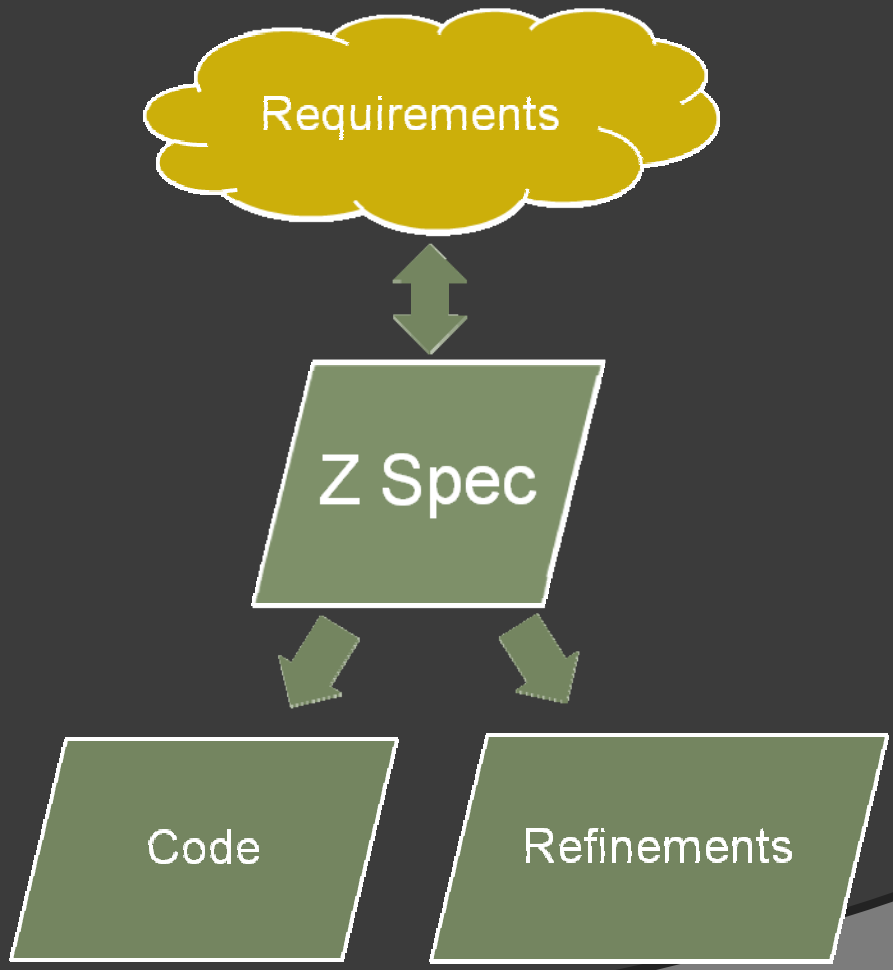
Mark Utting — The University of Waikato
Petra Malik — Victoria University of Wellington

UNIT TESTING FOR Z

Overview

- Our Methodology
- Theory: Testing Impls vs Specs
- Our “Z Unit Test” style
- Demonstration
- Testing Promotion
- Conclusions

Our Methodology



Validation via ???

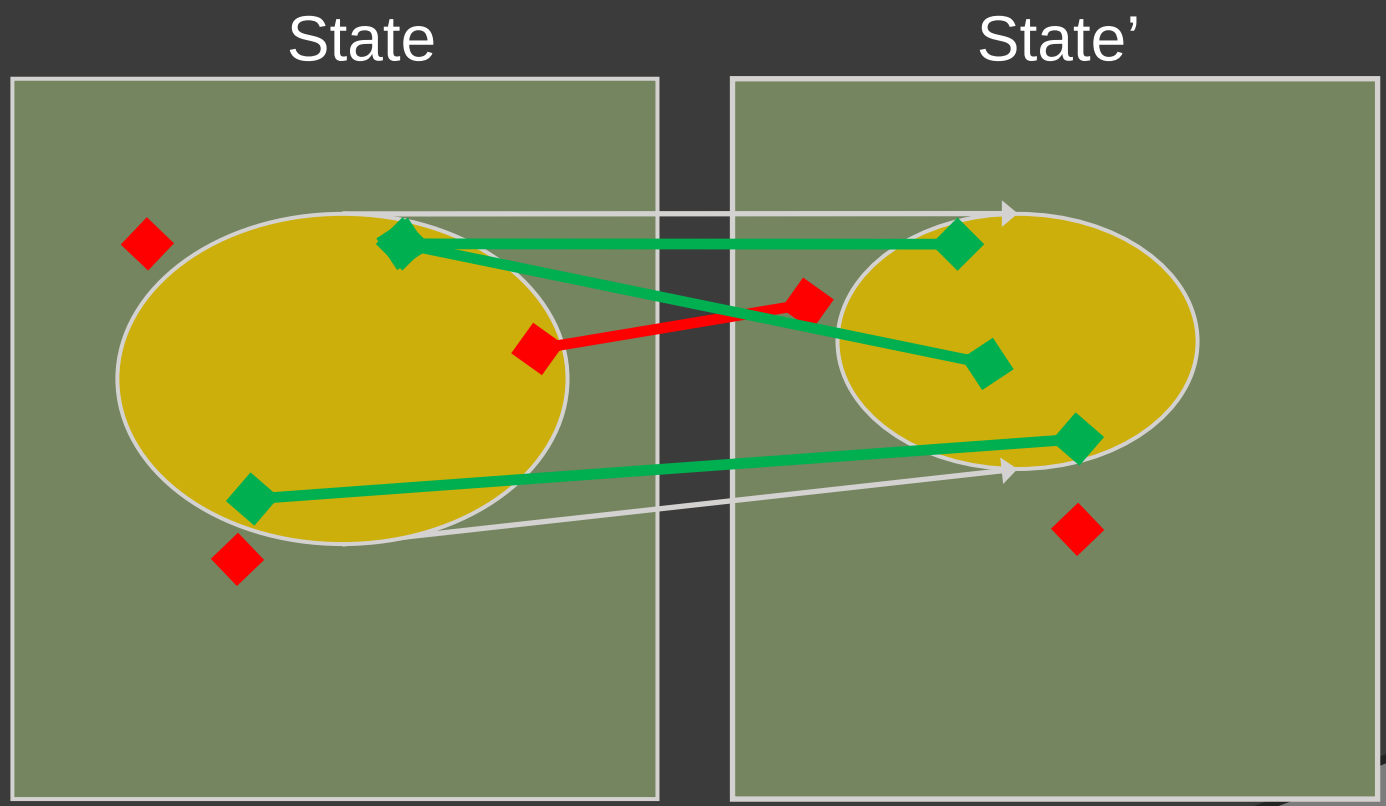
- Proof (∀Op•...)
  - Abstract properties (similar to the spec)
  - Harder to write
  - Proofs not automatic
  - Strong (universal) properties
- Test ⟨|x==3,x'==4|⟩∈Op
  - Very concrete (complementary to spec)
  - Easy to write (example-oriented)
  - Can be automated
  - Weak properties

Conclude: these are *complementary*.
We should use both validation strategies...
We focus on support for testing in this paper

cf. JML

```
/*@ public normal_behavior
 @     assignable \nothing;
 @     ensures \result == pennies % 100;
 @ for_example
 @     requires pennies == 703;
 @     assignable \nothing;
 @     ensures \result == 3;
 @   also
 @     requires pennies == -503;
 @     assignable \nothing;
 @     ensures \result == -3;
 @*/
public long cents();
```

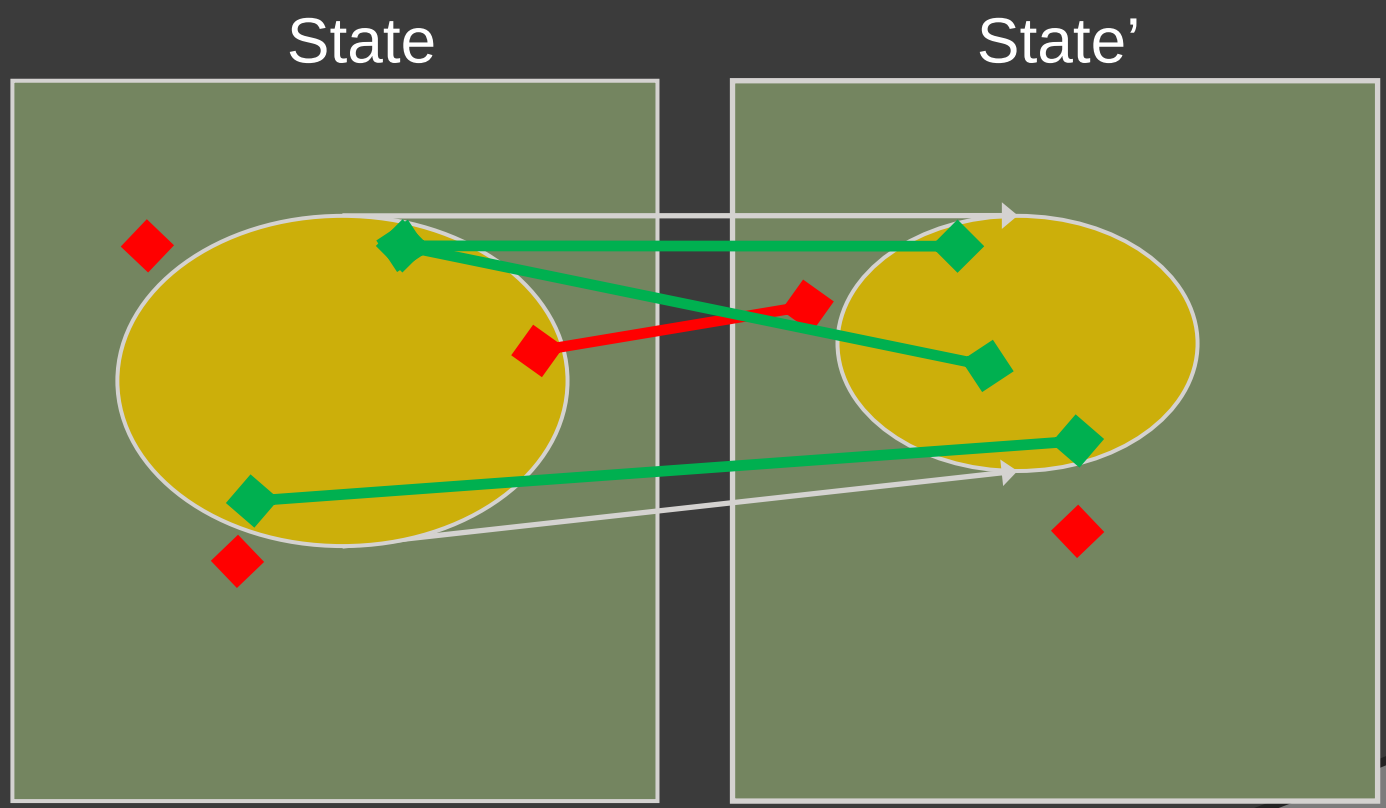
Testing A Spec. (Goal: validate spec)



The Spec

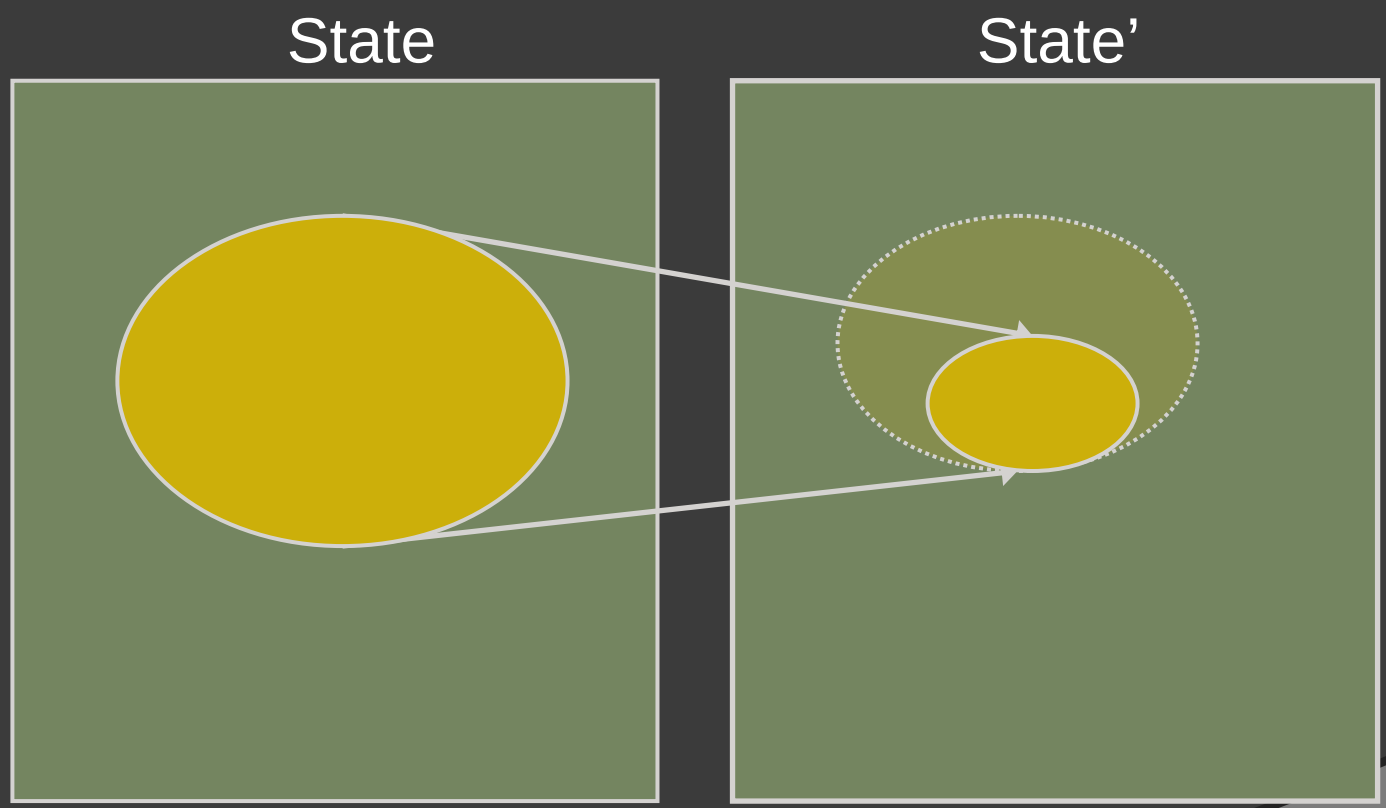
Requirements

Testing an Impl. (Goal: verify impl)



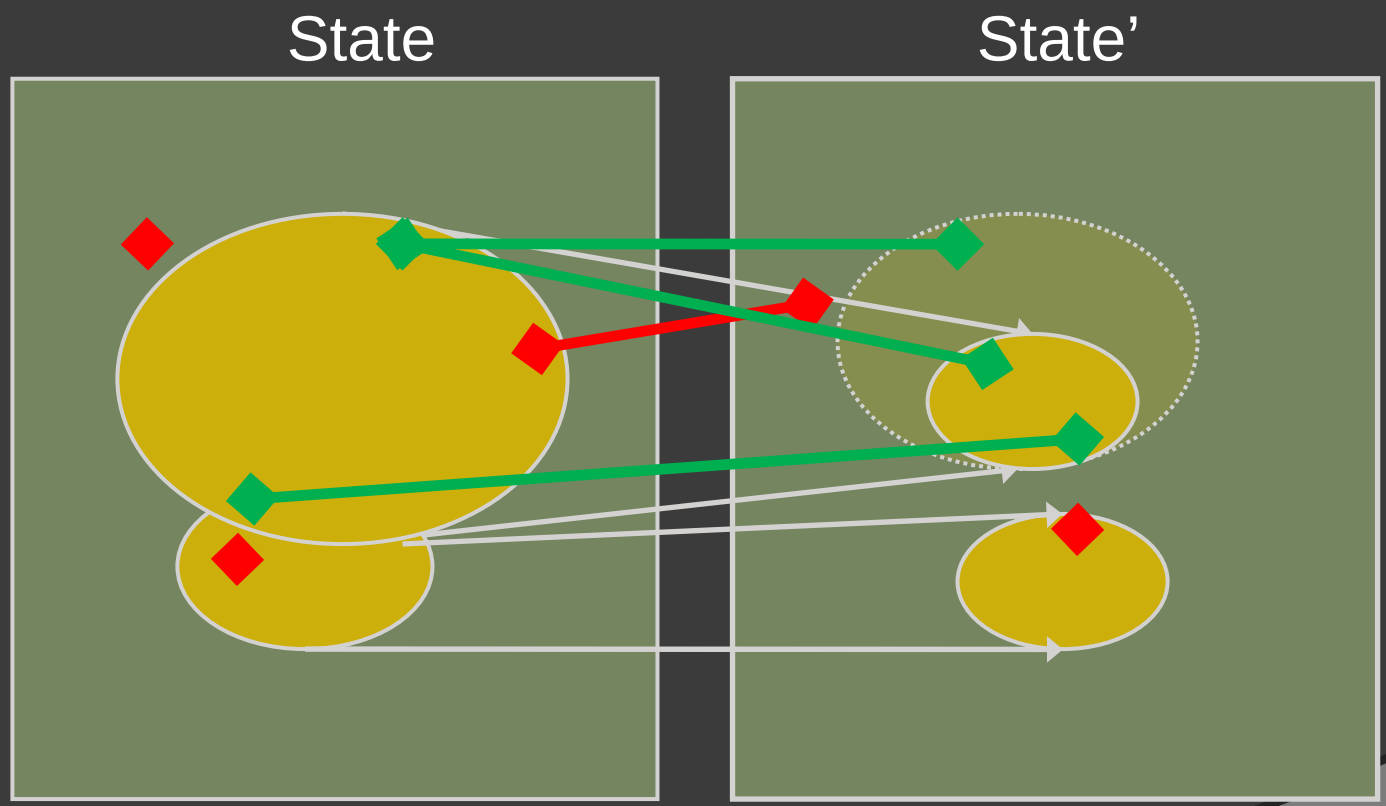
Start with the Spec

Testing an Impl. (Goal: verify impl)



Refinement: strengthen post

Testing an Impl. (Goal: verify impl)



Refinement: weaken pre

So...

- Validation-testing a spec is completely different to testing an implementation
- When validating a spec, we want to manually design our test examples from the informal requirements
- Many of the spec-validation tests would not be useful for testing an impl.
  - Tests outside of pre(Op) may fail on impl.
  - Tests for non-determinism may fail on impl.

Our Z Unit Test Style

- Positive Tests
  - $\langle\!| x==3, x'==4 |\!\rangle \in Op$
  - or $\{ eg1, eg2, eg3 \} \subseteq Op$
- Negative Tests
  - $\langle\!| x==3, x'==5 |\!\rangle \notin Op$
  - $\langle\!| x==3, x'==5 |\!\rangle \in \neg Op$ *(equivalent)*
  - $\langle\!| x==3 |\!\rangle \notin pre\ Op$
  - or $\{ neg1, neg2, neg3 \} \cap Op = \{\}$

Thanks to: Community Z Tools
Eclipse/CZT
ZLive

Demonstration

- Test-Driven Development of Sq==[x,x':N | x*x=x']
- Eclipse/CZT detects syntax/type errors continually
- Separate Z section (DemoTest) for each spec section (Demo)
- Eclipse can run 'zlive –load Demo.tex –test DemoTest', to check all tests

Promoting our Style

- Recall Promotion
  - PromOp == Op ∧ ΦProm

- ΦTestProm == [ΦProm | in?=...]
- PromOpTests == OpTests ∧ ΦTestProm
- We know: OpTests ⊆ Op
- We want: PromOpTests ⊆ PromOp
  - True by construction!
  - Can check: (ΦTestProm ⍀ PromOpTests)
    = PromOpTests

Conclusions

- A Practical TDD Style for Z
- Make Students Use It
  - Better quality Z specifications
  - Easier to mark the Specifications!
- Zlive/Jaza animation tools useful for evaluating the tests