Pinaka: Symbolic Execution Meets Incremental Solving

Eti Chaudhary Saurabh Joshi

Department of Computer Science and Engineering
IIT Hyderabad, India

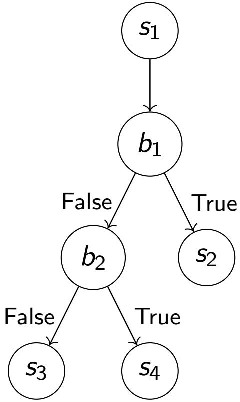
TOOLympics, TACAS, 2019

Underlying Approach: An Example

```
int x,y,output;

if(x>0){
   output = x;
}
else{
 if(y>0){
         output = y;
   }
 else{
         output = 0;
   }
}

assert(output>=0);
```

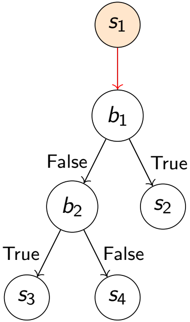


Underlying Approach: Partial Incremental Mode

```
int x,y,output;

if ( x > 0 ){
   output = x;
}
else {
  if ( y > 0 ){
         output = y;
   }
   else{
         output = 0;
   }
}

assert(output ≥ 0);
```

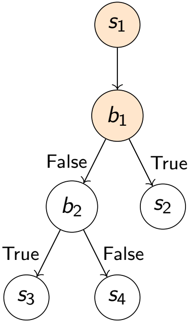


Underlying Approach: Partial Incremental Mode

```
int x,y,output;

if ( x > 0 ){
   output = x;
}
else {
  if ( y > 0 ){
        output = y;
  }
  else{
        output = 0;
  }
}

assert(output ≥ 0);
```

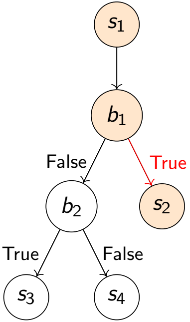


Underlying Approach: Partial Incremental Mode

```
int x,y,output;

if ( x > 0 ){
   output = x;
}
else {
 if ( y > 0 ){
       output = y;
  }
  else{
       output = 0;
  }
}

assert(output ≥ 0);
```



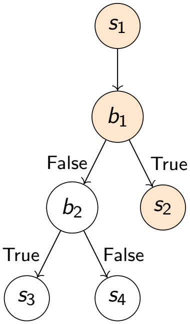
$$(x_1 > 0) \wedge (output_1 == x_1)$$

Underlying Approach: Partial Incremental Mode

```
int x,y,output;

if ( x > 0 ){
   output = x;
}
else {
  if ( y > 0 ){
          output = y;
  }
  else{
          output = 0;
  }
}

assert(output ≥ 0);
```



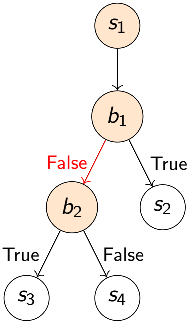
$$(x_1 > 0) \wedge (output_1 == x_1) \wedge \neg(output_1 \geq 0)$$

Underlying Approach: Partial Incremental Mode

```
int x,y,output;

if ( x > 0 ){
   output = x;
}
else {
  if ( y > 0 ){
        output = y;
  }
  else{
        output = 0;
  }
}

assert(output ≥ 0);
```



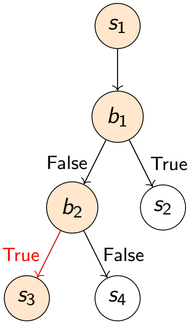
$(x_1 \leq 0)$

Underlying Approach: Partial Incremental Mode

```
int x,y,output;

if ( x > 0 ){
   output = x;
}
else {
  if ( y > 0 ){
        output = y;
   }
   else{
        output = 0;
   }
}

assert(output ≥ 0);
```



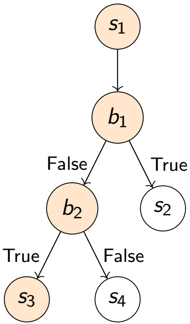
$$(x_1 \leq 0) \wedge (y_1 > 0) \wedge (output_2 == y_1)$$

Underlying Approach: Partial Incremental Mode

```
int x,y,output;

if ( x > 0 ){
   output = x;
}
else {
  if ( y > 0 ){
         output = y;
    }
    else{
          output = 0;
    }
}

assert(output ≥ 0);
```



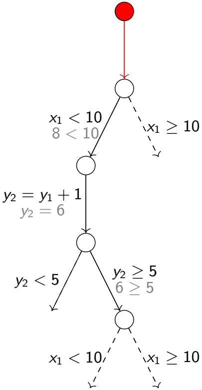
$$(x_1 \leq 0) \wedge (y_1 > 0) \wedge (output_2 == y_1) \wedge \neg(output_2 \geq 0)$$

Underlying Approach: Handling Loops

Non-Terminating Case

```
while ( x < 10 )
{
  y = y+1;
  if (  y < 5 )
  {
    x = x+1;
  }
}
```

Suppose by the loop is reached, we have $x_1 = 8$ and $y_1 = 5$.

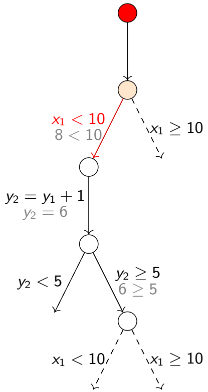


Underlying Approach: Handling Loops

Non-Terminating Case

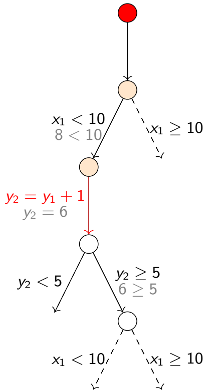
```
while ( x < 10 )
{
   y = y+1;
   if (  y < 5 )
   {
     x = x+1;
   }
}
```

Suppose by the loop is reached, we have $x_1 = 8$ and $y_1 = 5$.



```
while ( x < 10 )
{
  y = y+1;
  if (  y < 5 )
  {
    x = x+1;
  }
}
```

Suppose by the loop is reached, we have $x_1 = 8$ and $y_1 = 5$.

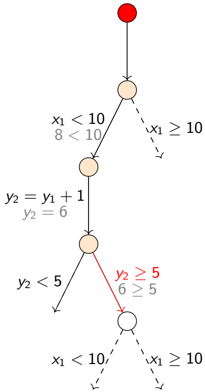


Underlying Approach: Handling Loops

Non-Terminating Case

```
while ( x < 10 )
{
   y = y+1;
   if (  y < 5 )
   {
     x = x+1;
   }
}
```

Suppose by the loop is reached, we have $x_1 = 8$ and $y_1 = 5$.

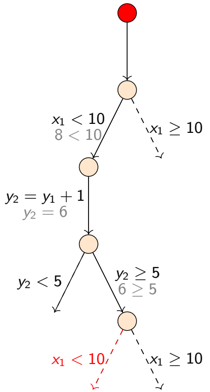


Underlying Approach: Handling Loops

Non-Terminating Case

Suppose by the loop is reached, we have $x_1 = 8$ and $y_1 = 5$.

```
while ( x < 10 )
{
   y = y+1;
   if (  y < 5 )
   {
     x = x+1;
   }
}
```



Verification Outcomes

	Safe Program	Unsafe Program
Terminating	Safe	Unsafe
Non-Terminating	Won't Terminate	Might report unsafe

Note:

- For a safe program, if Pinaka terminates, it also ensures that the program is terminating.
- If Pinaka reports a bug, it will indeed be a bug.
- For an unsafe non-terminating program, Pinaka might not terminate if it takes the non-terminating path *first* along the search.

Results

SVCOMP Configuration: DFS + PI

Category	Rank
ReachSafety-Arrays	4
ReachSafety-Floats	**2**
ReachSafety-Heaps	9
ReachSafety-ProductLines	9
ReachSafety-Recursive	8
ReachSafety-BitVectors	10
ReachSafety-ControlFlow	11
ReachSafety-ECA	10
ReachSafety-Loops	14

Thank You!

Available at:

https://github.com/sbjoshi/Pinaka

Developed at:

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Funded by:
SERB, DST, India