Automatic Derivation of Loop Bounds

In many industries including robotics, consumer electronics, avionics, automotive, and manufacturing, the system components must interact according to a stringent real-time schedule. It is therefore crucial for system engineers to have a good understanding of the worst case execution time (WCET). Recent years have seen a rapid development in automatic termination/liveness provers, most notably Terminator. The goal of this dissertation is to leverage these methods for WCET. Current techniques for termination are not constructive, i.e., they do in general not give an explicit time bound when the program is guaranteed to terminate. It is therefore crucial to extend the mathematical and logical techniques to obtain constructive bounds.

```
while (x > 0 ∧ y > 0)
  if nondet() { x--; }
  else {y--; x = read-pos-int(); }
```

Terminates, but no Bound exists

```
while (x > 0 ∧ y > 0)
  if nondet() { x--; }
  else {y--; x =+ 2; }
```

Bound: x + 3y

```
assume (n ≥ 1); x = n; y = n;
while (x > 0)
  /* Invariant: n ≥ y ≥ 1
                n ≥ x ≥ 1    */
  y--;
  if (y == 0)  {y = n;  x--;}
```

Bound: n^2

```
while (x < z)
  if (x  < y) {x =+ 1}
  else {y =+ 1}
```

Bound: max(0,z-x) + max(0,z-y)

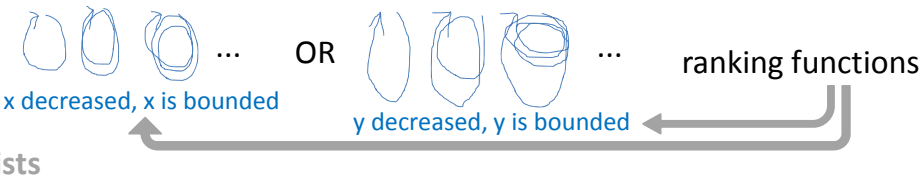
```
while (x ≥ 0)
  z =+ 1;
  x =- z;
```

Bound: max(0,-2z - 1) + ⌈sqrt(x)⌉

```
while (x ≤ n)
  x =* 2;
```

Bound: $\lceil \log_2(n) \rceil - \lceil \log_2(x) \rceil + 1$

Standard termination argument: Disjunctive well-foundedness



Used successfully in frameworks
- Model Checking with CEGAR (TERMINATOR)
- Abstract Interpretation (ranking functions as abstract domain)
- Disjunctive relational transitive hull overapproximation

Standard methods can prove termination, but cannot provide bounds
- termination argument is not constructive
- uses only linear ranking functions

Approaches
- single (more complicated) ranking function ($\rightarrow$ harder to find)
- ranking functions on finite domains ($\rightarrow$ strong program invariants needed)

Challenges
- polynomial bounds
- disjunctive bounds (max)
- non-polynomial bounds (sqrt(n), log(n), n*log(n), ...)

Automatic Techniques We Investigate
- underapproximation to guide formal techniques
- bound templates
- computeralgebra for symbolic reasoning
- static analysis for invariant generation
- abstract numerical domains for max and more generally disjunctiveness

Current Project: