Extracting a Petri Net Representation of Java Concurrency

Travis Pouarz
Anita J. Bateman
EE382C9 Embedded Software Systems
Spring 2002
Automated extraction of Petri Net representations from concurrent Java systems

- Use Petri Nets for analysis of concurrent Java systems
  - Petri Nets are well-suited for analyzing asynchronous and concurrent systems

- Current research does not provide an automated method to extract a Petri Net from a Java system
Adding BETS to the picture

**Bandera**: Generates Bandera Intermediate Representation (BIR) from Java source, isolating relevant concurrent interrelations (structures and actions)

**BETS**: BIR Extraction to Transition System

**Petrify**: Generates Petri Net from State Graph
How BETS works

- Result state graph must have all transitions to and from reachable states: enumeration of all reachable states is inevitable.
  - Bandera performs state space reductions
- Visit each reachable state only once
- Use Breadth-First Search to find all states reachable from the current state by evaluating permissible actions
- Sets of states encoded as Binary Decision Diagram (BDD)
Evaluation—“Yep, works great!”

- Method for developing code from human-readable BIR
- Compatible with Bandera BIR API
- Output is Petrify-ready

Petri Net generated from Deadlock example with 2 threads
Conclusion and Future Work

- **Deliverables**
  - Working system to convert a BIR representation into a state graph (Petrify input)
  - Uses hand-coding of text BIR for specific examples
  - Successful proof-of-concept
  - Deadlock detection is free (reachability analysis)

- **Future Work**
  - Full automation/integration with Bandera: “simple matter of programming”
  - Bright ideas to avoid state enumeration