# Alloy Scope Selection, Examples

Curt Clifton
Rose-Hulman Institute of Technology

# Scope Selection

# Scope Selection–1/3

- Start with the default scope
- Counterexample too complex to grok?
  - Reduce scope to get smaller example
- No counterexample found?
  - Increase scope to increase confidence

Begin Q1

# Scope Selection–2/3

- Make bound big enough for constants
  - **some disj** start, end: Node | ... ← Need at least 2 Nodes
- Use signature multiplicities when appropriate

```
// BAD!
abstract sig Color {}
sig Red, Green, Blue
                extends Color {}
fact {
    one Red
    one Green
    one Blue
}
pred show[] {}
run show for 2
```

← No error, but no instance found

```
// Good!
abstract sig Color {}
one sig Red, Green, Blue
                extends Color {}
pred show[] {}
run show for 2
```

Sig. multiplicity overrides scope

Cont. Q1

# Scope Selection–3/3

- Don't set scope larger than needed if all relevant values are named

```
sig Name, Address {}
sig Book { addr: Name -> Address }
pred add[b,b': Book, n: Name, a: Address] {
    b'.addr = b.addr + n->a
}
run add for 3 but 2 Book
```

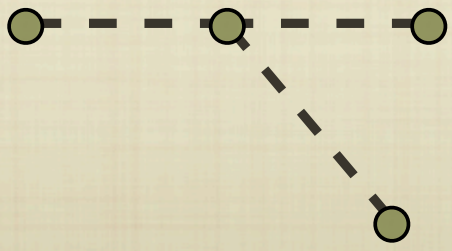
Why 2?

```
sig Node {adj: Node}
fact { all n, n': Node | n' in n.*adj }
pred show[] { some Node }
run show for 3
```

Why not 2?

- Let known shapes suggest constraints

```
sig Point {}
sig Segment { from, to: Point }
...
run show for 4 Point, 3 Segment
```



End Q1

# Careful!

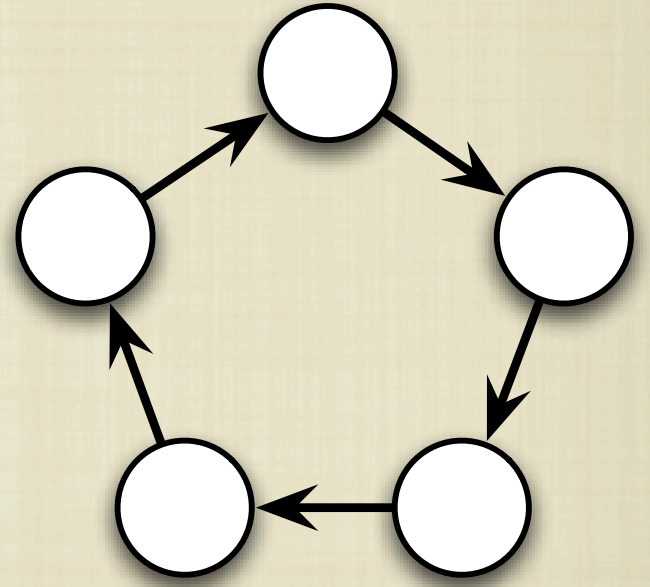
- **exactly** lets you specify an exact scope bound
  - Will omit **SMALLER** counterexamples!
- Use **exactly** for generating sample instances
- Don't use it for checking assertions

Q2

# Example: Leader Election

# Leader Election in Ring

- Problem: Concurrent processes running same code must collaborate to identify a "leader"
- Solution:
  - Assume unique PIDs and make highest the leader
  - Pass tokens containing PIDs (start with own)
  - Receive token:
    - < own PID?, discard
    - > own PID?, pass along
    - = own PID?, declare self elected



See File → Open Sample Models...,
book/chapter6/ringElection2.als