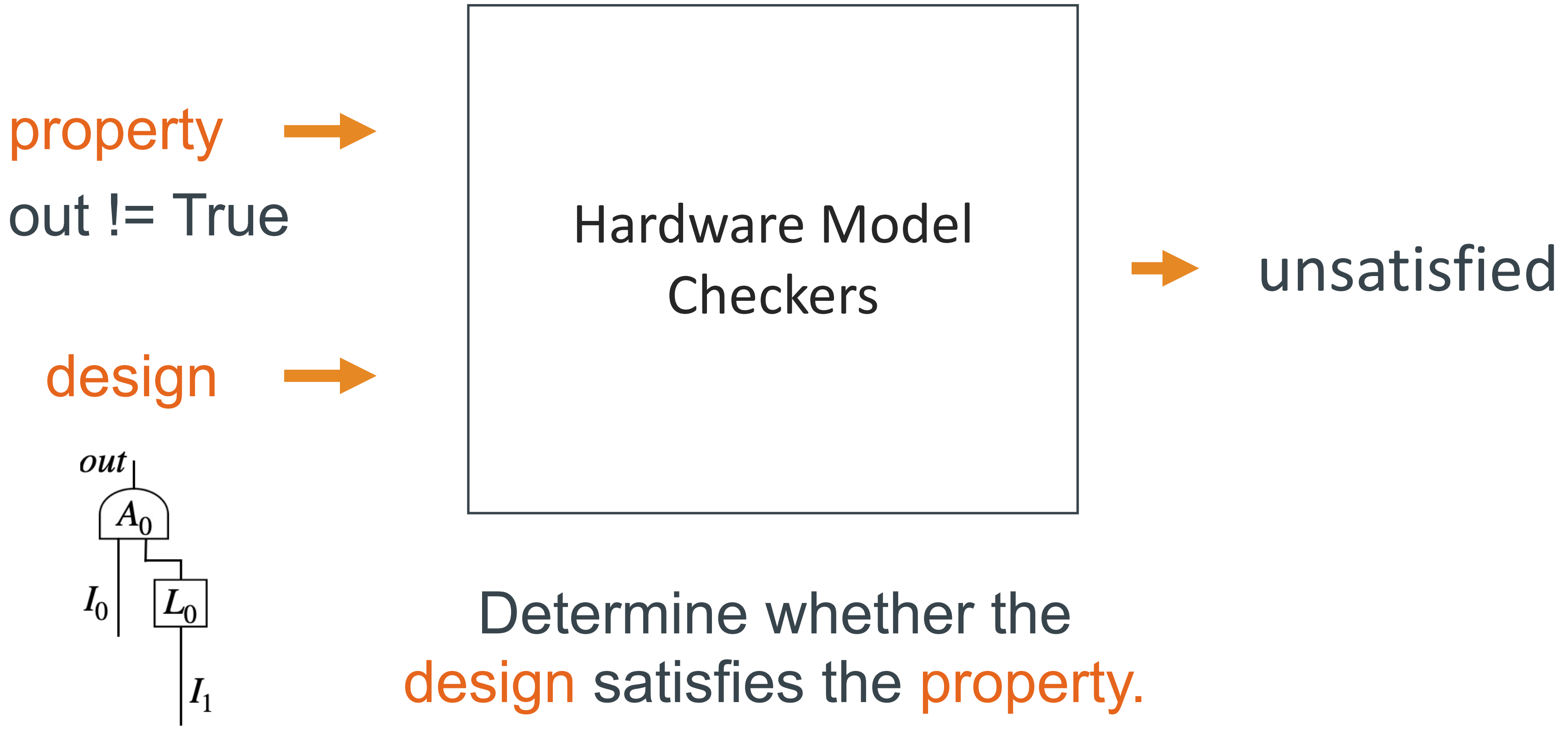
AIGROW: A Feedback-Driven Test Generation Framework for Hardware Model Checkers

Wenjing Deng

Shanghai Key Laboratory of Trustworthy Computing
East China Normal University, China

Hardware Model Checkers

the tools for validating the implementation of hardware designs



property

out != True

design

Hardware Model Checkers

unsatisfied

Determine whether the design satisfies the property.

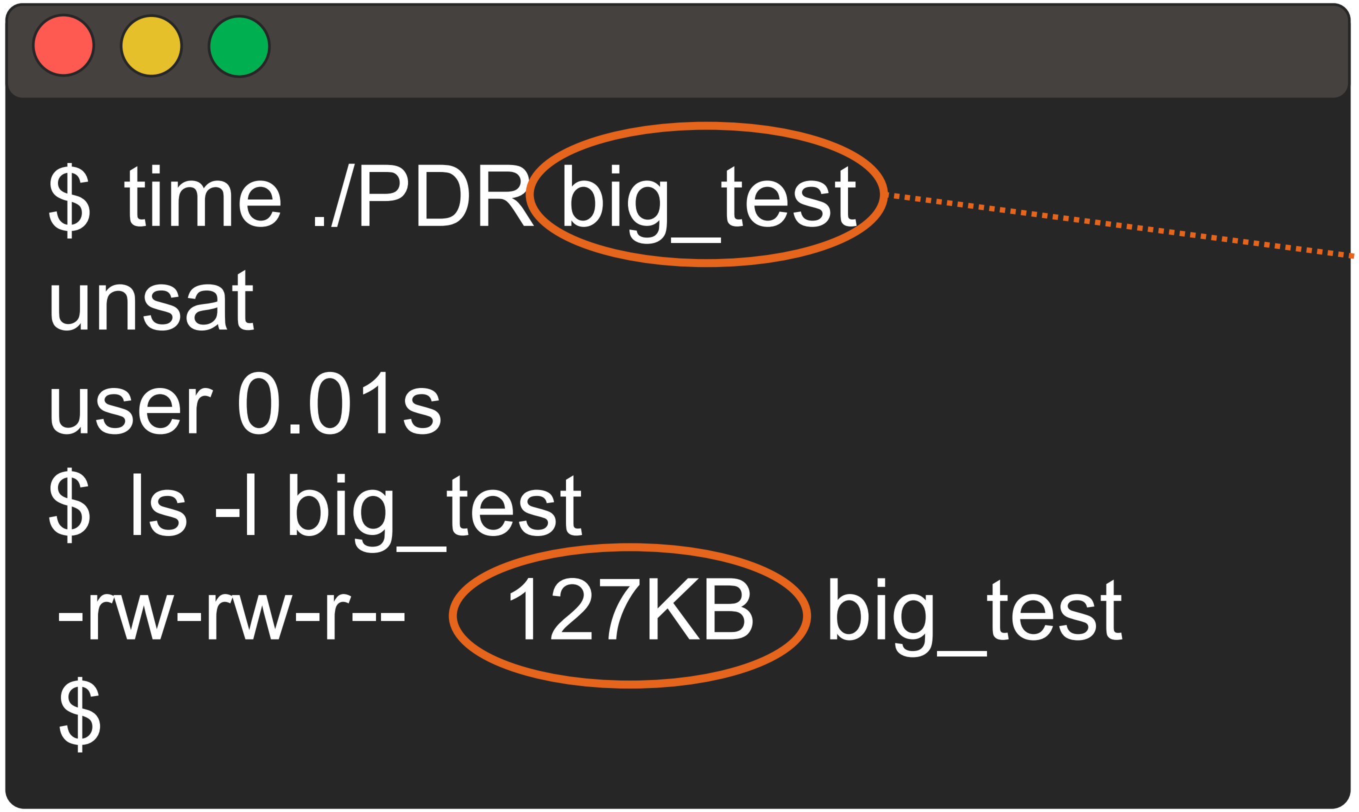
Hardware Model Checkers

the tools for validating the implementation of hardware designs

They should be intensively **evaluated!**

Hardware Model Checkers

PDR: a mature hardware model checker

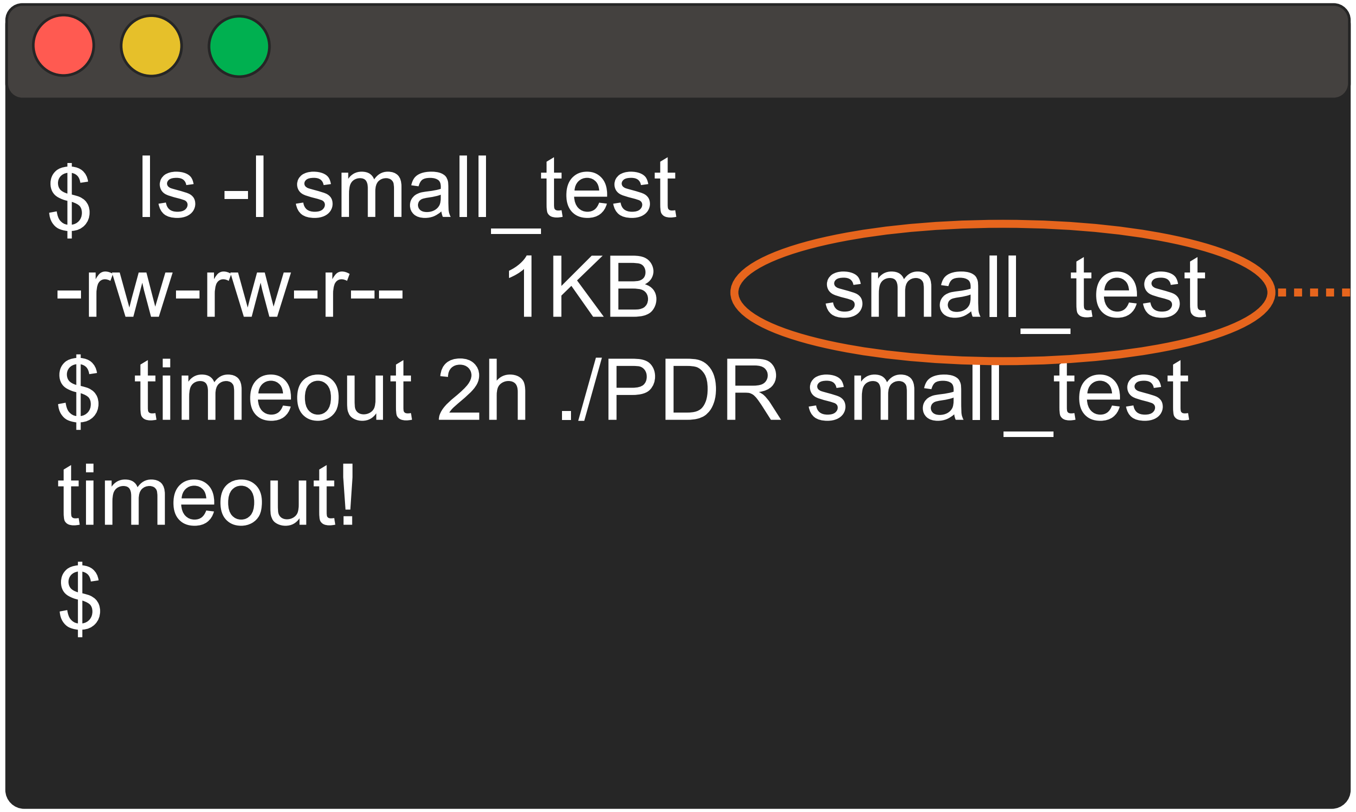


```
$ time ./PDR big_test
unsat
user 0.01s
$ ls -l big_test
 -rw-rw-r--   127KB  big_test
 $
```

From Intel.zip

Hardware Model Checkers

PDR: a mature hardware model checker



```
$  ls -l small_test
-rw-rw-r--    1KB    small_test
$ timeout 2h ./PDR small_test
timeout!
$
```

High-quality test

Difficult

Small size

Diverse

Hard to obtain!

Introduction

Why high-quality tests are hard to obtain?

Source of Tests	Availability	Difficulty	Avg. size	Diversity
Real world benchmarks	no	**yes**	large	**high**
generation tools				
Our approach				

Introduction

Why high-quality tests are hard to obtain?

Source of Tests	Availability	Difficulty	Avg. size	Diversity
Real world benchmarks	no	**yes**	large	**high**
AIGEN[1]				
AIGFUZZ[2]				
Our approach				

[1] Jacobs, S., Sakr, M.: Aigen: Random generation of symbolic transition systems. In: CAV (2021).
[2] Aiger, http://fmv.jku.at/aiger/

Introduction

Why high-quality tests are hard to obtain?

Source of Tests	Availability	Difficulty	Avg. size	Diversity
Real world benchmarks	no	**yes**	large	**high**
AIGEN[1]	**yes**	no	**small**	low
AIGFUZZ[2]				
Our approach				

[1] Jacobs, S., Sakr, M.: Aigen: Random generation of symbolic transition systems. In: CAV (2021).

[2] Aiger, http://fmv.jku.at/aiger/

Introduction

Why high-quality tests are hard to obtain?

Source of Tests	Availability	Difficulty	Avg. size	Diversity
Real world benchmarks	no	**yes**	large	**high**
AIGEN[1]	**yes**	no	**small**	low
AIGFUZZ[2]	**yes**	no	large	**high**
Our approach				

[1] Jacobs, S., Sakr, M.: Aigen: Random generation of symbolic transition systems. In: CAV (2021).

[2] Aiger, http://fmv.jku.at/aiger/

Introduction

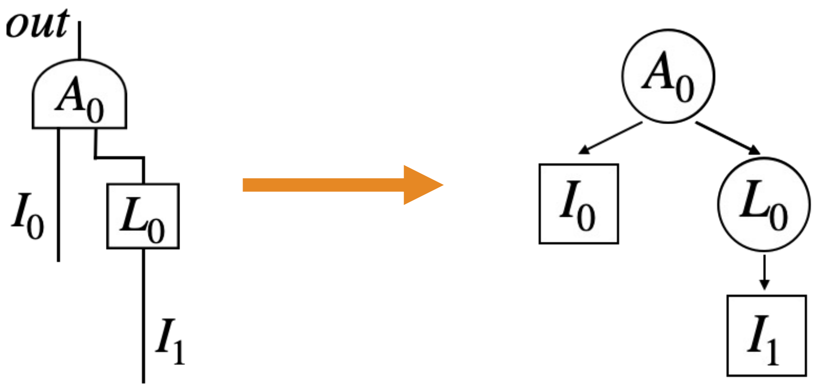
Why high-quality tests are hard to obtain?

Source of Tests	Availability	Difficulty	Avg. size	Diversity
Real world benchmarks	no	**yes**	large	**high**
AIGEN[1]	**yes**	no	**small**	low
AIGFUZZ[2]	**yes**	no	large	**high**
Our approach	**yes**	**yes**	**small**	**high**

[1] Jacobs, S., Sakr, M.: Aigen: Random generation of symbolic transition systems. In: CAV (2021).
[2] Aiger, http://fmv.jku.at/aiger/

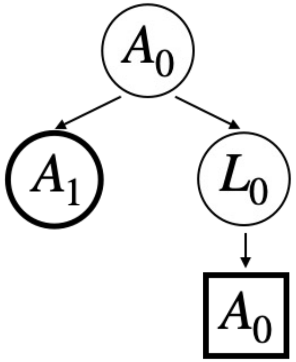
Our approach is to **incrementally generate** tests with **feedback**.

Our Approach: Incremental Generation



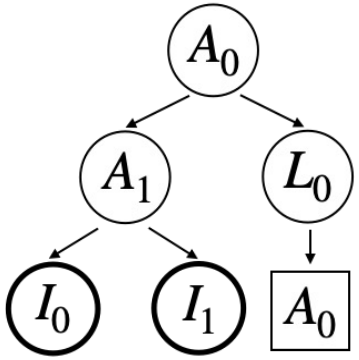
Remove Inputs

Our Approach: Incremental Generation



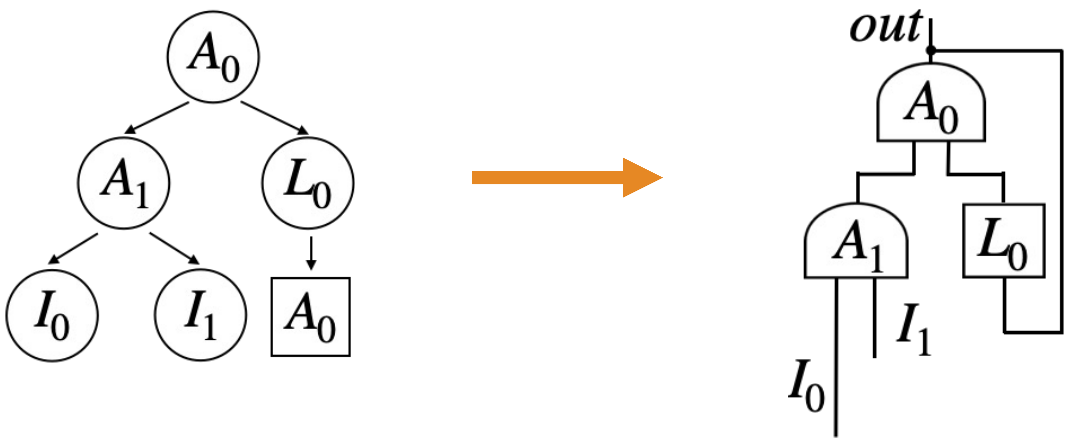
Add Nodes

Our Approach: Incremental Generation

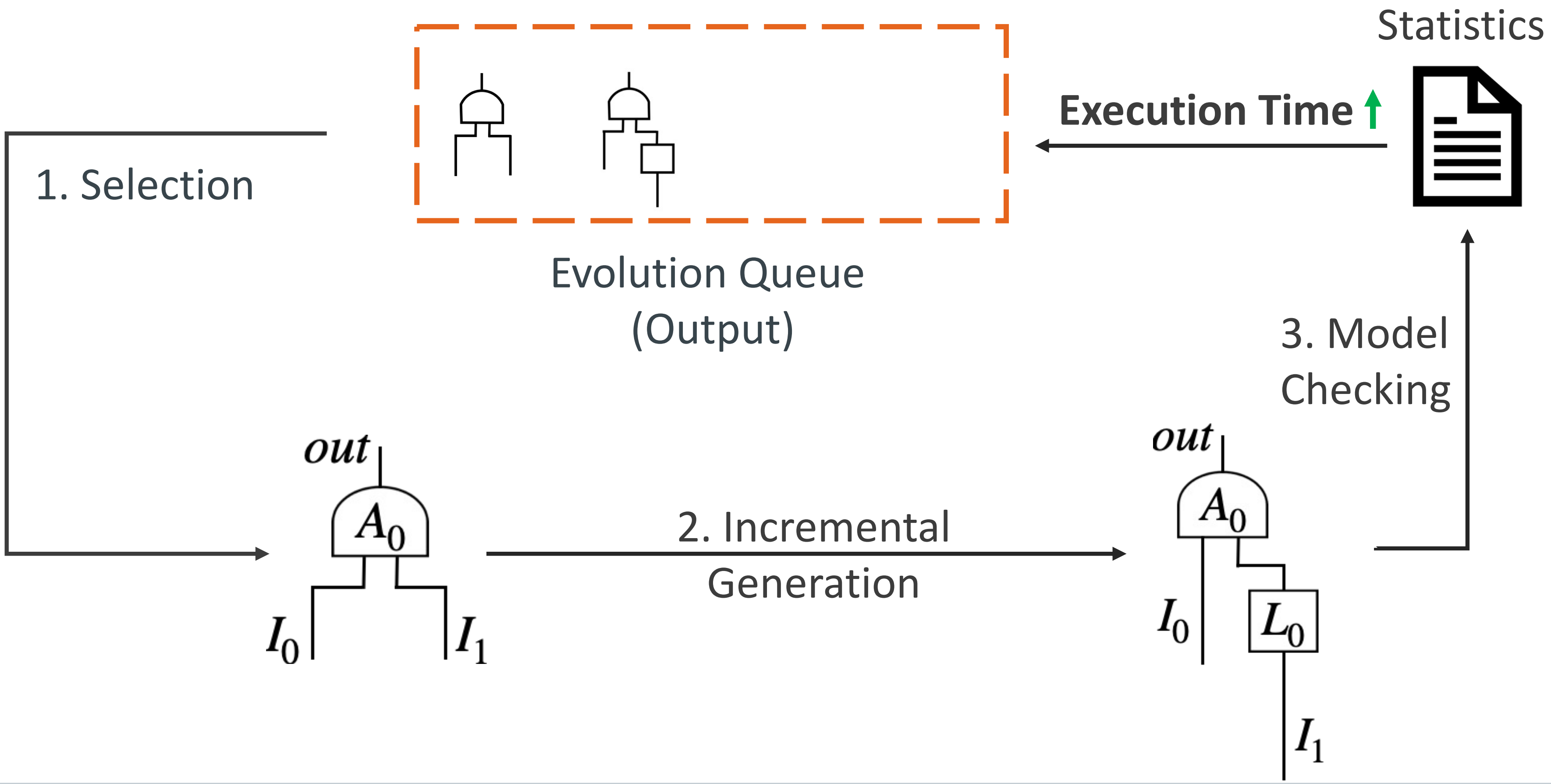


Add Inputs

Our Approach: Incremental Generation



Our Feedback-Driven Framework: AIGROW



Statistics

Execution Time ↑

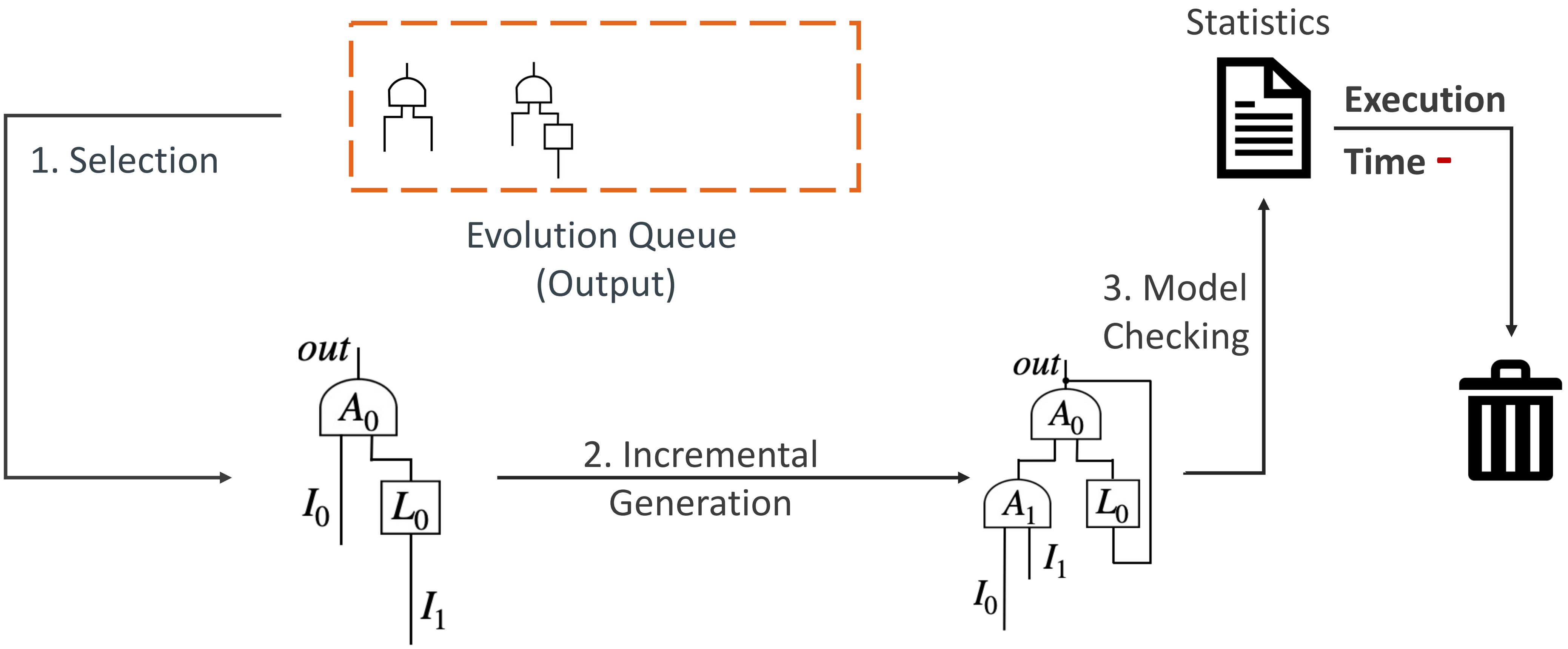
1. Selection

Evolution Queue (Output)

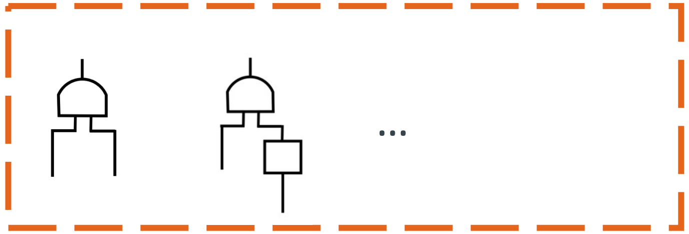
2. Incremental Generation

3. Model Checking

Our Feedback-Driven Framework: AIGROW



Our Feedback-Driven Framework: AIGROW



Evolution Queue
(Output)

Evaluation Setting

Testing targets

PDR: a mature hardware model checker

IC3ref: a reference implementation of IC3 algorithm

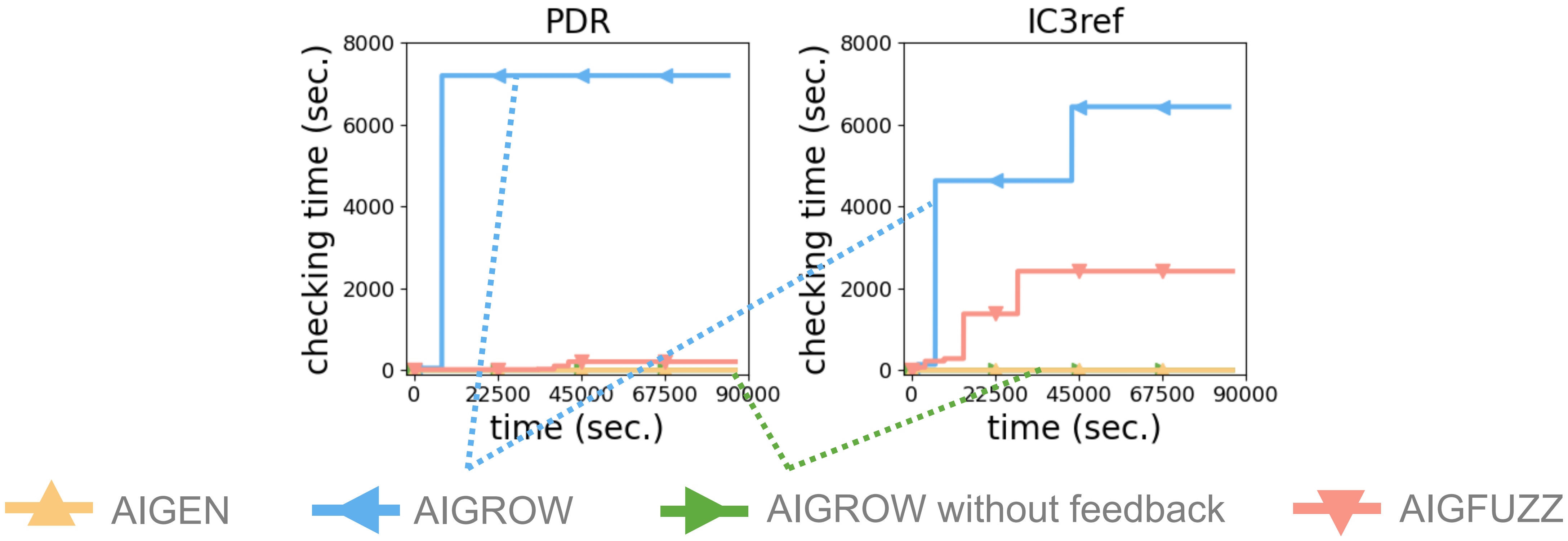
Evaluation metrics

Difficulty: checking time (sec.)

Size: #units

Diversity: line coverage (%)

Result: Difficulty



AIGROW generates **more difficult** tests.

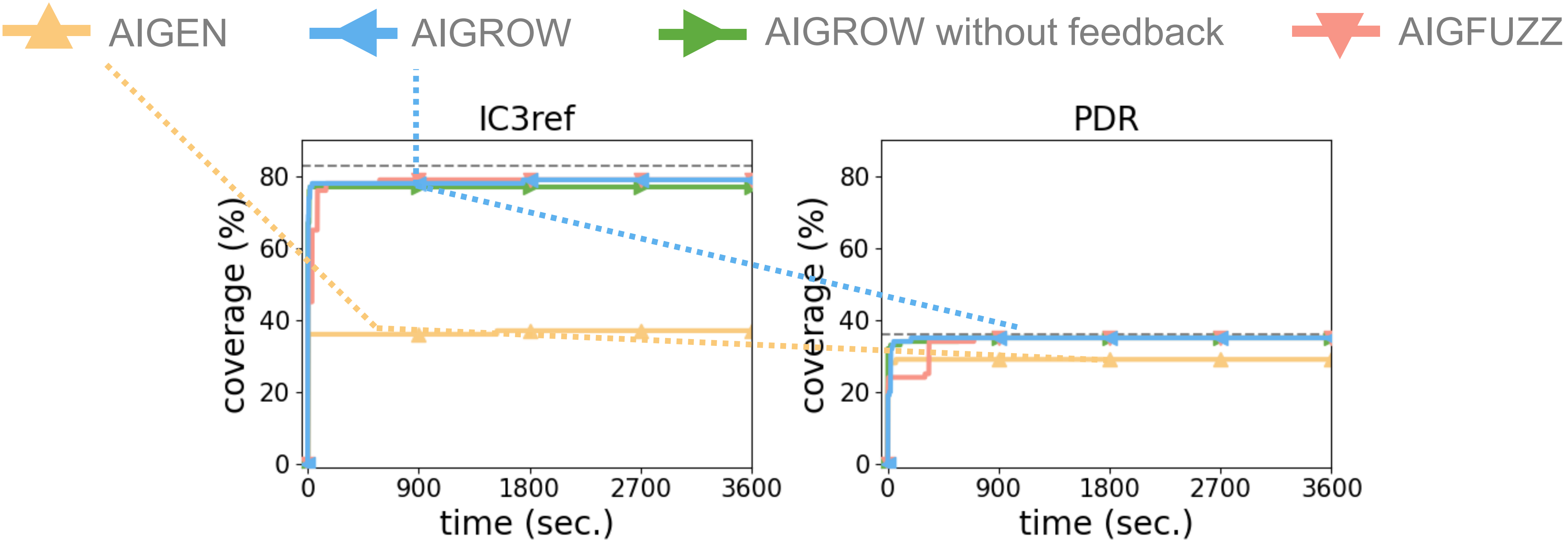
Feedback strategy make difference in generating difficult tests.

Result: Avg. Size (#units)

Hardware Model Checker	AIGROW	AIGEN	AIGFUZZ
PDR	**109**	6,871	1,910
IC3ref	**40**	6,867	1,940

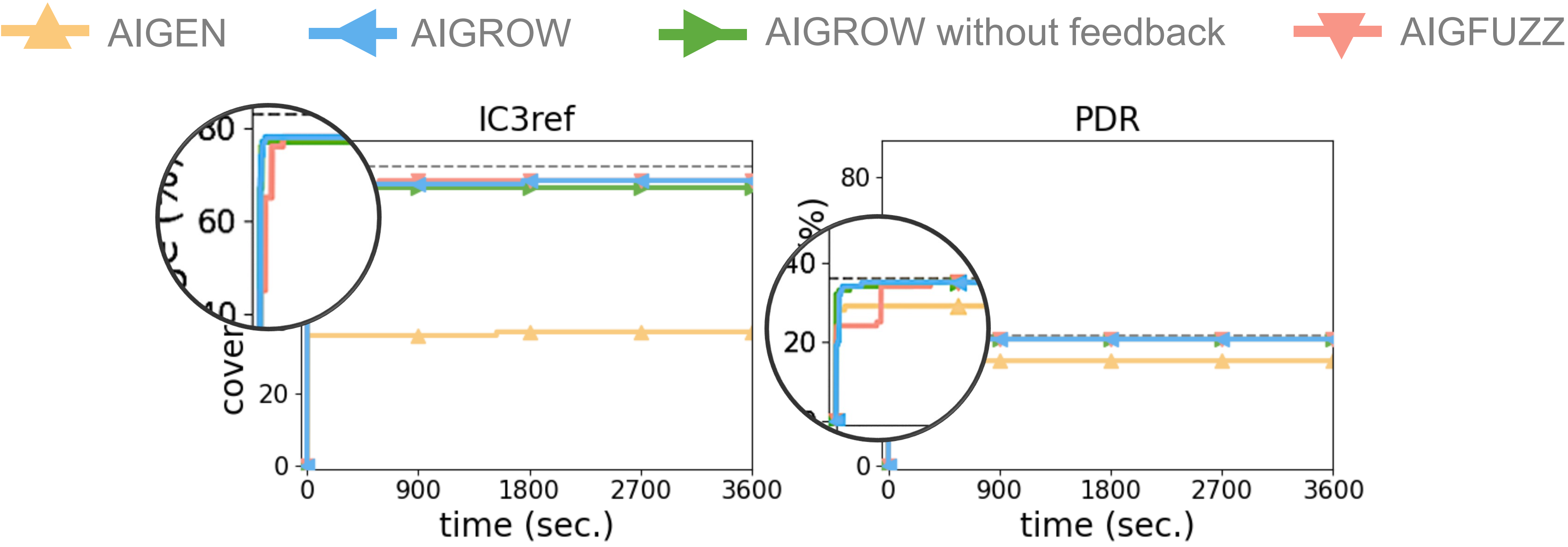
The avg. size of the tests generated by AIGROW is **10x smaller** than AIGEN and AIGFUZZ

Result: Coverage



AIGROW achieves **higher coverage** than AIGEN.

Result: Coverage



AIGROW achieves **higher coverage** than AIGEN.

AIGROW achieves the maximum coverage **faster** than AIGFUZZ.

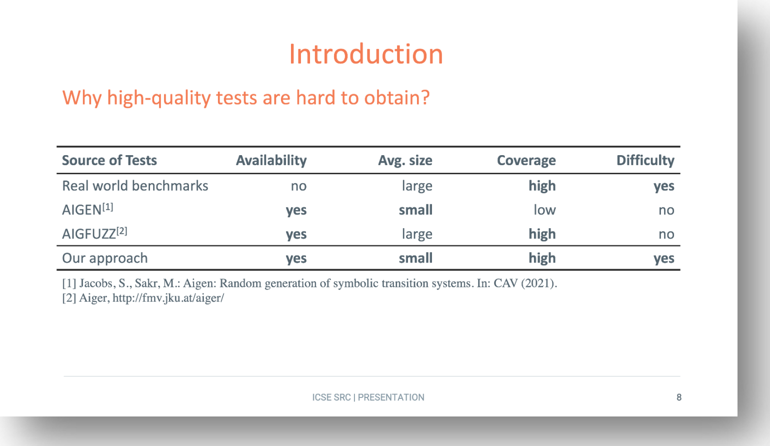
Future Work

- Apply this approach to test other **model-based design tools.**

SIMULINK VIVADO™

- Our preliminary investigation found **21 critical bugs** in the model-based design tool of our industry partner.

Conclusion



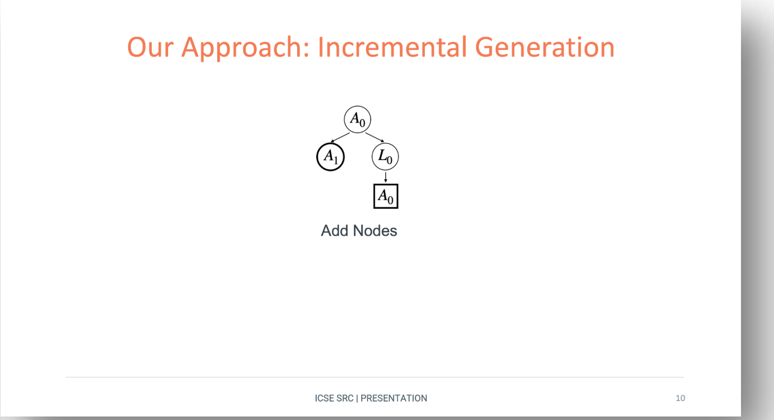
Introduction

Why high-quality tests are hard to obtain?

Source of Tests	Availability	Avg. size	Coverage	Difficulty
Real world benchmarks	no	large	**high**	**yes**
AIGEN[1]	**yes**	**small**	low	no
AIGFUZZ[2]	**yes**	large	**high**	no
Our approach	**yes**	**small**	**high**	**yes**

[1] Jacobs, S., Sakr, M.: Aigen: Random generation of symbolic transition systems. In: CAV (2021).
[2] Aiger, http://fmv.jku.at/aiger/

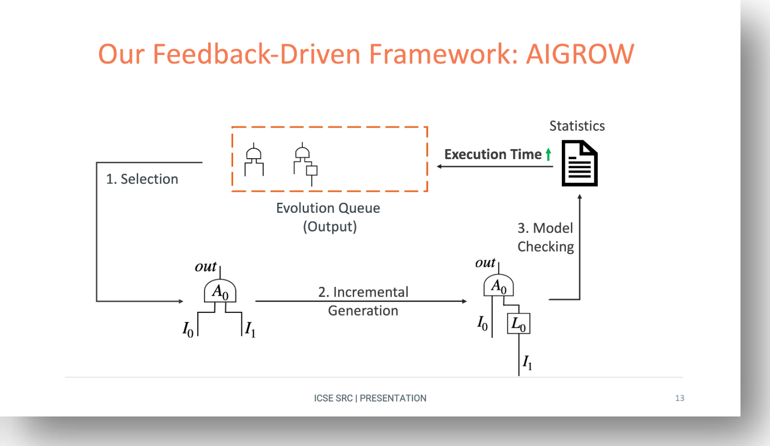
ICSE SRC | PRESENTATION 8



Our Approach: Incremental Generation

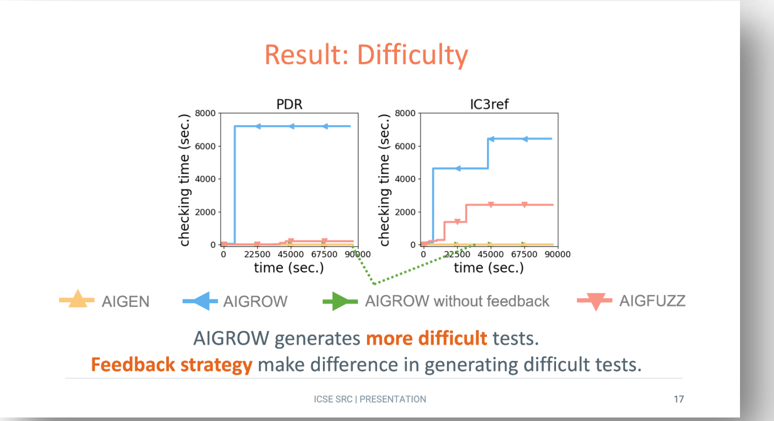
Add Nodes

ICSE SRC | PRESENTATION 10



Our Feedback-Driven Framework: AIGROW

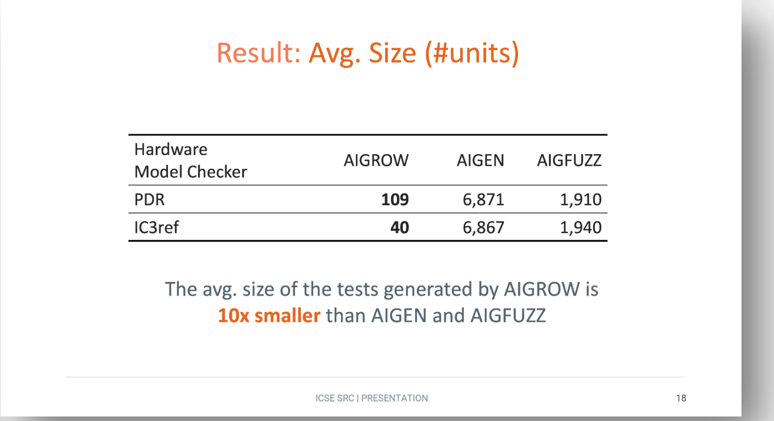
ICSE SRC | PRESENTATION 13



Result: Difficulty

AIGROW generates **more difficult** tests.
Feedback strategy make difference in generating difficult tests.

ICSE SRC | PRESENTATION 17

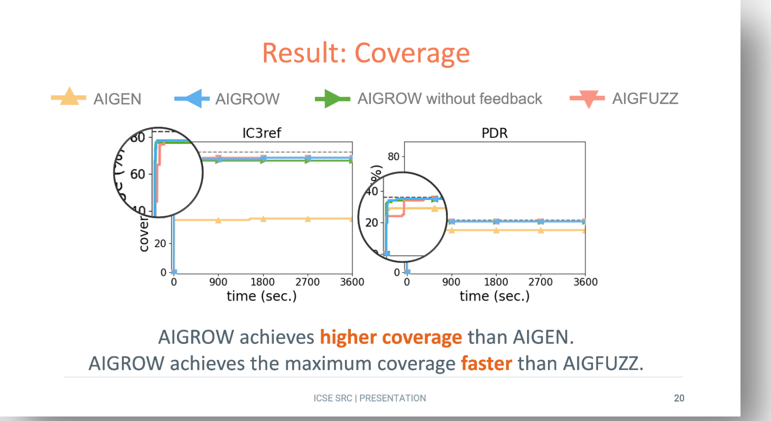


Result: Avg. Size (#units)

Hardware Model Checker	AIGROW	AIGEN	AIGFUZZ
PDR	**109**	6,871	1,910
IC3ref	**40**	6,867	1,940

The avg. size of the tests generated by AIGROW is **10x smaller** than AIGEN and AIGFUZZ

ICSE SRC | PRESENTATION 18



Result: Coverage

AIGROW achieves **higher coverage** than AIGEN.
AIGROW achieves the maximum coverage **faster** than AIGFUZZ.

ICSE SRC | PRESENTATION 20

Take-aways

Approach

- **Incremental generation** guided by **feedback**

Perform better than existing tools

- Generate **more difficult** test cases.
- Achieve maximum coverage **faster.**
- **Fewer redundant units** in a single test.

Introduction

Why high-quality tests are hard to obtain?

Def. High-quality Tests.

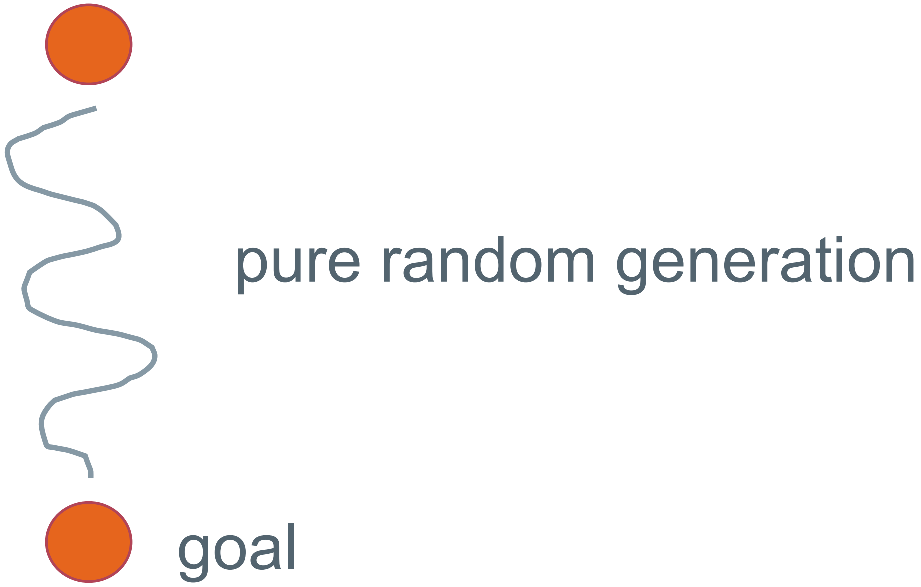
• **Hard to solve**	**Evaluation**
• **Diverse** among tests	**Validation**
• **Fewer redundant units** in a single test	**Debug**

Source of Tests	Always available	Fewer Redundant Units	Diverse	Hard to Solve
Real world	**no**	**no**	**yes**	**yes**
AIGEN	**yes**	**yes**	**no**	**no**
AIGFUZZ	**yes**	**no**	**yes**	**no**

Our Approach avoid these shortcomings.

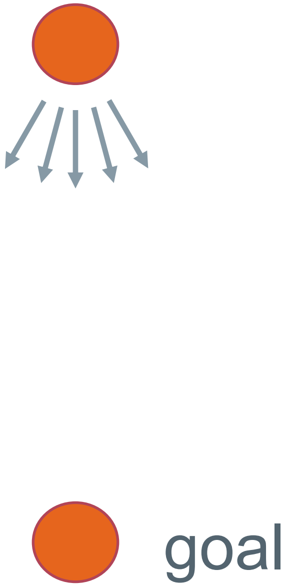
Approach

Our Approach is to **incrementally** generate tests with **feedback.**



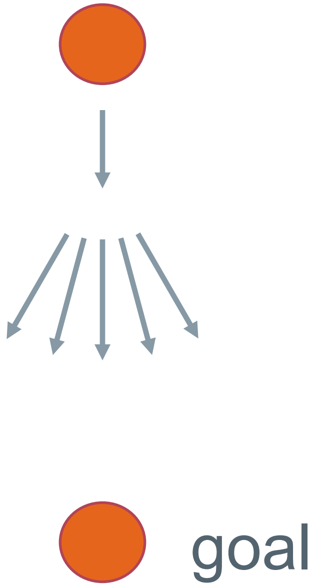
Approach

Our Approach is to **incrementally** generate tests with **feedback.**



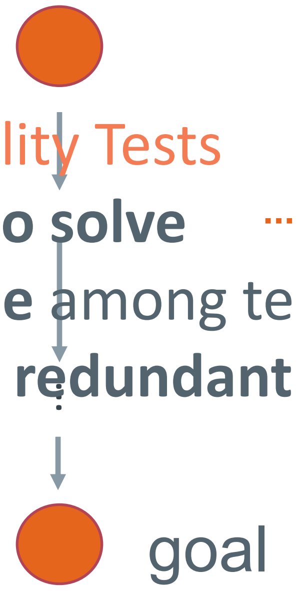
Approach

Our Approach is to **incrementally** generate tests with **feedback.**



Approach

Our Approach is to **incrementally** generate tests with **feedback.**



High-quality Tests

- **Hard to solve** .... Solving time
- **Diverse** among tests .... Coverage
- **Fewer redundant units** in a single test

Layout

Background

Introduction

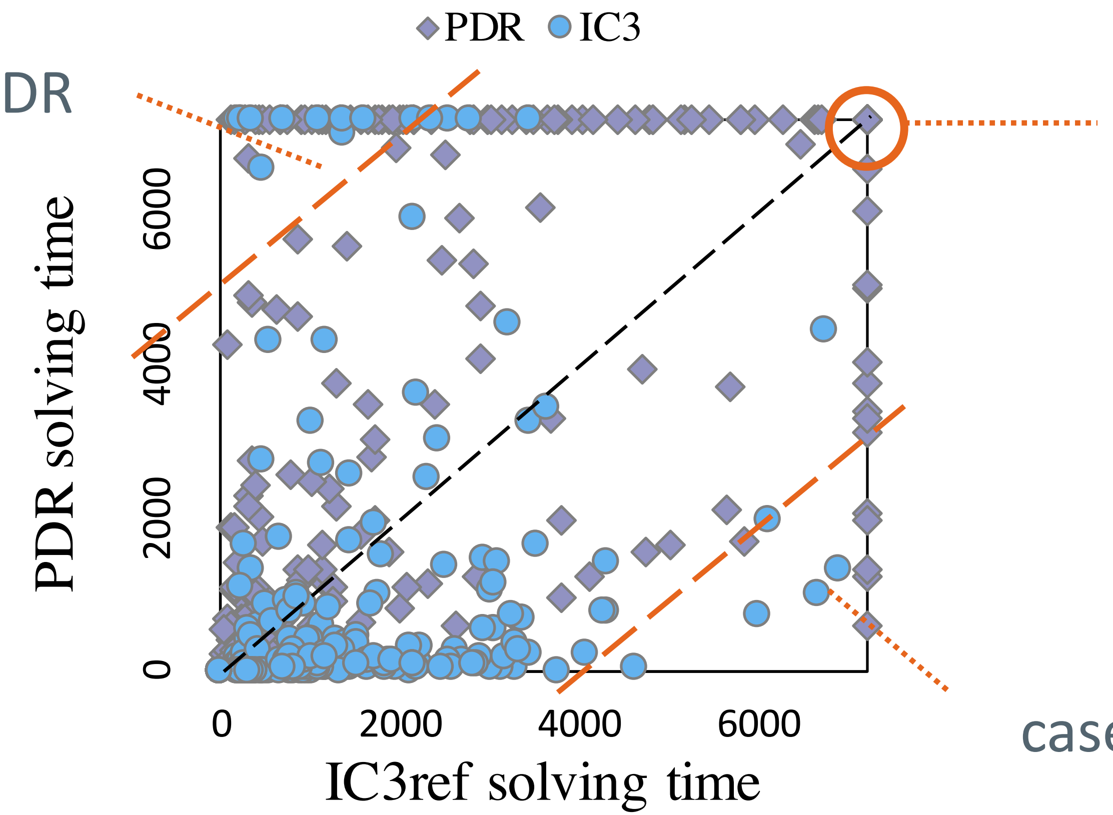
Approach

Result

Take-away

Diversity of Our Benchmarks

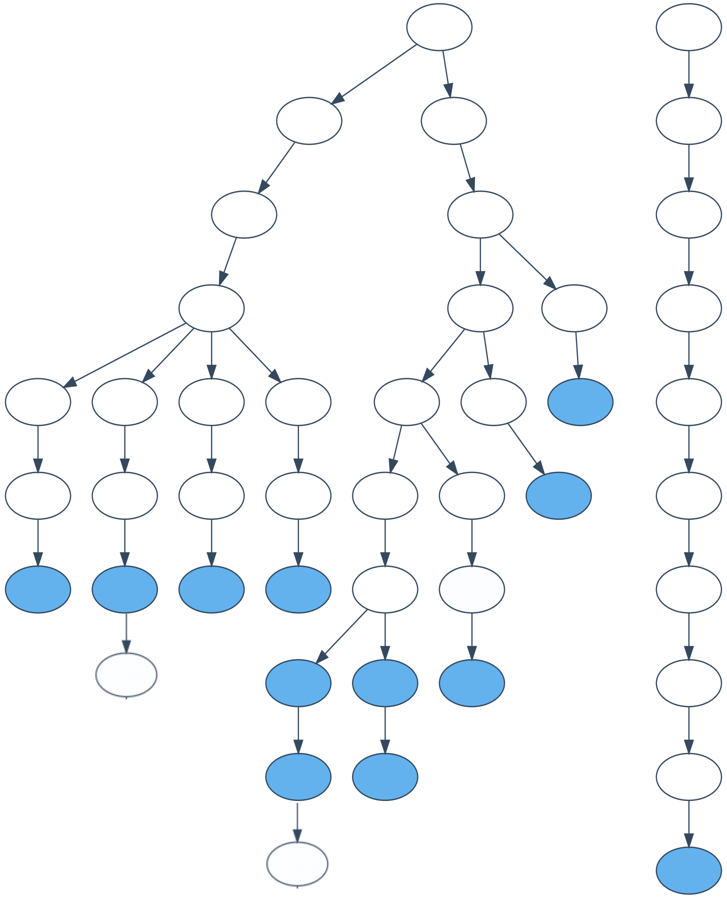
cases: IC3ref << PDR



73 cases: hard to solve for both

cases: IC3ref >> PDR

Generation Process for PDR



Background

Def. Hardware Model Checkers.

The tools for validating the implementation of hardware designs.

Hardware model checkers should be intensively **validated** and **evaluated.**

High-quality tests

hard to obtain!