Games Based Safety Checking with MAGE

Adam Bakewell & Dan Ghica
University of Birmingham, UK
Games Based ...

★ Compositional Model:
Games Based ...

★ Compositional Model:
Games Based ...

★ Compositional Model:
Games Based ...

★ Black-Box Model:
Games Based ...

★ Black-Box Model:
Games Based

★ Black-Box Model:
Games Based ...

★ Black-Box Model:
Games Based

★ Black-Box Model:
Games Based ...

★ Fully abstract:

- Error in program

<=>

Reachable error action in model
Safety Checking

★ Program unsafe iff reachable error action in model

Program → Model → f.s.m. → Check → Counterexample

★ Implemented [2003-4] - works ... in principle!
Safety Checking

★ GAMECHECKER [2005-6] adds CEGAR:

- Approximate program
- Refine approximation
- Model
- Check

- Verdict
- Analyse
- Counterexample

f.s.m.
★ Lazy Modelling:
- compute model parts by need

★ Lazy Checking:
- stop checking at first error
Symbolic Modelling:

- actually build the model
- make an interpreter to implement the model

... with MACE
Cheap Counterexample Test:
- no theorem prover needed
- accept approximated counterexamples
Cheap Refinement:

- adjust types of some leaf terms
oflo : com,  
uflo : com,  
input : nat32,  
output : nat32,  
analyse : com -> com -> com

new buffer : nat32[64] in  
new top : nat32 := 0 in  
let push=if top=64 then oflo else buffer[top++]:=input  
let pop=if top=0 then uflo else output:=buffer[--top]  
analyse(push,pop)