SVMRanker: A General Termination Analysis Framework of Loop Programs via SVM

Authors: **Xie Li**, Yi Li, Yong Li, Xuechao Sun
Andrea Turrini and Lijun Zhang

October 23, 2020

Verification

Checking termination is neccessary.

Example

```
assume(True);
int x = 1;
while(x>0) x++;
assertion(x<=0);
```

The assertion can be checked once the loop is terminating.
However theoretically...
In the practice the program return `UNKNOWN` once we cannot prove or disprove the termination.
Our tool: focus on the synthesis ranking function.
What is **ranking function**?

Ranking Function

```
assume(True);
int x = 5;
while(x > 0) x--;
assertion(x<=0);
```

ranking function is the function used for proof of termiation.

$$\Omega(\mathrm{x}, \mathrm{x}') := x > 0 \wedge x' = x - 1$$

Ranking function:

$$f(\mathrm{x}) = x$$

- Informally: $f : S \to \mathbb{R}$ where S is the set of states and $\mathbb{R}$ is a well-founded ordered set.
- strictly decreases at each iteration.
- has a lower bound in the loop state space.

More Powerful Ranking Functions

- Linear ranking function: $f(\mathrm{x}) = \mathrm{ax} + b$.
- Polynomial ranking function.
- k-phase- nested or multiphase ranking function:

$$\langle f_1, \cdots, f_k \rangle$$

Synthesis of Ranking Function

- Prove the termination is UNDECIDABLE.
- Synthesis of a certain class of ranking function is DECIDABLE.

State-of-the-Art Tools

Termination problem is essential and basic to program verification. State-of-the-art tools:

- LassoRanker in Ultimate Automizer.
- iRankFinder: A tool for ranking function inferring.
- Loopster: A static loop termination analyzer.
- CPAChecker
- ...

Competition SV-COMP also has a termination track for this problem.

SVMRanker: Synthesize Ranking Function via SVM

Technique previous tools use.

We reduce the synthesis of ranking functions into the SVM problem.

Advantage? Disadvantage?

SVMRanker: Architecture