Reducing the Evolutionary Analysis Cost of Alloy

Hamid Bagheri

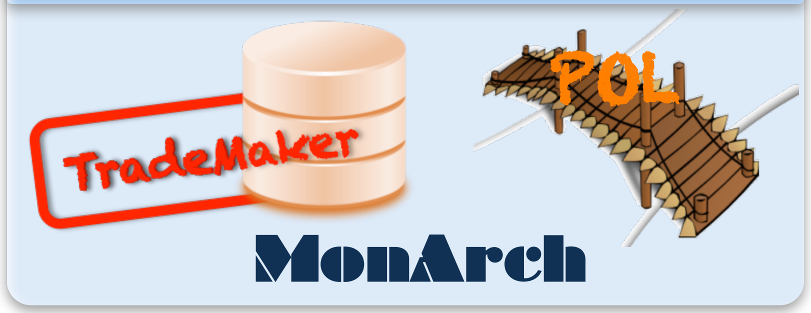
Workshop on the Future of Alloy

April 30 & May 1, 2018. Cambridge, MA

UNIVERSITY OF Nebraska® Lincoln

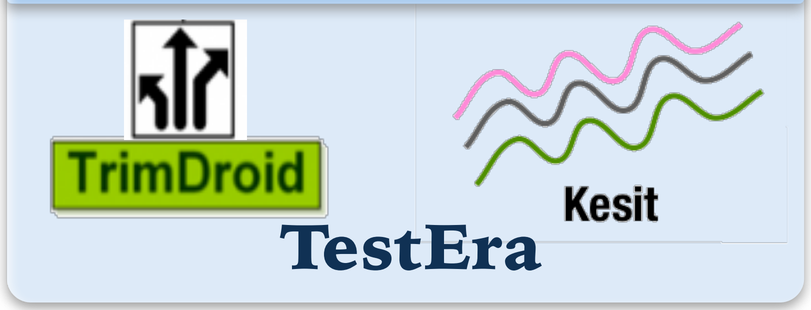
Alloy’s widespread applications

Design modeling and analysis



Program verification

Test-case generation



Security analysis

Challenges

- No support for analysis of evolving specifications even if they are substantially overlapping

- Recompute results in each analysis

- Especially problematic in online analyses where specifications are kept in sync with running systems

Objective

Improve bounded analysis of evolving specifications

Envision

- Bound adjustment
- Constraint reduction & solution reuse
- Parallelization

Bound adjustment

Each change by itself is not likely to invalidate all the prior analysis results

Insights

Each change by itself is not likely to invalidate all the prior analysis results

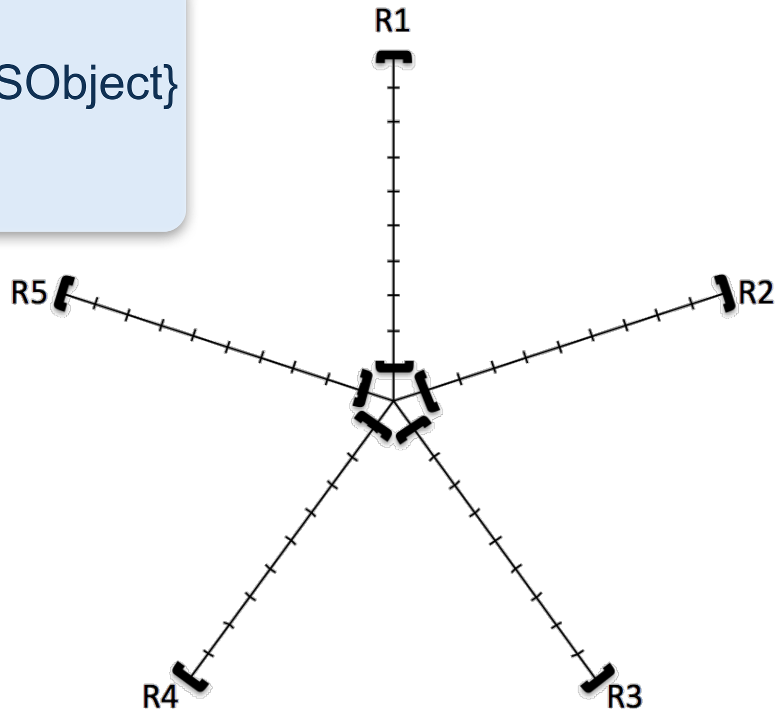
Results from previous analyses can be used to narrow the exploration space of the revised specification

A sample Alloy specification

```
sig FSObject {}
sig Dir extends FSObject {contents: set FSObject}
sig File extends FSObject {}
one sig Root extends Dir {}

fact hierarchy {
  no contents.Root
  all obj: FSObject | lone contents.obj
  FSObject in Root.*contents
  File + Dir = FSObject
}

run model {} for 4
```

A sample Alloy specification

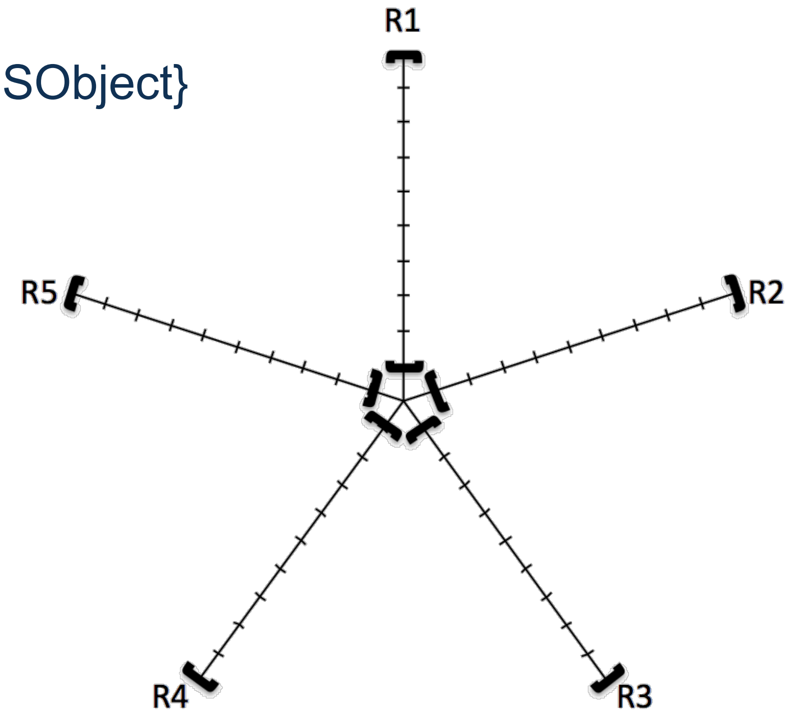
```
sig FSObject {}
sig Dir extends FSObject {contents: set FSObject}
sig File extends FSObject {}
one sig Root extends Dir {}

fact hierarchy {
  no contents.Root
  all obj: FSObject | lone contents.obj
  FSObject in Root.*contents
  File + Dir = FSObject
}

run model {} for 4
```



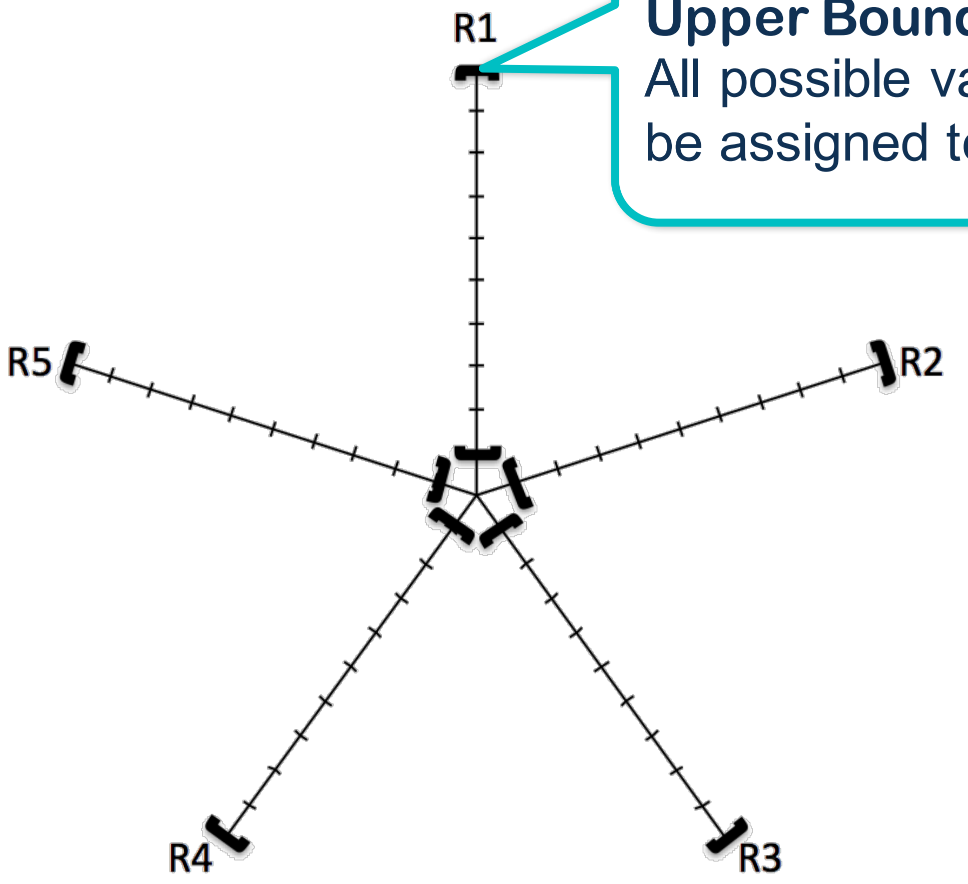
A sample Alloy specification

```
sig FSObject {}
sig Dir extends FSObject {contents: set FSObject}
sig File extends FSObject {}
one sig Root extends Dir {}

fact hierarchy {
  no contents.Root
  all obj: FSObject | lone contents.obj
  FSObject in Root.*contents
  File + Dir = FSObject
}
```

```
run model {} for 4
```

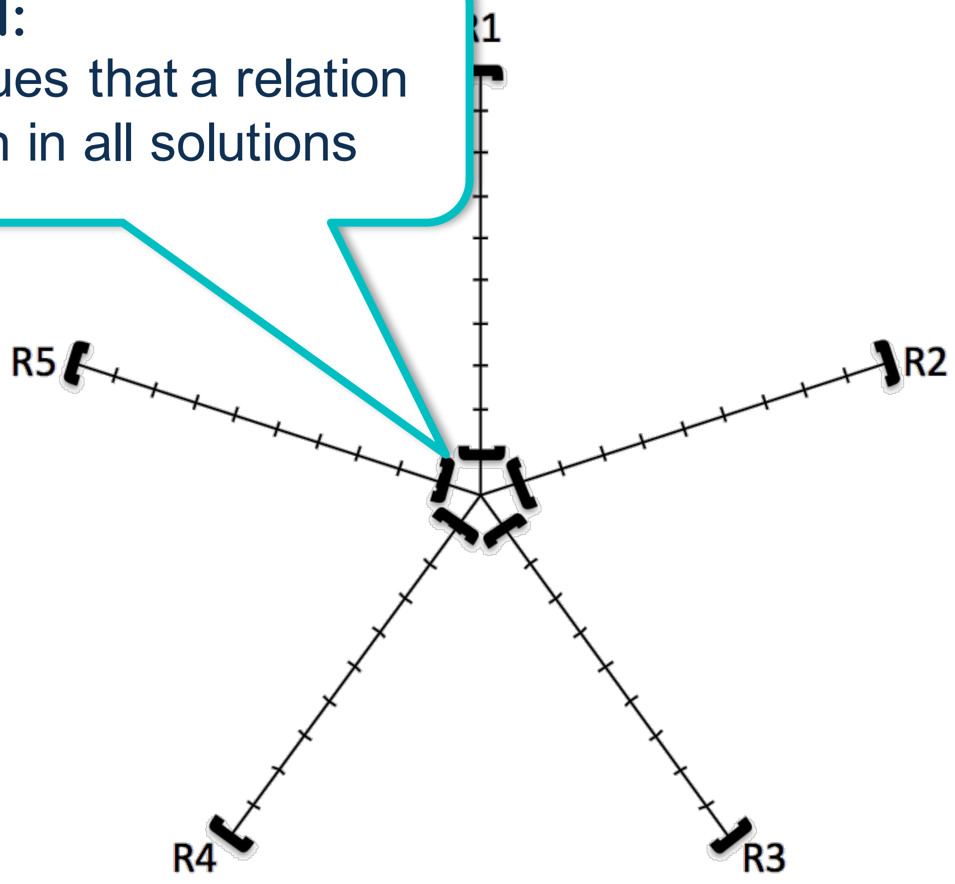


Relational variables and bounds



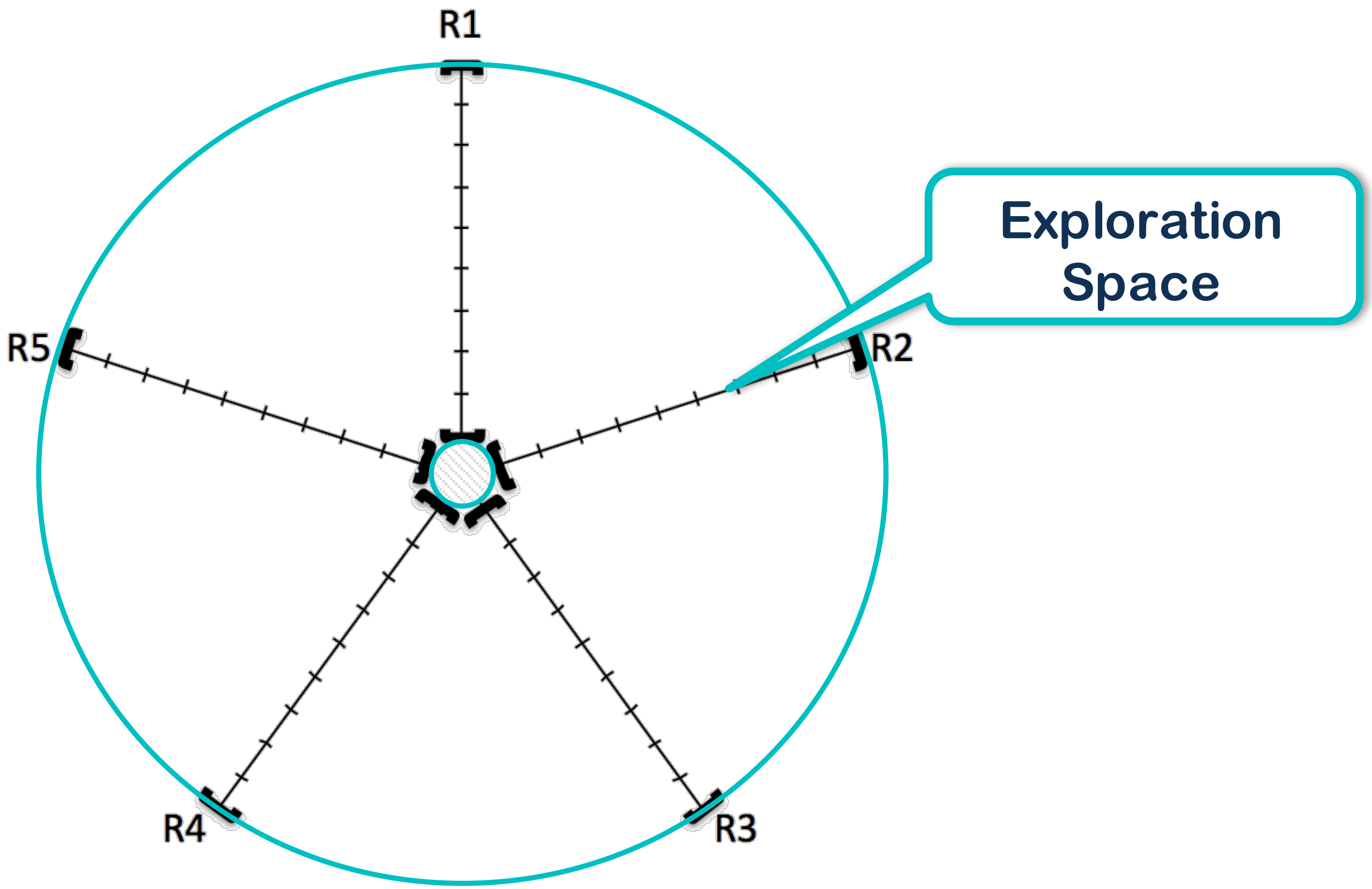
Upper Bound:
All possible values that can be assigned to the relation

Relational variables and bounds

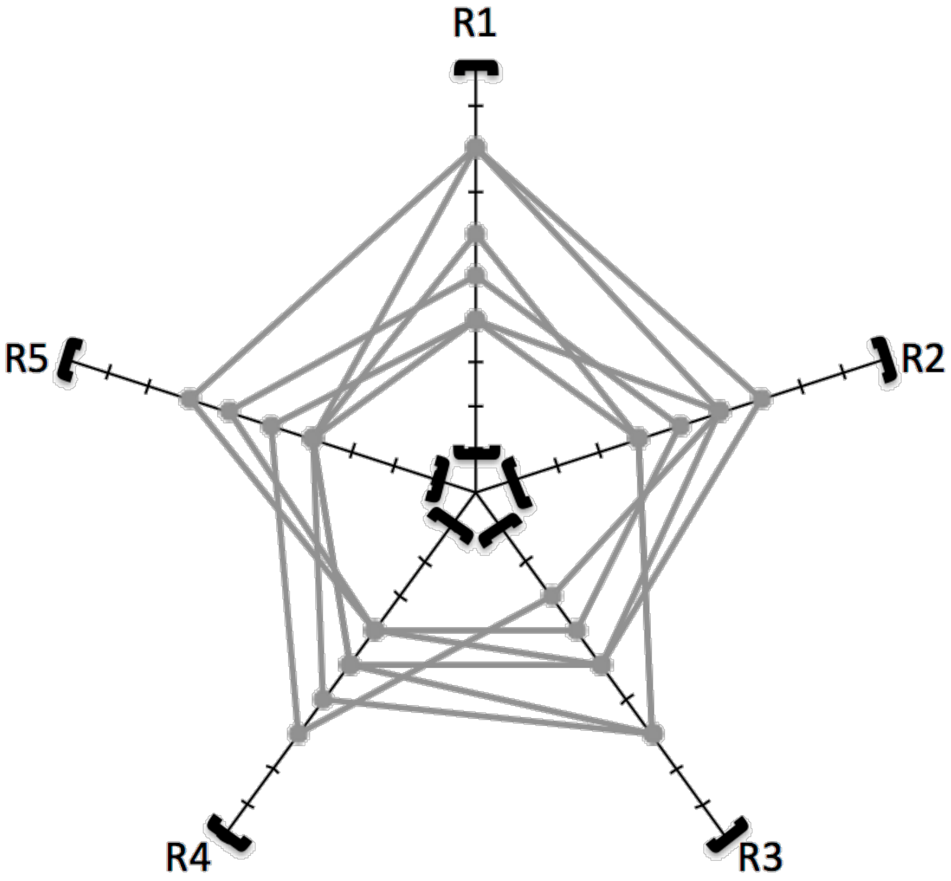


Lower Bound:
The set of values that a relation should contain in all solutions

Relational variables and bounds



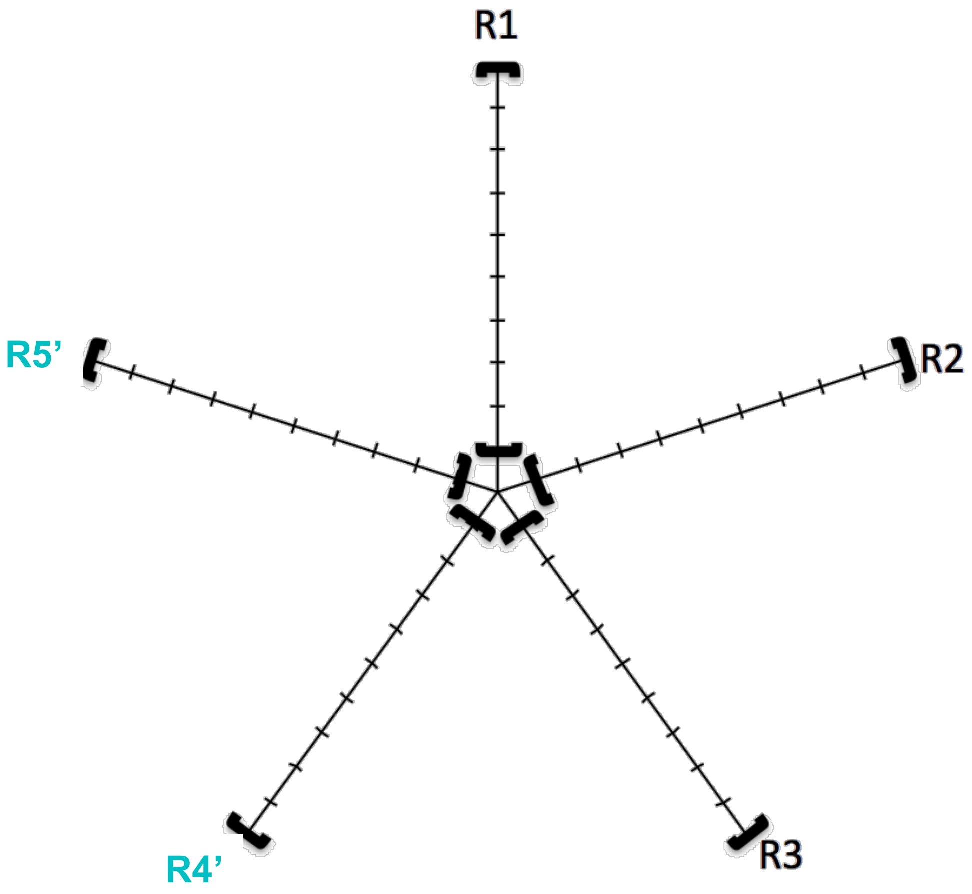
Solutions within relational bounds



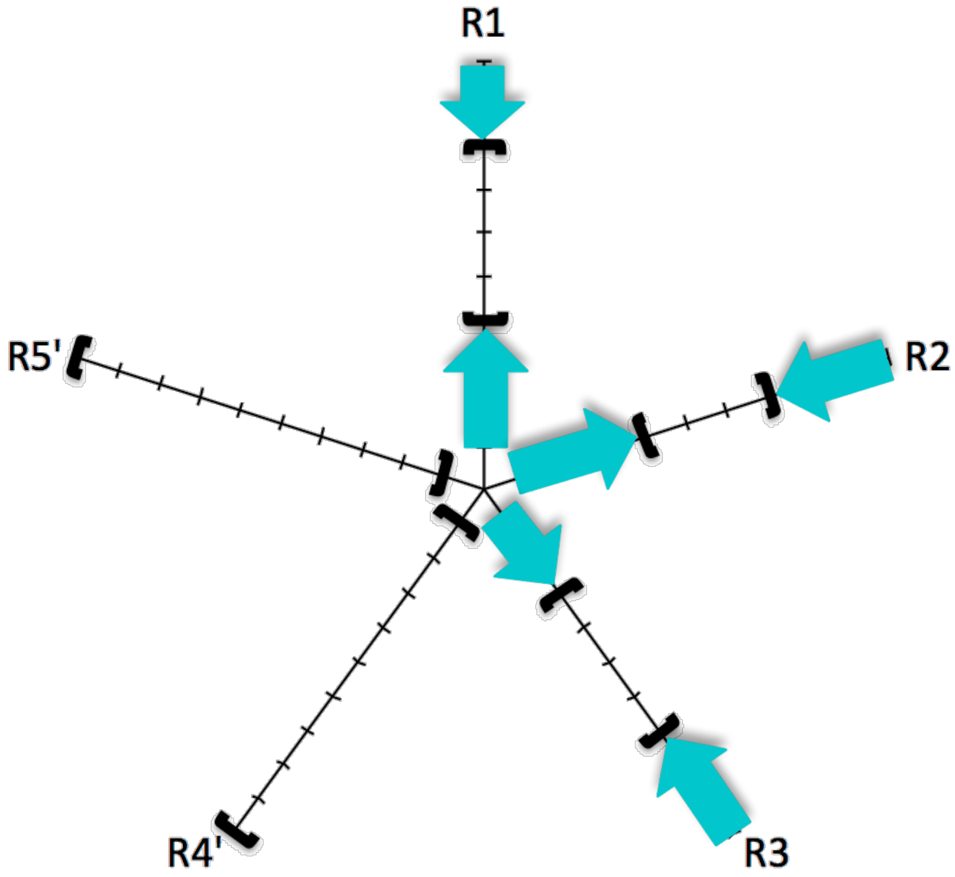
[ ] Upper/lower bound

⬠ Model instance

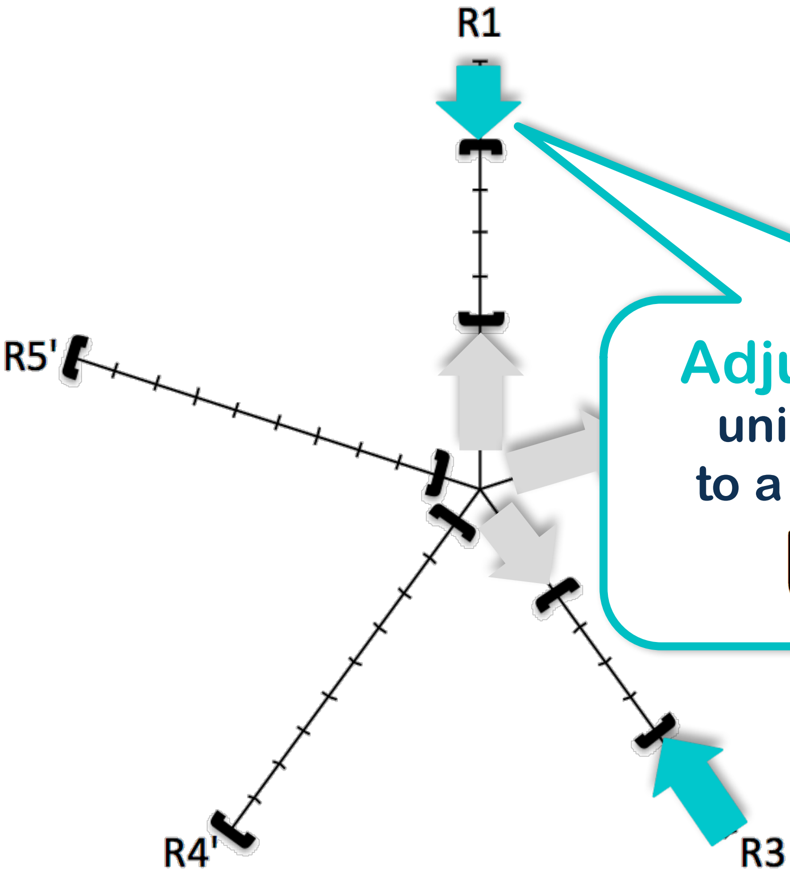
Change in the relational specification



Tighten bounds on relational variables



Tighten bounds on relational variables

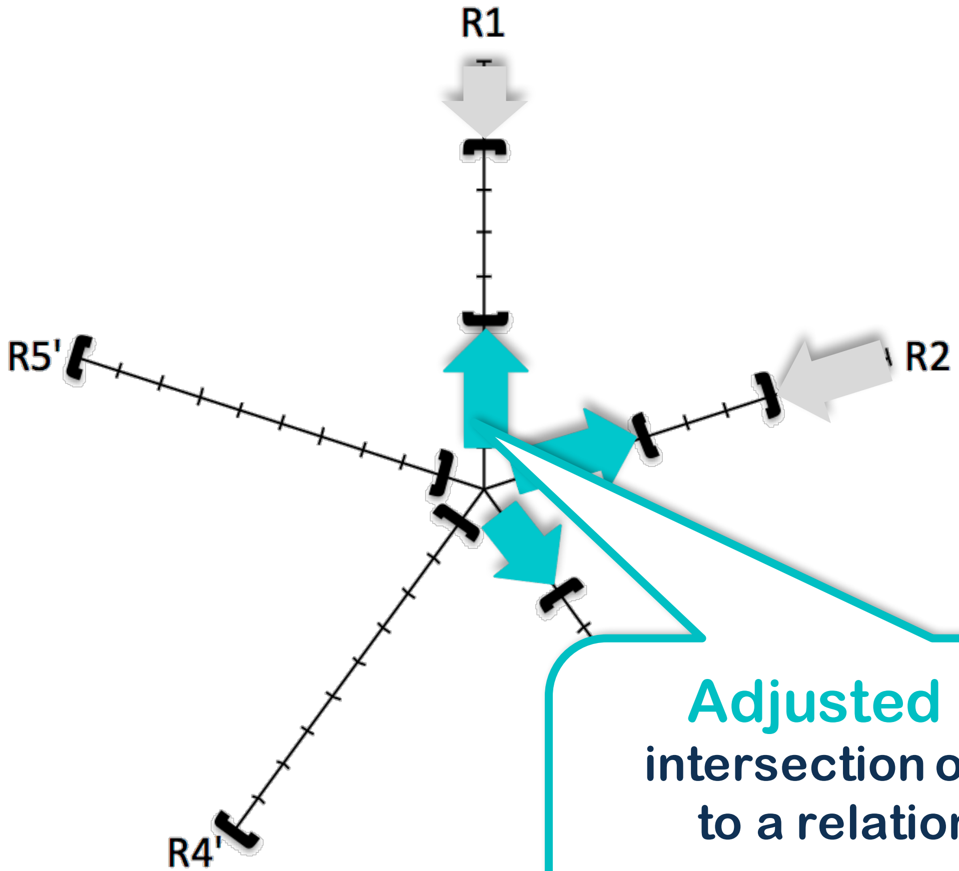


Adjusted upper bound:

union of values assigned to a relation in all solutions

$$\bigcup_{i\in I} i.val(r)$$

Tighten bounds on relational variables

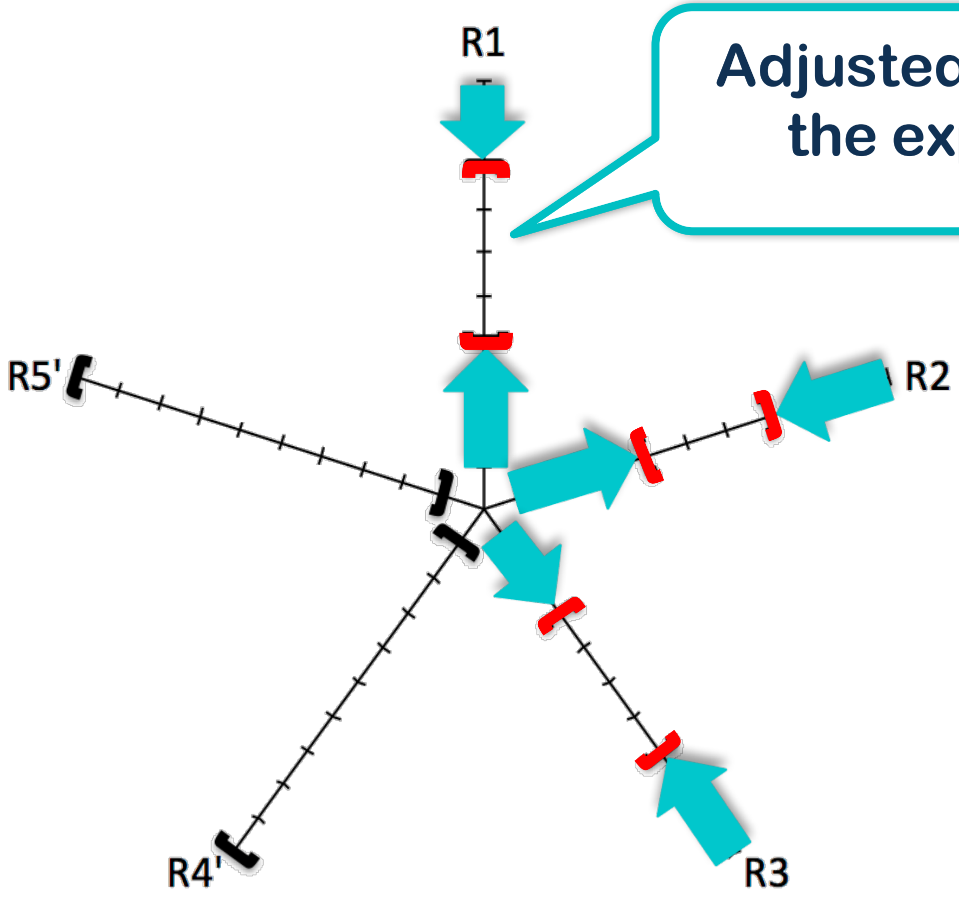


Adjusted lower bound:

intersection of values assigned to a relation in all solutions

$$\bigcap_{i \in I} i.val(r)$$

Tighten bounds on relational variables



Adjusted bounds reduce the exploration space

Constraint reduction & solution reuse

- Constraints recur during evolutionary analyses
- Incrementally store the constraints already solved, and retrieved them within the evolutionary analysis
- Prior work: memoization-based approaches in symbolic execution

Thank you