Compositional Safety Verification with Max-SMT

What means compositional safety verification?

- Explanation of safety verification (assure every evaluation path in a program graph is safe for an assertion)
- Explanation of compositional (SCCs)

Motivation

- Why compositional? (scalability)
- Why is it not trivial to do it compositional? (by example)

Definitions

Presentation of recurring program example graph

At this point I will present an example of a program graph, which will be as easy as possible but at the same point covers all necessary cases for everything following. It will probably an example with three SCCs in a row, where the first has a transition to the second, the second has a transition to the third and the first has also a transition directly to the third. This way there is an SCC with multiple entry transitions, which is crucial to show the tree-like exploration of SCCs. There should be SCCs with just one node, but also more complex ones. The first to show the basic idea, the second to cover all parts of the algorithm. The example should be an actual working program doing something useful, to enable understanding.

Program graph

A concise formal definition (with example in the graph) of

- Locations
- Transitions
- Assertions
- SCCs
  - Entry transitions
  - Exit transitions

and the semantics

Invariants

Presentation of the different types of invariants linking to the example graph.

- Invariants
- Inductive Invariants
- Conditional Inductive Invariants

For inductive invariants I will show additionally the possible construction directions top-down and bottom-up.

Algorithm by example

The idea is to explain it the other way around than in the paper. This way I can cover the whole picture and then dive into detail. The explanation will be done by the presented example.

CheckSafe

Here we will view the SCCs as components and abstract from the details of CondSafe.

Basic Algorithm:

- Begin at the end to reach the initial states in a tree which is to discover
- We must descend the tree to all of the initial states to prove safety
- If we can not descend further in a path of the tree, find other conditions (narrow)

CondSafe

Here we will inspect the different SCCs and apply CondSafe to them

Definition of the algorithm

At this point I will link the presented example to the algorithm code and give the definition of the algorithms. Therefore I will also introduce Max-SMT-Solving of hard and soft constraints.