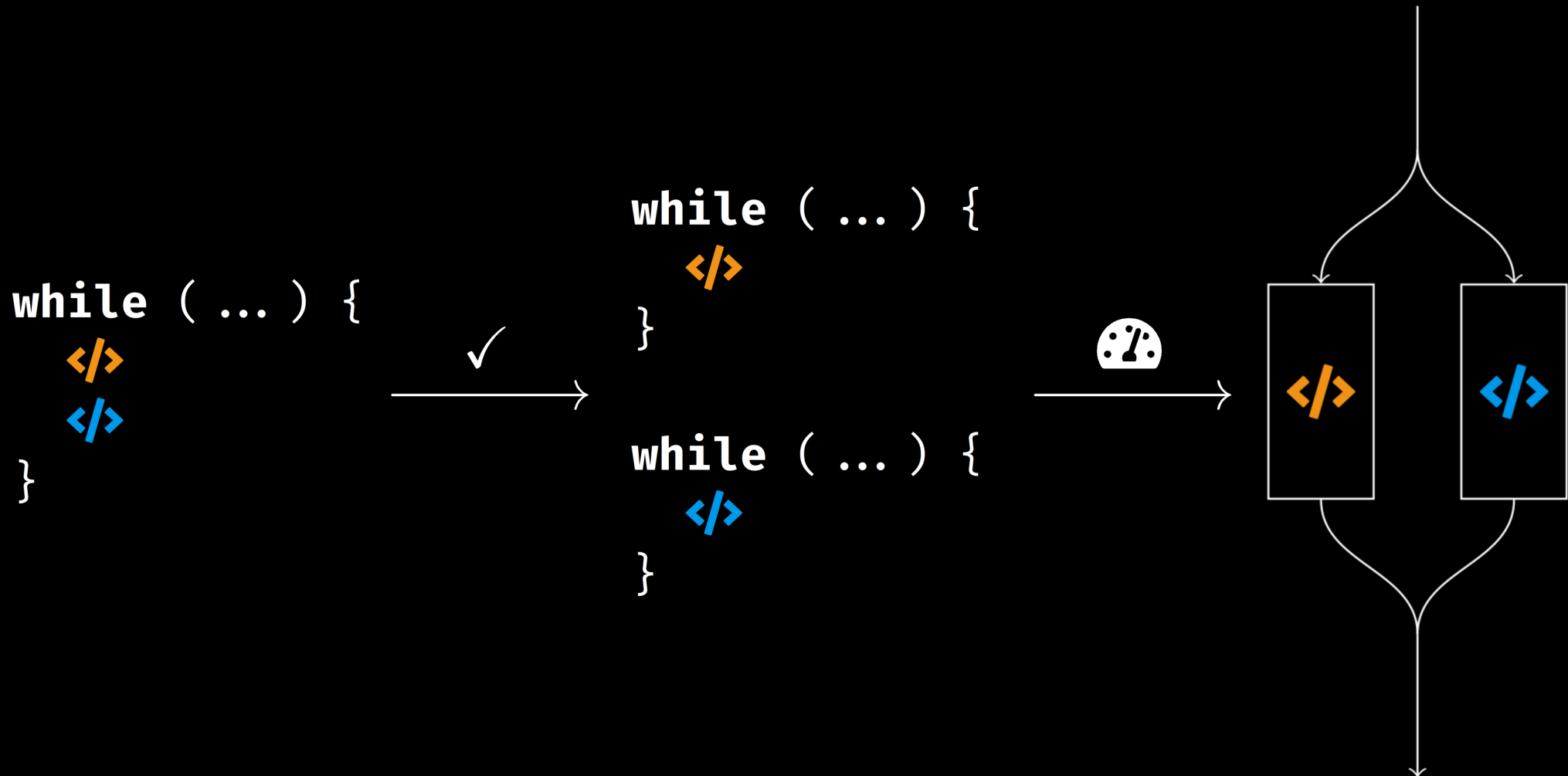

Abstract Cost Analysis

```
while ( ... ) {  
  </>  
  </>  
}
```

$\$ \$ \$ \$ \$ \$ \$$
→

```
</>  
while ( ... ) {  
  </>  
}
```

Restructuring for Parallelization





An aerial photograph of a vast wind farm. Numerous white wind turbines are scattered across a lush green, rolling landscape. In the background, a body of water is visible under a sky filled with large, grey clouds. The overall scene conveys a sense of clean energy and natural beauty.

REFINITY Resources

Photo by Tyler Casey on Unsplash

Download

  The KeY Project

Automatically Proving the Correctness of Refactoring Rules with Abstract Execution and REFINITY

Documentation

[Online documentation for REFINITY](#) (*under ongoing development*) is available.

Downloads

[Download KeY with REFINITY](#)

[Get Support](#)

You need **at least a Java Development Kit (JDK) 11** to run the Jar above. Then, just run “`java -jar key-2.7-AE.jar`”. Have a look at the descriptions below to get started.

Example

To see an abstract program model for the equivalence proof of a refactoring in action, download and run KeY-AE and load the example “Extract Prefix” (see screenshot below) from the examples dialog. This will open the REFINITY window for inspection of the

key-project.org/REFINITY

REFINITY Documentation

REFINITY is a workbench for modeling and verifying Java-based program transformation rules. It is based on KeY and Abstract Execution. Here, we describe the user interface of REFINITY, how to specify program transformation rules with pre- and postconditions using the tool, and its file format. At the end of the document, you find helpful references / scientific publications (which can also be helpful).

This documentation is under ongoing development.

Last update: 2020-08-21.

Getting Started

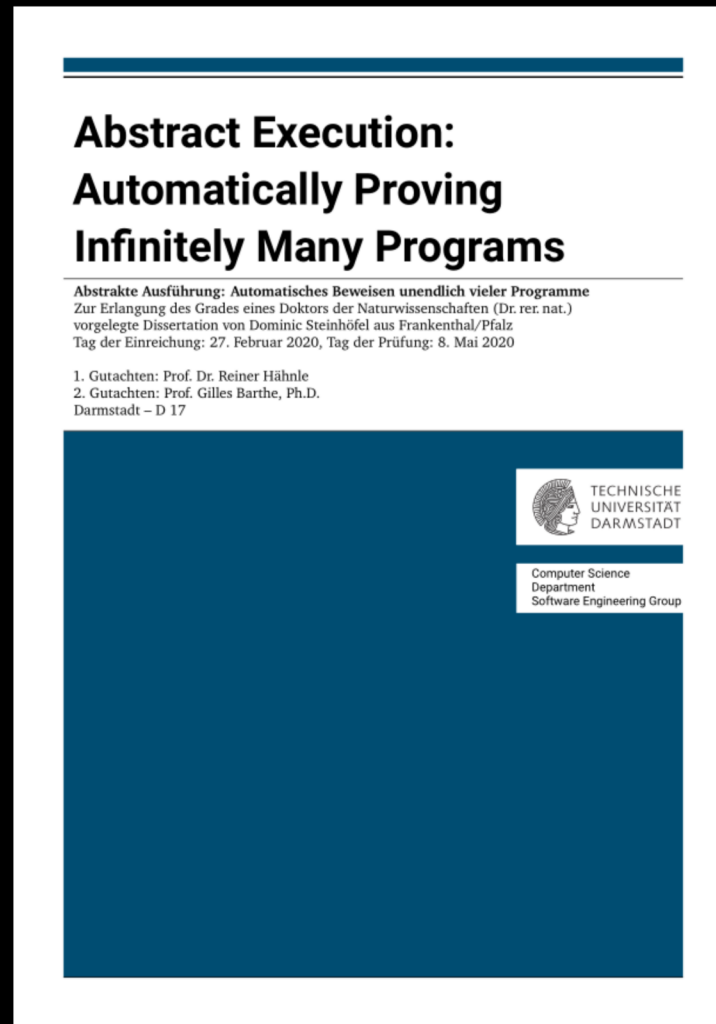
REFINITY is an add-on to the [KeY program prover](#). As of Aug 2020, it is available in the "[AbstractExecution](#)" [branch](#) of KeY. It is planned to merge it into the master branch until end 2020.

You can start REFINITY from within the graphical KeY user interface by pressing the big "REFINITY" button. If you do not see this button, check that you (1) use an up-to-date KeY version with Abstract Execution, and (2) the REFINITY extension is activated: In the GUI, choose Options > Show Settings > Extensions, check AE-Relational, apply and restart.

The User Interface

`key-project.org/material/REFINITYDoc/`

More Details & Background



<https://doi.org/10.25534/tuprints-00008540>

A tropical scene featuring a hammock strung between two palm trees. The hammock is made of light-colored rope and is empty. The palm trees have thick, textured trunks and green fronds. The background is a bright blue sky with scattered white clouds. The overall mood is peaceful and relaxing.

CONCLUSION

Photo by Mohamed Ajufaan on Unsplash

Transformation
rules as pairs of
schematic programs

  →  

Proof generation
and certificate
checking



REFINITY

Editing support:
Shortcuts, syntax
checking, tool tips, ...

Instance checking

✉ steinhoefel@cs.tu-darmstadt.de

🐦 [@DomSteinhoefel](https://twitter.com/DomSteinhoefel)

🌐 dominic-steinhoefel.de