A Constraint-Based Approach to Multi-Threaded Program Location Reachability

Konstantinos Athanasiou

Northeastern University, Boston

FMCAD 2015 Student Forum

```
unsigned value, m = 0; // shared

unsigned count() {
  unsigned v = 0; //local
  acquire(m);
  if(value == 0u−1) {
    release(m);
    return 0;
  }
  else{
    v = value;
    value = v + 1;
    release(m);
    assert(value > v);
    return v + 1;
  }
}

int main {
  while (1) { thread(&count) }
}
```

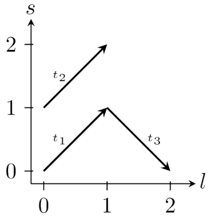
```
unsigned value, m = 0; // shared

unsigned count() {
  unsigned v = 0; //local
  acquire(m);
  if(value == 0u−1) {
    release(m);
    return 0;
  }
  else{
    v = value;
    value = v + 1;
    release(m);
    assert(value > v);
    return v + 1;
  }
}

int main {
  while (1) { thread(&count) }
}
```

Goal

Verify safety properties of multi-threaded programs, run by an unknown number of threads.

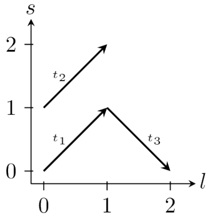
Thread-Transition Systems(TTS)

Thread-Transition Systems(TTS)



- Finite-state models extracted through predicate abstraction of recursion-free, finite-data procedures executed by threads.
- Each state (s, l) has a *shared s* and *local l* component.
- Configurations of the form $(s|l_0, \ldots, l_n)$

Thread-Transition Systems(TTS)

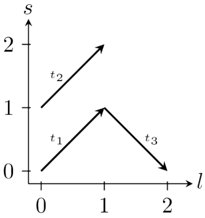


- Finite-state models extracted through predicate abstraction of recursion-free, finite-data procedures executed by threads.
- Each state (s, l) has a *shared s* and *local l* component.
- Configurations of the form $(s|l_0, \ldots, l_n)$

Problem Statement

Given a target thread state $t_F = (s_F, l_F)$, can the transition system reach a configuration of the form $(s_F|l_1, \ldots, l_F, \ldots)$?

Thread-Transition Systems(TTS)



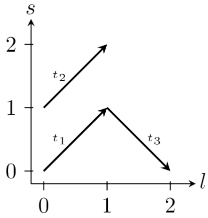
- Finite-state models extracted through predicate abstraction of recursion-free, finite-data procedures executed by threads.
- Each state (s, l) has a *shared s* and *local l* component.
- Configurations of the form $(s|l_0, \dots, l_n)$

Problem Statement

Given a target thread state $t_F = (s_F, l_F)$, can the transition system reach a configuration of the form $(s_F|l_1, \dots, l_F, \dots)$?

Decidable.

Thread-Transition Systems(TTS)



- Finite-state models extracted through predicate abstraction of recursion-free, finite-data procedures executed by threads.
- Each state (s, l) has a *shared* s and *local* l component.
- Configurations of the form $(s|l_0, \ldots, l_n)$

Problem Statement

Given a target thread state $t_F = (s_F, l_F)$, can the transition system reach a configuration of the form $(s_F|l_1, \ldots, l_F, \ldots)$?

Decidable. But $\mathrm{EXPSPACE}$ complete.

A Constraint-Based Approach

A Constraint-Based Approach

- Model necessary conditions for the target state to be reachable as integer linear constraints.

A Constraint-Based Approach

- Model necessary conditions for the target state to be reachable as integer linear constraints.

  If the constraints are unsatisfiable, then the target state t_F is unreachable and the safety property is verified.

A Constraint-Based Approach

- Model necessary conditions for the target state to be reachable as integer linear constraints.

  If the constraints are unsatisfiable, then the target state t_F is unreachable and the safety property is verified.

  Incomplete! The constraints might be satisfiable, even for safe target states.

A Constraint-Based Approach

- Model necessary conditions for the target state to be reachable as integer linear constraints.

  If the constraints are unsatisfiable, then the target state t_F is unreachable and the safety property is verified.

  Incomplete! The constraints might be satisfiable, even for safe target states.

- Use satisfying assignment to check if the instance is unsafe.

A Constraint-Based Approach

- Model necessary conditions for the target state to be reachable as integer linear constraints.

  If the constraints are unsatisfiable, then the target state t_F is unreachable and the safety property is verified.

  Incomplete! The constraints might be satisfiable, even for safe target states.

- Use satisfying assignment to check if the instance is unsafe.

  Still incomplete!

A Constraint-Based Approach

- Model necessary conditions for the target state to be reachable as integer linear constraints.

  If the constraints are unsatisfiable, then the target state t_F is unreachable and the safety property is verified.

  Incomplete! The constraints might be satisfiable, even for safe target states.

- Use satisfying assignment to check if the instance is unsafe.

  Still incomplete!

- Decides efficiently a large number of instances.

A Constraint-Based Approach

- Model necessary conditions for the target state to be reachable as integer linear constraints.

  If the constraints are unsatisfiable, then the target state t_F is unreachable and the safety property is verified.

  Incomplete! The constraints might be satisfiable, even for safe target states.
- Use satisfying assignment to check if the instance is unsafe.

  Still incomplete!
- Decides efficiently a large number of instances.

Thank you!