# Progess Report 10

Xie Li

January 12, 2021

# Overview of the Progress

- Termination track of SV-COMP.
- Problem of adapting termination checking.
- Clang/LLVM?

# Termination Track

**Input:**

- Yaml file: can be parsed into paths of property file and src file.
- Property: `CHECK(init(main()), LTL(F end))`
- Src: C file.

**Output:**

- YES + WITNESS.
- FALSE + Counterexample.
- UNKNOWN.

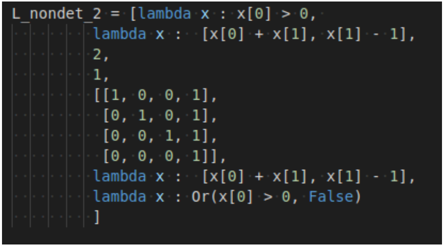
Counterexample: should be a infinite execution.

# SVMRanker and What is Needed?

- SVMRanker assume the input file is a single loop program. SV-COMP: may contain several loops or nested calls of functions.
- SVMRanker originally use boogie toolchain and can only support part of program.
  Clang&LLVM: a more powerful toolchain with

# Data Structures

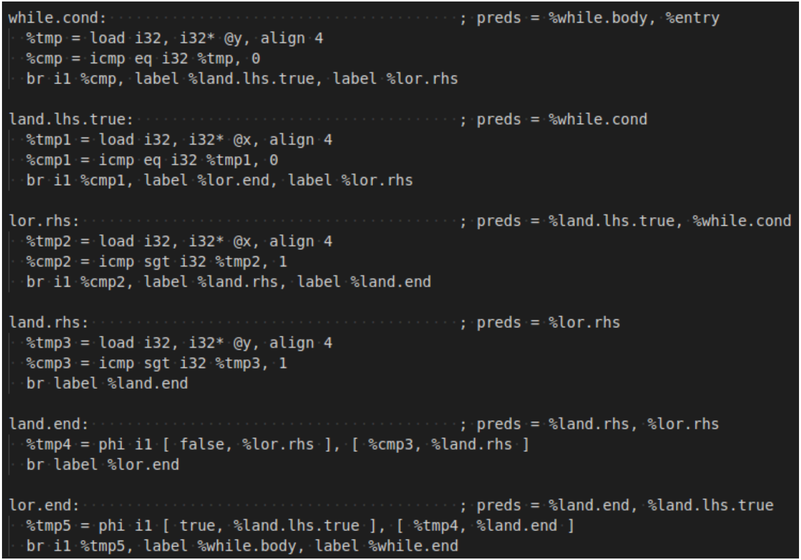
Python in SVMRanker:



```
L_nondet_2 = [lambda x : x[0] > 0,
         lambda x :  [x[0] + x[1], x[1] - 1],
         2,
         1,
         [[1, 0, 0, 1],
          [0, 1, 0, 1],
          [0, 0, 1, 1],
          [0, 0, 0, 1]],
         lambda x :  [x[0] + x[1], x[1] - 1],
         lambda x : Or(x[0] > 0, False)
         ]
```

# Data Structures

IR:



```
while.cond:                                       ; preds = %while.body, %entry
  %tmp = load i32, i32* @y, align 4
  %cmp = icmp eq i32 %tmp, 0
  br i1 %cmp, label %land.lhs.true, label %lor.rhs

land.lhs.true:                                    ; preds = %while.cond
  %tmp1 = load i32, i32* @x, align 4
  %cmp1 = icmp eq i32 %tmp1, 0
  br i1 %cmp1, label %lor.end, label %lor.rhs

lor.rhs:                                          ; preds = %land.lhs.true, %while.cond
  %tmp2 = load i32, i32* @x, align 4
  %cmp2 = icmp sgt i32 %tmp2, 1
  br i1 %cmp2, label %land.rhs, label %land.end

land.rhs:                                         ; preds = %lor.rhs
  %tmp3 = load i32, i32* @y, align 4
  %cmp3 = icmp sgt i32 %tmp3, 1
  br label %land.end

land.end:                                         ; preds = %land.rhs, %lor.rhs
  %tmp4 = phi i1 [ false, %lor.rhs ], [ %cmp3, %land.rhs ]
  br label %lor.end

lor.end:                                          ; preds = %land.end, %land.lhs.true
  %tmp5 = phi i1 [ true, %land.lhs.true ], [ %tmp4, %land.end ]
  br i1 %tmp5, label %while.body, label %while.end
```

# Boogie Toolchain

# Currently Implemented