State Extensions for Java PathFinder

Background
- Java PathFinder (JPF) is an explicit-state model checker for Java programs
- Used to find bugs in programs or verify properties
- Backtrackable Java Virtual Machine (JVM)
- Runs on the top of the host JVM
- Operations on the state representation:
  - Bytecode execution
  - State backtracking
  - State comparison
- First open-source tool released by NASA
  - http://javapathfinder.sourceforge.net/

Goal and summary
- Increase the applicability and performance of JPF
- Developed four state extensions:
  - Untracked state - affects backtracking
  - Undo backtracking - affects backtracking and execution
  - Delta execution - affects all three operations
  - Mixed execution - affects execution

Example
- Driver for exploration of states of a TreeMap object:

```java
// an empty tree, the root object for exploration
TreeMap t = new TreeMap();
for (int i = 0; i < N; i++) {
    int methodNum = Verify.getInt(0, 2);
    switch (methodNum) {
        case 0: t.put(Verify.getInt(1, N), 0); break;
        case 1: t.remove(Verify.getInt(1, N)); break;
    }
    Verify.ignoreIfPreviouslySeen(t);
    incrementCounters(methodNum == 1);
}
```

Undo backtracking
- Incrementally stores and restores states:
  - Keeps track of the state changes (not entire state)
  - Restores the state by undoing these changes

Mixed execution
- Execute some parts of the program being checked directly on the host JVM
  - Only deterministic blocks can be executed on the host
- Translating the state (eagerly or lazily)

Untracked state
- New functionality in JPF
  - Allows the user to mark that certain parts of the state should not be restored during backtracking
  - Useful for collecting some info about all execution paths

Delta execution
- Execution paths during state-space exploration overlap
- Key idea:
  - Share overlapping parts of multiple executions
  - Separately execute only parts that differ
- Use special state representation that can encode several states/trees into one

Conclusions
- Four state extensions for JPF
  - Untracked state extends JPF functionality
  - The other three extensions can speed up JPF
    - Undo state: speedup 1.42x-6.62x
    - Mixed execution: speedup up to 1.73x
    - Delta execution: speedup 0.88x-126.80x
- Untracked state and Undo backtracking have been recently included in the JPF codebase

http://mir.cs.uiuc.edu/jpf