XACML Access Control Policies

Automated Verification, Recommendation and Model Repair

Alexandr Murashkin
Matthew Ma
University of Waterloo

Problem and Motivation

Managing *access control policies* (**ACP**) in modern computer systems can be challenging and error-prone *(Hughes & Bultan, 2008)*

Inconsistency roots in policies are hard to determine

Even if the error is localized, there are various ways of repairing policies

Explore techniques for semi-automated policy repair *(Kolovski & Hendler, 2008)*

Previous and Related Work

1. **Automated Verification of Access Control Policies Using a SAT Solver** (*Hughes & Bultan, 2008)* - proposed first-order logic and Alloy for XACML access control policies verification

2. **Access Control Policies: Modeling and Validation** (*Logrippo & Mankai*) - defined subset of XACML in Alloy

3. Synthesising verified access control systems in XACML *(Zhang et al.)*

4. XACML Policy Analysis Using Description Logics *(Kolovski and Hendler)*

XACML Access Control Policies (ACP)

PolicySet

set Policy

Policy

set Rule

ruleCombiningAlgo = {DenyOverrides, PermitOverrides}

Rule

set Subject

set Action

set Resource

Effect = {Permit, Deny}

Request

one Subject

one Action

one Resource

PolicySet {P1, P3}

Verify the property:
For any request, no two policies return different decisions (one Permit, one Deny)

Policy P1	Policy P3
Rules: 1) Student ⇒ Read ⇒ Marks ⇒ *Permit* 2) Professor ⇒ Read | Modify ⇒ Marks ⇒ *Permit*	**Rules:** 1) Professor ⇒ Read | Modify ⇒ Marks ⇒ *Deny* 2) Professor ⇒ Modify ⇒ Marks ⇒ *Permit* 3) Professor ⇒ Read ⇒ Marks ⇒ *Permit*
RuleCombiningAlgorithm: DenyOverrides	**RuleCombiningAlgorithm:** DenyOverrides

Request

Professor ⇒ Read ⇒ Marks

Policy P1	Policy P3
Rules: 1) Student ⇒ Read ⇒ Marks ⇒ *Permit* 2) Professor ⇒ Read | Modify ⇒ Marks ⇒ *Permit*	**Rules:** 1) Professor ⇒ Read | Modify ⇒ Marks ⇒ *Deny* 2) Professor ⇒ Modify ⇒ Marks ⇒ *Permit* 3) Professor ⇒ Read ⇒ Marks ⇒ *Permit*
RuleCombiningAlgorithm: DenyOverrides	**RuleCombiningAlgorithm:** DenyOverrides

Request

Professor ⇒ Read ⇒ Marks

Policy P1	Policy P3
Rules: 1) Student ⇒ Read ⇒ Marks ⇒ *Permit* 2) Professor ⇒ Read | Modify ⇒ Marks ⇒ *Permit*	**Rules:** 1) Professor ⇒ Read | Modify ⇒ Marks ⇒ *Deny* 2) Professor ⇒ Modify ⇒ Marks ⇒ *Permit* 3) Professor ⇒ Read ⇒ Marks ⇒ *Permit*
RuleCombiningAlgorithm: DenyOverrides	**RuleCombiningAlgorithm:** DenyOverrides

Request

Professor ⇒ Read ⇒ Marks

Policy P1	Policy P3
Rules: 2) Professor ⇒ Read | Modify ⇒ Marks ⇒ ***Permit***	**Rules:** 1) Professor ⇒ Read | Modify ⇒ Marks ⇒ ***Deny*** 3) Professor ⇒ Read ⇒ Marks ⇒ ***Permit***
DenyOverrides	DenyOverrides

Request

Professor ⇒ Read ⇒ Marks

Policy P1	Policy P3
Rules: 2) Professor ⇒ Read | Modify ⇒ Marks ⇒ ***Permit***	**Rules:** 1) Professor ⇒ Read | Modify ⇒ Marks ⇒ ***Deny*** 3) Professor ⇒ Read ⇒ Marks ⇒ ***Permit***
DenyOverrides	**Deny**Overrides
PERMIT	**DENY**

How to Repair: Original Model

Professor ⇒ Read ⇒ Marks

Policy P1	Policy P3
Rules: 2) Professor ⇒ Read | Modify ⇒ Marks ⇒ ***Permit***	**Rules:** 1) Professor ⇒ Read | Modify ⇒ Marks ⇒ ***Deny*** 3) Professor ⇒ Read ⇒ Marks ⇒ ***Permit***
DenyOverrides	DenyOverrides
PERMIT	**DENY**

How to Repair: Way 1 of 4

Professor ⇒ Read ⇒ Marks

Policy P1	Policy P3
Rules: 2) Professor ⇒ Read | Modify ⇒ Marks ⇒ ***Permit***	**Rules:** 1) Professor ⇒ Read | Modify ⇒ Marks ⇒ ***Deny*** 3) Professor ⇒ Read ⇒ Marks ⇒ ***Permit***
DenyOverrides	PermitOverrides
PERMIT	**PERMIT**

How to Repair: Original Model

Professor ⇒ Read ⇒ Marks

Policy P1	Policy P3
Rules: 2) Professor ⇒ Read | Modify ⇒ Marks ⇒ ***Permit***	**Rules:** 1) Professor ⇒ Read | Modify ⇒ Marks ⇒ ***Deny*** 3) Professor ⇒ Read ⇒ Marks ⇒ ***Permit***
DenyOverrides	DenyOverrides
PERMIT	**DENY**

How to Repair: Way 2 of 4

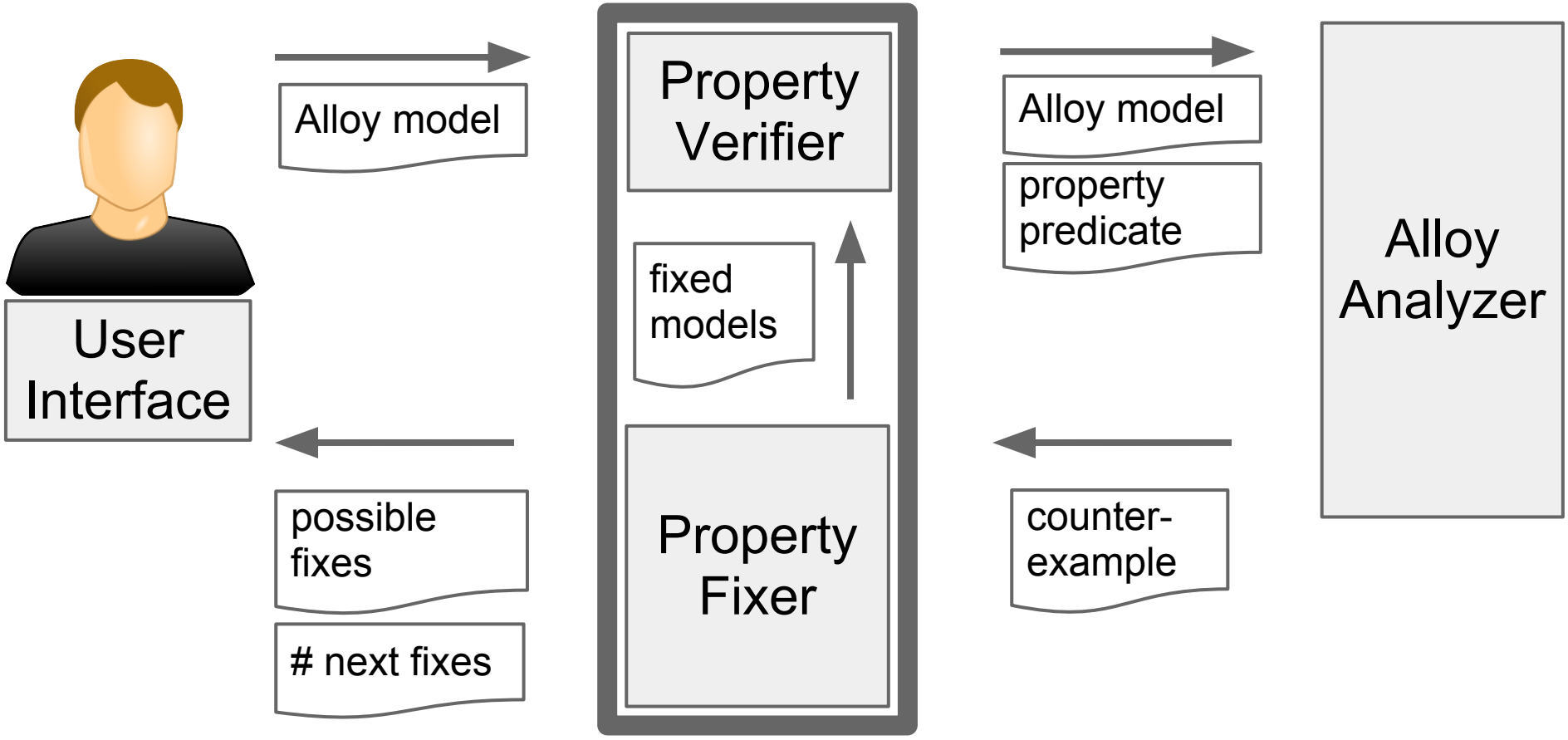
Professor ⇒ Read ⇒ Marks

Policy P1	Policy P3
Rules: 2) Professor ⇒ Read | Modify ⇒ Marks ⇒ ***Permit***	**Rules:** 1) Professor ⇒ Read | Modify ⇒ Marks ⇒ ***Permit*** 3) Professor ⇒ Read ⇒ Marks ⇒ ***Permit***
DenyOverrides	DenyOverrides
PERMIT	**PERMIT**

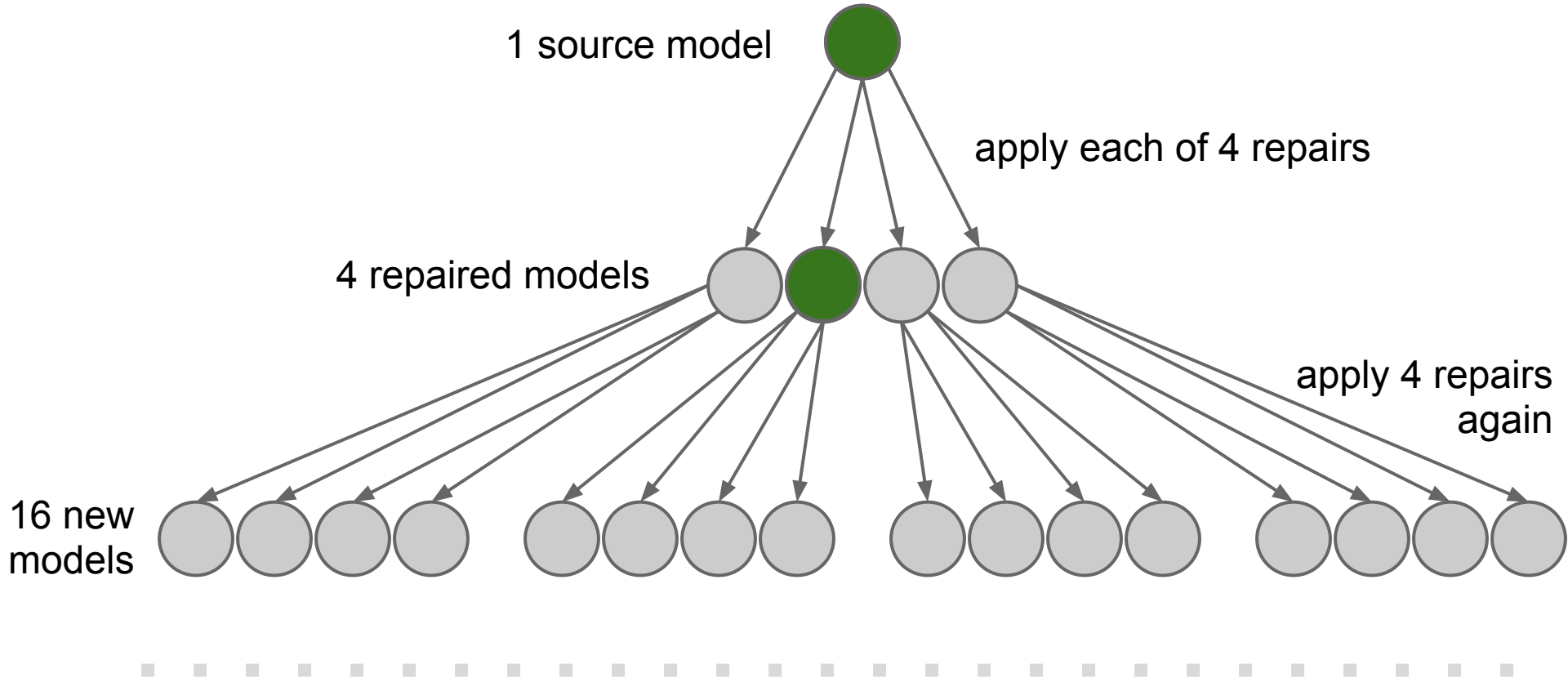
Demo

Everyone's on board!

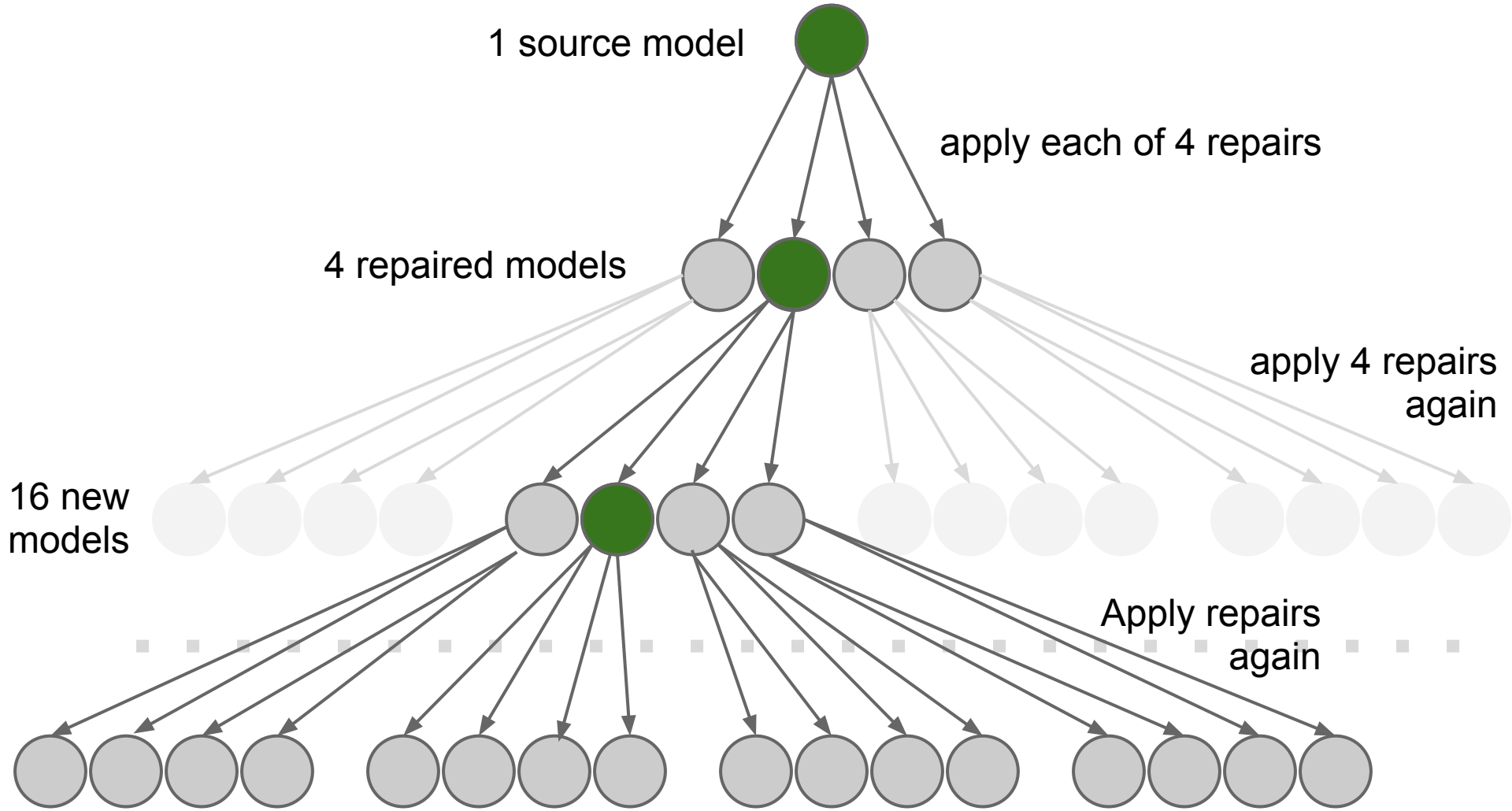
Dataflow Diagram



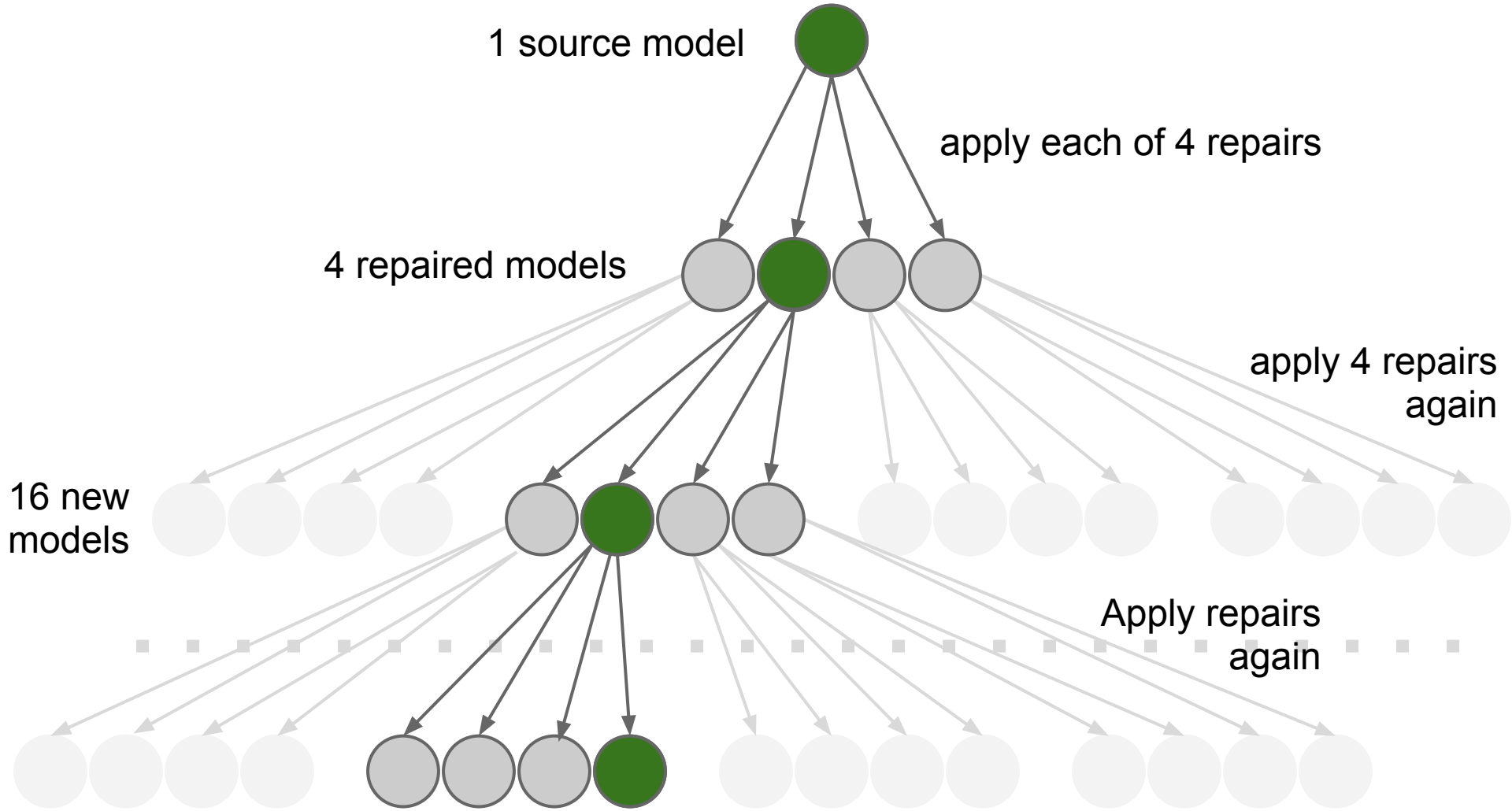
State Explosion Problem: Incremental Approach



State Explosion Problem: Incremental Approach



State Explosion Problem: Incremental Approach



Results and Conclusions

1. User-interactive repair of ACP is feasible

2. Fixes has been done automatically to Alloy file, increasing the efficiency

3. State explosion problem in semi-automated repair is handled successfully in most cases

4. The tool is extensible and will work for several properties and outside the domain of ACP

Future Work

1. Verification is bounded

2. Can verify one property in a time until first counterexample is found

3. Subset of XACML is covered

4. Repair procedure depends on the property and requires pre-defined prospective repair ways

Questions and Advices