Discussion 4: Experiment: Comparation to SVMRanker and LassoRanker

June 9, 2020

Experiment Result

Experiment Configurations: Test cases are grabbed from data set of conference paper. Number of cases: 134.
Sampling Strategy: Lowerbound of number of points $= 10$, expand sampling zone until reach the lowerbound.
Template Strategy:

- Mulitphase: Linear templates: single variable plus linear combination of all.
- SVMRanker: Linear + NonLinear
- LassoRanker: Only Support Linear

	2-Multiphase	5-Multiphase	SVMRanker	LassoRanker
FINITE	39	42	30	24
INFINITE	34	34	34	37
UNKNOWN	61	58	70	73
TIME	1107s	5709s	162s	1695s

Our algorithm can solve more cases than SVMRanker and LassoRanker.

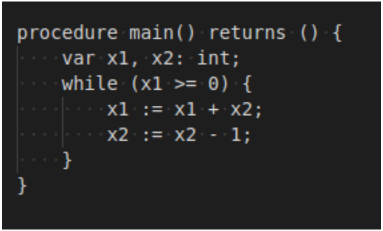
Specifying Cases

- Multiphase with depth 2 solved & SVMRanker not solved:
  $2, 51, 66, 67, 72, 73, 144, 152, 161$
- SVMRanker solved & Multiphase not solved:
  159
- Multiphase with depth 5 solved:
  $7, 155, 126$
- Advantage of Multiphase with depth 5 to LassoRanker:
  $1, 7, 58, 66, 67, 83, 144, 148, 149, 150, 151, 152, 153, 155, 156, 157$
  $158, 160, 161, 163, 164, 166, 167, 171$
- LassoRanker's advantage to Multiphase:
  Termination: $146, 147, 129, 74, 71, 15$
  Non-termination: $85, 25, 26$

Multiphase to SVMRanker

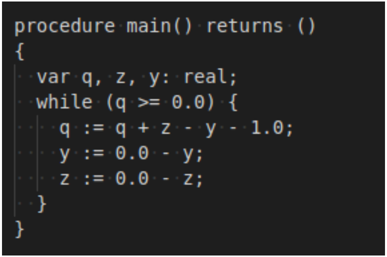
We solve two types of new loops:

- Decreased by a variable: $2, 51, 72, 73, 144, 152, 161$



```
procedure main() returns () {
    var x1, x2: int;
    while (x1 >= 0) {
        x1 := x1 + x2;
        x2 := x2 - 1;
    }
}
```

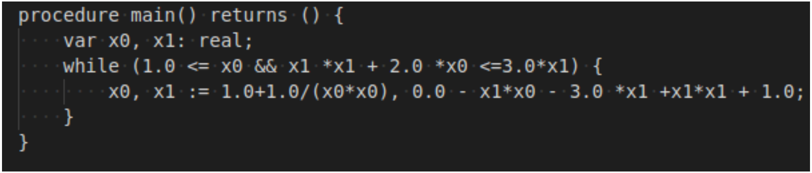
- Flipping between negative and positive: $66, 67$



```
procedure main() returns ()
{
  var q, z, y: real;
  while (q >= 0.0) {
    q := q + z - y - 1.0;
    y := 0.0 - y;
    z := 0.0 - z;
  }
}
```

SVMRanker to Multiphase

$159.bpl$

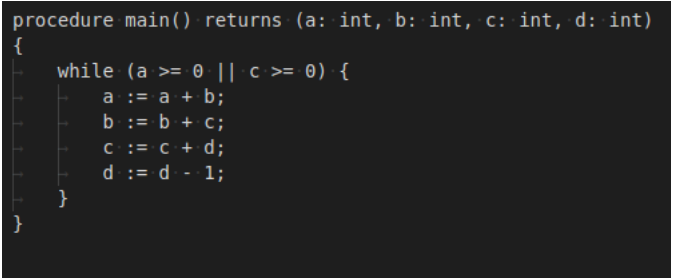


```
procedure main() returns () {
    var x0, x1: real;
    while (1.0 <= x0 && x1 *x1 + 2.0 *x0 <=3.0*x1) {
        x0, x1 := 1.0+1.0/(x0*x0), 0.0 - x1*x0 - 3.0 *x1 +x1*x1 + 1.0;
    }
}
```

Reason: we use the linear templates.

Multiphase with Depth 5

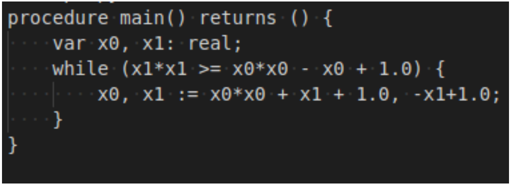
$7, 126$



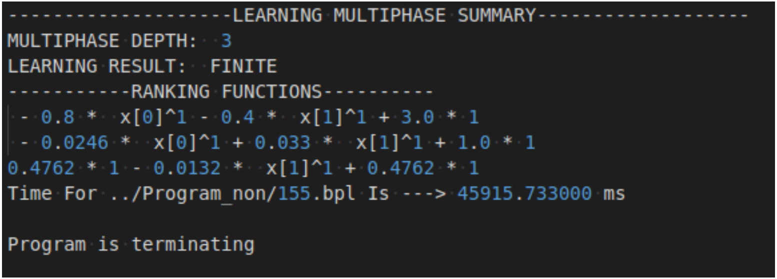
```
procedure main() returns (a: int, b: int, c: int, d: int)
{
    while (a >= 0 || c >= 0) {
        a := a + b;
        b := b + c;
        c := c + d;
        d := d - 1;
    }
}
```

Continue

155



```
procedure main() returns () {
    var x0, x1: real;
    while (x1*x1 >= x0*x0 - x0 + 1.0) {
        x0, x1 := x0*x0 + x1 + 1.0, -x1+1.0;
    }
}
```



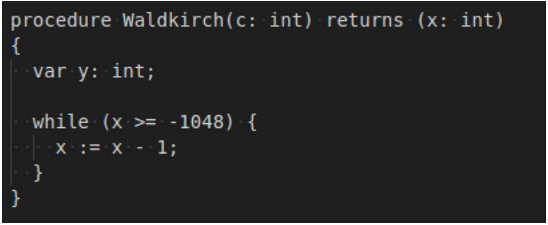
```
--------------------LEARNING MULTIPHASE SUMMARY-------------------
MULTIPHASE DEPTH:  3
LEARNING RESULT:  FINITE
----------RANKING FUNCTIONS---------
 - 0.8 *  x[0]^1 - 0.4 *  x[1]^1 + 3.0 * 1
 - 0.0246 *  x[0]^1 + 0.033 *  x[1]^1 + 1.0 * 1
0.4762 * 1 - 0.0132 *  x[1]^1 + 0.4762 * 1
Time For ../Program_non/155.bpl Is ---> 45915.733000 ms

Program is terminating
```

Not sure whether the result of this case is correct.

Termination: LassoRanker

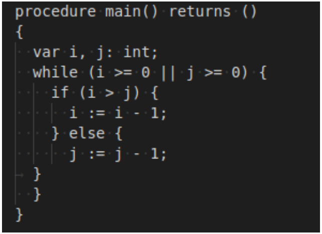
Termination: $146, 147, 129, 74, 71, 15$

- $146, 147$: large scale cases, where the number of variables is 51. Our algorithm goes timeout.
- 129: Might be the problem of incompatible types or redundant varibles. To be investigated later.



```
procedure Waldkirch(c: int) returns (x: int)
{
  var y: int;

  while (x >= -1048) {
    x := x - 1;
  }
}
```

- 74



```
procedure main() returns ()
{
  var i, j: int;
  while (i >= 0 || j >= 0) {
    if (i > j) {
      i := i - 1;
    } else {
      j := j - 1;
    }
  }
}
```

TODOs

- Make the problems of cases above clear.
- Do more experiment on different experiment configurations.
- Write the experiment part of the paper.