Università degli Studi di Napoli Federico II

Luca Giamattei

Reasoning-Based Software Testing

Tutor: Prof. Roberto Pietrantuono

Cycle: XXXVII

Year: Third

Candidate's information

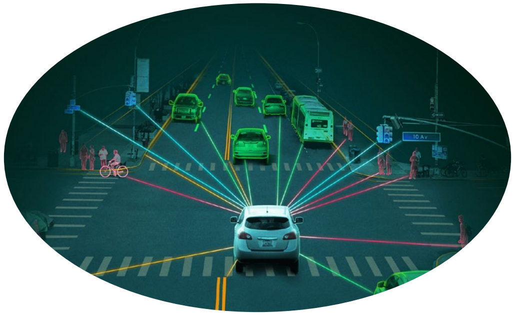
- MSc degree: Computer Engineering
- Research group: DESSERT
- PhD start date – end date: 01/11/2021 – 31/10/2024
- Fellowship: Unina
- Periods abroad: 3 months to Università della Svizzera Italiana (Prof. Paolo Tonella)
- Periods in company: 4 months to Panel Sistemas Informaticos (Madrid) – 1 month to Silensec (Cyprus)

uDevOps

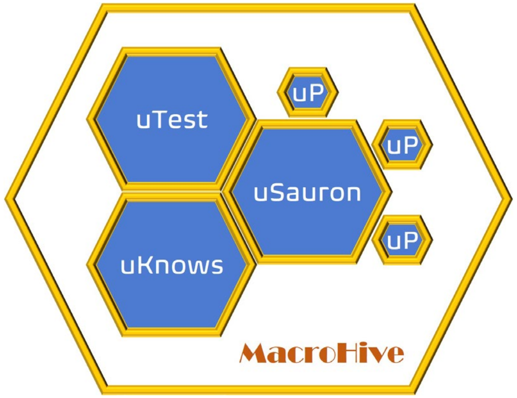
Summary of study activities

- **PhD schools:**
  - International Winter School on Blockchain Technology and Applications – Hyperledger, Università di Camerino
  - Advanced Course on Data Science & Machine Learning, Proff. Nicosia, Pardalos
- **Courses:**
  - Machine Learning (MSc course)
  - Statistical data analysis for science and engineering research (PhD course)
  - Virtualization technologies and their applications (PhD course, Dr. De Simone)
  - IoT Data Analysis (PhD course), Prof. Raffaele Della Corte
- **Conferences attended:**
  - International Conference on the Quality of Information and Communications Technology (QUATIC), Talavera de la Reina, Spain, September 2022
  - 22nd IEEE International Conference on Software Quality, Reliability and Security (QRS), Guangzhou, China, 2022 (online)
  - 45th IEEE/ACM International Conference on Software Engineering (ICSE), Melbourne, Australia, 2023
  - 34th IEEE International Symposium on Software Reliability Engineering (ISSRE), Florence, Italy, 2023
  - 46th IEEE/ACM International Conference on Software Engineering (ICSE), Lisbon, Portugal, 2024

Research area:
Software Testing



- Testing of Autonomous Driving Systems



- Testing of Microservices Architectures
  - Product:

    MacroHive, a prototype for automated grey-box MSA testing and Root Cause Analysis

Research results

- Major contribution*:* Definition, formalization, and evaluation of a novel software testing mothodology based on causal reasoning called **Reasoning-Based Software Testing**
- *Minor contributions:*
  - *Evaluation and implementation of monitoring infrastructures for microservices*
  - *Anomaly detection and Root cause analysis of energy consumption in microservices*

Research products (1)

[P1]	L. Giamattei, A. Guerriero, R. Pietrantuono, S. Russo, *Causality-driven Testing of Autonomous Driving Systems,* **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Vol 33, 3, 2024, DOI: 10.1145/3635709.
[P2]	L. Giamattei, A. Guerriero, R. Pietrantuono, S. Russo, I. Malavolta, T. Islam, M. Dînga, A. Koziolek, S. Singh, M. Armbruster, J.M. Gutierrez-Martinez, S. Caro-Alvaro, D. Rodriguez, S. Weber, J. Henss, E. Fernandez Vogelin, F. Simon Panojo, *Monitoring tools for DevOps and microservices: A systematic grey literature review,* **Journal of Systems and Software (JSS)**, Vol 208, 2024, 111906, ISSN 0164-1212, DOI: 10.1016/j.jss.2023.111906
[P3]	L. Giamattei, A. Guerriero, R. Pietrantuono, S. Russo, *Automated functional and robustness testing of microservice architectures,* **Journal of Systems and Software (JSS)**, Vol 207, 2024, 111857, ISSN 0164-1212, DOI: 10.1016/j.jss.2023.111857.
[P4]	L. Giamattei, M. Biagiola, R. Pietrantuono, S. Russo, P. Tonella, *Reinforcement Learning for Online Testing of Autonomous Driving Systems: a Replication and Extension Study,* **Empirical Software Engineering (EMSE)**, Accepted, to appear.

Research products (2)

[P5]	L. Giamattei, A. Guerriero, R. Pietrantuono, S. Russo, *Causal Reasoning in Software Quality Assurance: A Systematic Review,* **Information and Software Technology (IST),** Accepted, to appear.
[P6]	L. Giamattei, A. Guerriero, R. Pietrantuono and S. Russo, *Assessing Black-box Test Case Generation Techniques for Microservices*, **15th International Conference on the Quality of Information and Communications Technology (QUATIC),** Talavera de la Reina, Spain, Sep. 12-14, 2022, pp. 46-60, Springer, DOI: 10.1007/978-3-031-14179-9_4
[P7]	L. Giamattei, A. Guerriero, R. Pietrantuono, S. Russo, *Automated Grey-Box Testing of Microservice Architectures*, **22nd International Conference on Software Quality, Reliability and Security (QRS)**, Guangzhou, China, 2022, pp. 640-650, IEEE, DOI: 10.1109/QRS57517.2022.00070
[P8]	L. Giamattei, R. Pietrantuono, S. Russo, *Reasoning-Based Software Testing*, **45th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)**, Melbourne, Australia, May 14-20, 2023, pp. 66-71, IEEE, DOI: 10.1109/ICSE-NIER58687.2023.00018

Research products (3)

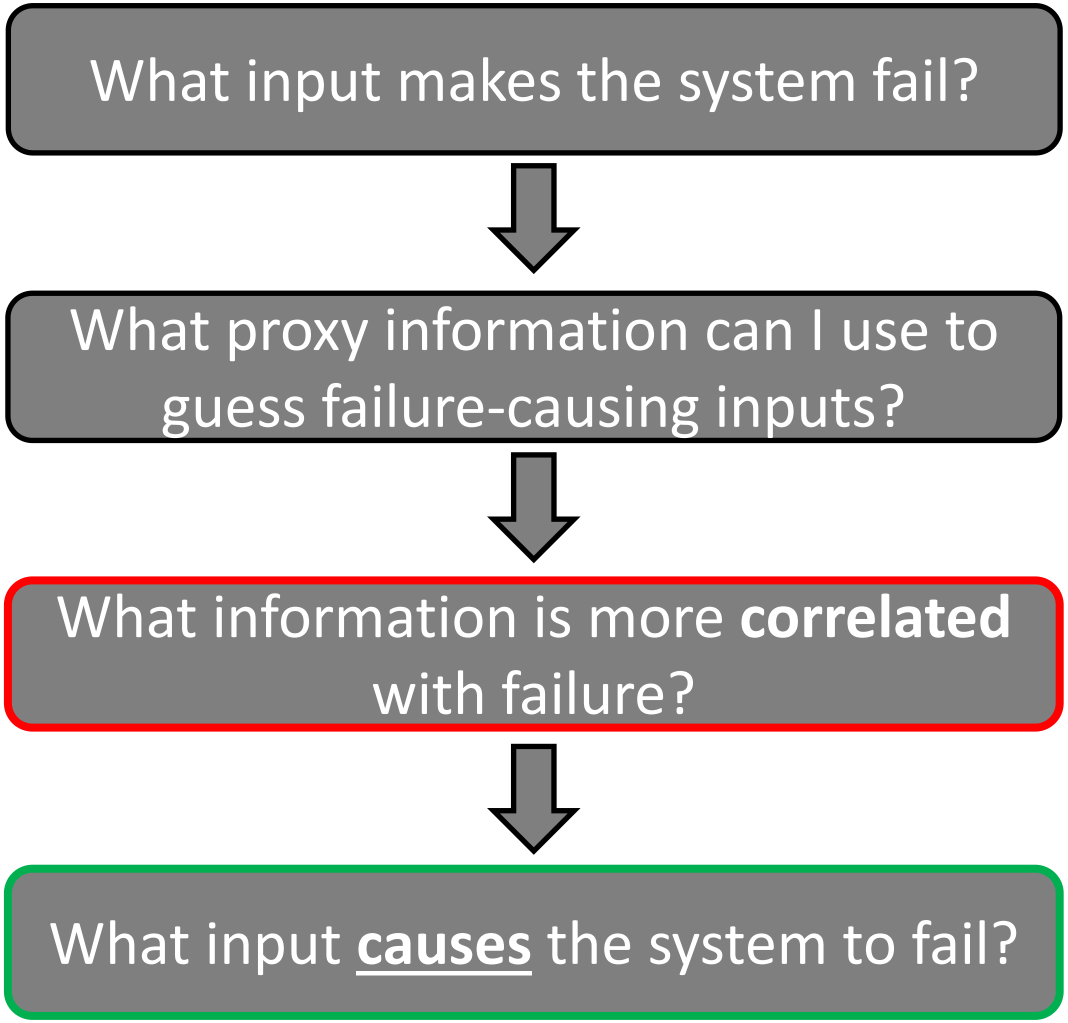
[P9]	M. Dinga, I. Malavolta, L. Giamattei, A. Guerriero, R. Pietrantuono, *An Empirical Evaluation of the Energy and Performance Overhead of Monitoring Tools on Docker-Based Systems*, **21st International Conference on Service-Oriented Computing (ICSOC)**, Rome, Italy, Nov. 28 – Dec. 1, 2023, pp. 181-196, Springer, DOI: 10.1007/978-3-031-48421-6_13
[P10]	L. Giamattei, A. Guerriero, I. Malavolta, C. Mascia, R. Pietrantuono, S. Russo, *Identifying Performance Issues in Microservice Architectures through Causal Reasoning*, **5th International Conference on Automation of Software Test (AST)**, Lisbon, Portugal, Apr. 15-16, 2024, pp. 149–153, ACM, DOI: 10.1145/3644032.3644460
[P11]	M. S. Floroiu, S. Russo, L. Giamattei, A. Guerriero, I. Malavolta, R. Pietrantuono, *Anomaly Detection and Root Cause Analysis of Microservices Energy Consumption*, **International Conference on Web Services (ICWS)**, Shenzhen, China, Jul. 7-13, 2024, IEEE, to appear.

PhD thesis overview

- **Motivation:**
  - Automation of software testing is key in modern systems
  - Evolution of Software Testing led to data-driven techniques that are ineffective in complex and dynamic environments
- **Problem statement:**
  - Data-driven techniques are not able to answer the fundamental question of software testing:

    “What input causes the system to fail?”
- **Objective:**
  - Shift from *data-driven* to *reasoning-driven* software testing
- **Methodology:**
  - *Reasoning-Based Software Testing (RBST)*
  - Validation trough stateless and stateful testing of Autonomous Driving Systems

Evolution of Software testing



Guessing/
Intuition

What input makes the system fail?

Belief/
Experience

What proxy information can I use to guess failure-causing inputs?

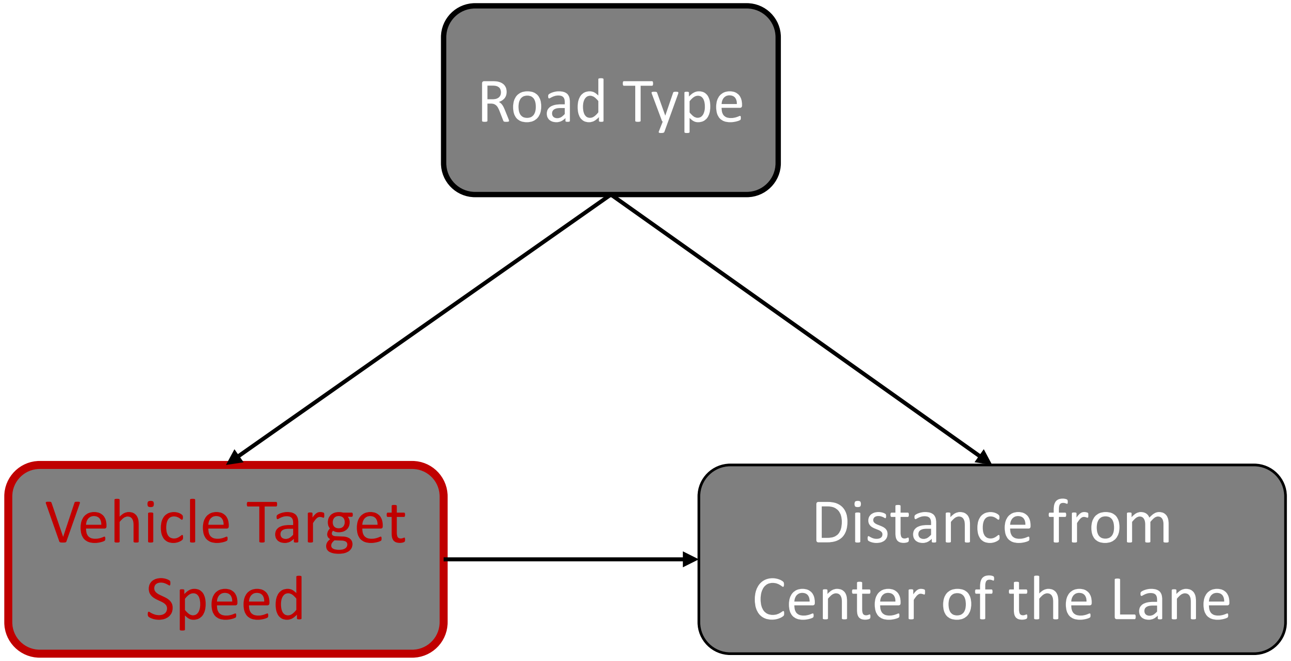
Past observations/
Past tests

What information is more **correlated** with failure?

Automated
Reasoning

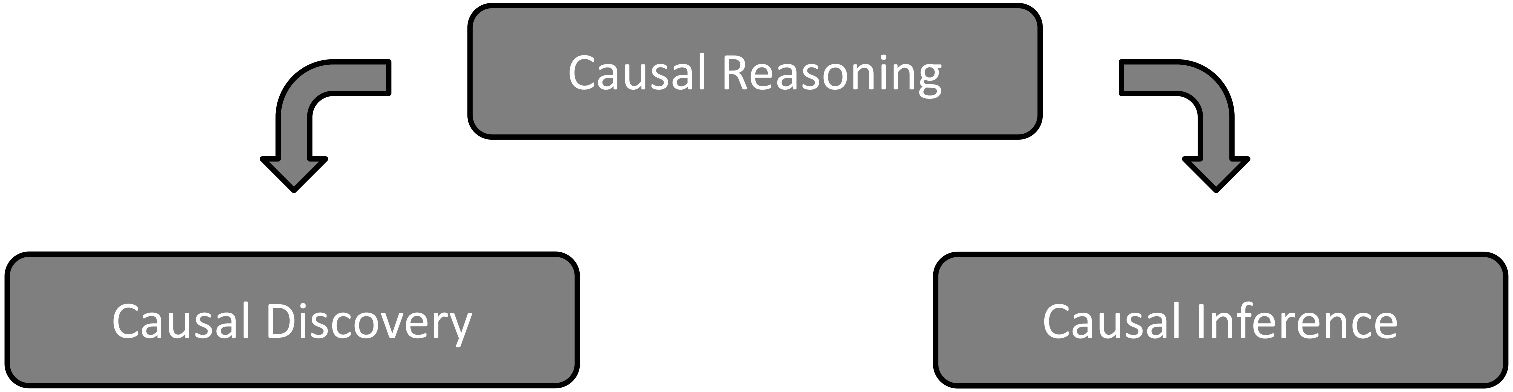
What input **causes** the system to fail?

Confounding bias



$$P(D|V = v)$$

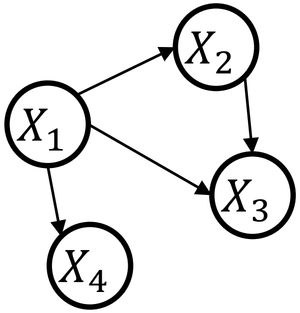
Causal Reasoning



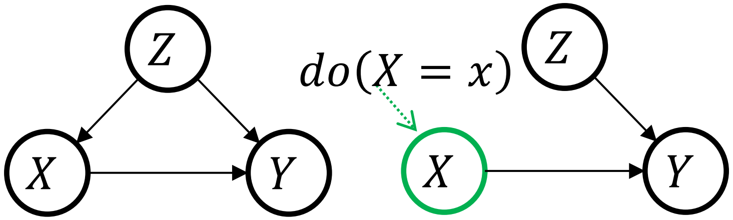
Causal Reasoning

Causal Discovery

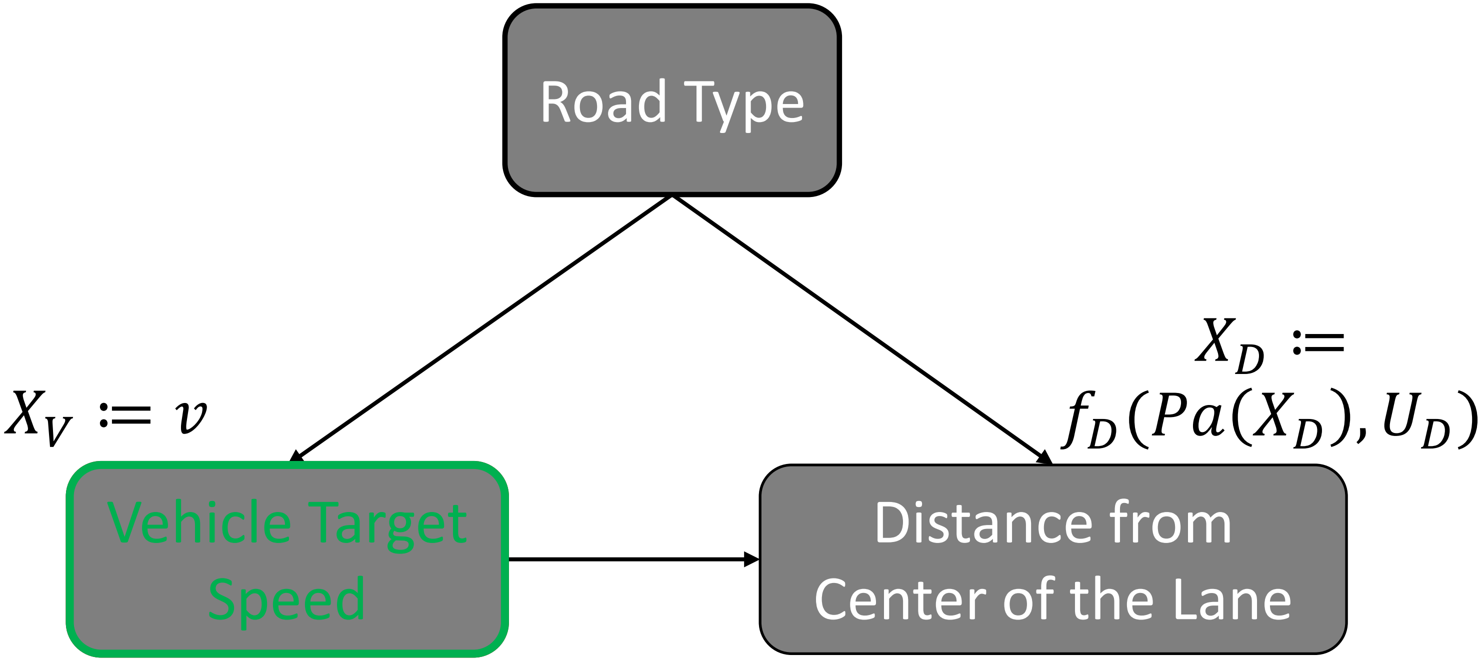
Causal Inference



$X_i \coloneqq f_i(Pa(X_i), U_i)$



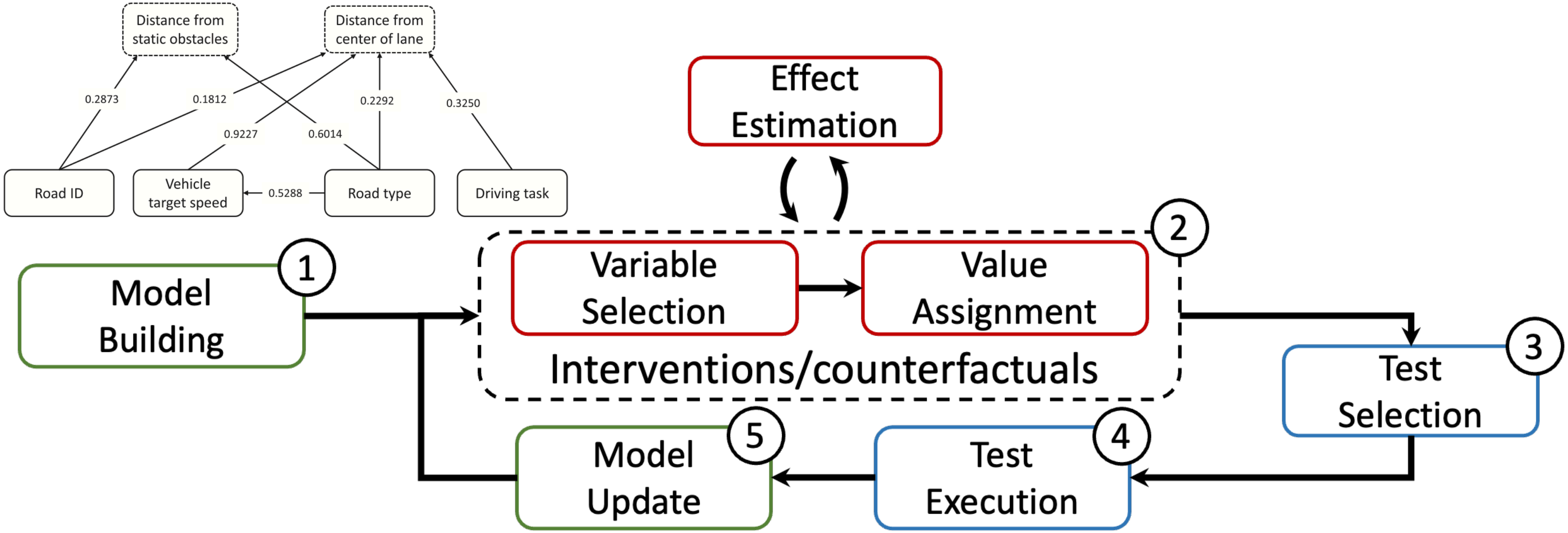
Confounding bias



$$P(D \mid do(V = v))$$

$$P(D \mid V = v)$$

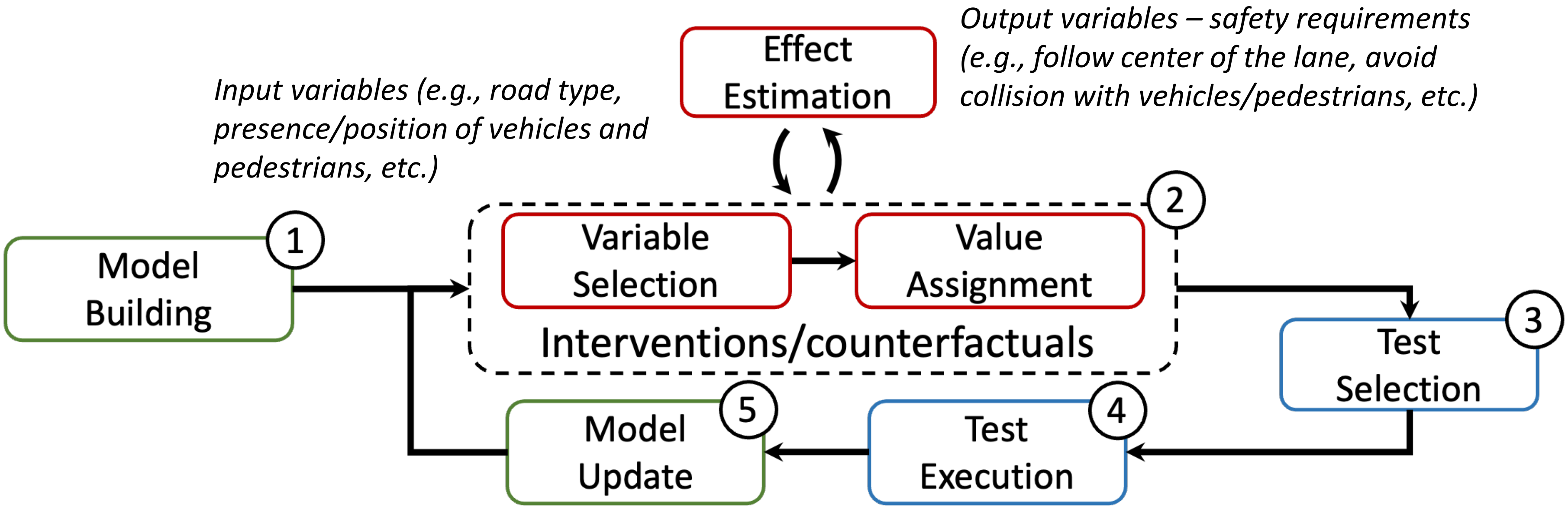
Reasoning-Based Software Testing



1 Build a causal model

- Manually
- Randomized Controlled Experiments
- Causal Discovery Algorithms

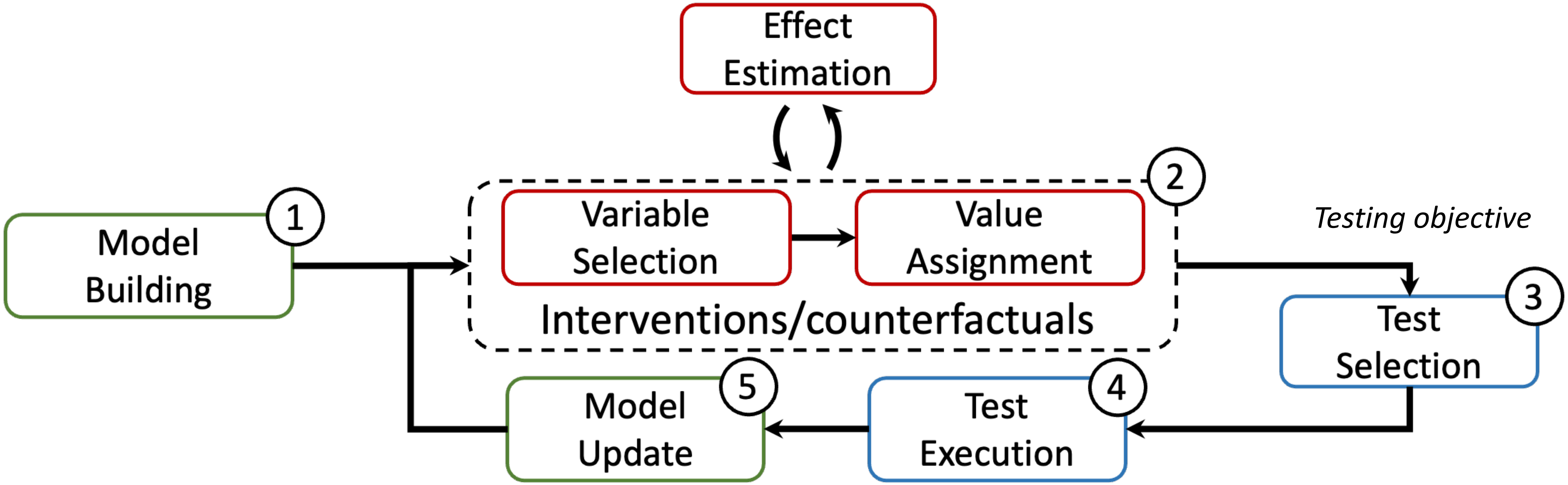
Reasoning-Based Software Testing


Effect Estimation
Output variables – safety requirements (e.g., follow center of the lane, avoid collision with vehicles/pedestrians, etc.)
Input variables (e.g., road type, presence/position of vehicles and pedestrians, etc.)
2
Model Building
1
Variable Selection
Value Assignment
Interventions/counterfactuals
Test Selection
3
Model Update
5
Test Execution
4

2 Query model via Interventions/counterfactuals

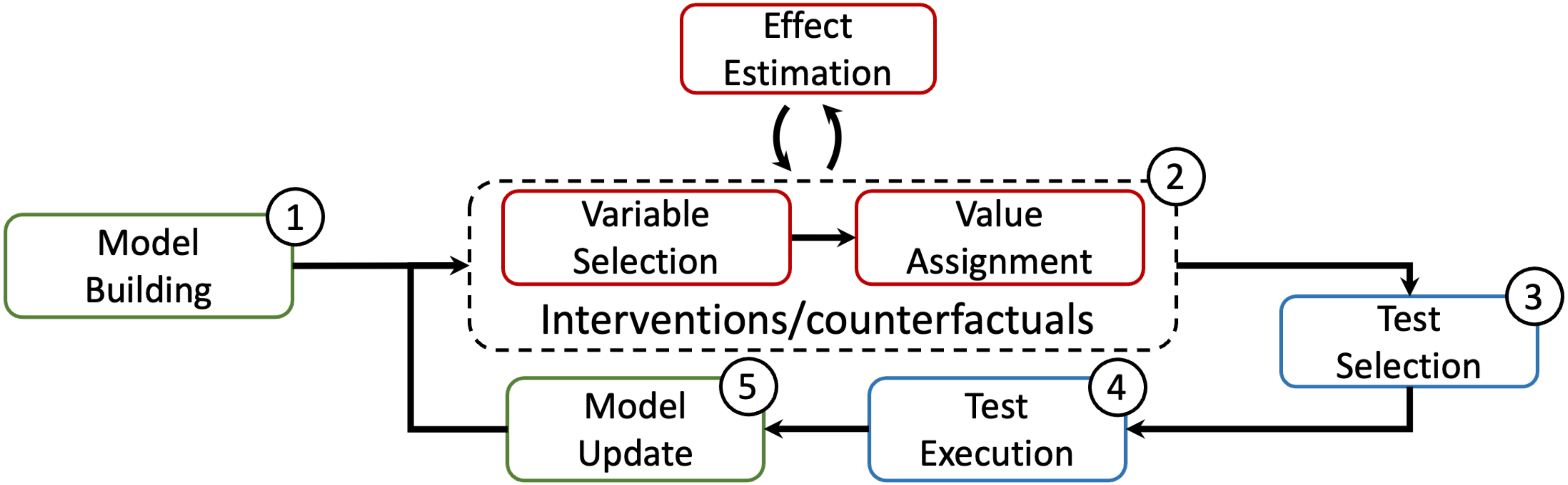
- Select variable on which intervene
- Assign value of intervention
- Estimate effect and generate *hypothetical tests*

Reasoning-Based Software Testing



3 Select set of *hypothetical* tests

Reasoning-Based Software Testing



4 Execute selected tests

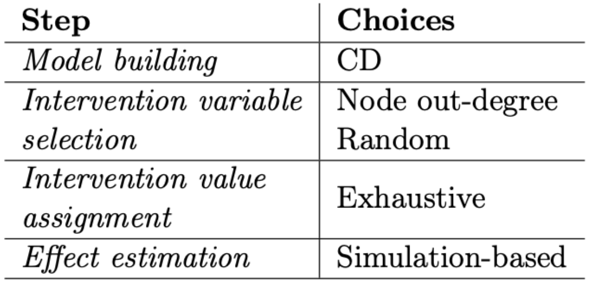
- Observe actual value (not hypothetical)

5 Update causal model

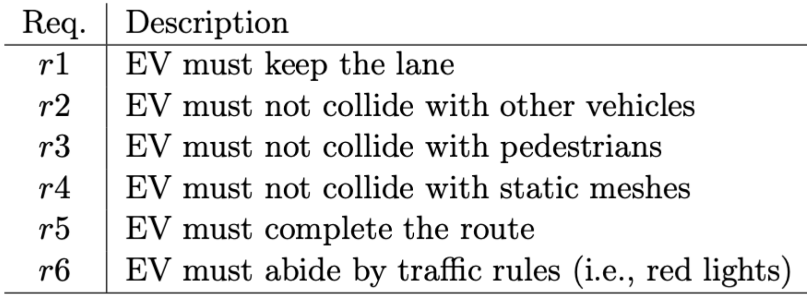
- Use *real* observations to refine the model

Reasoning-Based Software Testing instantiation for **stateless** testing

- RBST Instance:
  - CART (CAusal-Reasoning-driven Testing)

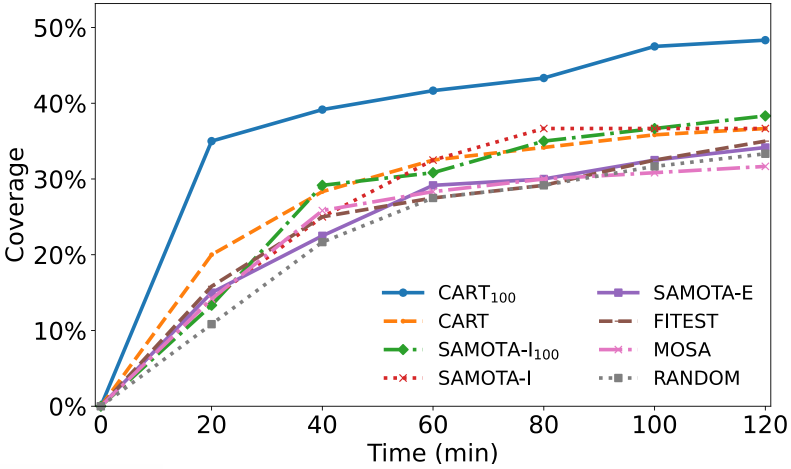


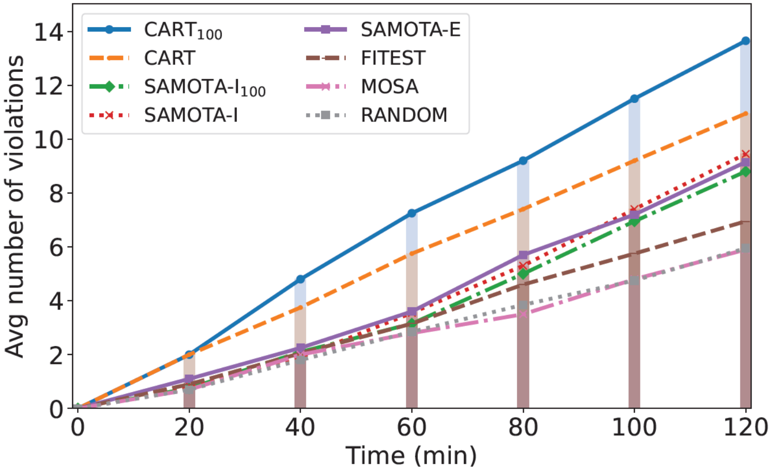
Step	**Choices**
Model building	CD
Intervention variable selection	Node out-degree Random
Intervention value assignment	Exhaustive
Effect estimation	Simulation-based



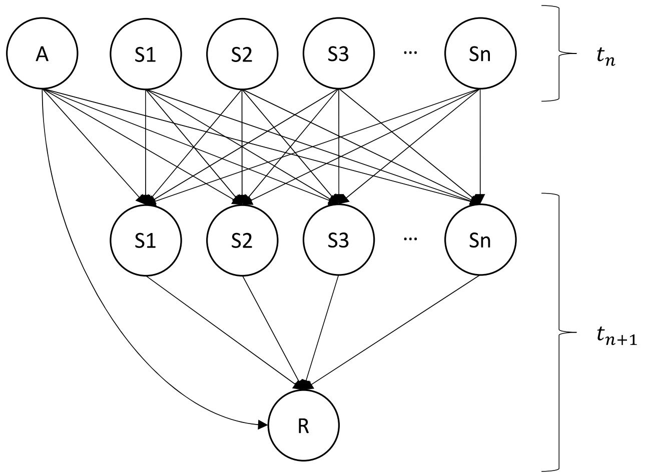
Req.	Description
$r1$	EV must keep the lane
$r2$	EV must not collide with other vehicles
$r3$	EV must not collide with pedestrians
$r4$	EV must not collide with static meshes
$r5$	EV must complete the route
$r6$	EV must abide by traffic rules (i.e., red lights)

- ADS safety and functional requirements



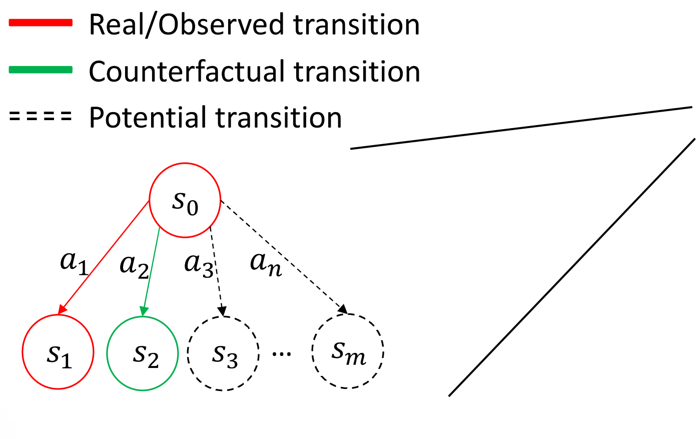


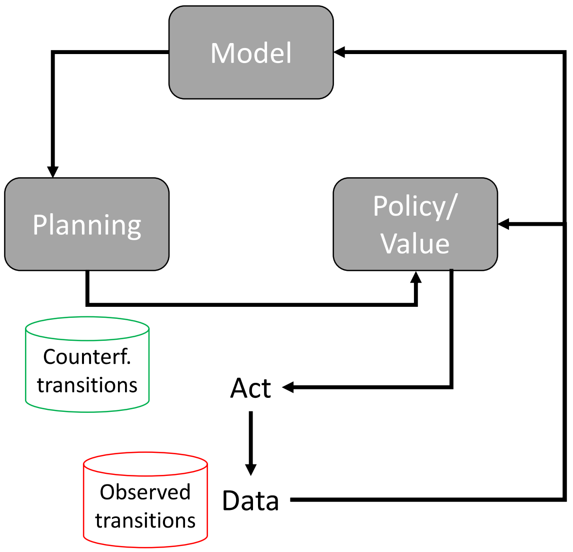
Reasoning-Based Software Testing instantiation for **stateful** testing



- Markov Decision Process mapping to Structural Causal Model

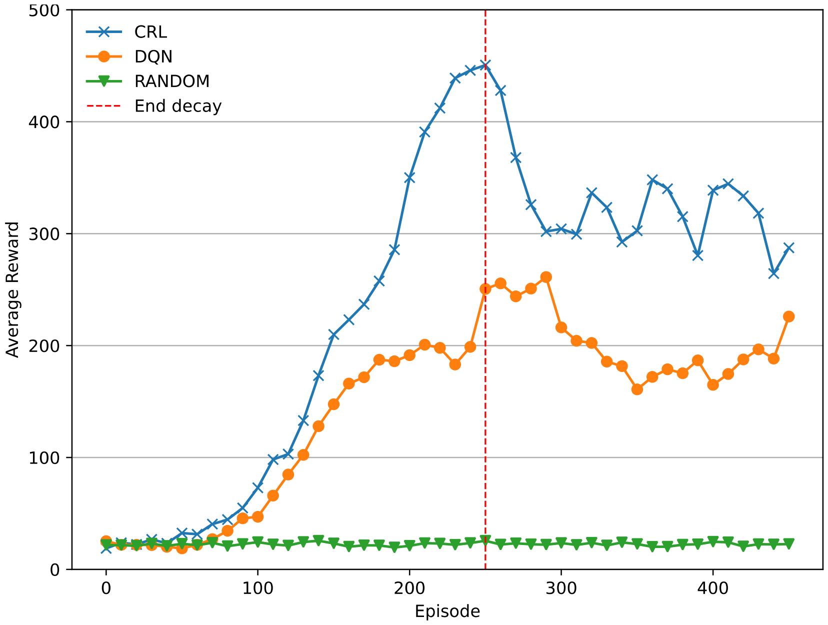
Step	**Choices**
Model building	Domain knowledge
Intervention variable selection	Fixed (action)
Intervention value assignment	Random
Effect estimation	Simulation-based





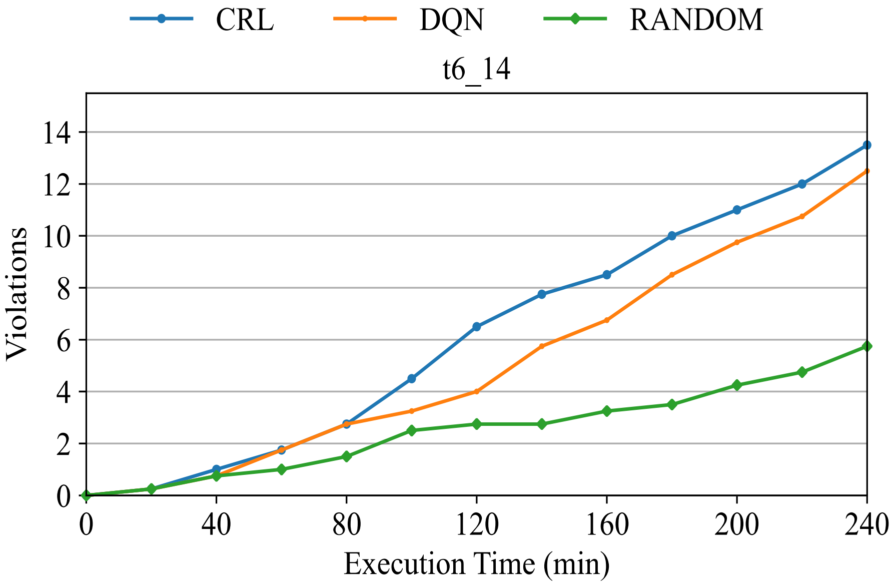
Reasoning-Based Software Testing instantiation for **stateful** testing

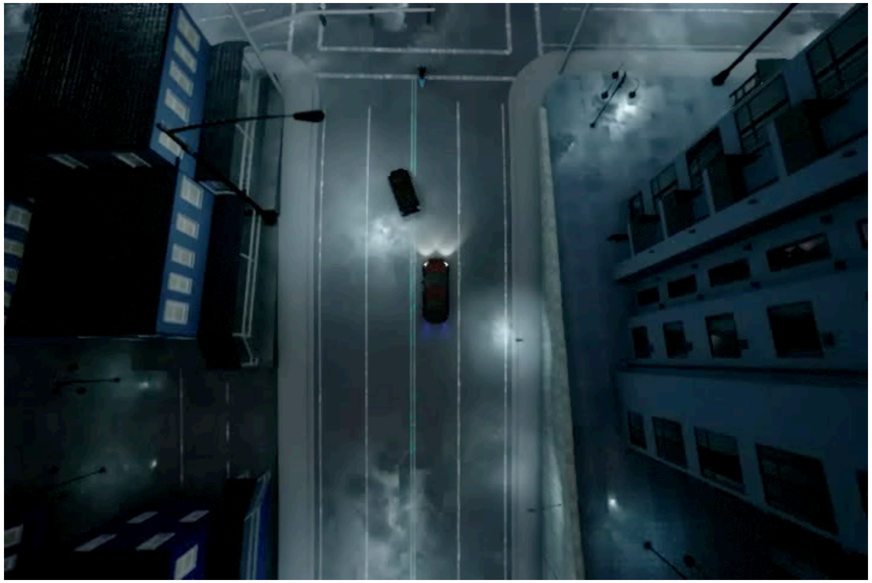
- Motivation study (cartpole env.)
  - *action: cart right/left*
  - *reward:* +1 if pole up
  - *state: cart/pole position, angle and speed*



Reasoning-Based Software Testing instantiation for **stateful** testing

- ADS testing
  - *scenario*: 4-way intersection
  - *r2:* collision with other vehicles
  - *actions:* vehicles lane offset and change lane right/left, speed up/down
  - *state:* vehicles speed, heading, and distance from EV





Thank you