WLP-based Testing

Progress Report

Orestis Melkonian

October 25, 2017

Universiteit Utrecht

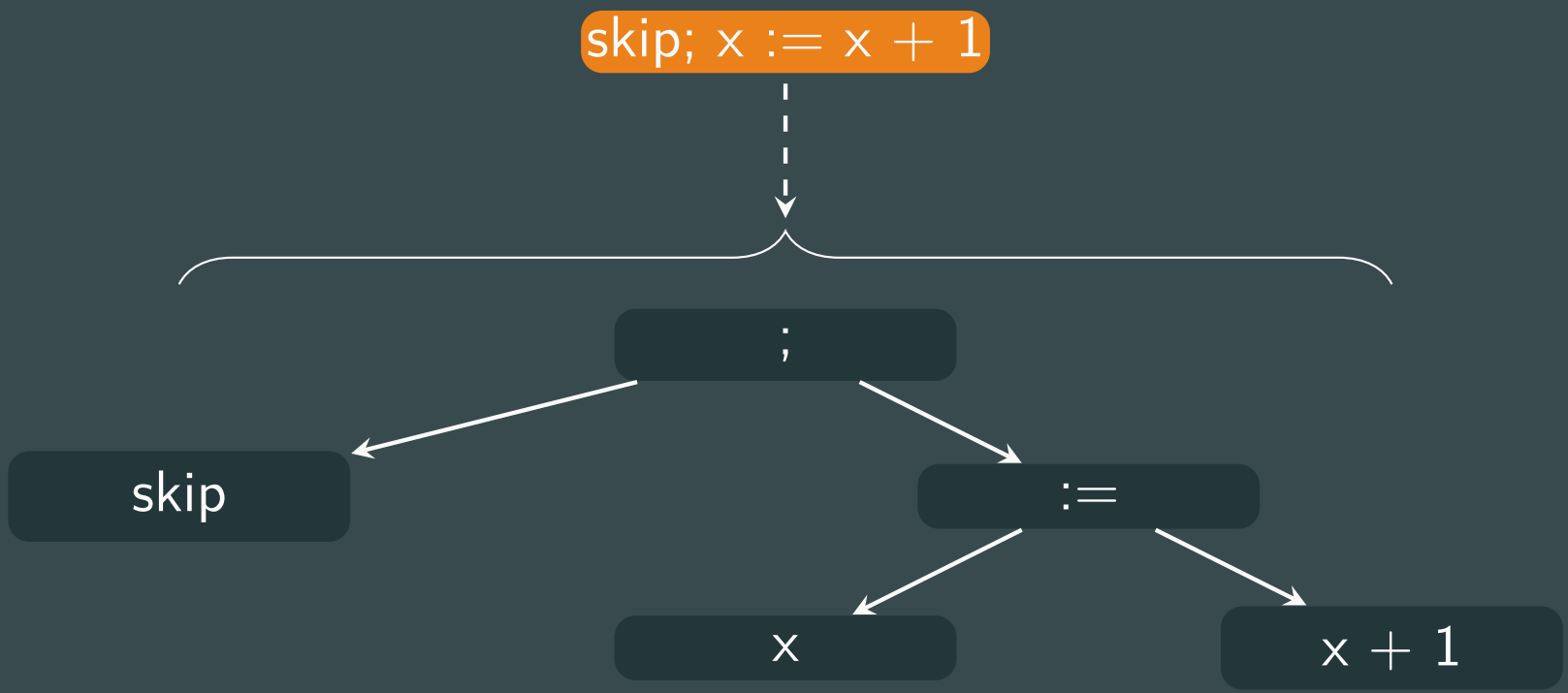
Overview

- Parsing
- Calculating paths
- Renaming
- Calculating WLP
- Replacing Conditionals
- Normalizing
- SMT-solving

Parsing

Transforms given GCL file to an internal AST.

Example

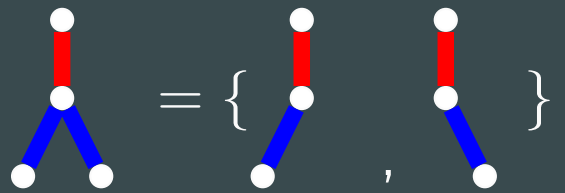


Paths

Calculate all possible program paths up to some length. Branching statements are:

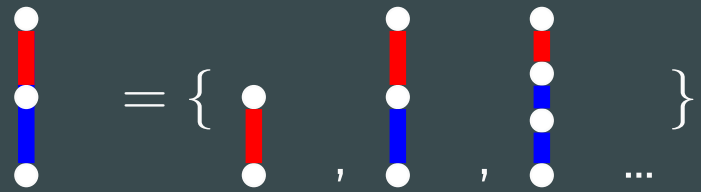
- If-then-else

Example



- While

Example



Renaming

Rename all locally-scoped variables in:

- Var statements

Example

$$\texttt{x := 0;}\ \texttt{var x in y := x} \dashrightarrow \texttt{x := 0;}\ \texttt{var \$0 in y := \$0}$$

- Forall expressions

Example

$$\texttt{x = 0} \Rightarrow\ \forall\ \texttt{x} :: \texttt{y = x} \dashrightarrow \texttt{x = 0} \Rightarrow\ \forall\ \texttt{\$0} :: \texttt{y = \$0}$$

WLP

For each path, calculate its weakest precondition.

Issue

Handle array assignments.

Solution

Use repby and, when indexed, convert to conditional expressions.

Conditional replacement

Array assignments introduce conditional expressions to the final logical formula.

Example

$$wlp\ (a[1] := 10)\ (a[5] = 1) \Longrightarrow (1 = 5 \rightarrow 10 \,|\, a[5]) = 1$$

Replace these with primitive logical formulas.

Example

$$(g \rightarrow lhs \,|\, rhs) = E \ \equiv \ (g \Rightarrow lhs = E) \wedge (\neg g \Rightarrow rhs = E)$$

Normalizing

Normalize the form of the result WLP formula to:

- A list of assumptions
- The final goal

Example

$$A_1 \Rightarrow A_2 \Rightarrow \ldots \Rightarrow A_n \Rightarrow G$$

$$\downarrow$$

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \Rightarrow G$$

SMT-solving

SAT-solve the conjuction of the assumptions:

- Unsatisfiable $\rightarrow$ Infeasible path
- Satisfiable with model $[x \rightsquigarrow x']$ $\rightarrow$
  - Pass, if $G[x'/x]$
  - Fail, if $\neg G[x'/x]$

Issue

Still have to handle arrays, as they appear in the formula.

Solution

Treat them as uninterpreted symbols.