Falsification of Cyber-Physical Systems with Robustness-Guided Black-Box Checking

Masaki Waga[1,2,3]

National Institute of Informatics[1], SOKENDAI[2],
JSPS Research Fellow[3]

April 2020, HSCC 2020

This work is partially supported by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), by JSPS Grants-in-Aid No. 15KT0012 & 18J22498

Testing is Important…

BBC | Sign in | News | Sport | Reel | Worklife | Travel | Future | More | Search

NEWS

Home | Video | World | Asia | UK | Business | Tech | Science | Stories | Entertainment & Arts | Health | World News TV | More

Technology

Tesla Model 3: Autopilot engaged during fatal crash

17 May 2019

Share

The Tesla Model 3 after the crash

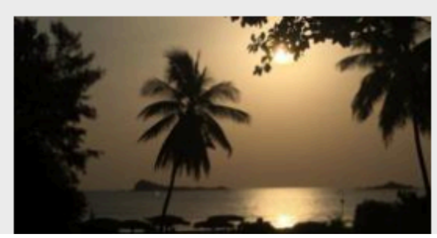
Top Stories

Trump lashes out at impeachment inquiry
Using a vulgarity, the Republican president accuses Democrats of dishonesty and even treason.
7 hours ago

'Grave moment' as N Korea tests submarine missile
27 minutes ago

HK rubber bullet blinds journalist in one eye
2 October 2019

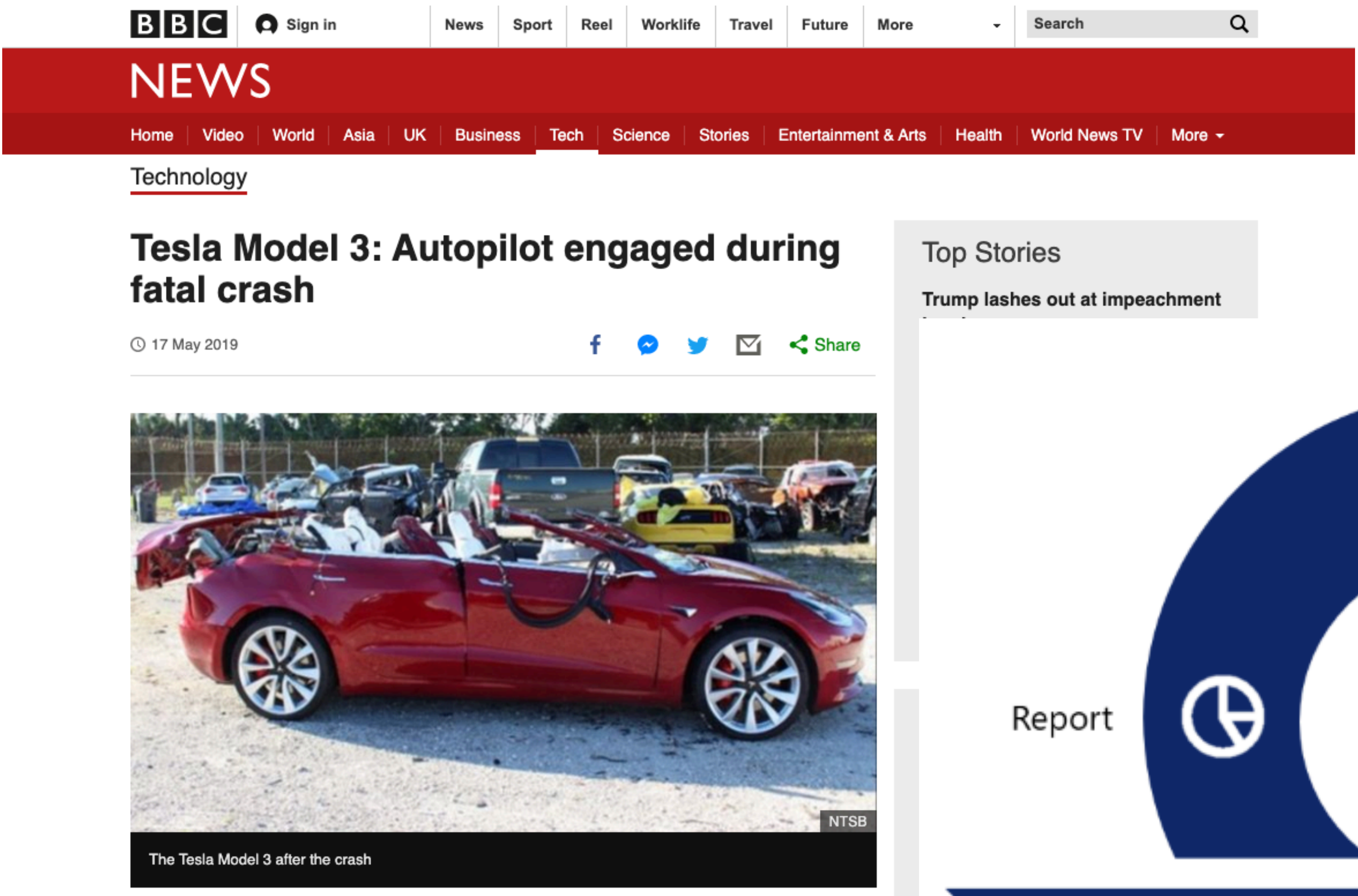
Features

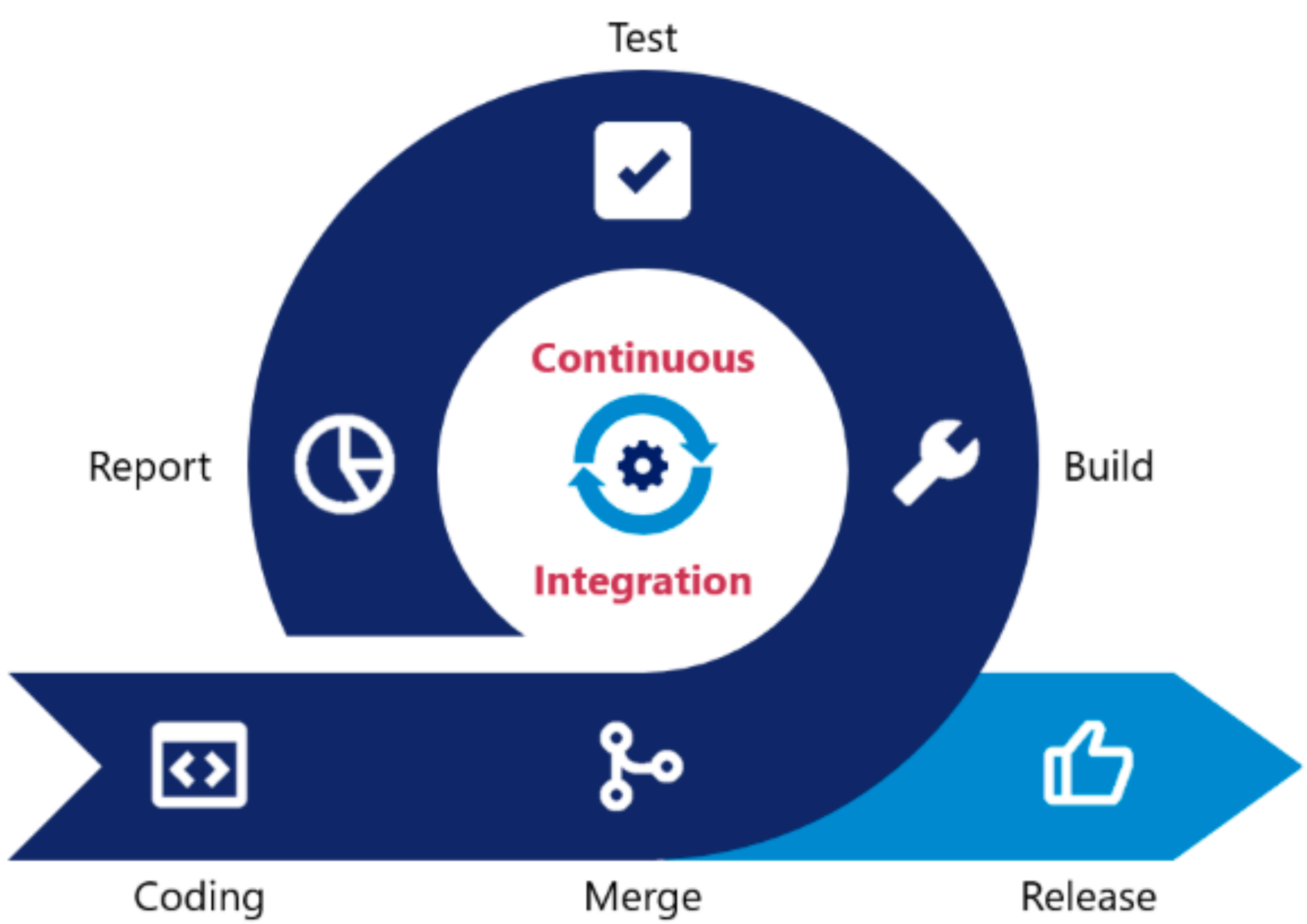


https://www.bbc.com/news/technology-48308852

https://www.qfs.de/fr/blog/article/2019/07/11/utiliser-qf-test-dans-des-systemes-continuous-integration-1.html

Testing is Important…



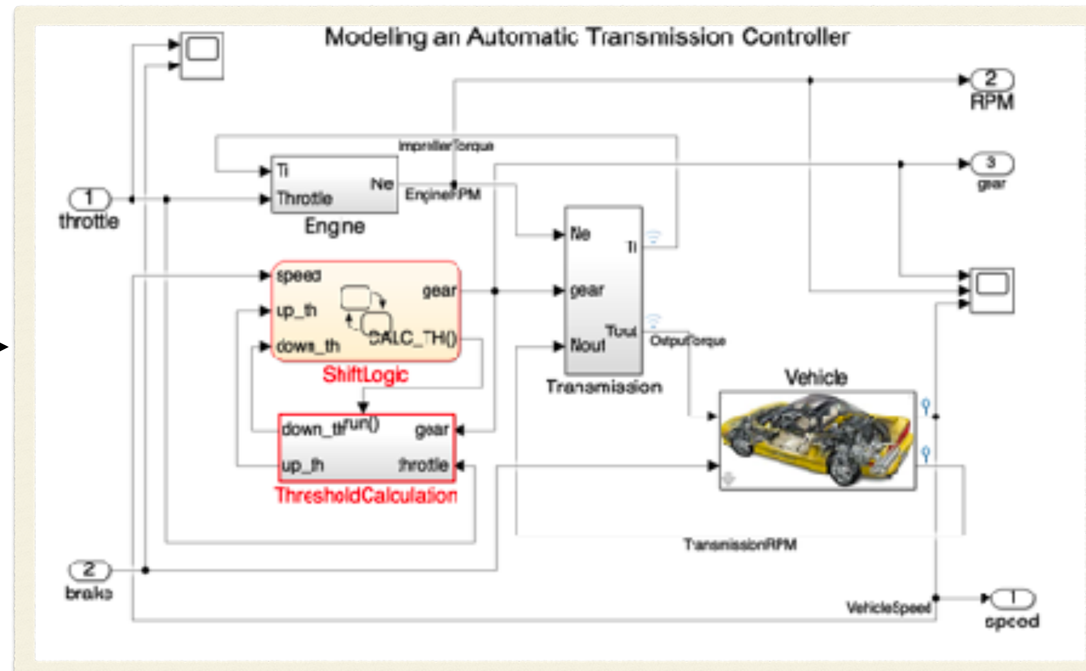


https://www.bbc.com/news/technology-48308852

https://www.qfs.de/fr/blog/article/2019/07/11/utiliser-qf-test-dans-des-systemes-continuous-integration-1.html

Falsification: Robustness + Optimization

$\mathscr{M}$



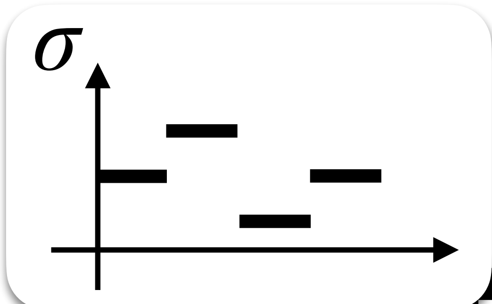
Robustness Monitor
Spec: φ in STL
(signal temporal logic)

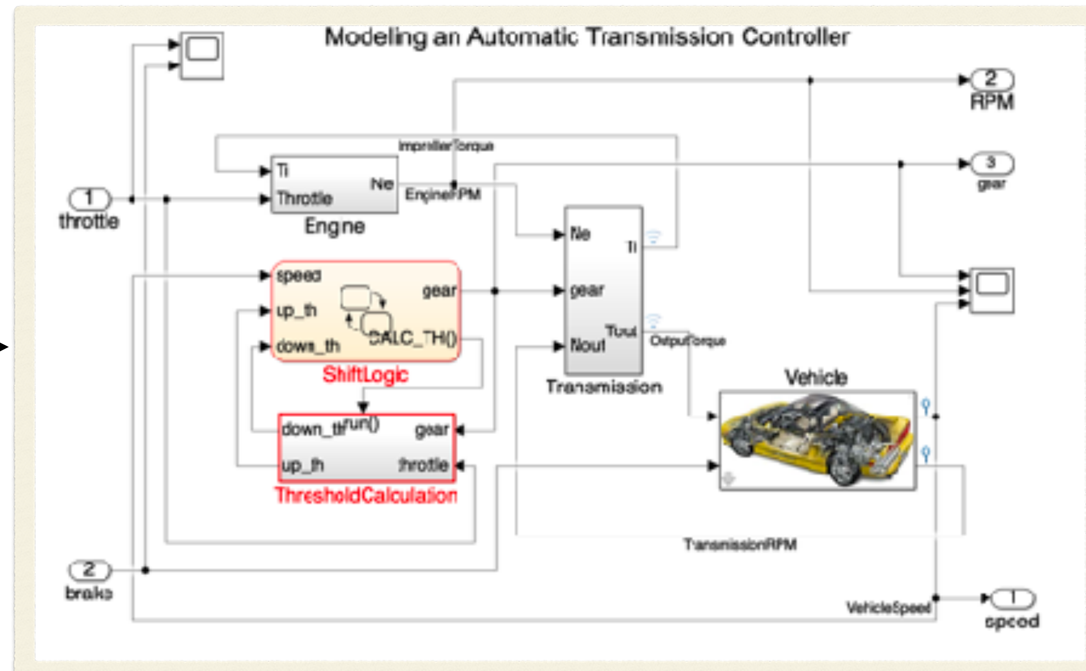
Optimizer
(Minimize Robustness)

M. Waga (NII)

Falsification: Robustness + Optimization

$\mathscr{M}$





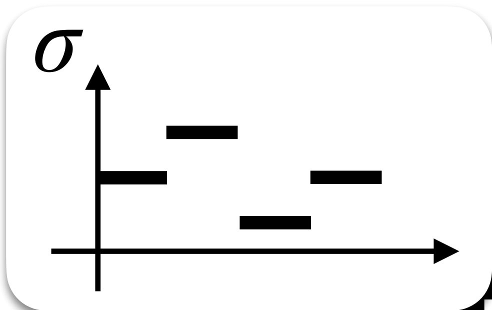
Robustness Monitor
Spec: φ in STL
(signal temporal logic)

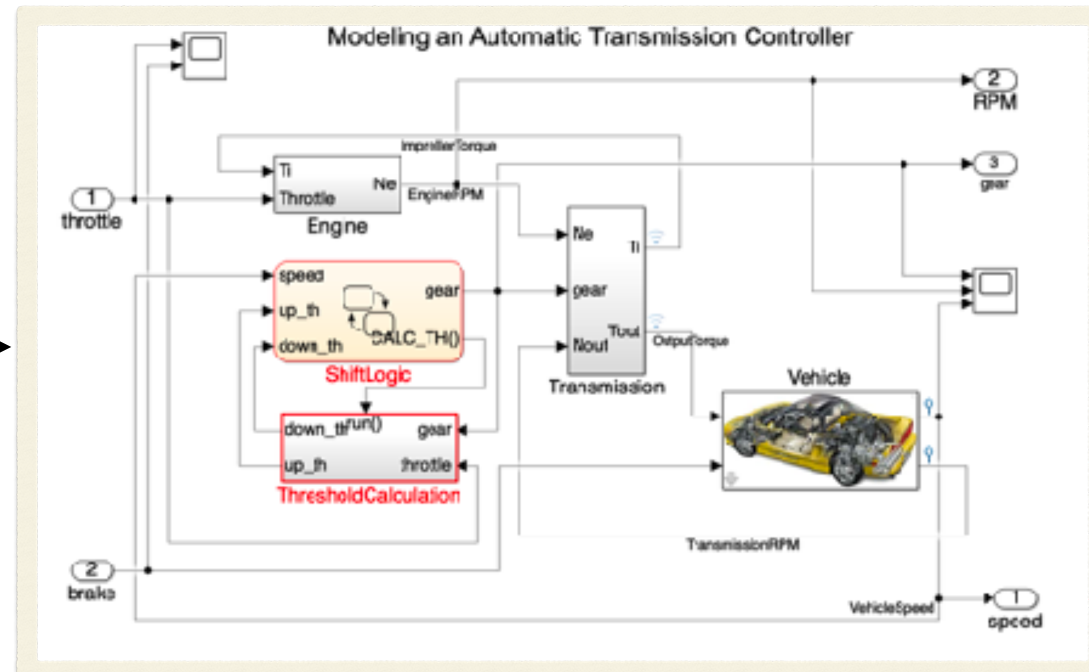
Optimizer
(Minimize Robustness)

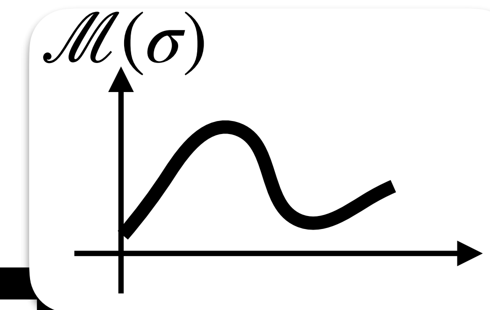
M. Waga (NII)

Falsification: Robustness + Optimization

$\mathscr{M}$





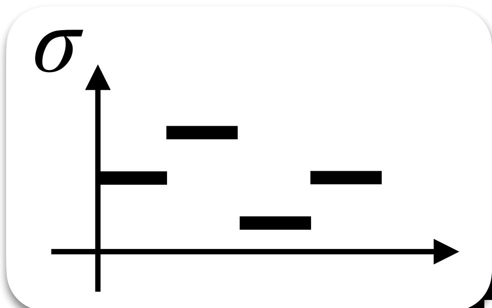


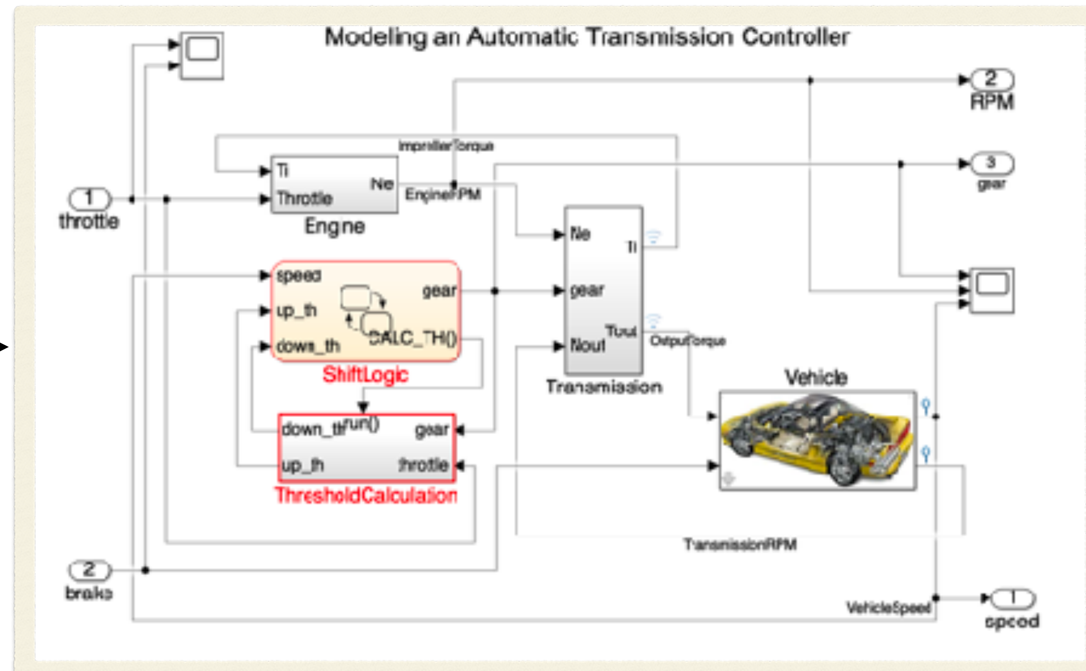
Optimizer
(Minimize Robustness)

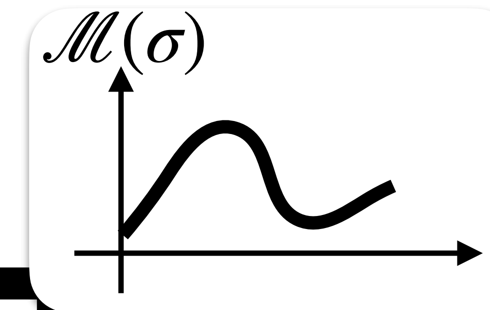
Robustness Monitor
Spec: φ in STL
(signal temporal logic)

Falsification: Robustness + Optimization

$\mathscr{M}$







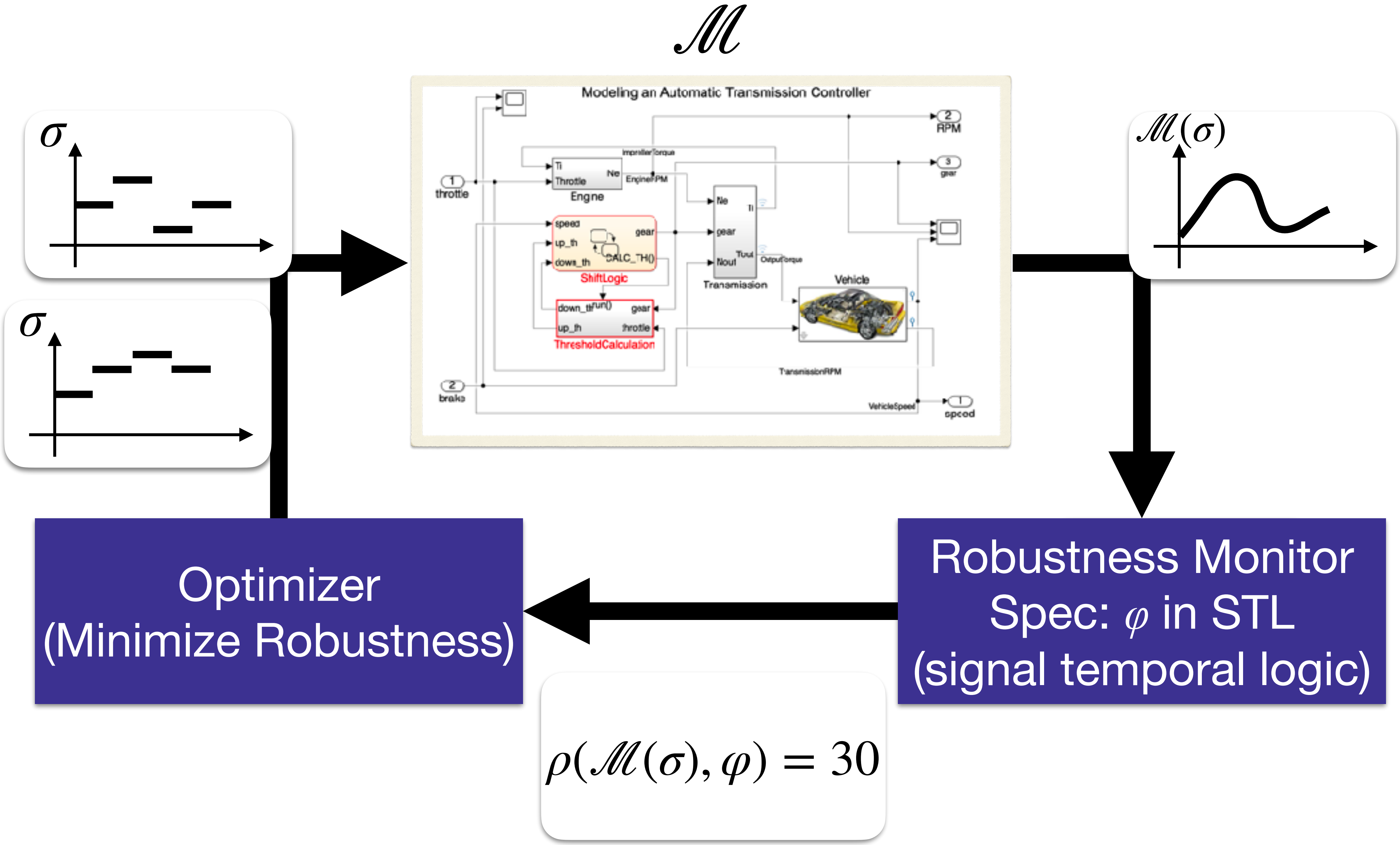
Robustness Monitor
Spec: φ in STL
(signal temporal logic)

$$\rho(\mathscr{M}(\sigma), \varphi) = 30$$

Optimizer
(Minimize Robustness)

M. Waga (NII)

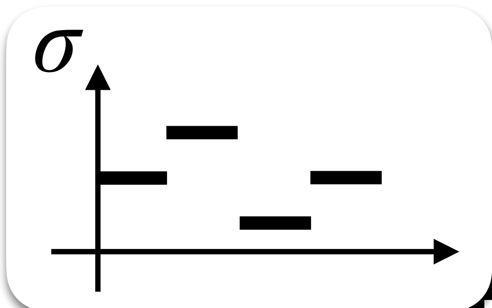
Falsification: Robustness + Optimization

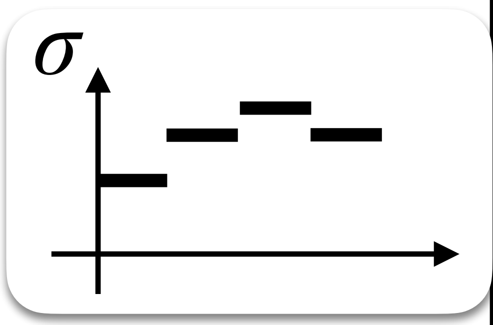


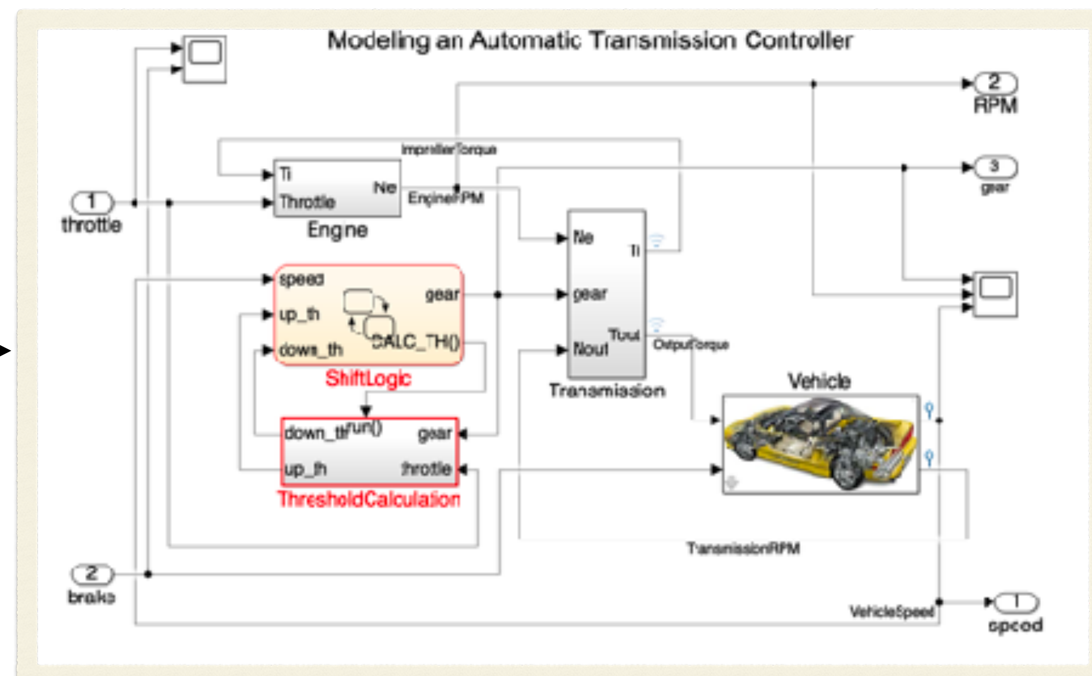
M. Waga (NII)

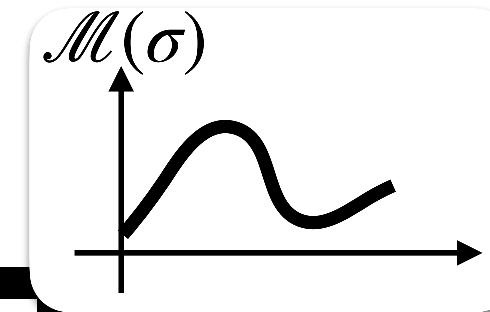
Falsification: Robustness + Optimization

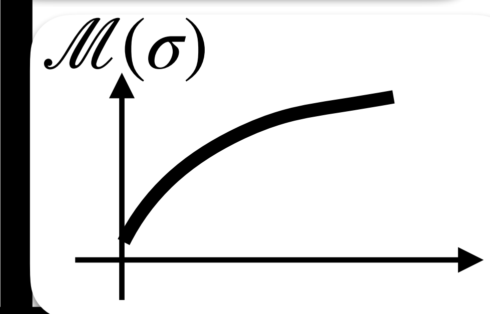
$\mathscr{M}$











Robustness Monitor
Spec: φ in STL
(signal temporal logic)

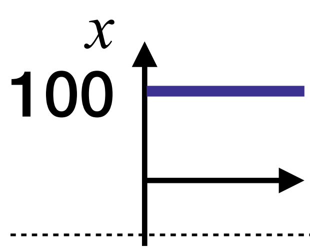
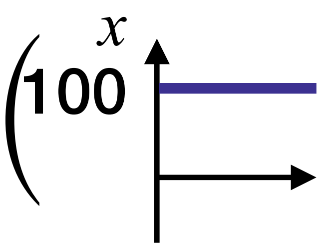
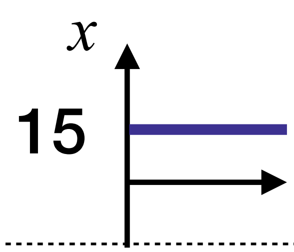
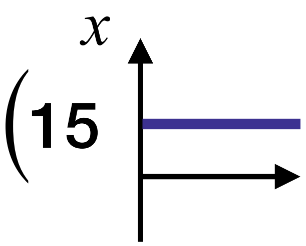
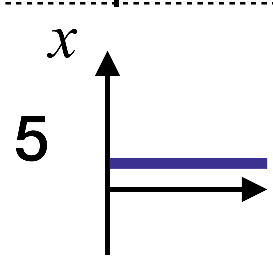
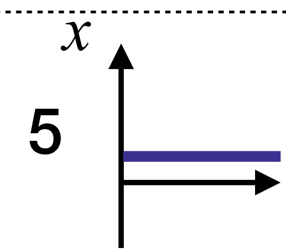
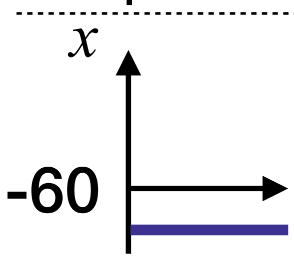
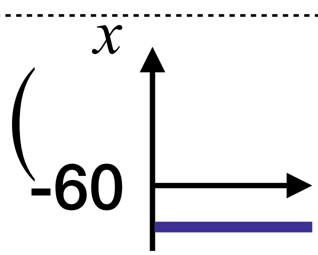
$\rho(\mathscr{M}(\sigma), \varphi) = 30$

Optimizer
(Minimize Robustness)

$\rho(\mathscr{M}(\sigma), \varphi) = -10$

M. Waga (NII)

Robustness of STL

Boolean	Robust
 $\models \Box\, x > 10$	$\rho\Big($  $, \Box\, x > 10\Big) =$ **90**
 $\models \Box\, x > 10$	$\rho\Big($  $, \Box\, x > 10\Big) =$ **5**
 $\not\models \Box\, x > 10$	$\rho\Big($  $, \Box\, x > 10\Big) =$ **-5**
 $\not\models \Box\, x > 10$	$\rho\Big($  $, \Box\, x > 10\Big) =$ **-70**

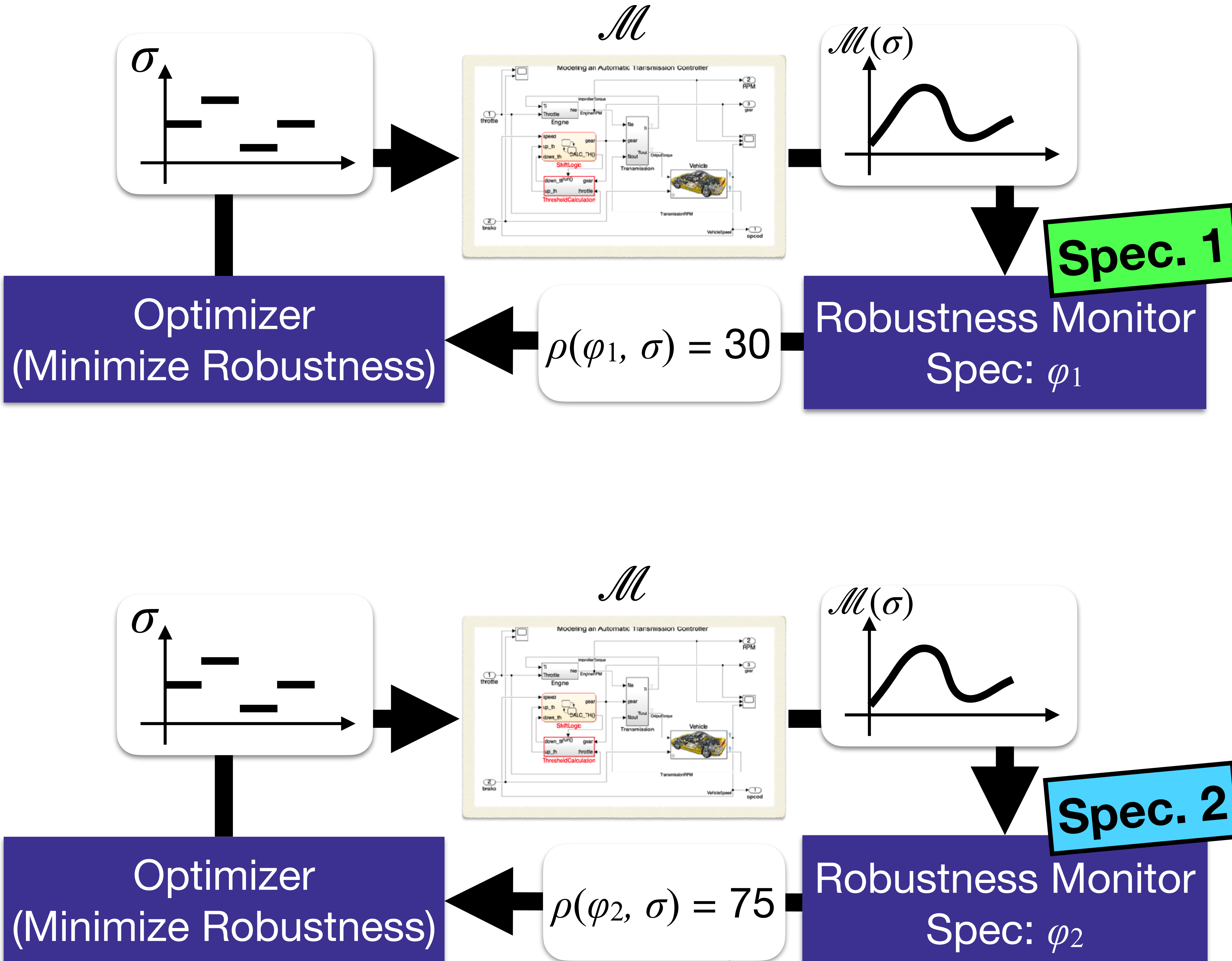
Clearly satisfied

Boarder

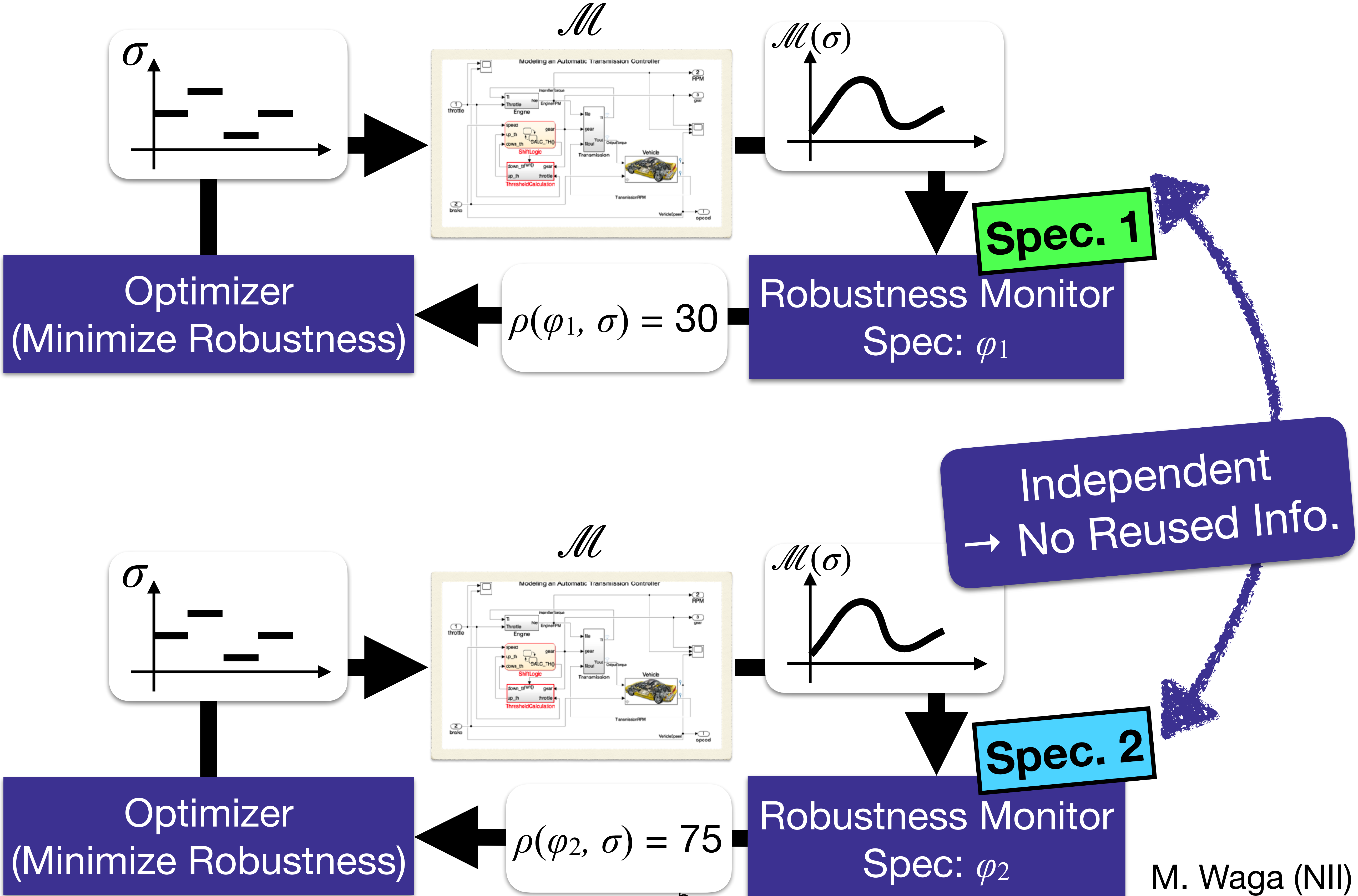
Clearly violated

M. Waga (NII)

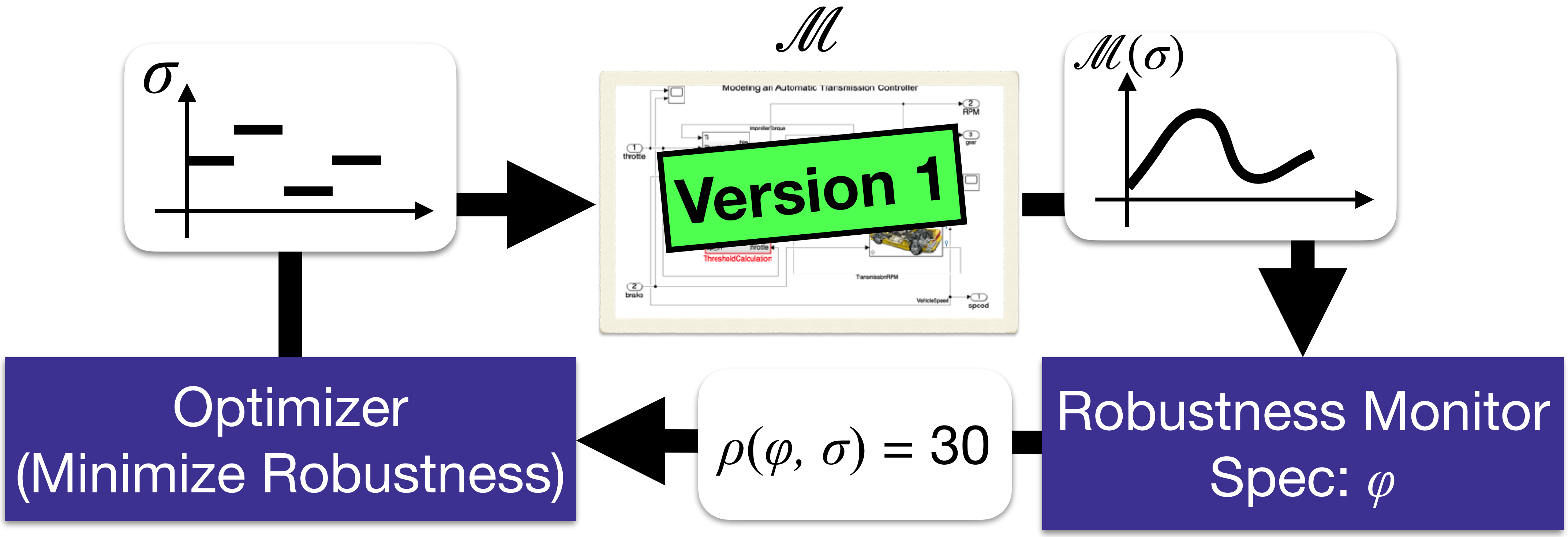
Falsification is Oneshot

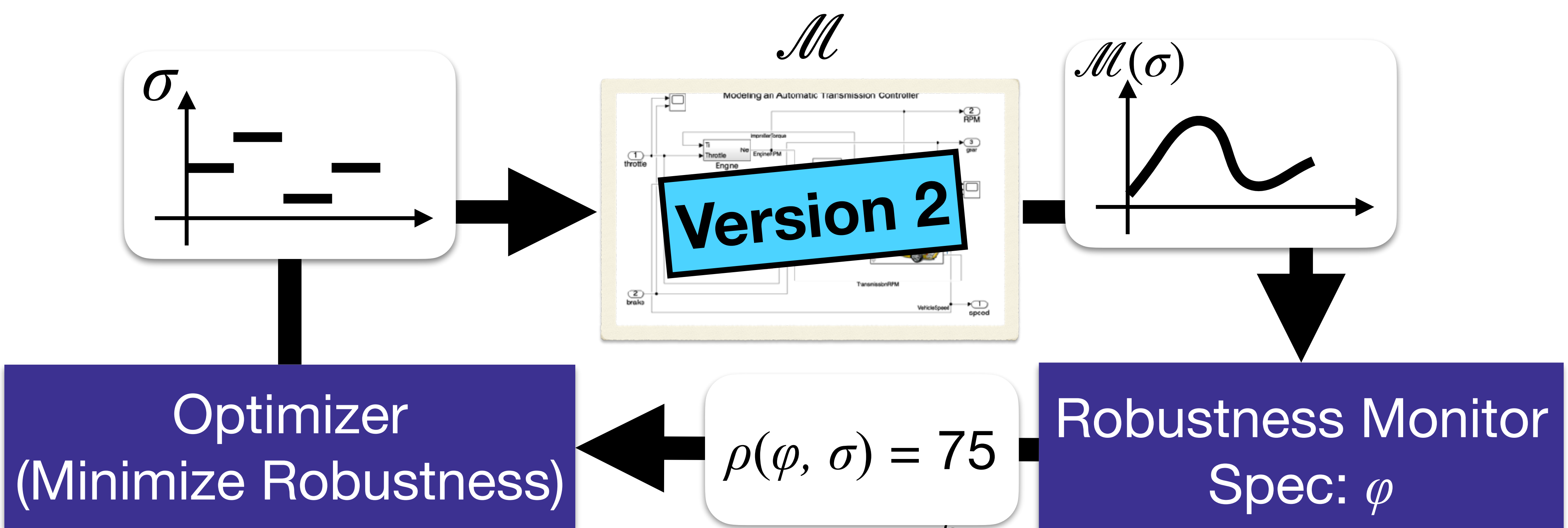


Falsification is Oneshot

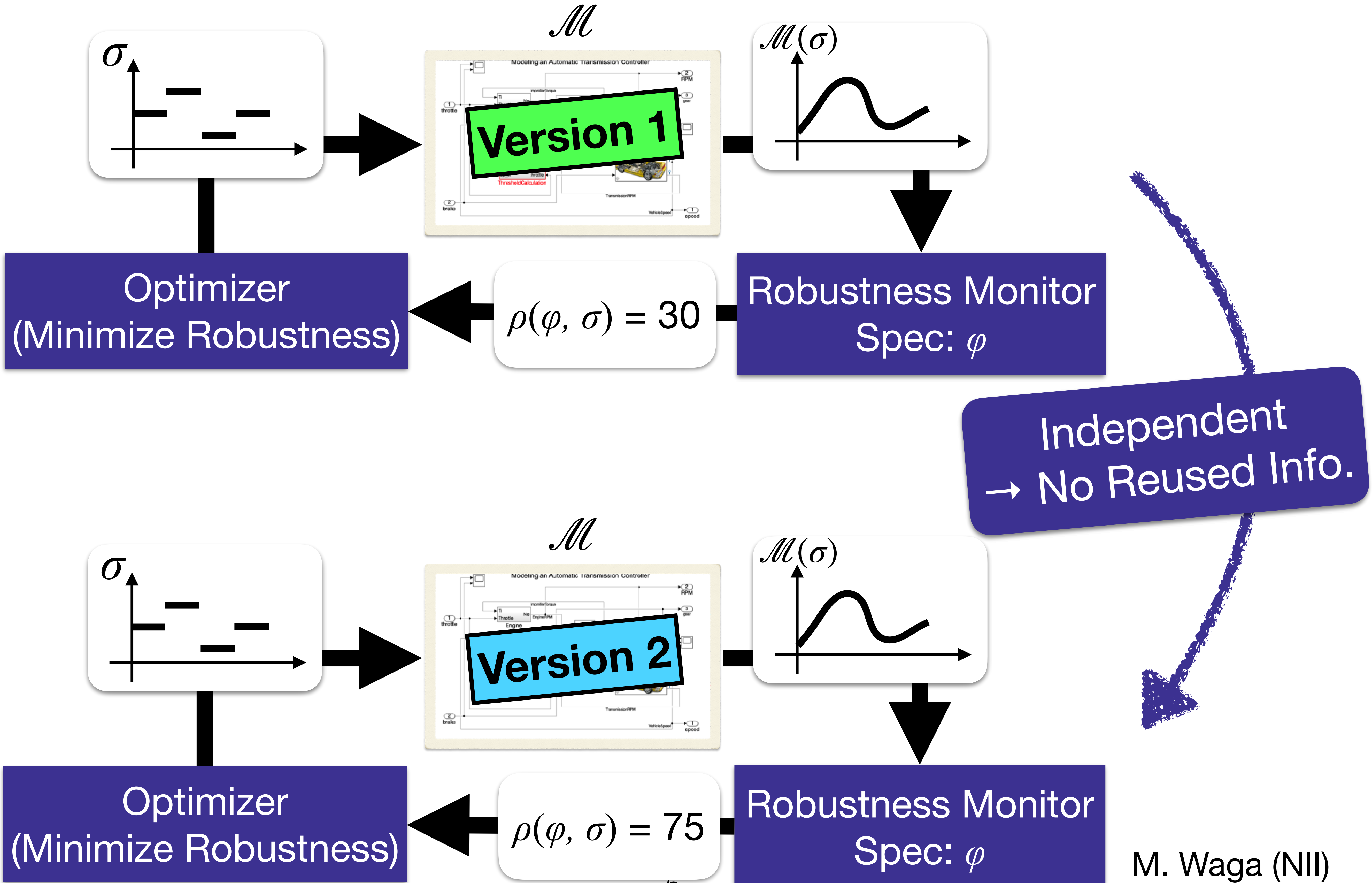


Falsification is Oneshot





Falsification is Oneshot



Q. How to reuse the previous exploration information

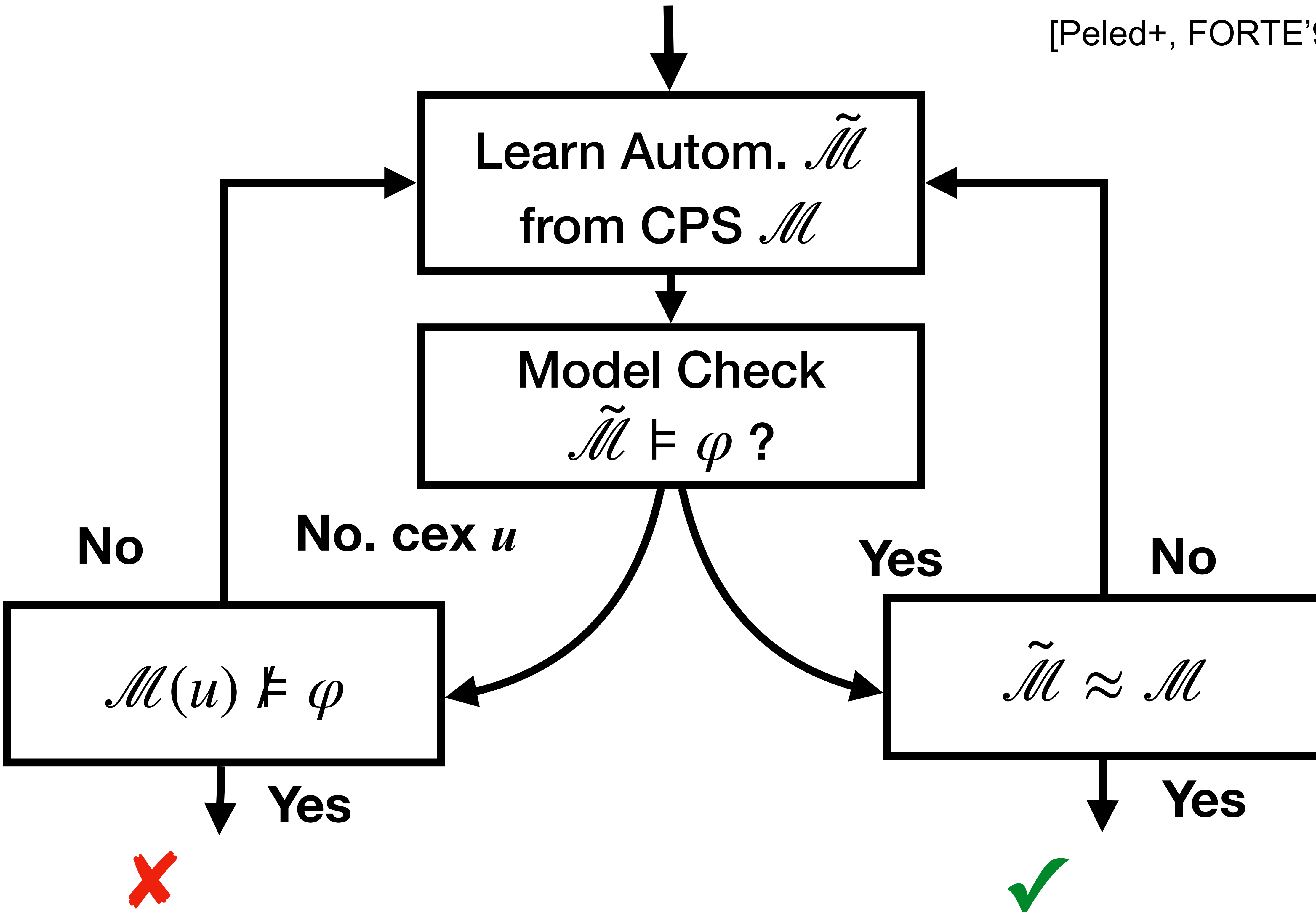
Q. How to reuse the previous exploration information

Our Approach:
Black-Box Checking (BBC)

M. Waga (NII)

Black-Box Checking (BBC)

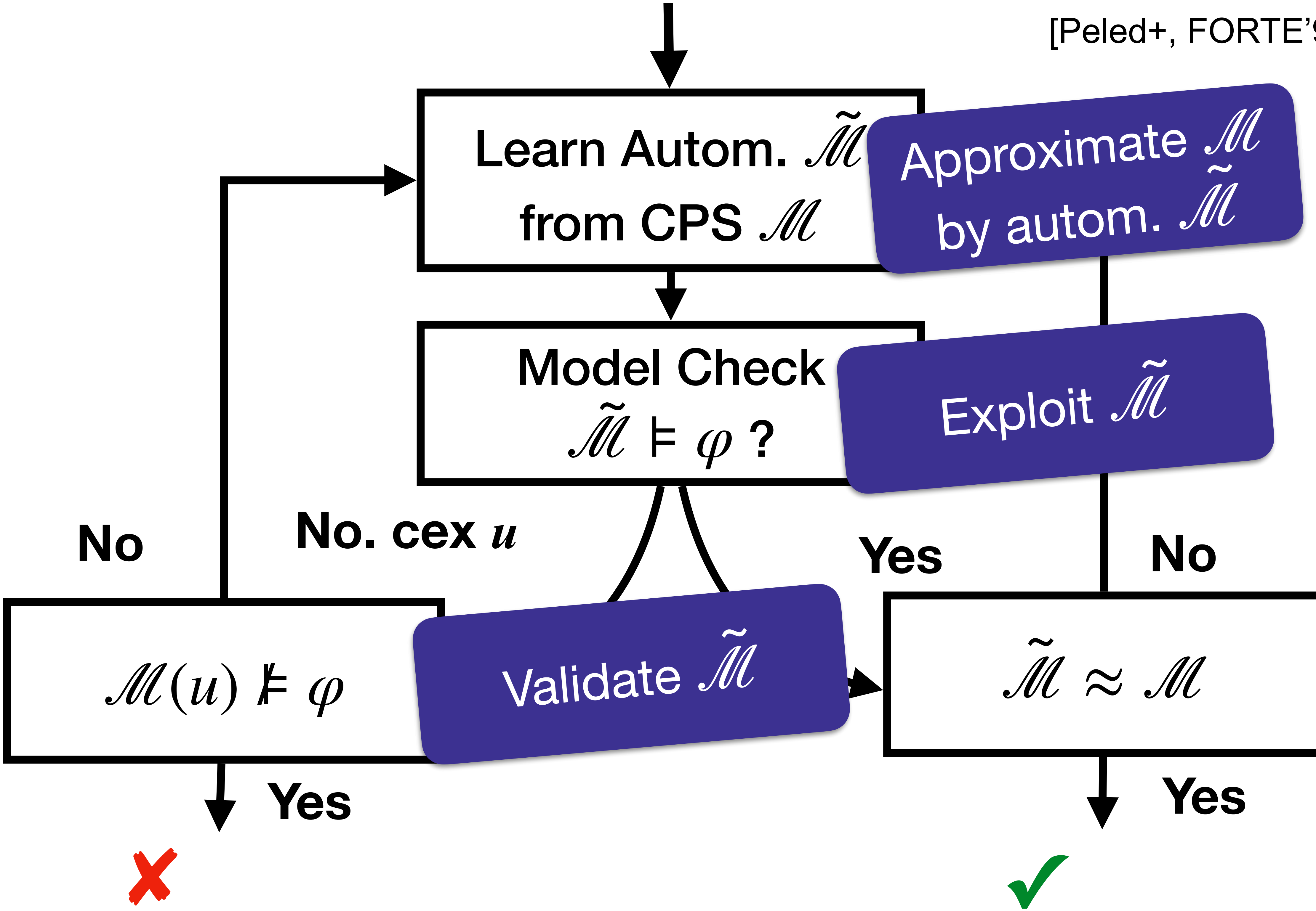
[Peled+, FORTE'99]



M. Waga (NII)

Black-Box Checking (BBC)

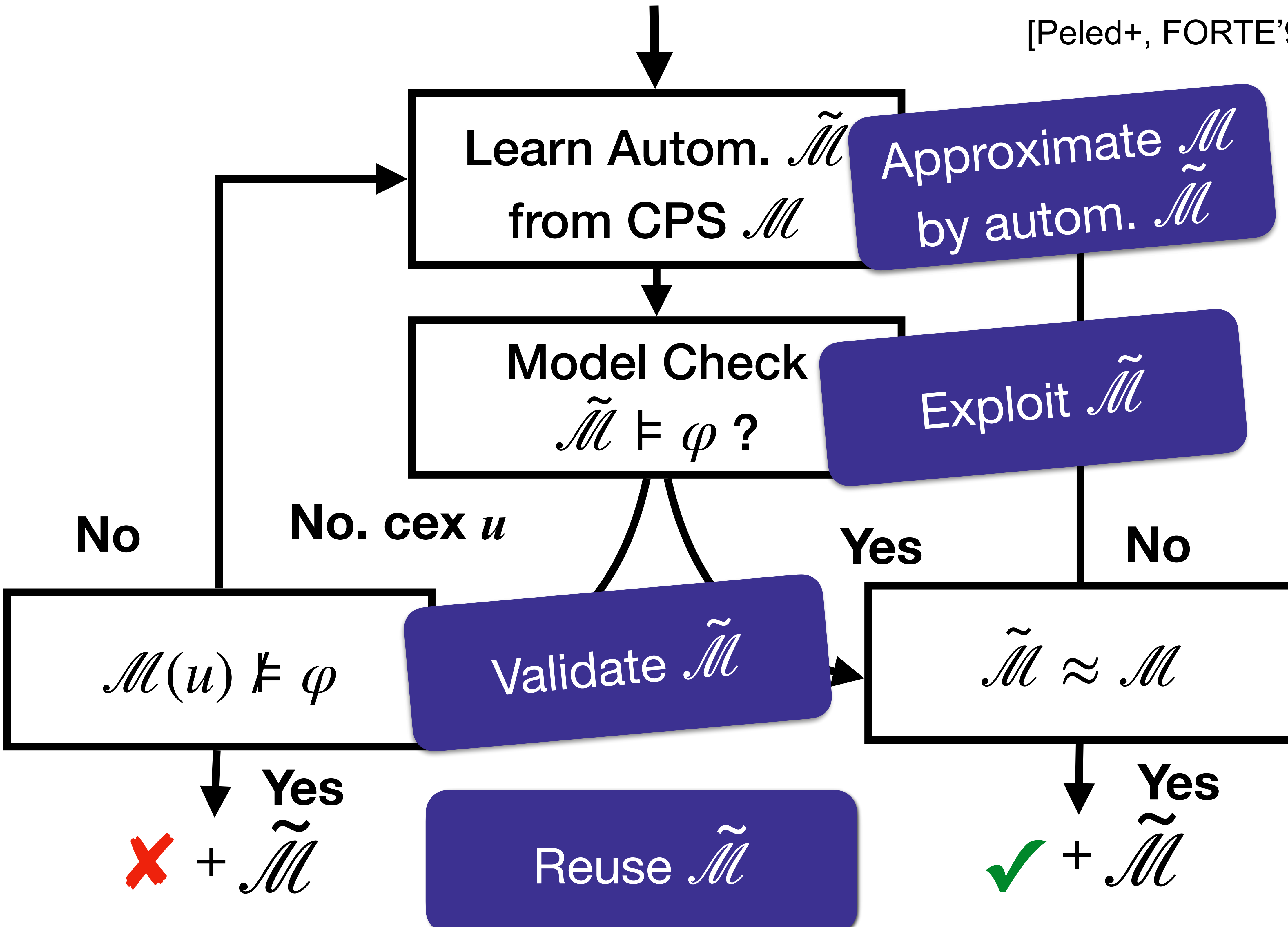
[Peled+, FORTE'99]



M. Waga (NII)

Black-Box Checking (BBC)

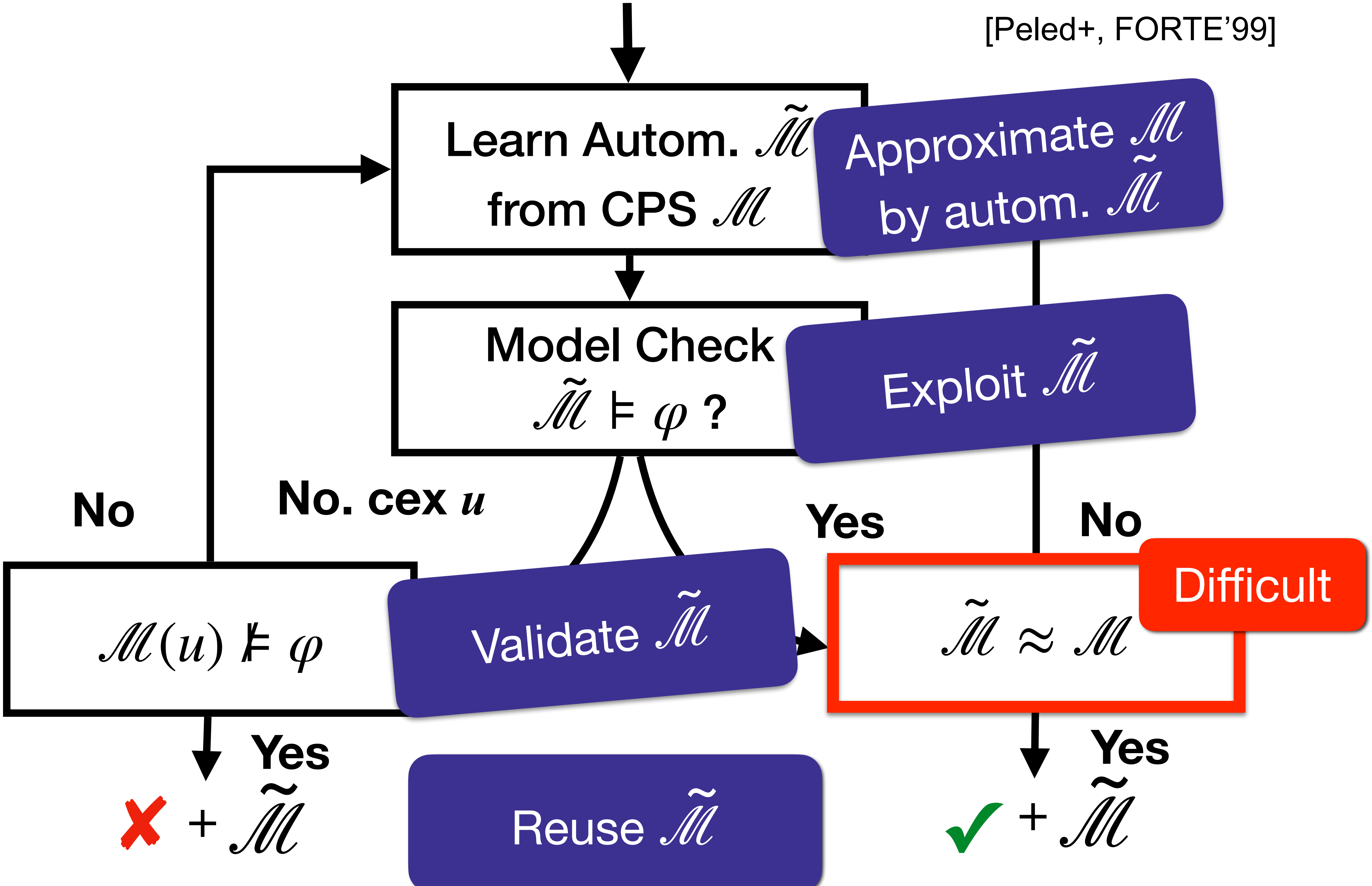
[Peled+, FORTE'99]



M. Waga (NII)

Black-Box Checking (BBC)

[Peled+, FORTE'99]



M. Waga (NII)

Equivalence Test: $\tilde{\mathscr{M}} \approx \mathscr{M}$

- **Theoretically**: Automata-based conformance test e.g., W-method [Chow, TSE'78], Wp-method [Fujiwara+, TSE'91]
  - ☹️ # of states is necessary for soundness
- **Practically**:
  - Random test
    - ☹️ Not good at "rare" cex

Equivalence Test: $\tilde{\mathscr{M}} \approx \mathscr{M}$

- **Theoretically**: Automata-based conformance test
  e.g., W-method [Chow, TSE'78], Wp-method [Fujiwara+, TSE'91]
  - ☹️ # of states is necessary for soundness

- **Practically**:
  - Random test
    - ☹️ Not good at "rare" cex

Q. How about hitting "rare" cex with robustness of STL?

M. Waga (NII)

Robustness-Guided Equivalence Test

[Contribution]

- Equivalence Test with Robustness + Optimization
  - Search for u s.t. $\tilde{\mathscr{M}}(u) \neq \mathscr{M}(u)$
  - Genetic algorithm or hill climbing
  - Fitness function: Robustness of STL
- $\tilde{\mathscr{M}}$ is not directly evaluated
  - Discrete behavior of $\tilde{\mathscr{M}}$ is not a problem

Detail is later

Contributions

- Presented robustness-guided equivalence test
  - Also introduced a new robustness suitable to model checking
- Used it for black-box checking of CPS
- Experimental evaluation: outperforms the baseline

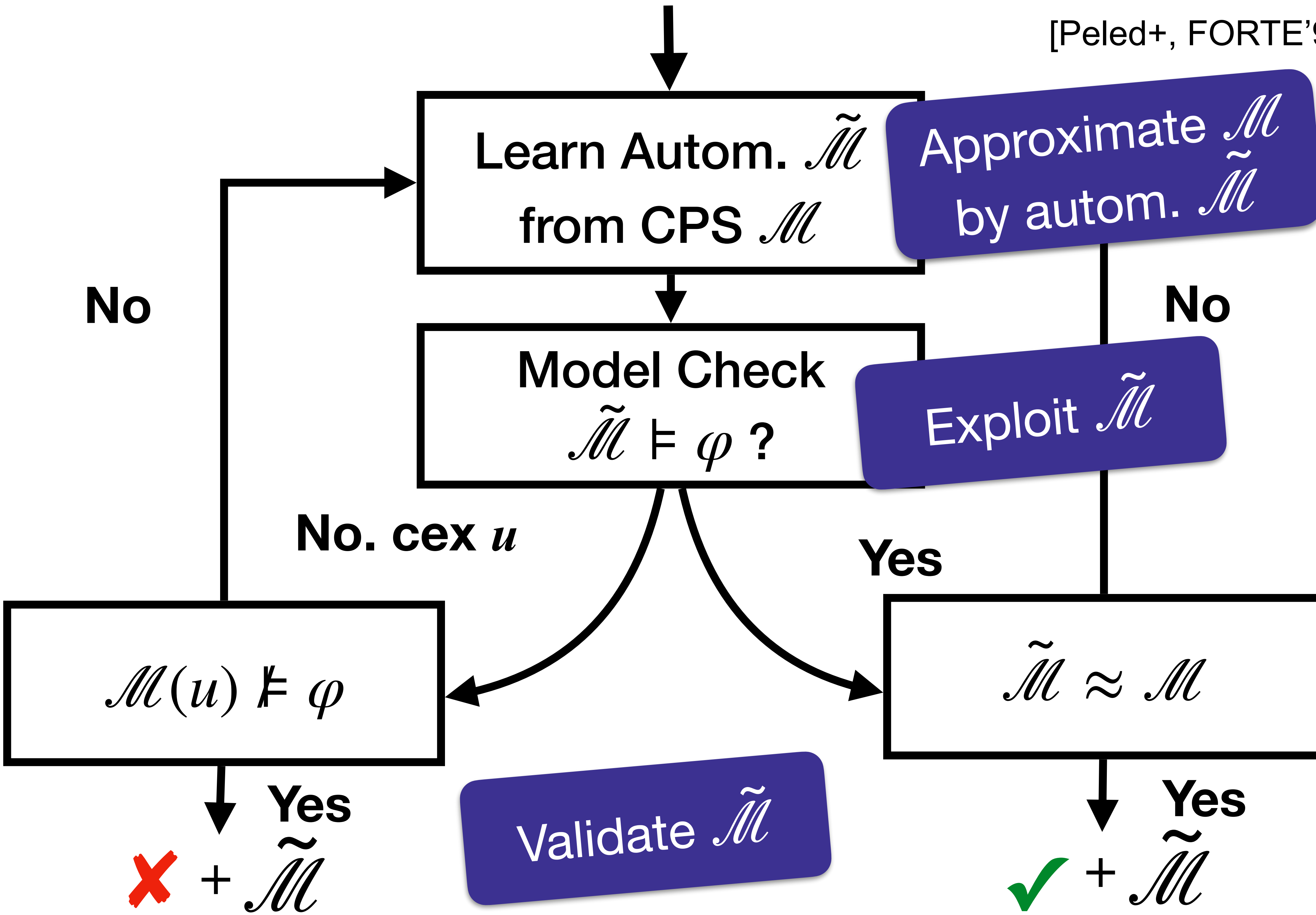
M. Waga (NII)

Outline

- Introduction
- Preliminary: Black-box checking
    - Automata learning, model checking, & equiv. test
- Main Contribution:
  Robustness-guided equivalence test
    - **idea:** Optimization + robustness
- Experiments

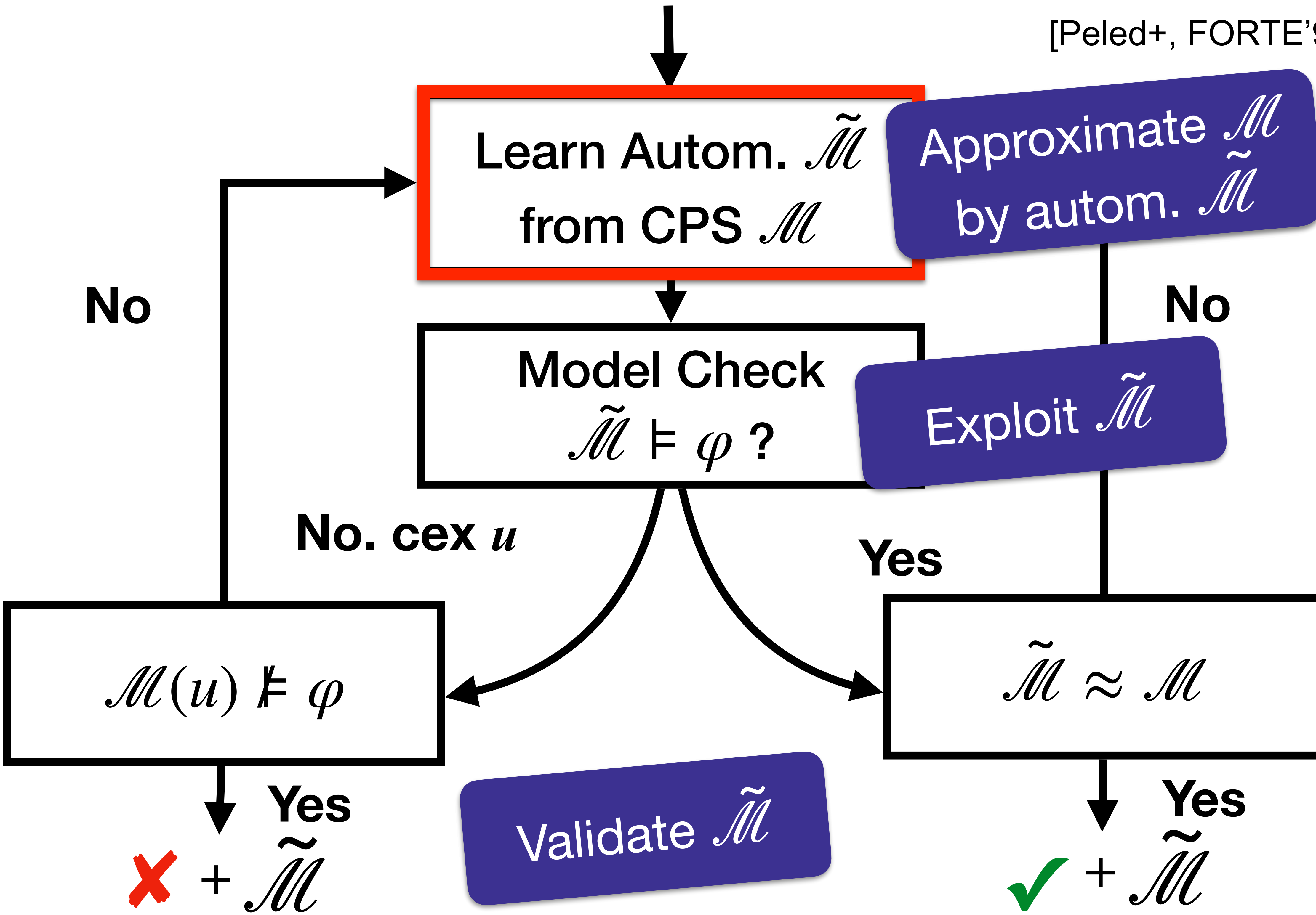
Black-Box Checking (BBC)

[Peled+, FORTE'99]



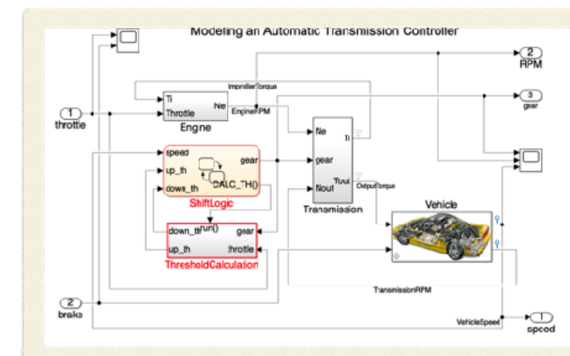
M. Waga (NII)

Black-Box Checking (BBC)



Automata Learning

1. Obtain suitable input/output pairs $\big(u, \mathscr{M}(u)\big)$ of $\mathscr{M}$
    - Input/output: finitely discretized

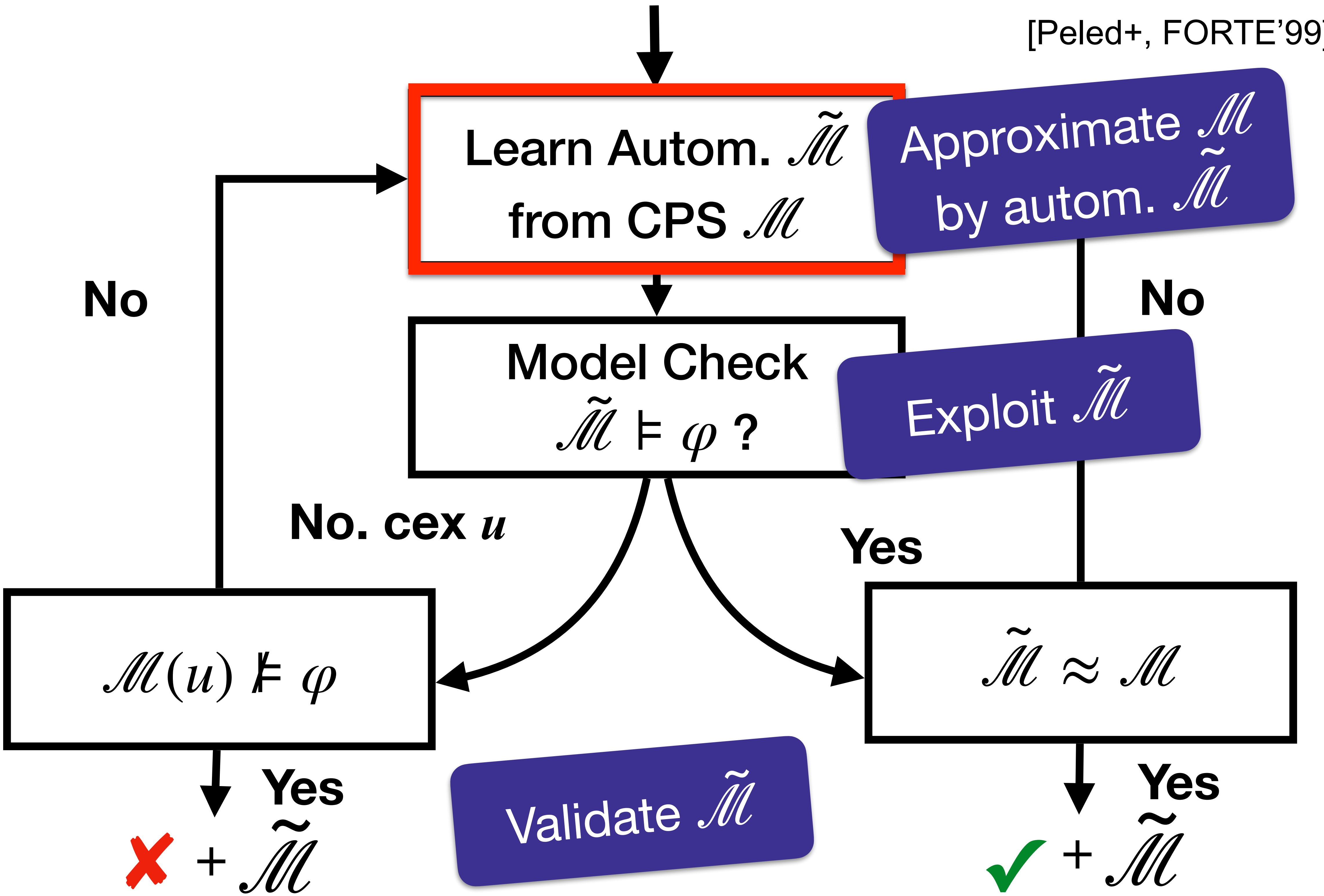


2. Generate a candidate Myhill-Nerode equiv. rel.

3. Construct the corresponding Mealy machine $\tilde{\mathscr{M}}$

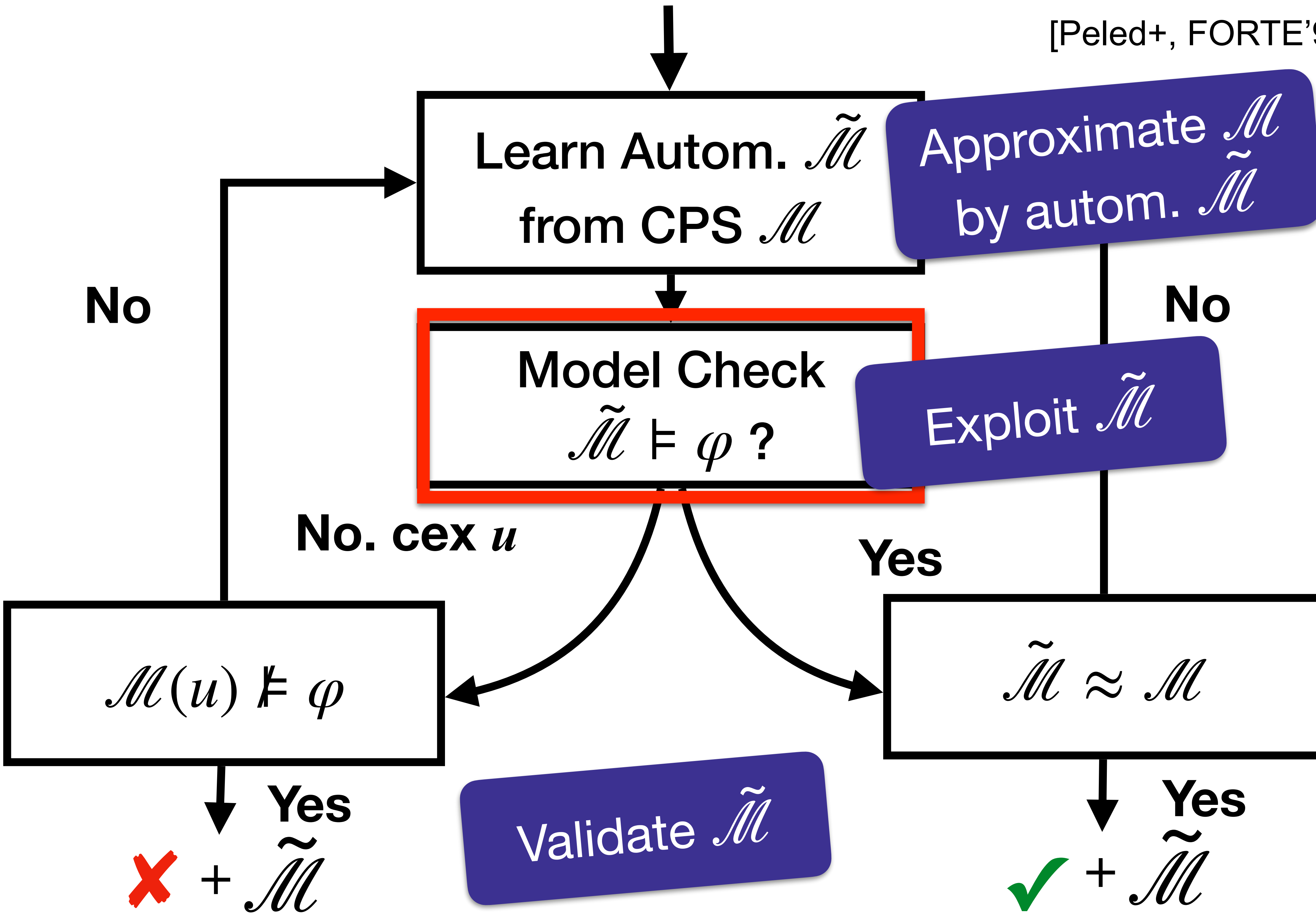
M. Waga (NII)

Black-Box Checking (BBC)



Black-Box Checking (BBC)

[Peled+, FORTE'99]



M. Waga (NII)

Model Check: $\tilde{\mathscr{M}} \models \varphi$?

Issue: φ is not in LTL but in STL

Our Solution: Encode to LTL model checking

STL

$\varphi ::= \underline{x \bowtie c} \mid \varphi \land \varphi \mid \cdots \mid \varphi \mathcal{U}_I \varphi$

Inequality

LTL

$\varphi ::= \underline{P} \mid \varphi \land \varphi \mid \cdots \mid \varphi \mathcal{U} \varphi$

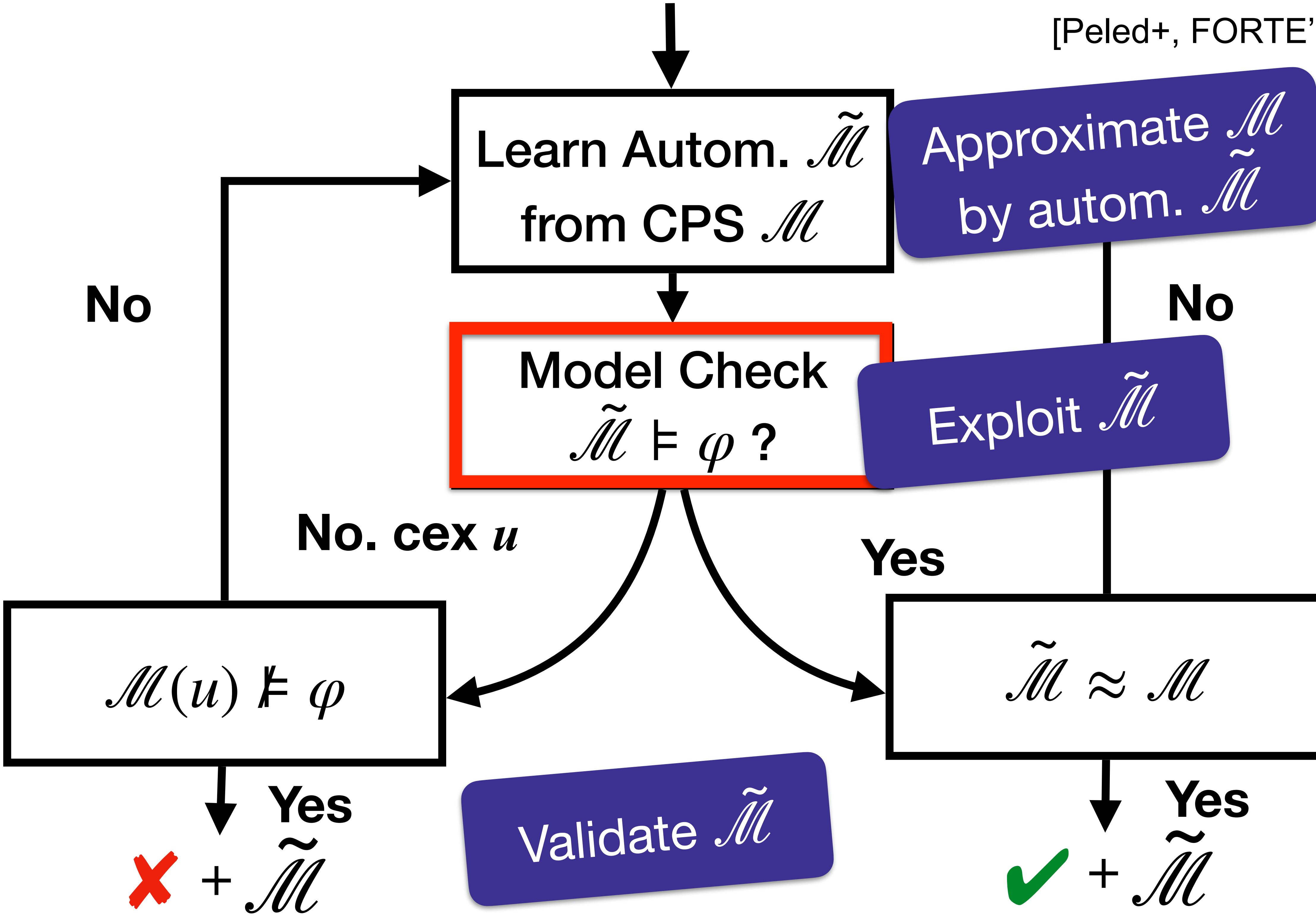
Proposition

Assumption:
signal is discrete-time with constant interval

M. Waga (NII)

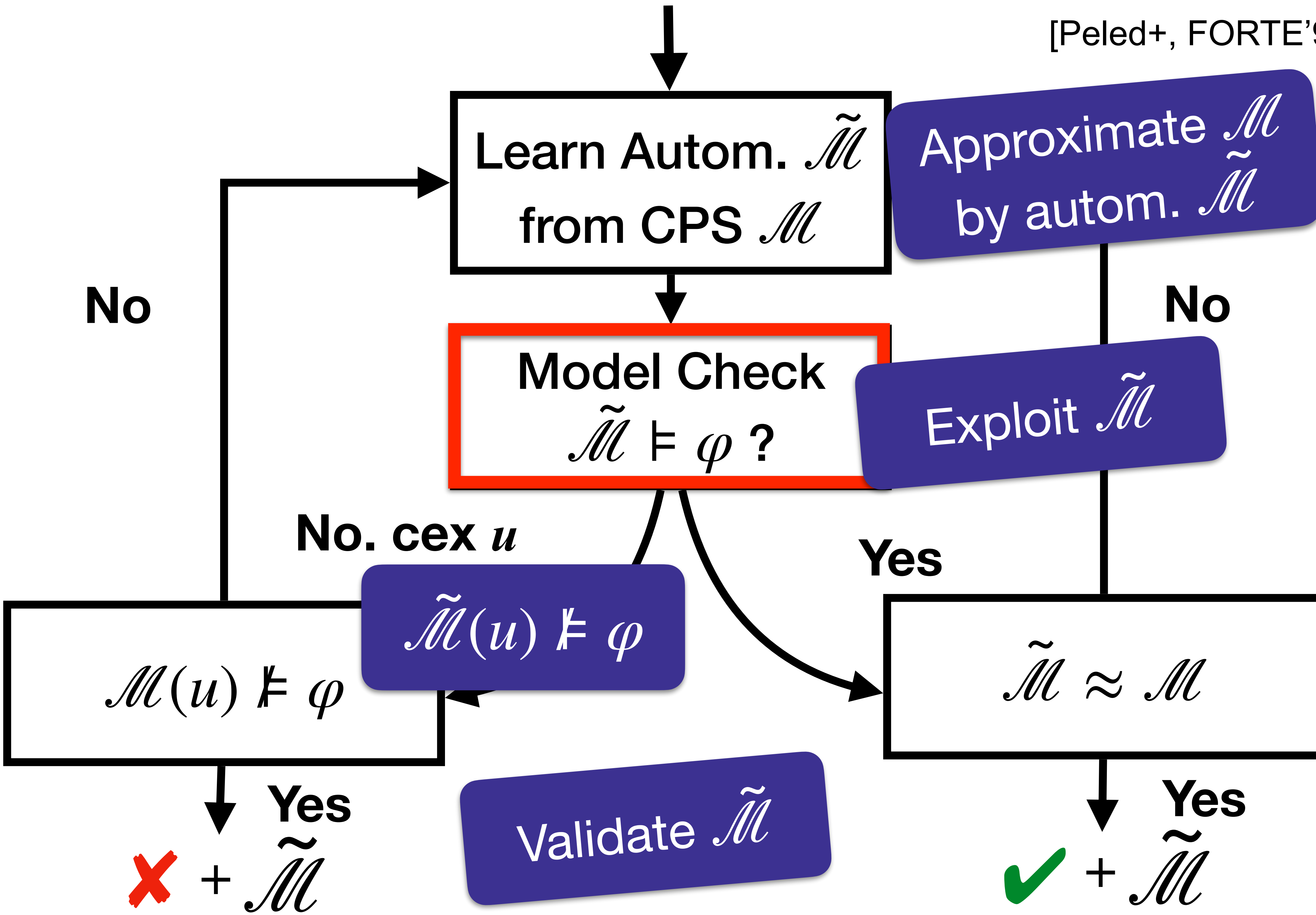
Black-Box Checking (BBC)

[Peled+, FORTE'99]



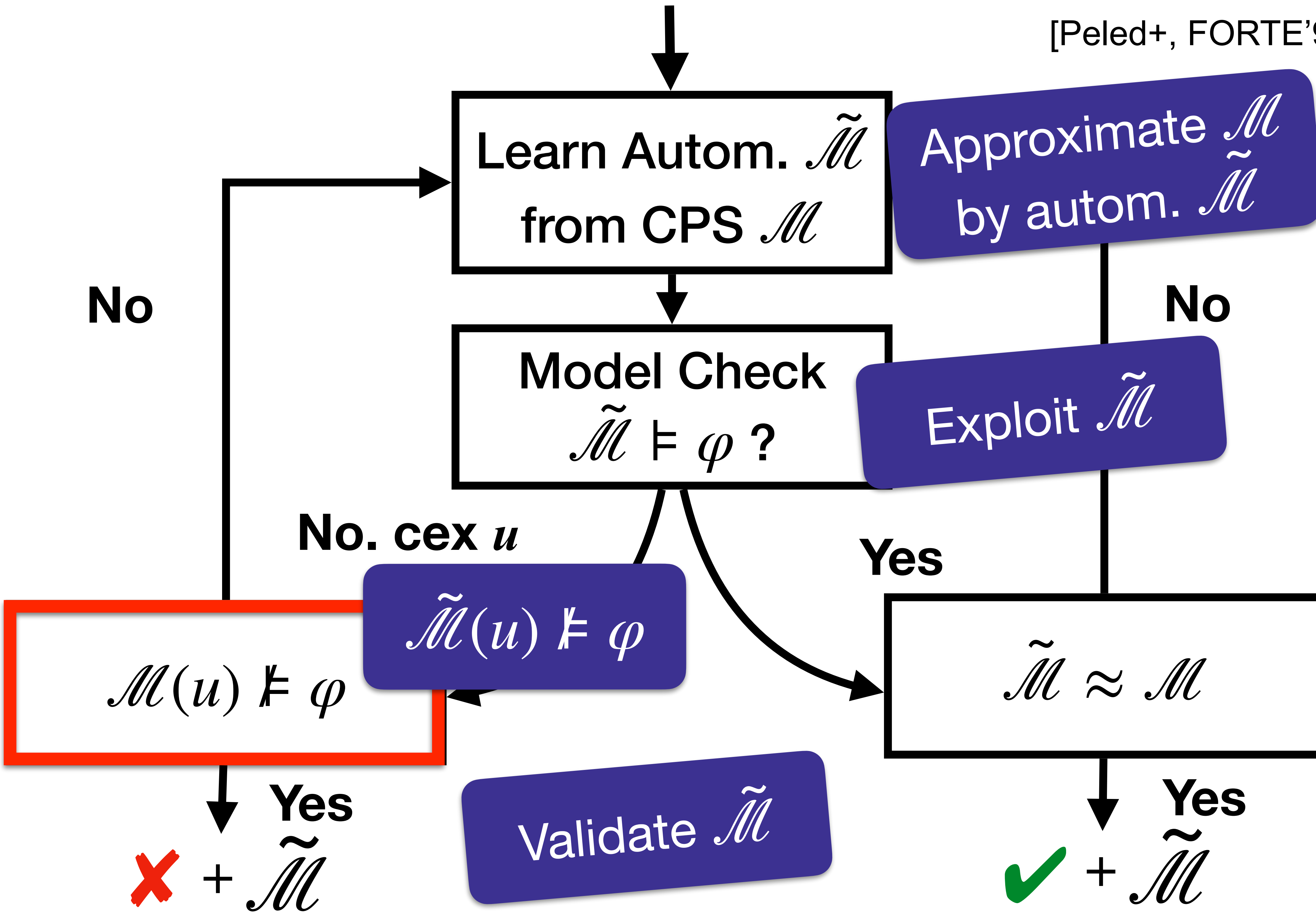
Black-Box Checking (BBC)

[Peled+, FORTE'99]



Black-Box Checking (BBC)

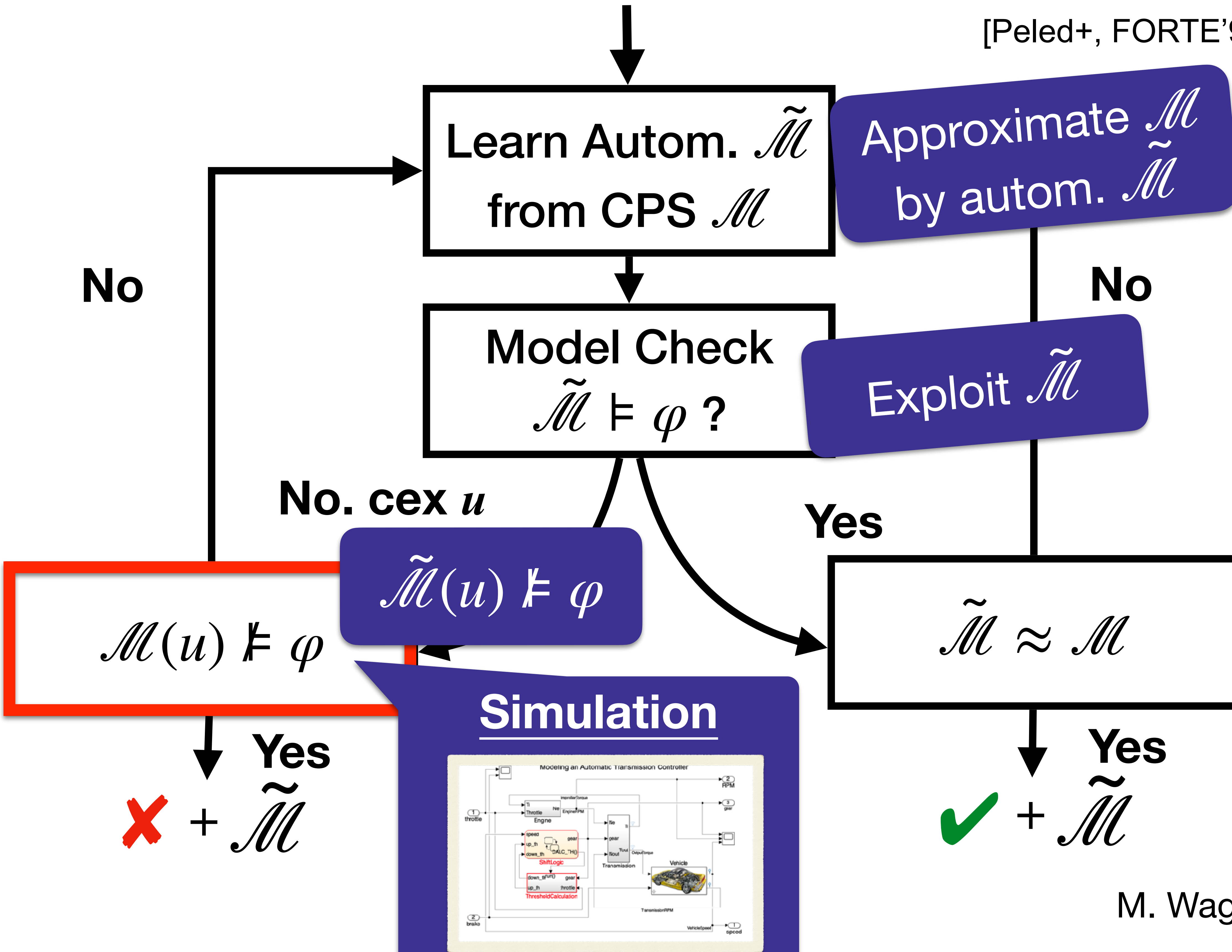
[Peled+, FORTE'99]



M. Waga (NII)

Black-Box Checking (BBC)

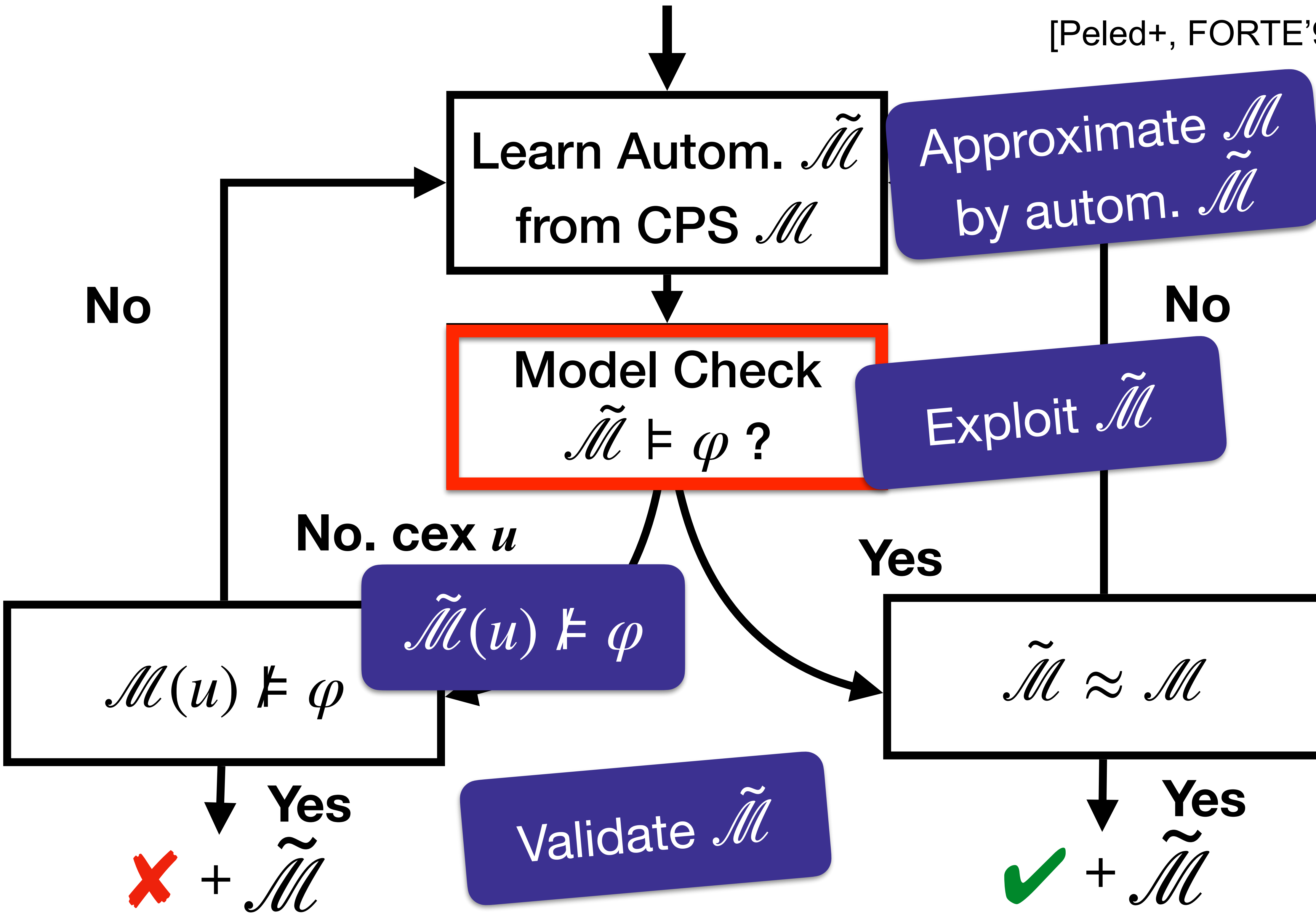
[Peled+, FORTE'99]



M. Waga (NII)

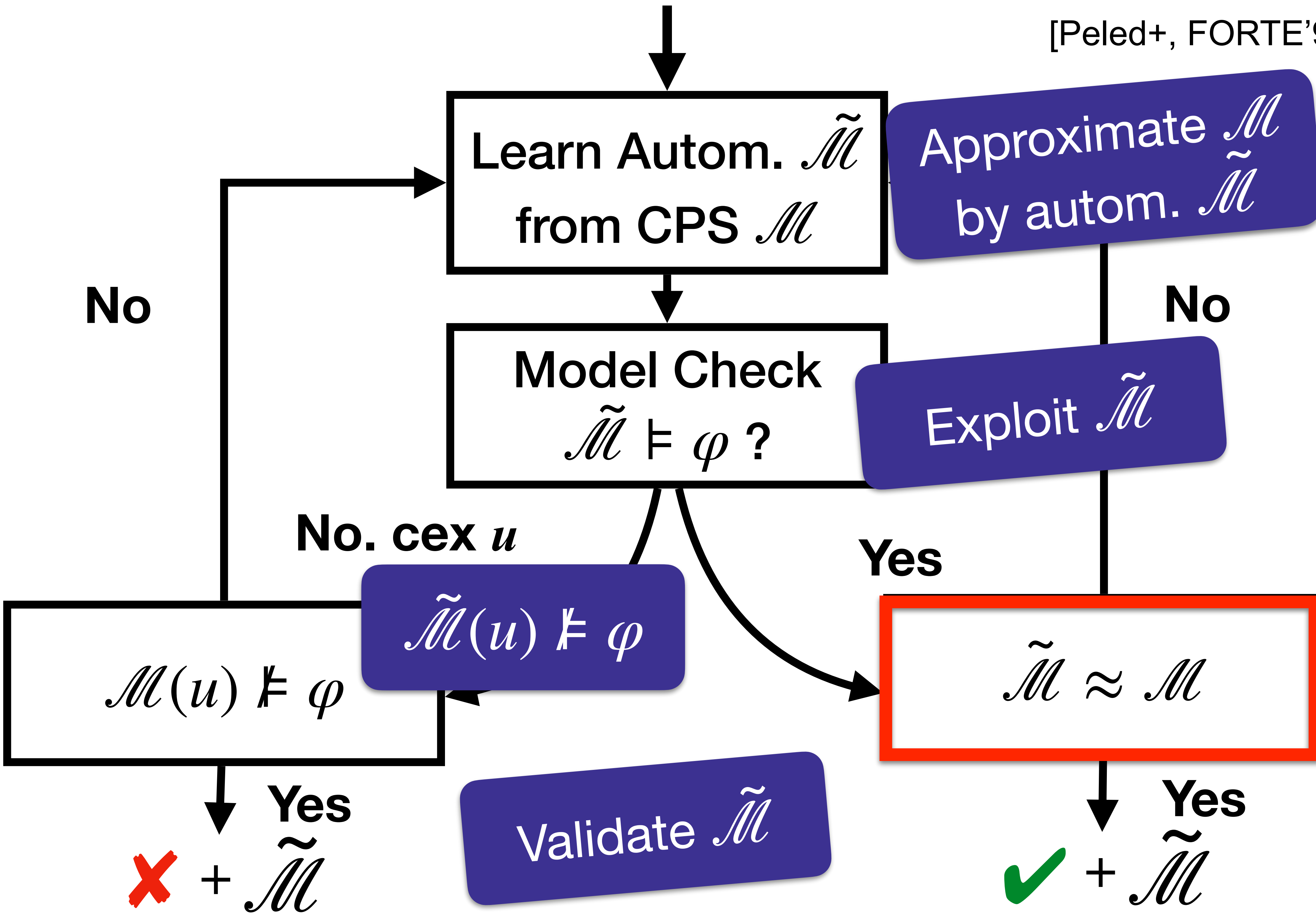
Black-Box Checking (BBC)

[Peled+, FORTE'99]

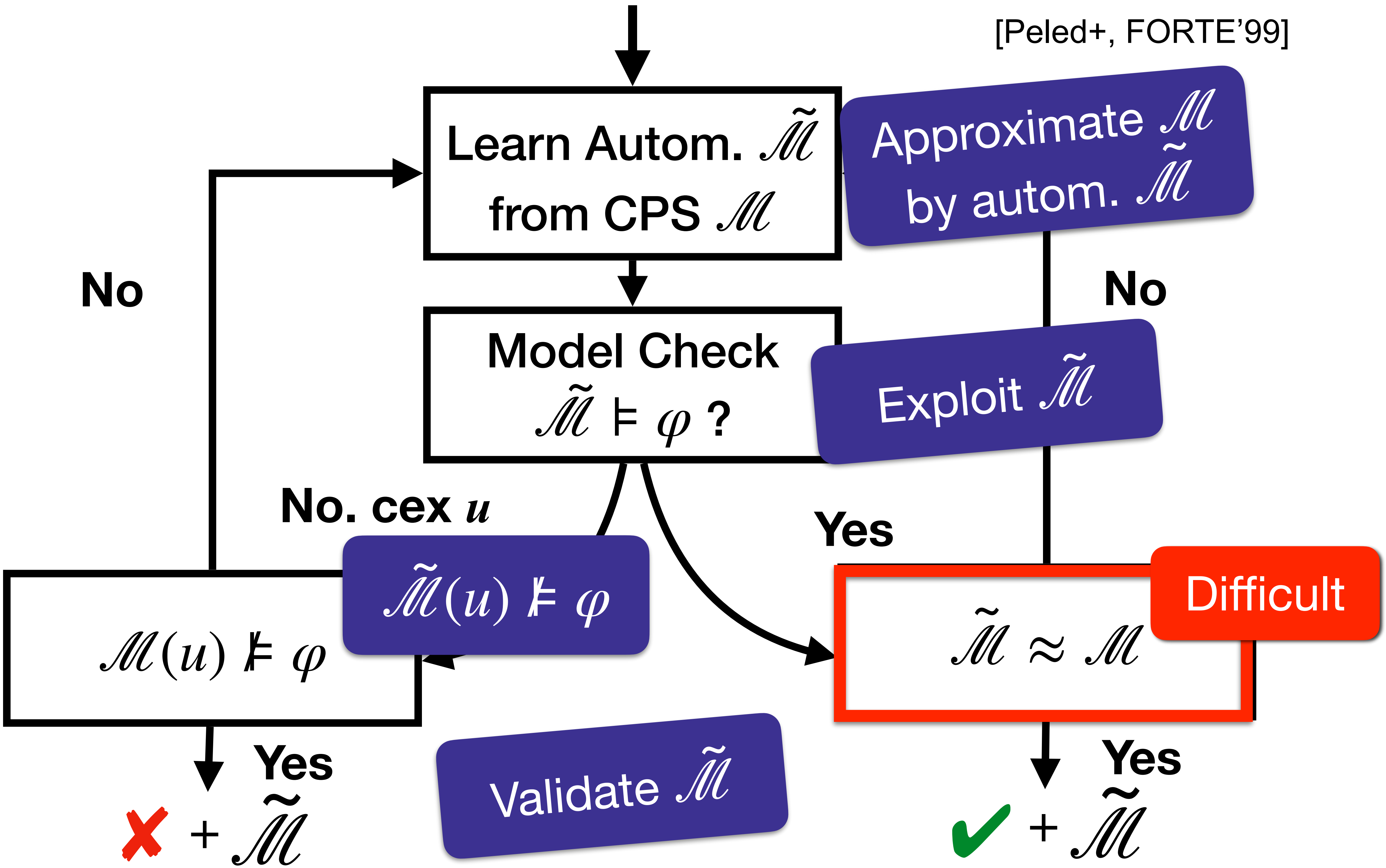


Black-Box Checking (BBC)

[Peled+, FORTE'99]



Black-Box Checking (BBC)



M. Waga (NII)

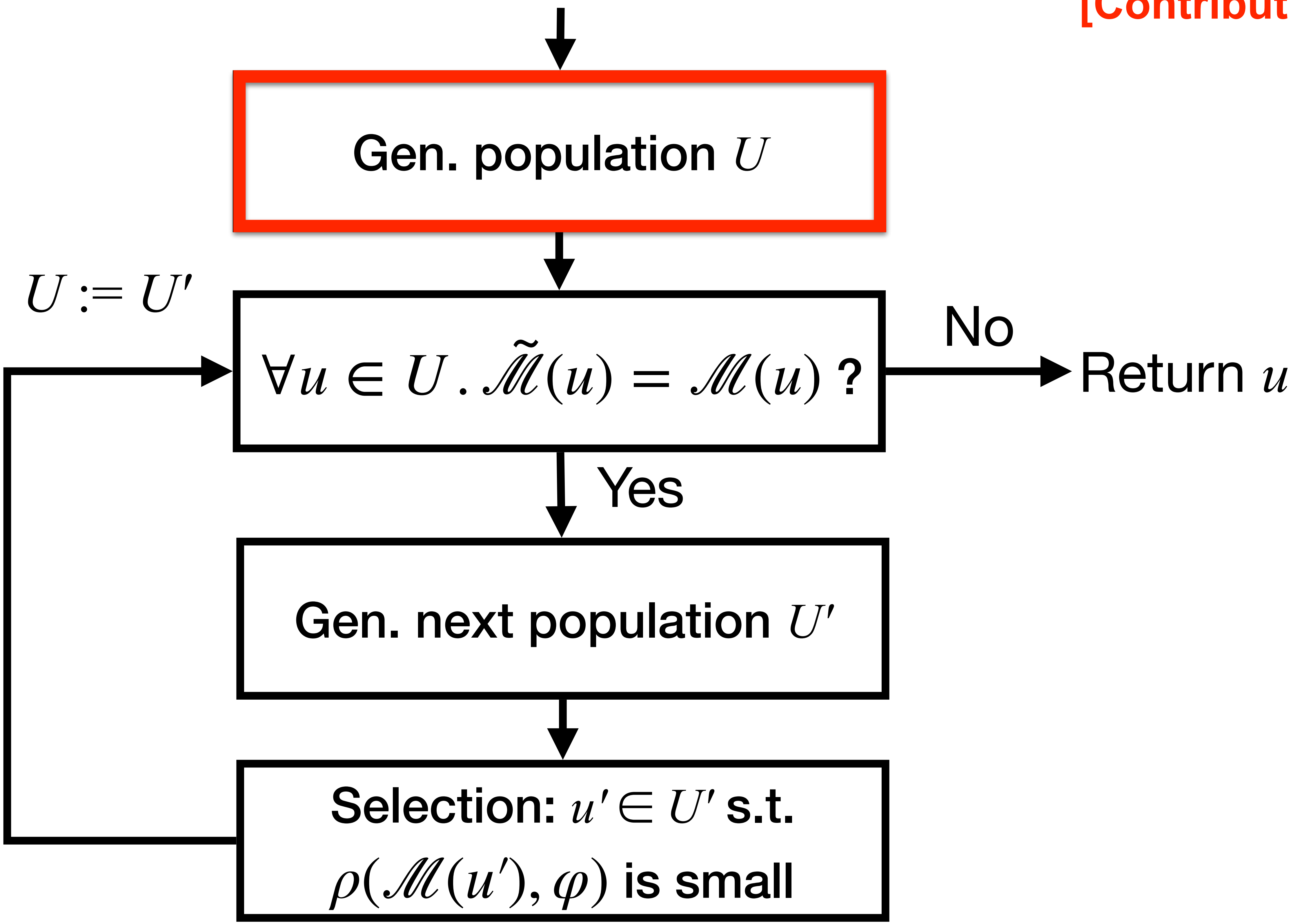
Outline

- Introduction
- Preliminary: Black-box checking
    - Automata learning, model checking, & equiv. test
- Main Contribution:
  Robustness-guided equivalence test
    - **idea:** Optimization + robustness
- Experiments

M. Waga (NII)

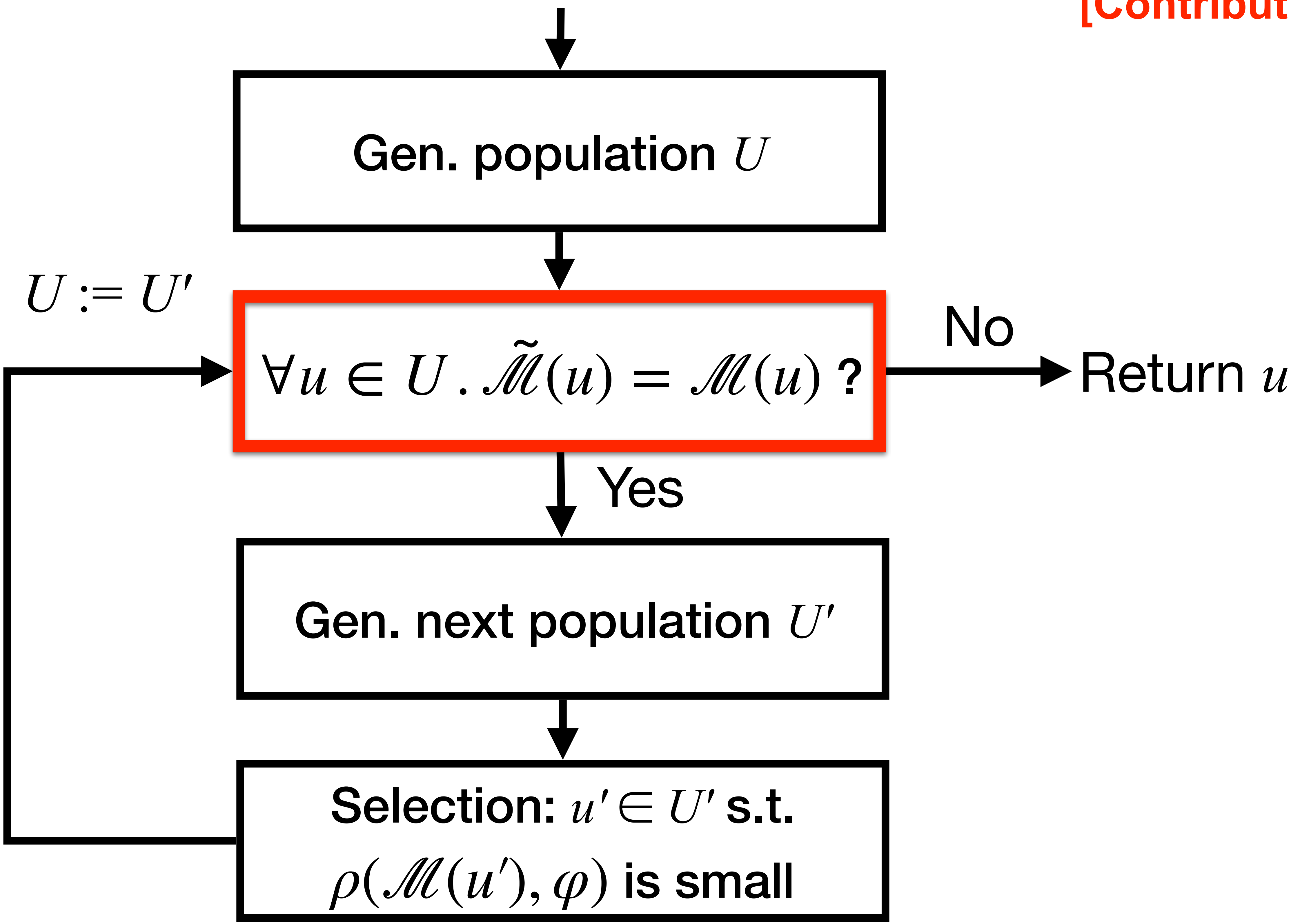
Robustness-Guided Equivalence Test

[Contribution]



Robustness-Guided Equivalence Test

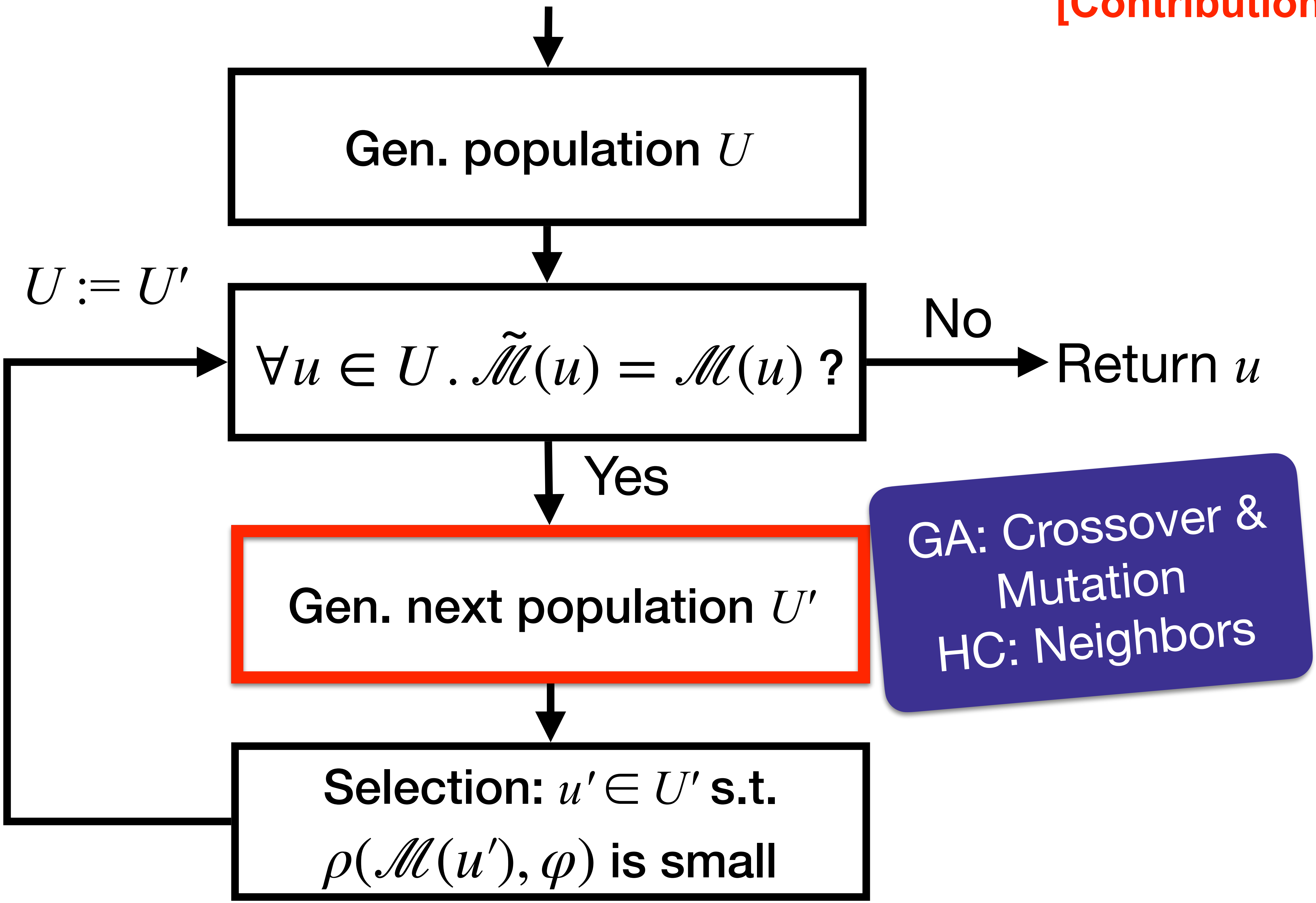
[Contribution]



M. Waga (NII)

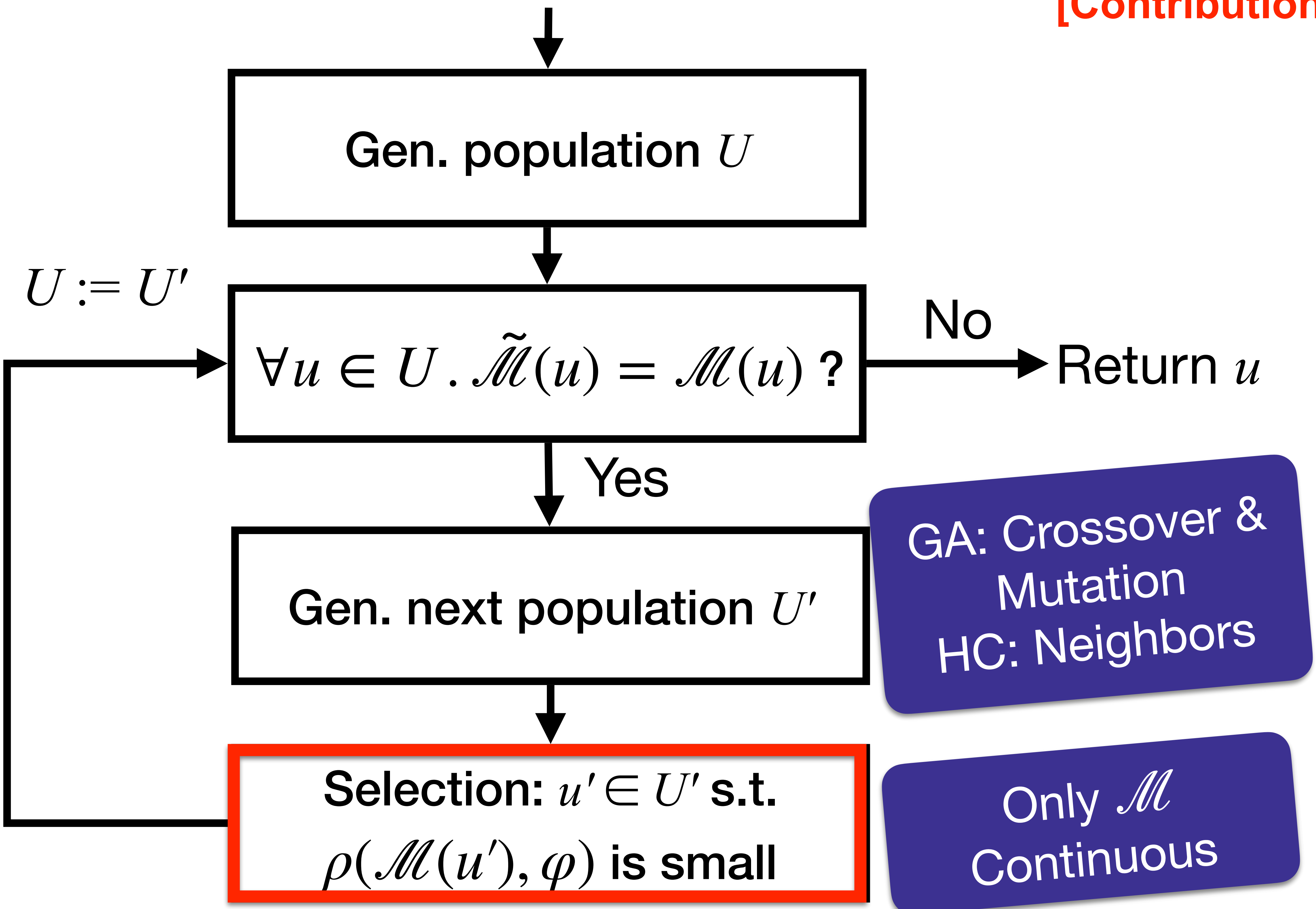
Robustness-Guided Equivalence Test

[Contribution]



Robustness-Guided Equivalence Test

[Contribution]



Why $\rho(\mathscr{M}(u'), \varphi)$ as Fitness?

Assumption: $\mathscr{M} \not\models \varphi$

i.e. $\exists u. \rho(\mathscr{M}(u), \varphi) \leq 0$

Fact: $\tilde{\mathscr{M}} \models \varphi$ i.e.

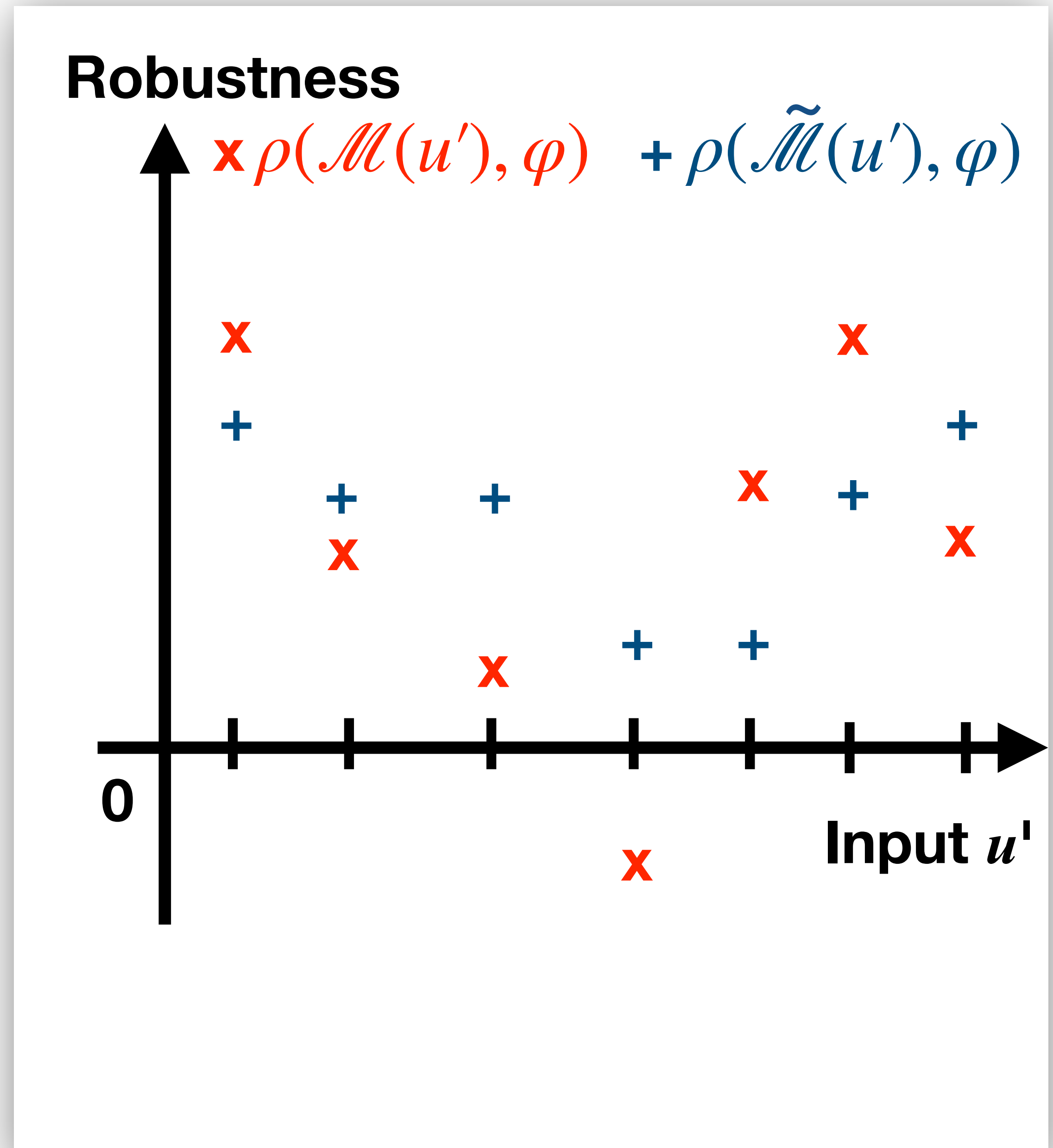
$\forall u \,.\, \rho(\tilde{\mathscr{M}}(u), \varphi) \geq 0$

by model checking

Heuristic: Find u' s.t.

$\tilde{\mathscr{M}}(u') \neq \mathscr{M}(u')$ in

$\rho(\mathscr{M}(u'), \varphi)$ is small



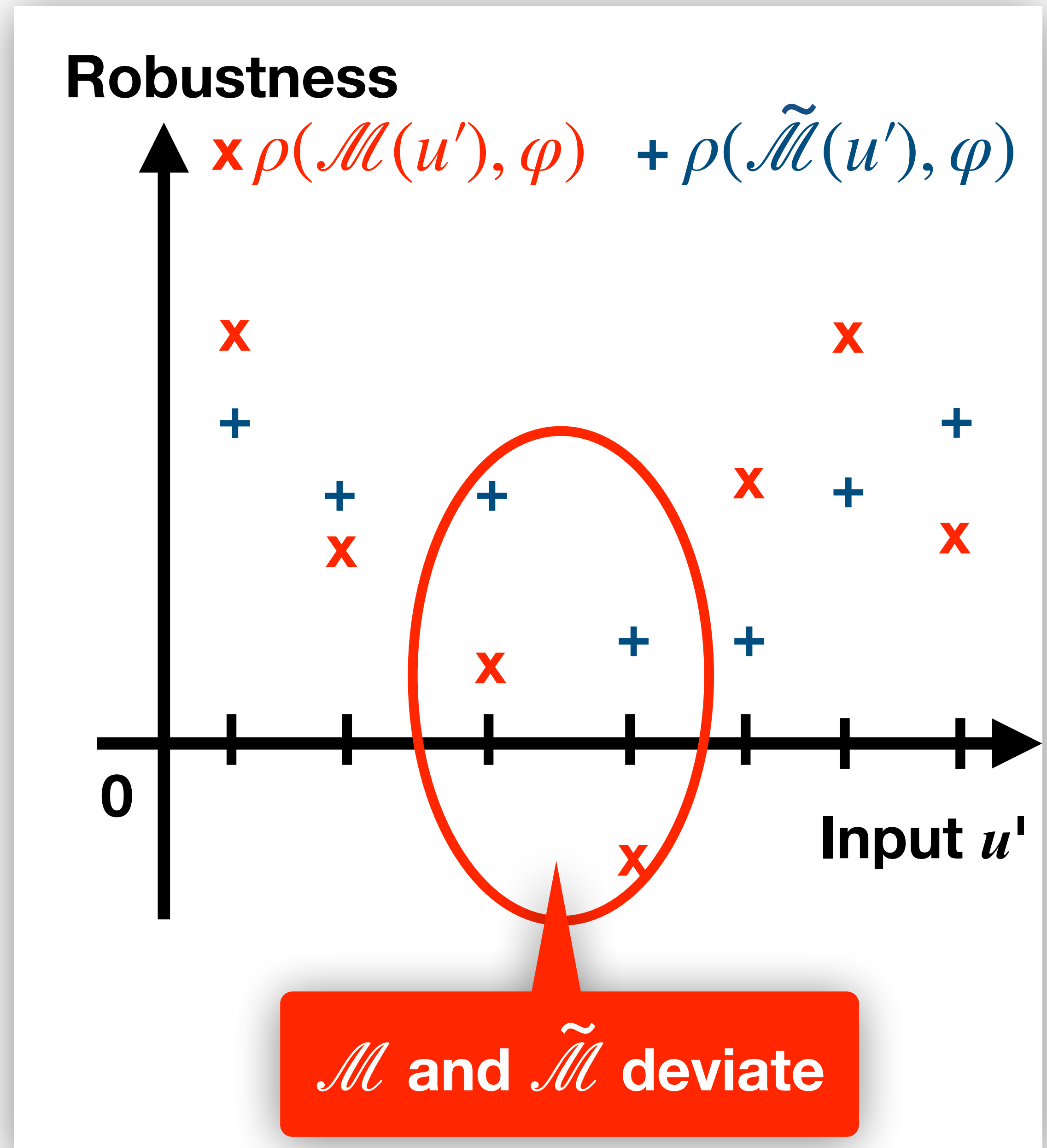
M. Waga (NII)

Why $\rho(\mathscr{M}(u'), \varphi)$ as Fitness?

Assumption: $\mathscr{M} \not\models \varphi$
i.e. $\exists u. \rho(\mathscr{M}(u), \varphi) \leq 0$

Fact: $\tilde{\mathscr{M}} \models \varphi$ i.e.
$\forall u \,.\, \rho(\tilde{\mathscr{M}}(u), \varphi) \geq 0$
by model checking

Heuristic: Find u' s.t.
$\tilde{\mathscr{M}}(u') \neq \mathscr{M}(u')$ in
$\rho(\mathscr{M}(u'), \varphi)$ is small

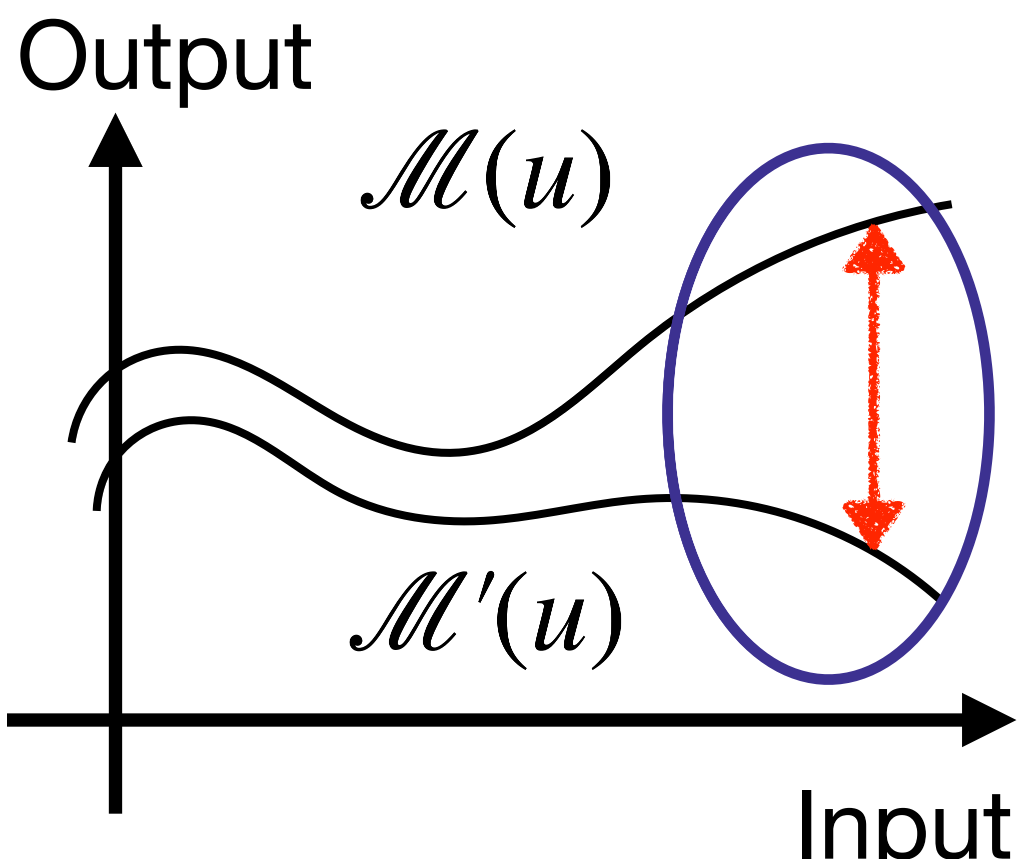


M. Waga (NII)

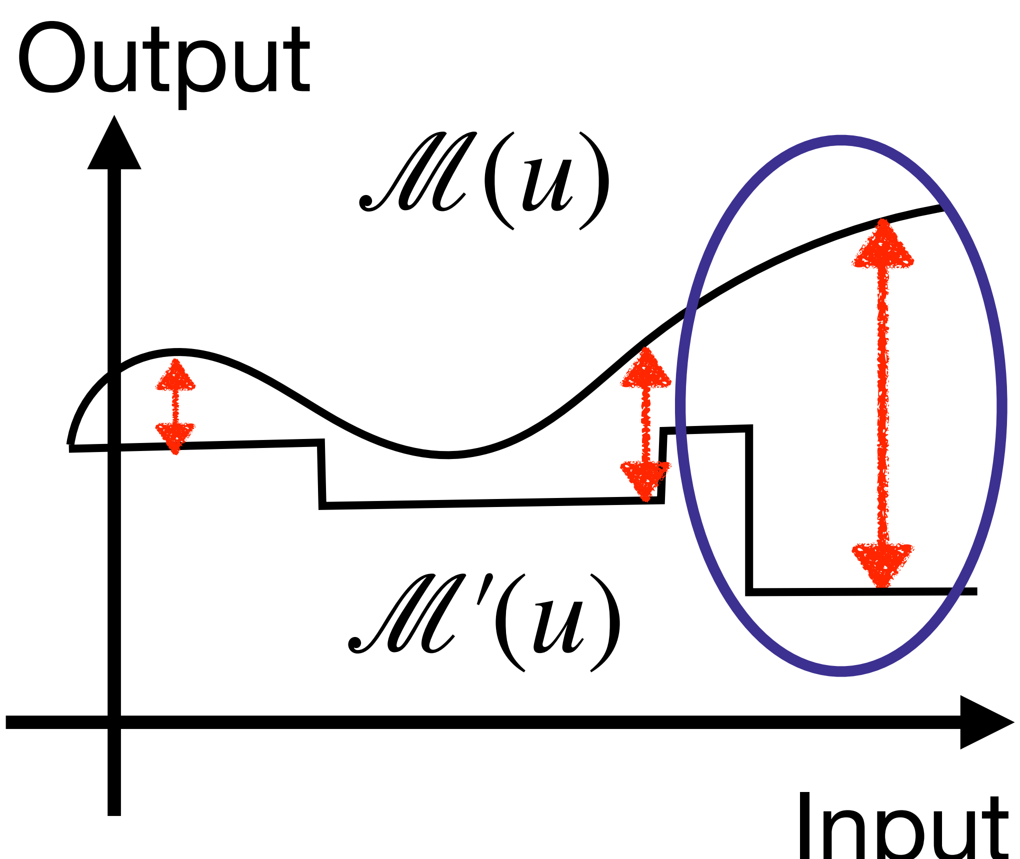
How About Direct Comparison?

- Direct comparison is also available e.g. [Abbas+, MEMOCODE'14]
- **Potential Issue**: more local maxima

Cont. vs Cont.



Cont. vs Disc.



Outline

- Introduction
- Preliminary: Black-box checking
    - Automata learning, model checking, & equiv. test
- Main Contribution:
  Robustness-guided equivalence test
    - **idea:** Optimization + robustness
- Experiments

Research Questions

RQ1: Does RobBBC falsify as many spec. as baseline?

RQ2: Which BBC is the best: Random, GA, and HC?

RQ3: Scalable wrt # of spec?

M. Waga (NII)

Experiment Setting: Algorithms

- Robustness-guided BBC with
  - Genetic Algorithm (RobBBC(GA))
  - Hill Climbing (RobBBC(HC))

Ours

- BBC (equivalence test by random search)
- Breach: one of the SoTA falsification tools
- Pure random search
  - show the difficulty of the specifications

Baseline

Experiment Setting: Benchmark

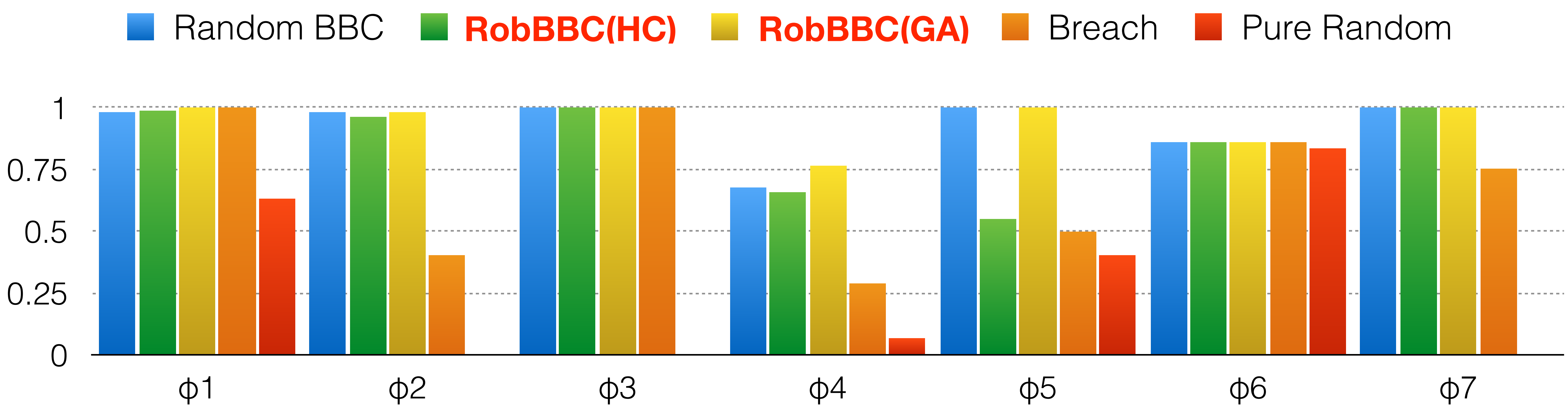
Model: Automatic Transmission [Hoxha+,ARCH'14-15]

Spec: Set of STL formulas

- Similar specifications with different thresholds
- Motivation: hard to decide the threshold

	STL template	parameter valuations	size
φ_1	$\Box(v < p)$	$p \in \{100, 102.5, 105, 107.5, 110, 112.5, 115, 117.5, 120\}$	9
φ_2	$\Box(g = 3 \Rightarrow v > p)$	$p \in \{20, 22.5, 25, 27.5, 30\}$	5
φ_3	$\Diamond_{[p_1,p_2]}(v < p_3 \lor v > p_4)$	$(p_1, p_2) \in \{(5, 20), (5, 25), (15, 30), (10, 30)\}, (p_3, p_4) \in \{(50, 60), (53, 57)\}$	8
φ_4	$\Box_{[0,26]}(v < p_1) \lor \Box_{[28,28]}(v > p_2)$	$p_1 \in \{90, 100, 110\}, p_2 \in \{55, 65, 75\}$	9
φ_5	$\Box(\omega < p_1 \lor \mathcal{X}(\omega > p_2))$	$p_1 \in \{4000, 4700\}, p_2 \in \{600, 1000, 1500\}$	6
φ_6	$\Box(v < p_1 \Rightarrow \Box_{[0,p_2]}(v < p_3))$	$p_1 \in \{30, 40, 50\}, p_2 \in \{6, 8, 10\}, p_3 \in \{60, 70, 80, 90\}$	36
φ_7	$\Box(((g \neq p_1) \land \mathcal{X}(g = p_1)) \Rightarrow \Box_{[0,p_2]}(g = p_1))$	$p_1 \in \{1, 2, 3, 4\}, p_2 \in \{1, 2, 3\}$	12

$\frac{\text{\# of Falsified Properties}}{\text{\# of Properties}}$

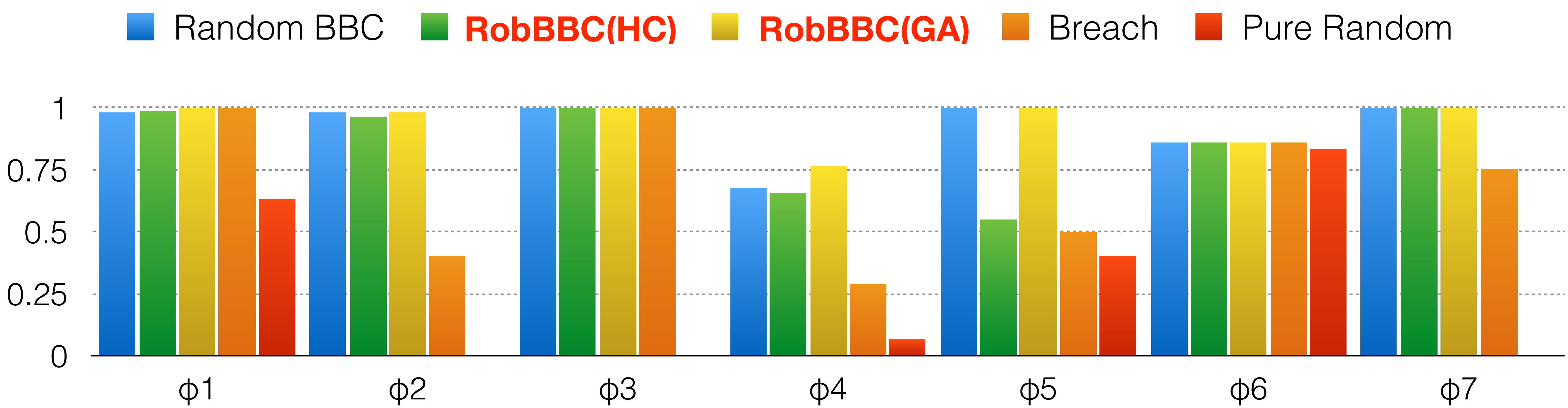


- RobBBC(HC) and Breach: Often fails
- Random BBC: sometimes fails
- RobBBC(GA): the largest for all specifications

M. Waga (NII)

of Falsified Properties / # of Properties

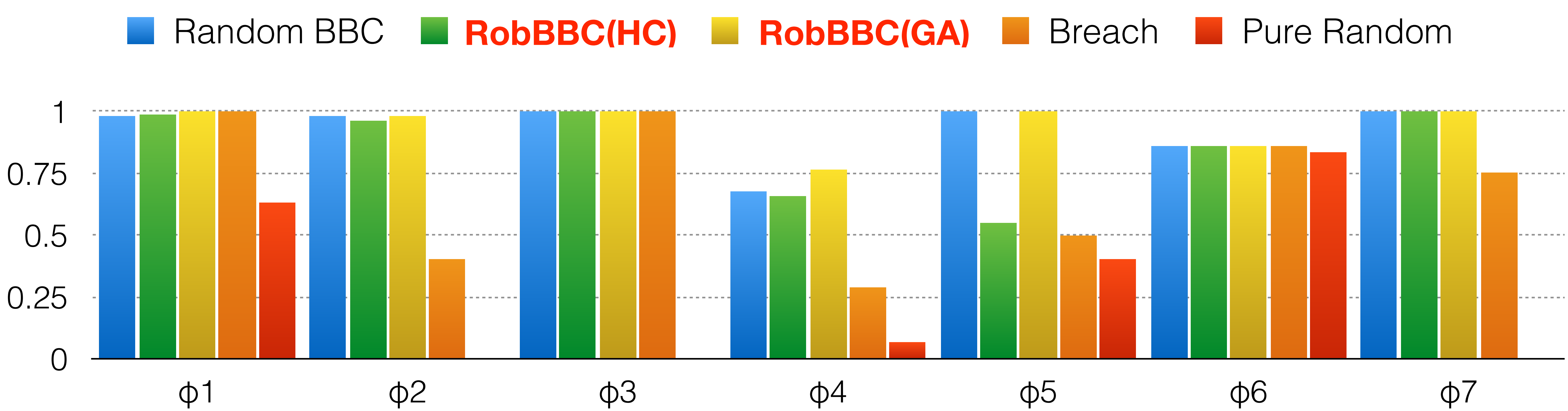
$$\frac{\text{\# of Falsified Properties}}{\text{\# of Properties}}$$



- RobBBC(HC) and Breach: Often fails
- Random BBC: sometimes fails
- RobBBC(GA): the largest for all specifications

RQ1: Yes, RobBBC can falsify as many as Breach

$\frac{\text{\# of Falsified Properties}}{\text{\# of Properties}}$

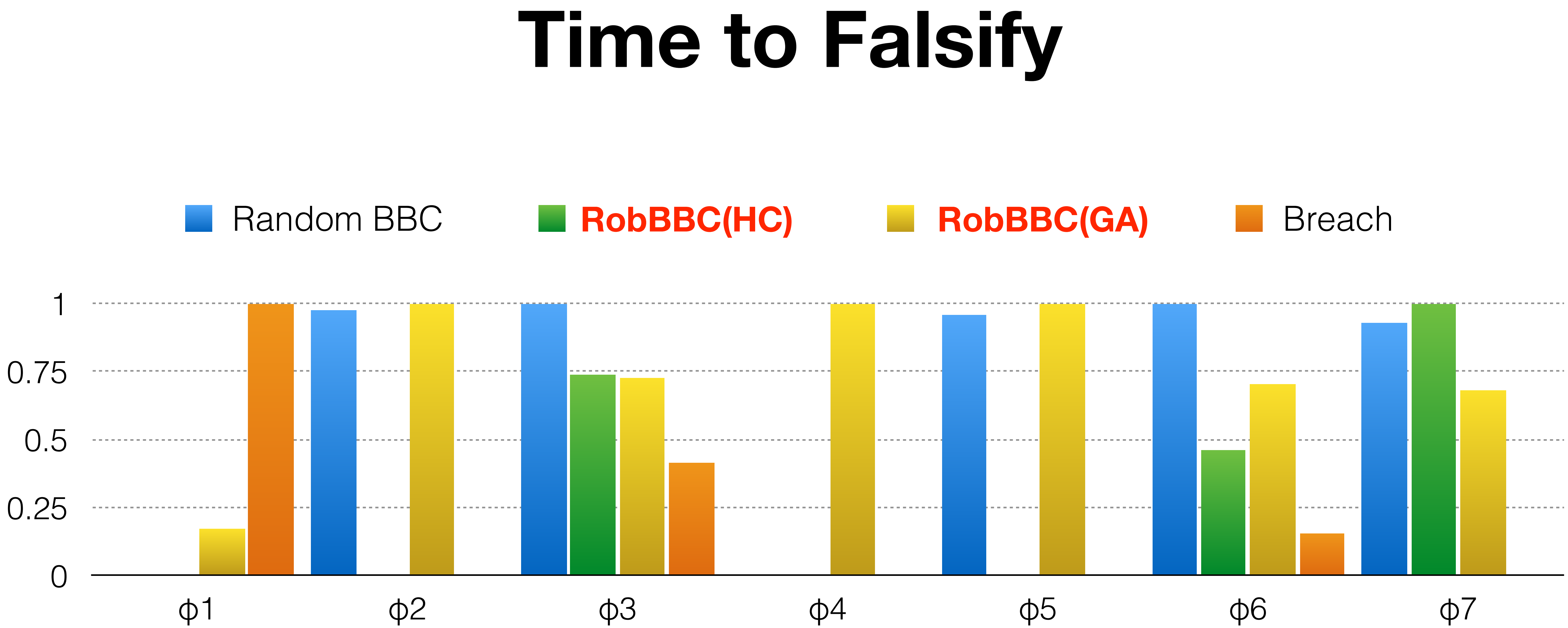


- RobBBC(HC) and Breach: Often fails
- Random BBC: sometimes fails
- RobBBC(GA): the largest for all specifications

RQ1: Yes, RobBBC can falsify as many as Breach

RQ2: GA falsified more than Random and HC…

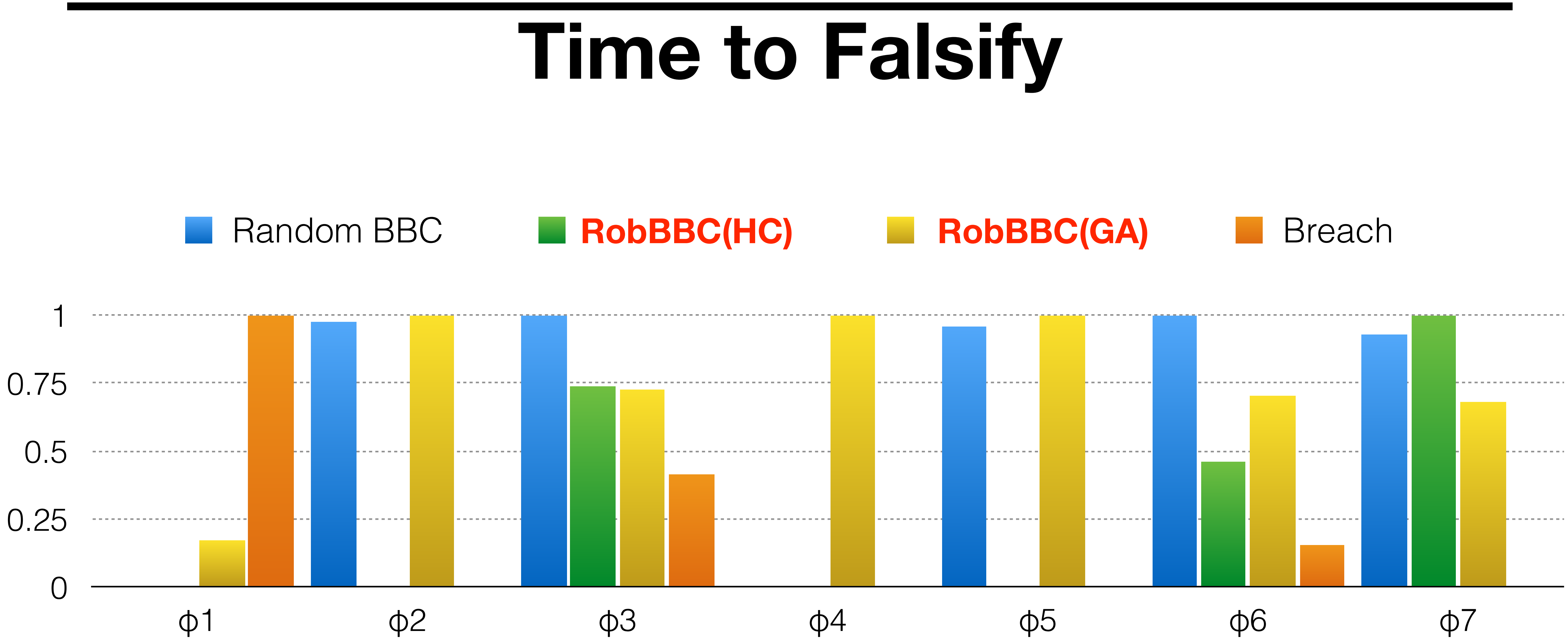
Time to Falsify by the Fastest Method



- RobBBC(GA) tends not be the fastest when others are available → not the best for easy spec.

M. Waga (NII)

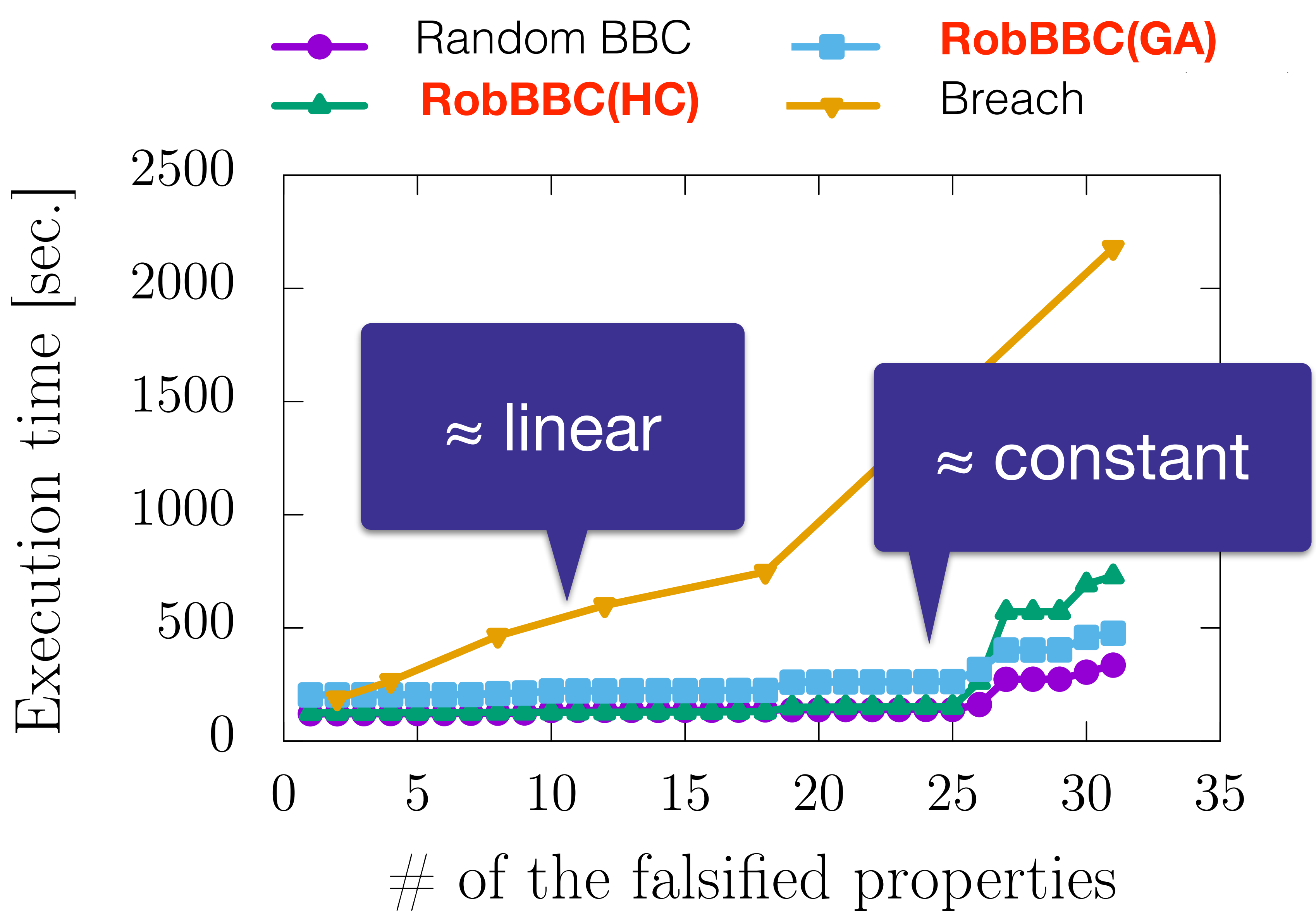
Time to Falsify by the Fastest Method



- RobBBC(GA) tends not be the fastest when others are available → not the best for easy spec.

RQ2: GA falsified more than Random HC
For easy spec. Random BBC can be an option

Scalability wrt # of Spec. in φ_6



RQ3: BBC is more scalable than Breach wrt # of Spec.

M. Waga (NII)

Conclusion

- Introduced robustness-guided BBC
  - Equivalence test using robustness
- Experimental evaluation
  - It outperforms Breach

Conclusion

- Introduced robustness-guided BBC
  - Equivalence test using robustness
- Experimental evaluation
  - It outperforms Breach

But this is the beginning of the story…

Direct Reuse of $\tilde{\mathscr{M}}$ Does Not Work

Model checking generates ***false*** counter examples

STL formula φ	# of $\varphi \not\models \tilde{\mathcal{M}}$	# of $\varphi \not\models \mathcal{M}$	Average of $[\![\varphi]\!]$	std. dev. of $[\![\varphi]\!]$
$\Box(v < 90)$	10	5	1.10	1.94
$\Box_{[0,26]}(v < 90) \lor \Box_{[28,28]}(v > 40)$	4	0	4.19	0.00
$\Box_{[0,26]}(v < 90) \lor \Box_{[28,28]}(v > 50)$	10	0	3.80	0.60
$\Box_{[0,26]}(v < 90) \lor \Box_{[28,28]}(v > 60)$	10	0	3.24	0.76

But robustness is low

M. Waga (NII)

Future Work

- How to (re-)use the generated Mealy machine?
- Black-box checking for multiple systems
    - Adaptive model checking?
- Try more benchmarks to figure out further issues

Appendix

Experiment Setting (Detail)

- Executed each same experiment setting 10 times
  - We used the average
- Amazon EC2 c4.large (2 vCPUs and 3.75 GiB RAM)