WOLVERINE 2023

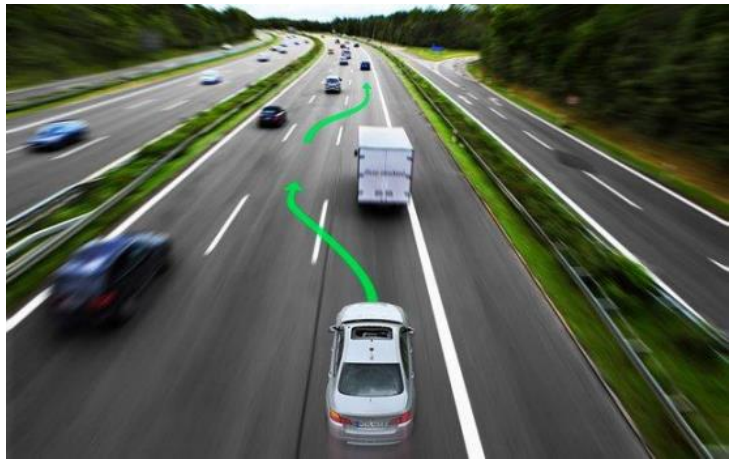
TU Graz

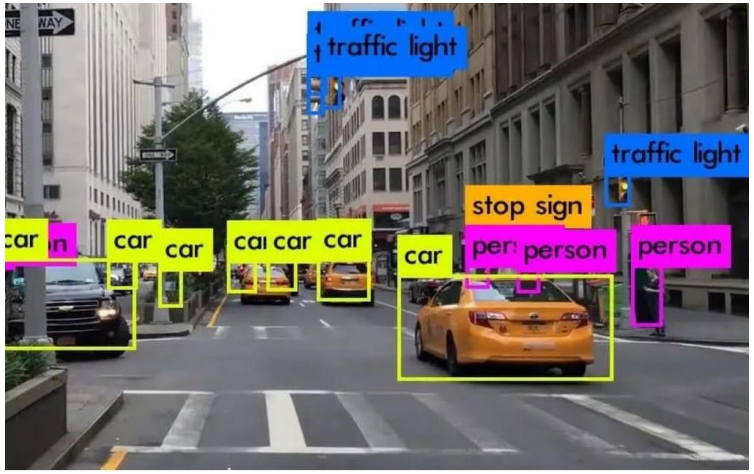
SCIENCE
PASSION
TECHNOLOGY

Trusted Reinforcement Learning

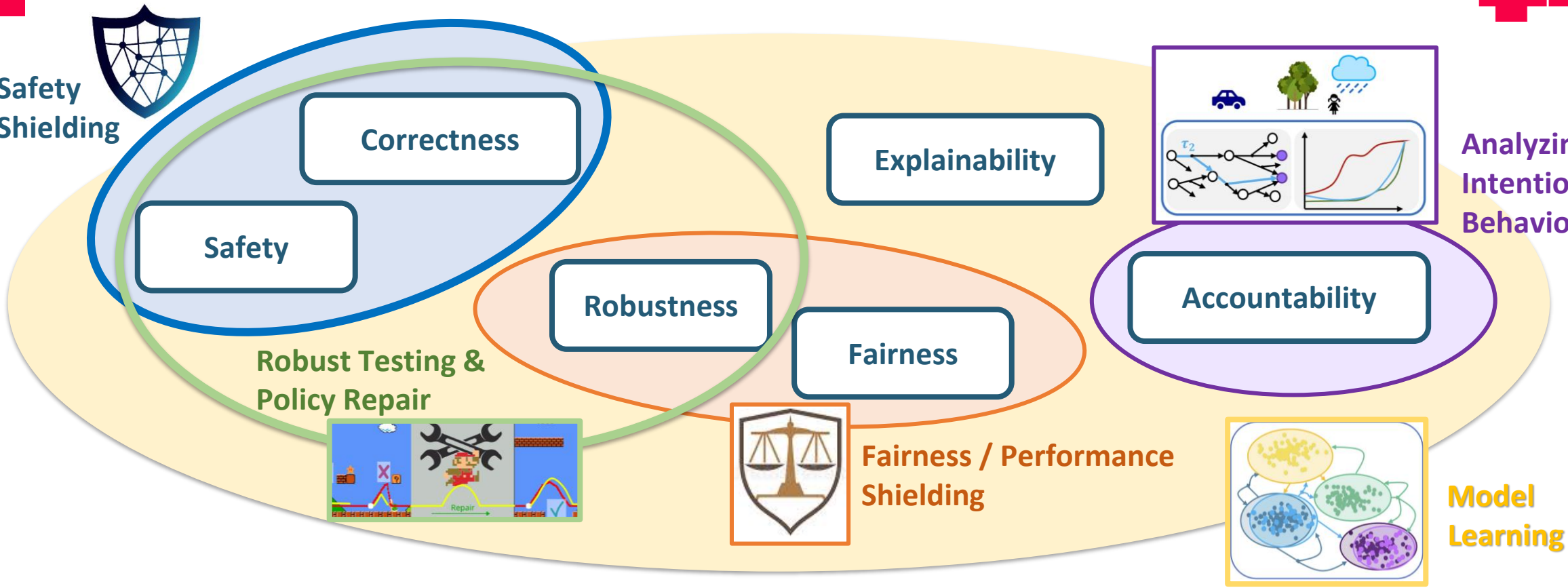
Bettina Könighofer
bettina.koenighofer@iaik.tugraz.at

Paris 17 July 2023



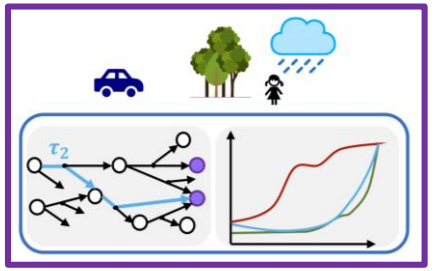
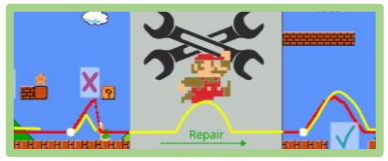


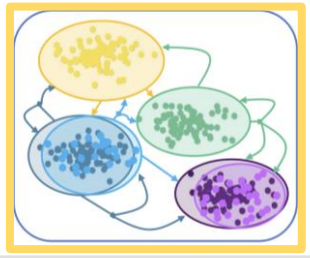
We need trustworthy AI!



We need trustworthy AI!

Outline

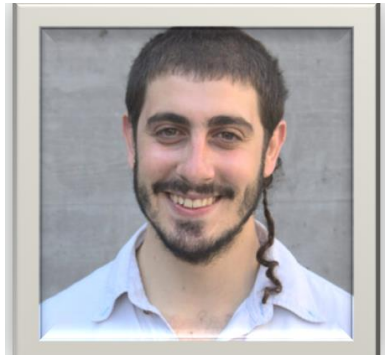
- Shielding for Safety 
- Shielding for Fairness / Performance 
- Analyzing Evidence of Intentional Behavior 
- Testing and Policy Repair 

- Model Learning 

Safety Shielding - Joint work with

Stefan Pranger

Filip Cano Cordoba



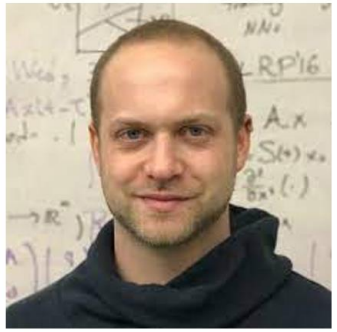
Roderick Bloem

Robert Könighofer

Ufuk Topcu

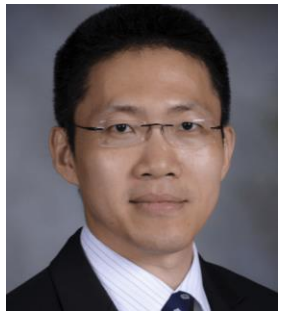
Rüdiger Ehlers

Nils Jansen



Sebastian Junges

Chao Wang



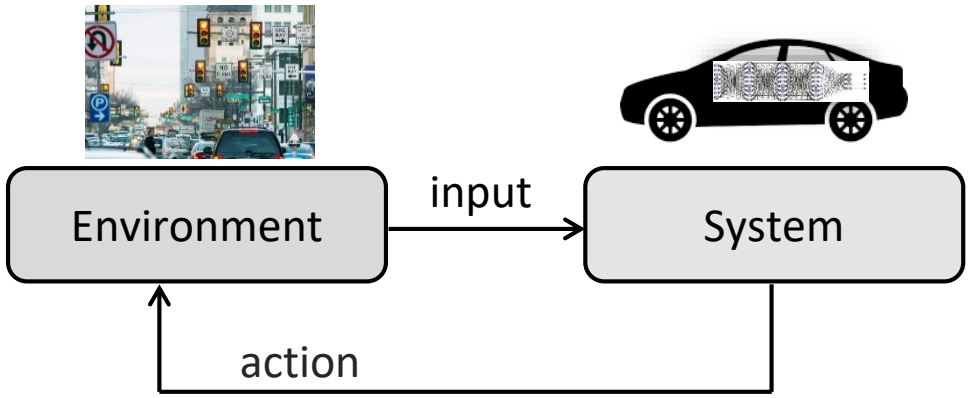
THE UNIVERSITY OF TEXAS AT AUSTIN

TU Clausthal
Clausthal University of Technology

Radboud University Nijmegen

VIRGINIA TECH

How to guarantee Safety?



?

Model of environment

Formal Safety Specification

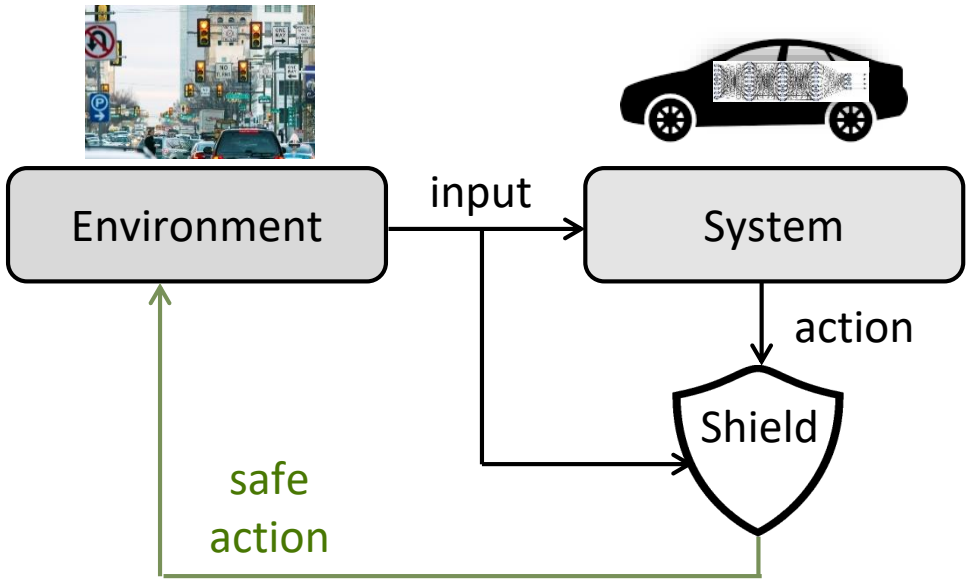
Verification inconclusive

System too complicated

… but we need to have absolute certainty

R. Bloem, B. Könighofer, R. Könighofer, C. Wang:
Shield Synthesis - Runtime Enforcement for Reactive Systems. **TACAS 2015**

How to guarantee Safety?



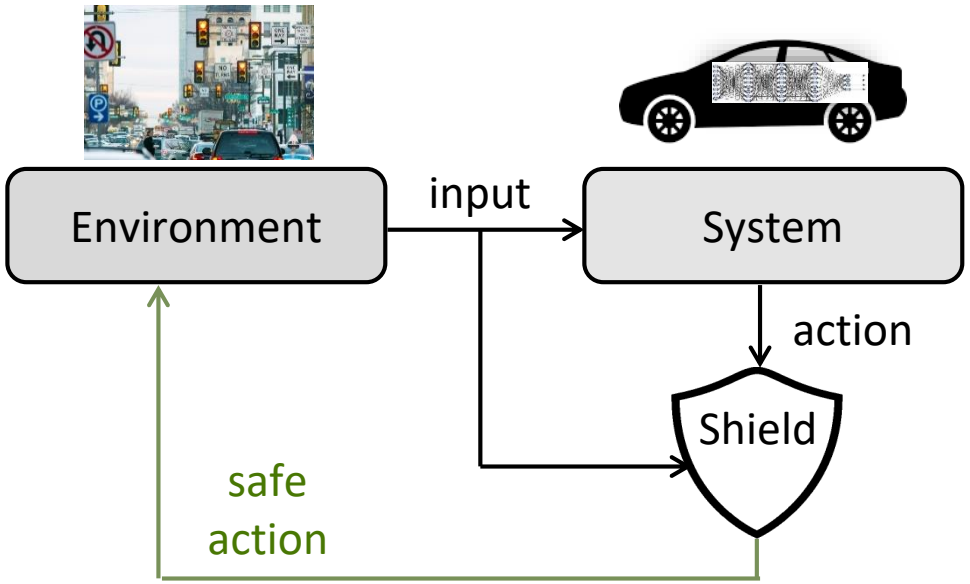
Model of environment

Formal Safety Specification

R. Bloem, B. Könighofer, R. Könighofer, C. Wang:
Shield Synthesis - Runtime Enforcement for Reactive Systems. **TACAS 2015**

Shielding - Properties

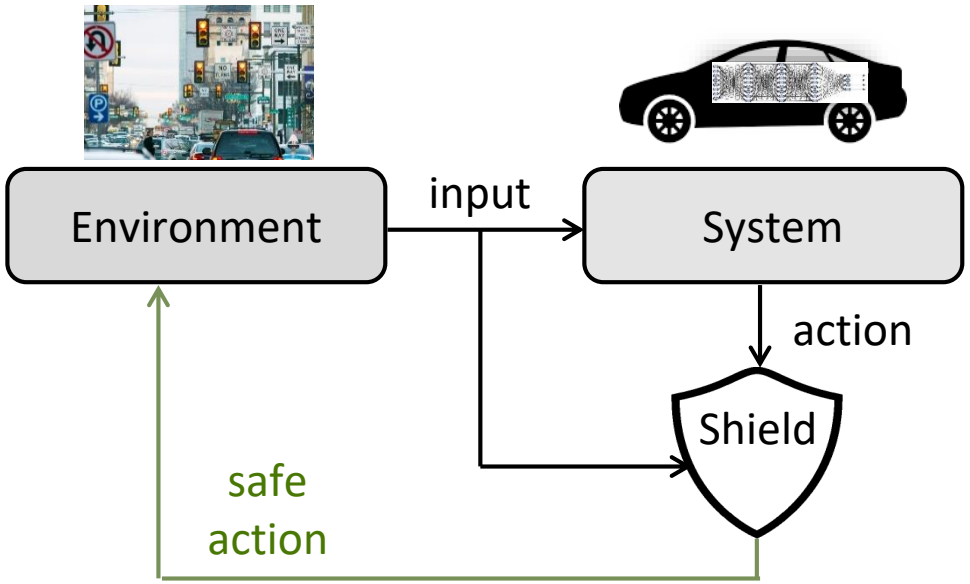
1. **Shields guarantee correctness**
2. **Shields are minimal interfering**



R. Bloem, B. Könighofer, R. Könighofer, C. Wang:
Shield Synthesis - Runtime Enforcement for Reactive Systems. **TACAS 2015**

Shielding - Properties

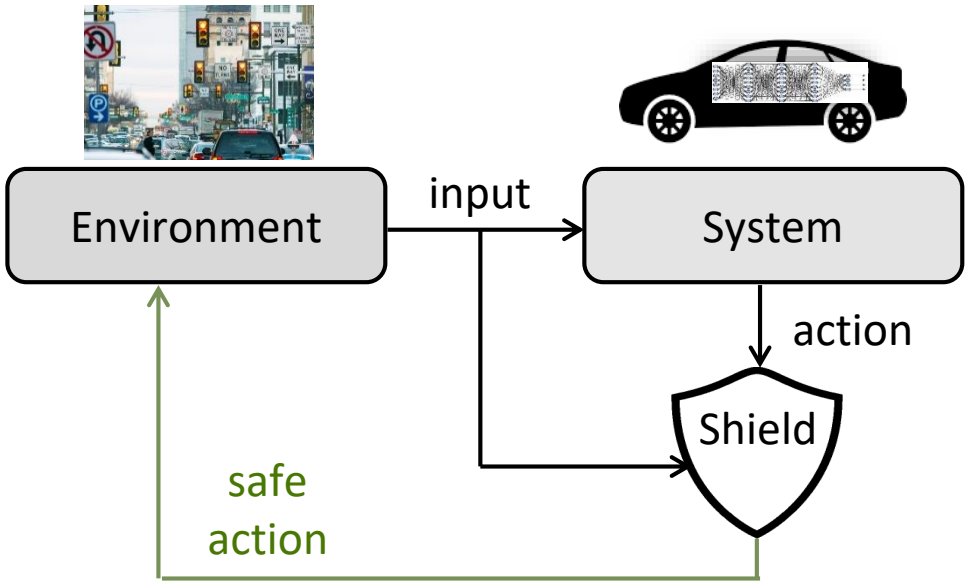
1. **Shields guarantee correctness**
   - Correct-by-construction
   - Predictive

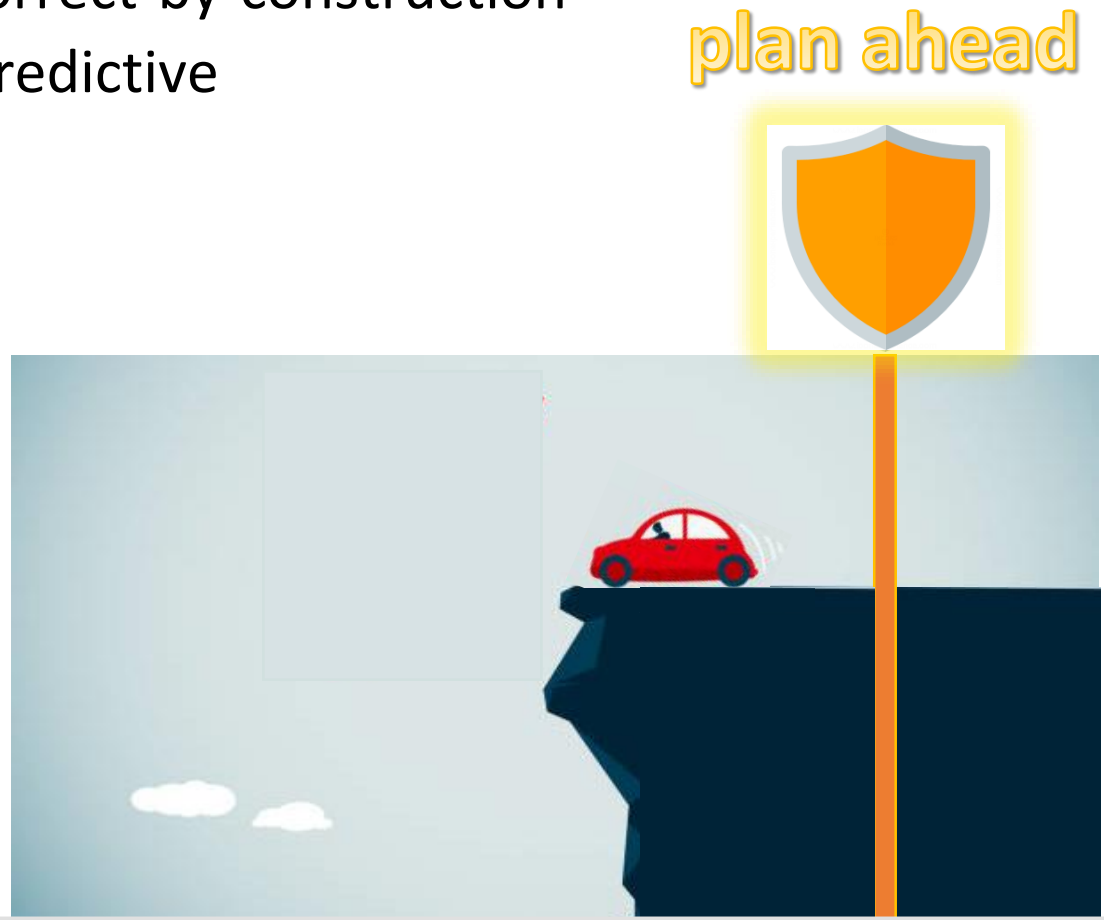


R. Bloem, B. Könighofer, R. Könighofer, C. Wang:
Shield Synthesis - Runtime Enforcement for Reactive Systems. **TACAS 2015**

Shielding - Properties

1. **Shields guarantee correctness**
   - Correct-by-construction
   - Predictive

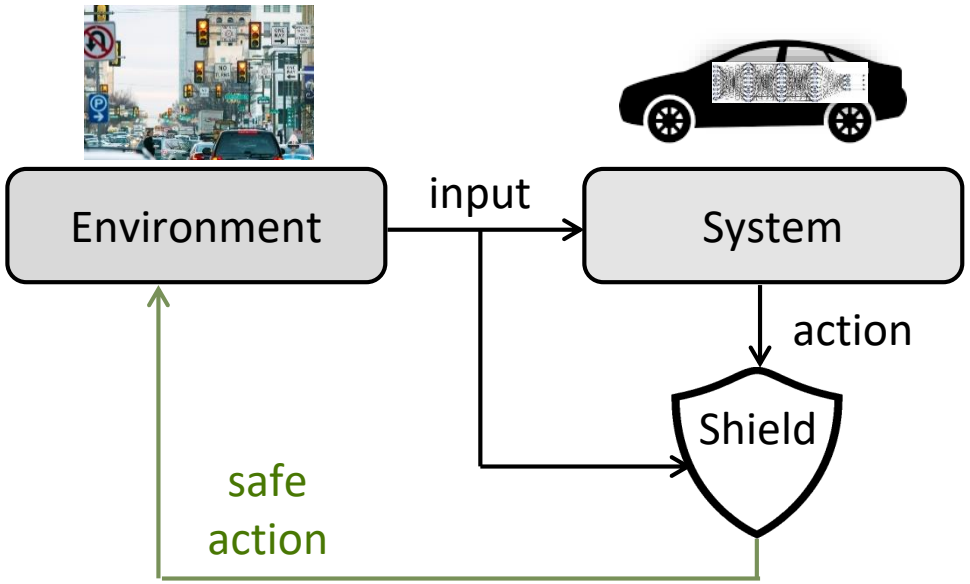


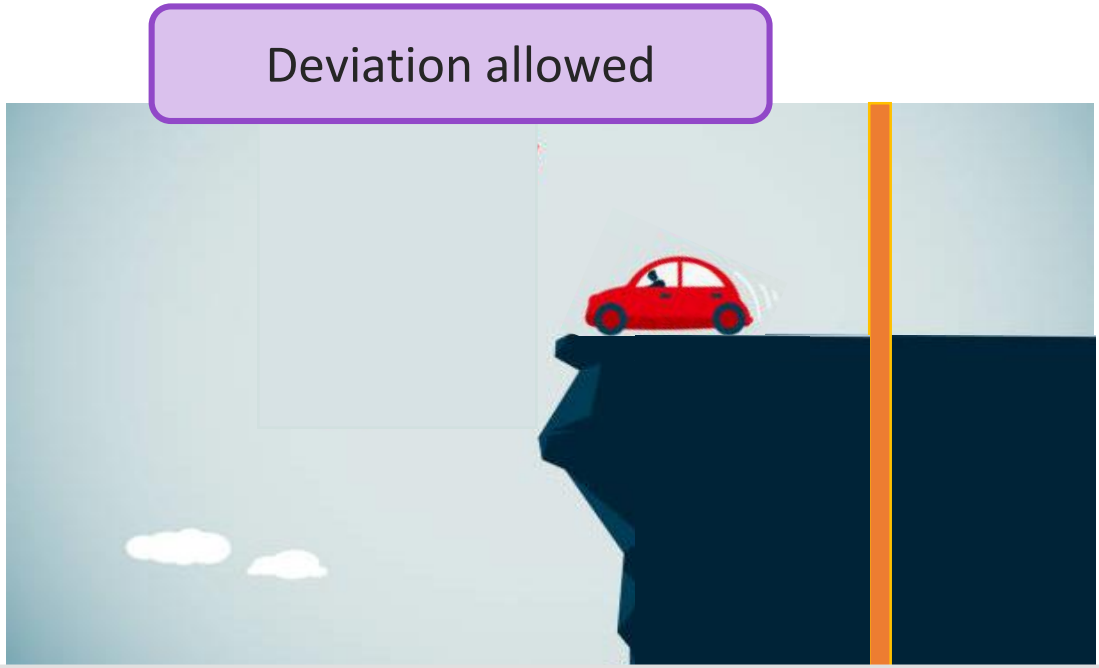


R. Bloem, B. Könighofer, R. Könighofer, C. Wang:
Shield Synthesis - Runtime Enforcement for Reactive Systems. **TACAS 2015**

Shielding - Properties

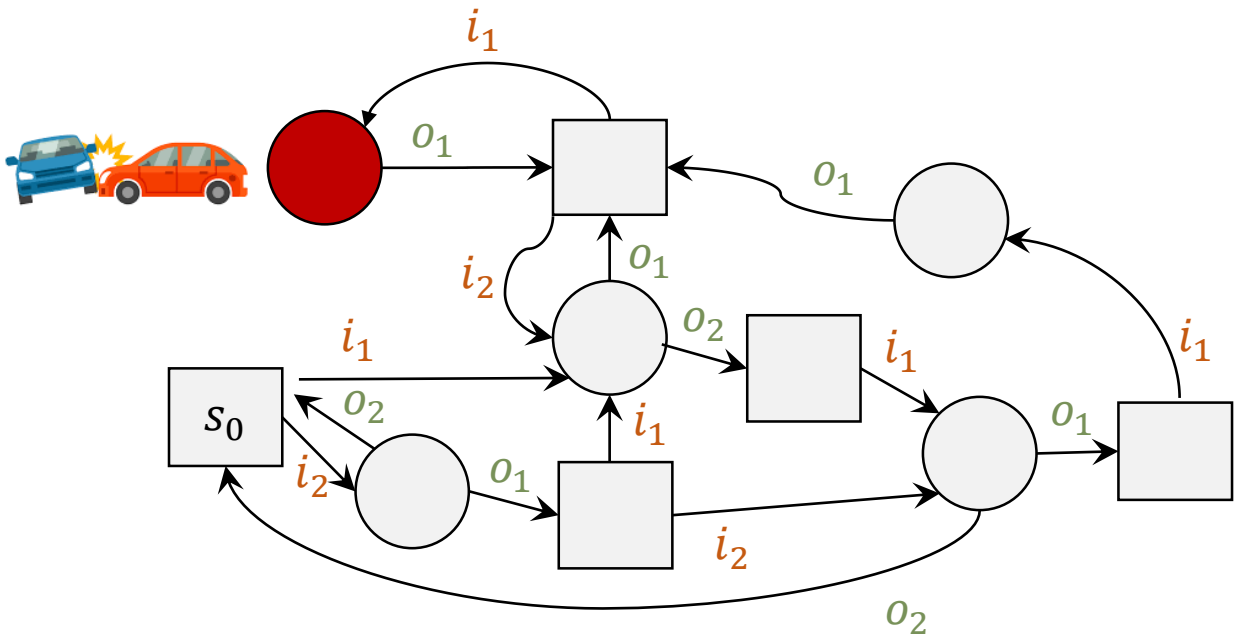
1. **Shields guarantee correctness**
   - Correct-by-construction
   - Predictive
2. **Shields are minimal interfering**





R. Bloem, B. Könighofer, R. Könighofer, C. Wang:
Shield Synthesis - Runtime Enforcement for Reactive Systems. **TACAS 2015**

Shield Construction – Synthesis is a Game

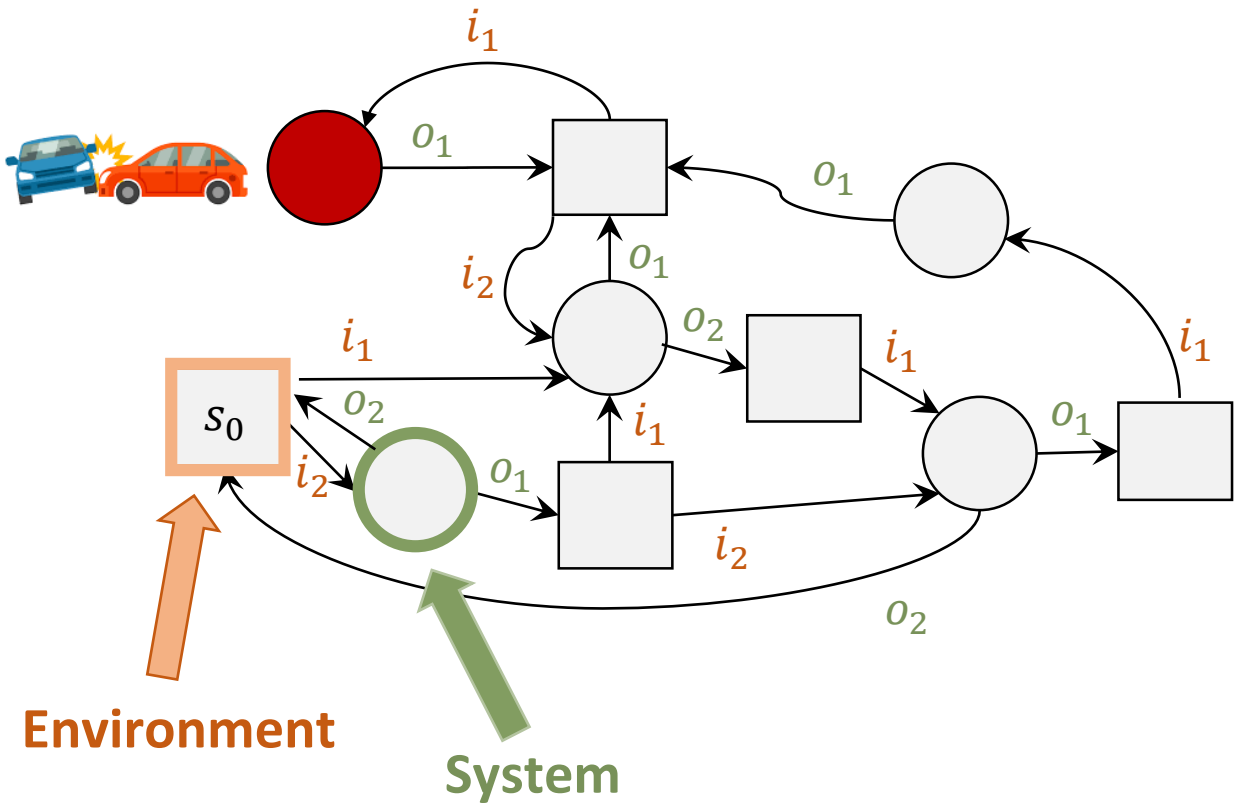


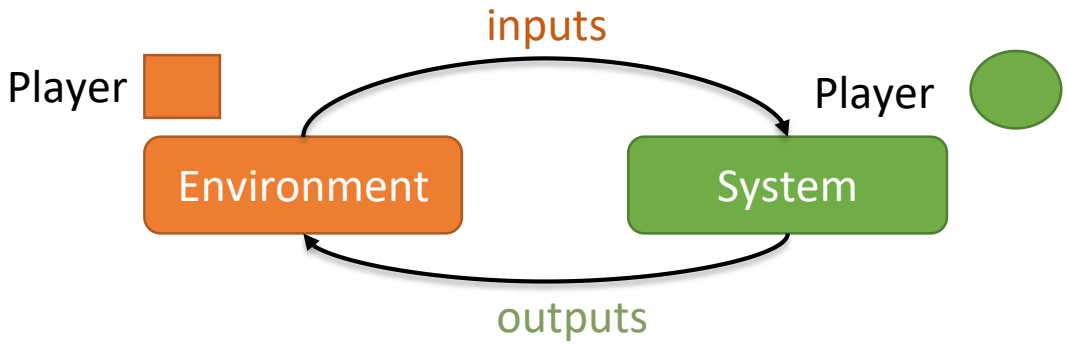
Formal safety specification

Model of environment

R. Bloem, B. Könighofer, R. Könighofer, C. Wang:
Shield Synthesis - Runtime Enforcement for Reactive Systems. **TACAS 2015**

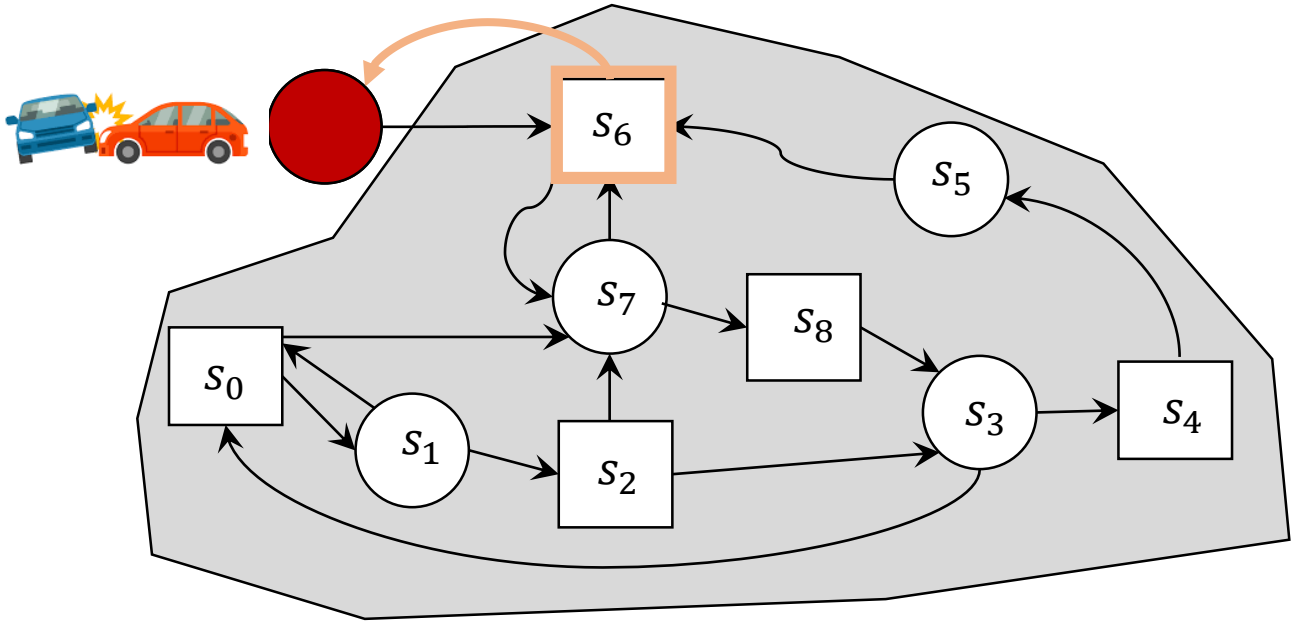
Shield Construction – Synthesis is a Game

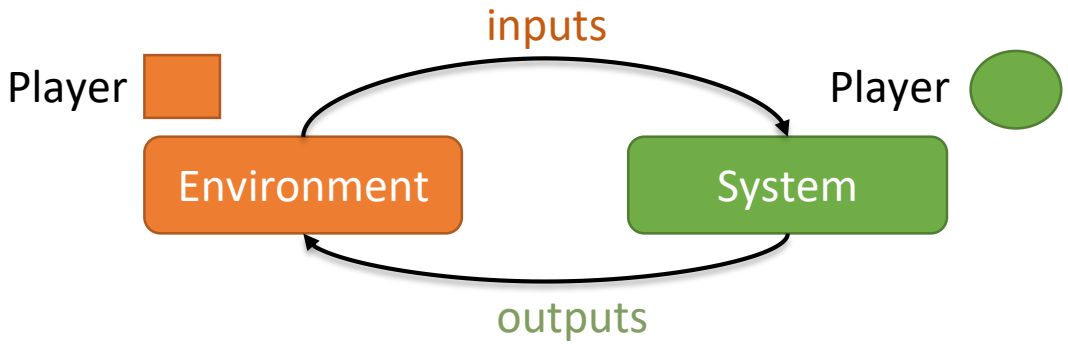




R. Bloem, B. Könighofer, R. Könighofer, C. Wang:
Shield Synthesis - Runtime Enforcement for Reactive Systems. **TACAS 2015**

Shield Construction – Synthesis is a Game

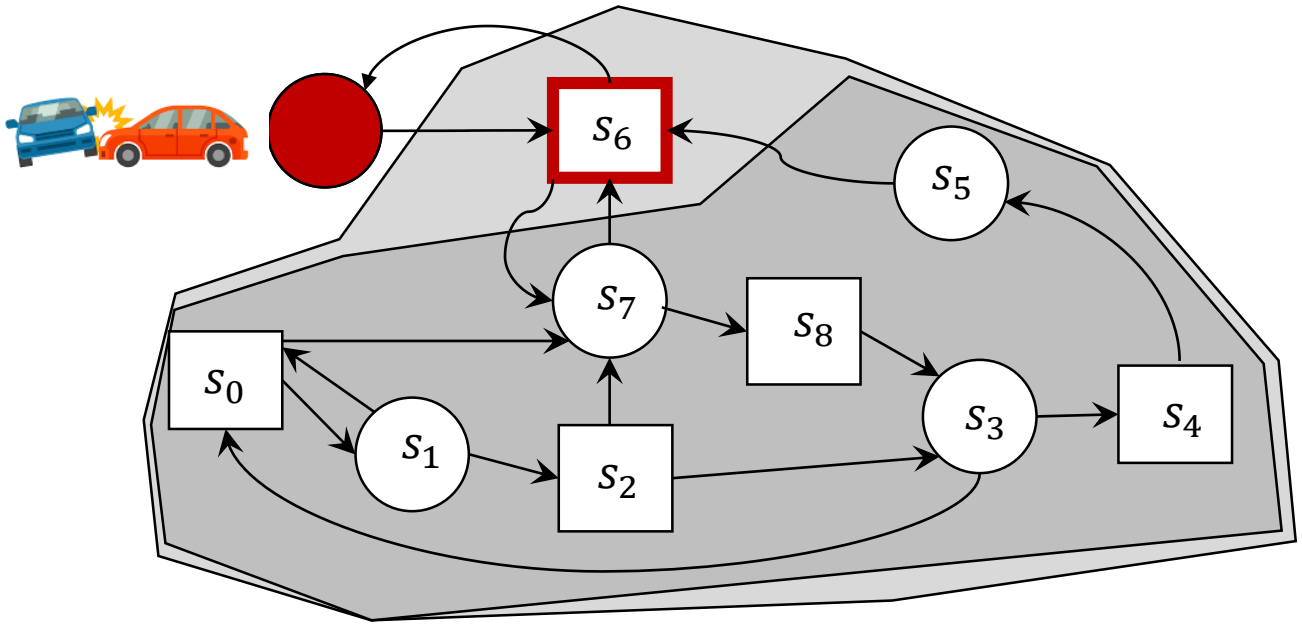


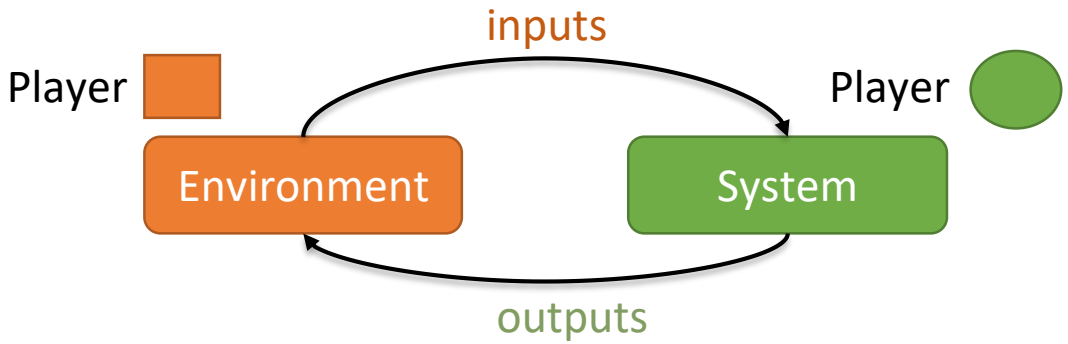


System Player wins, if  is **never** visited

Winning Region: States from which the system can enforce that  is **never** visited

Shield Construction – Synthesis is a Game

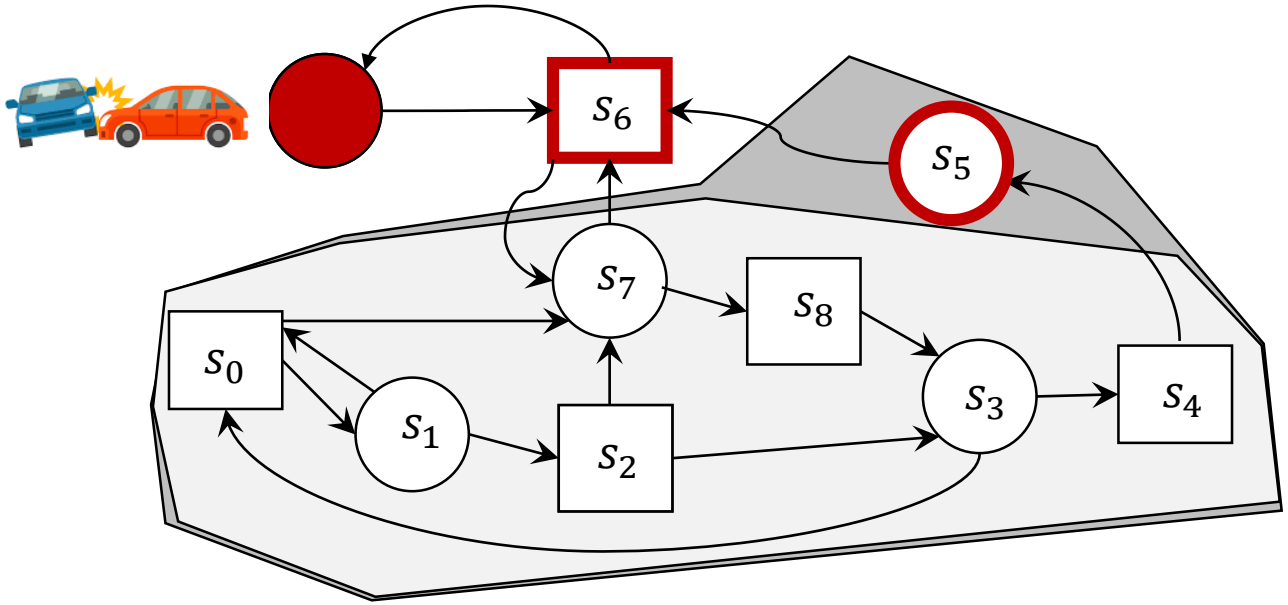


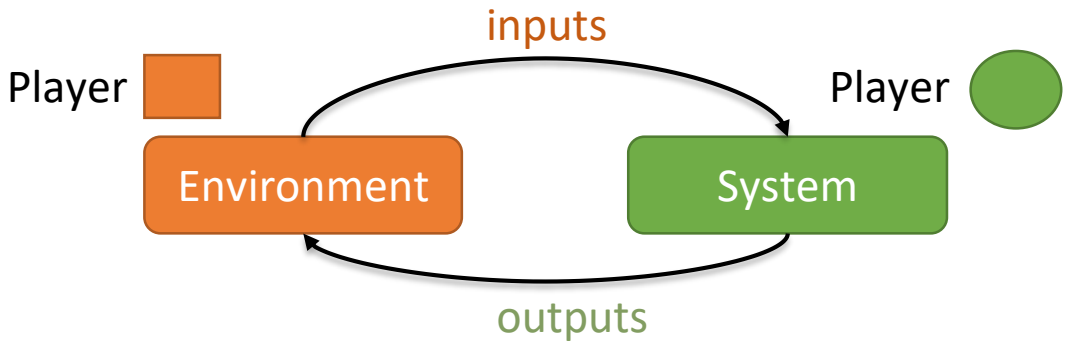


System Player wins, if  is **never** visited

Winning Region: States from which the system can enforce that  is **never** visited

Shield Construction – Synthesis is a Game

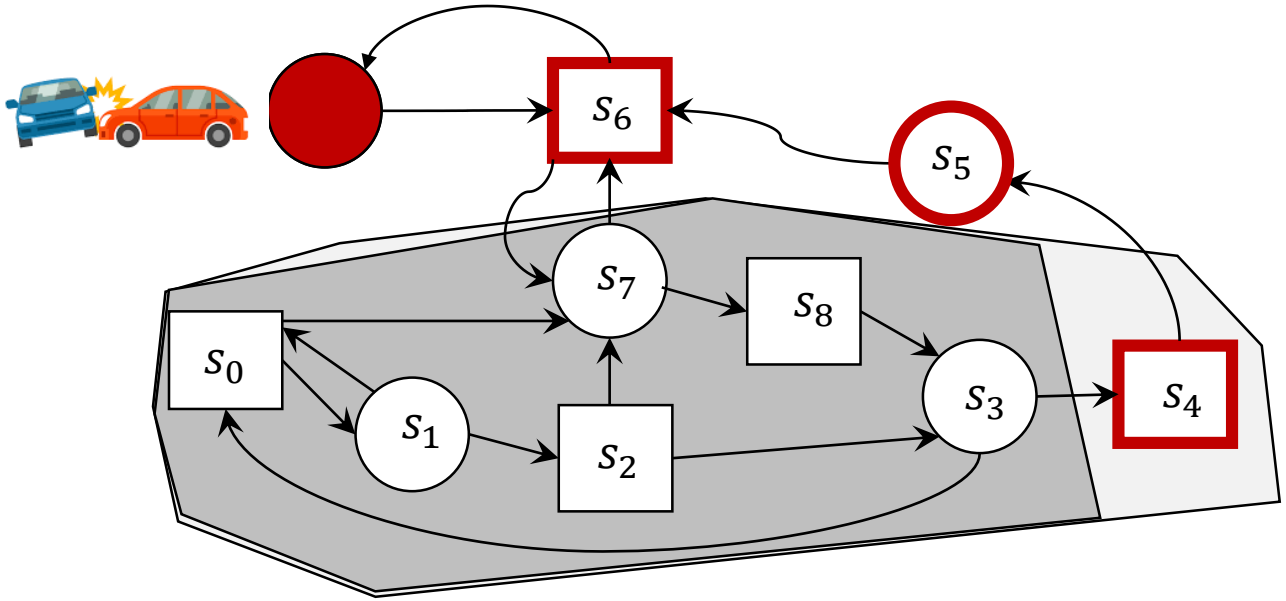


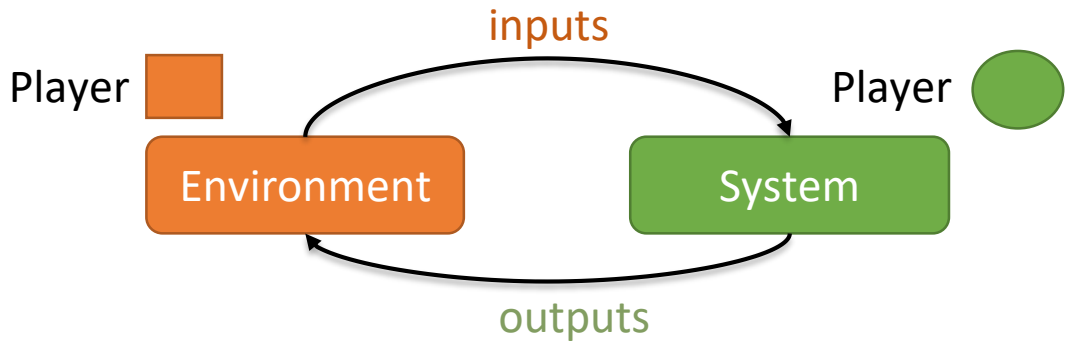


System Player wins, if  is **never** visited

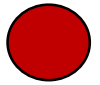
Winning Region: States from which the system can enforce that  is **never** visited

Shield Construction – Synthesis is a Game

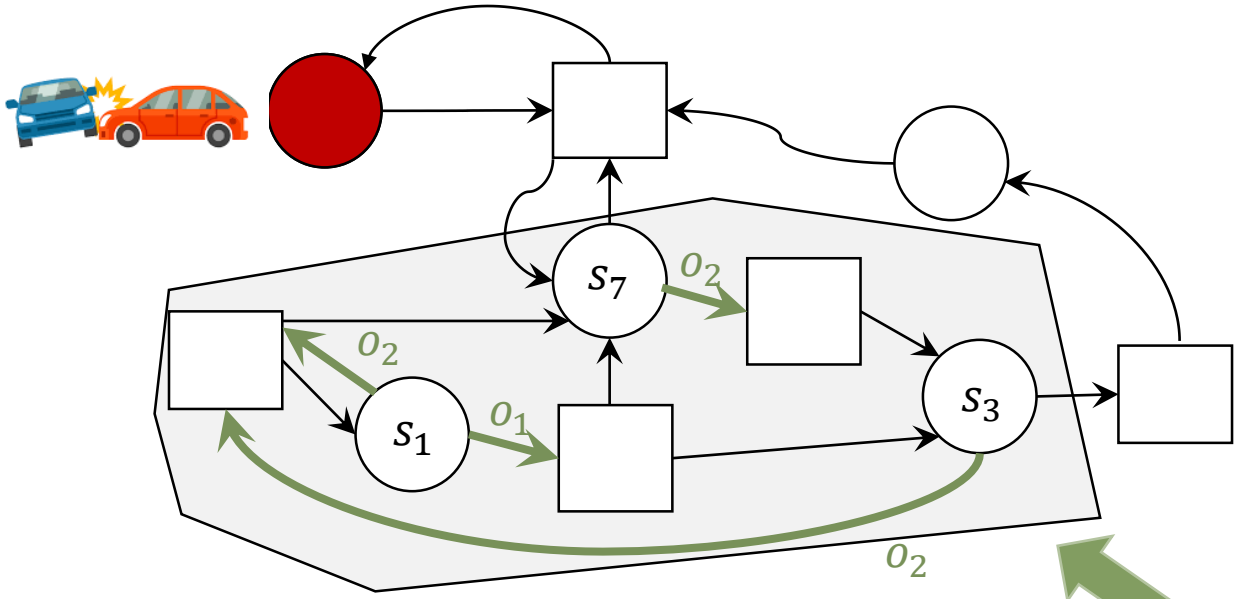


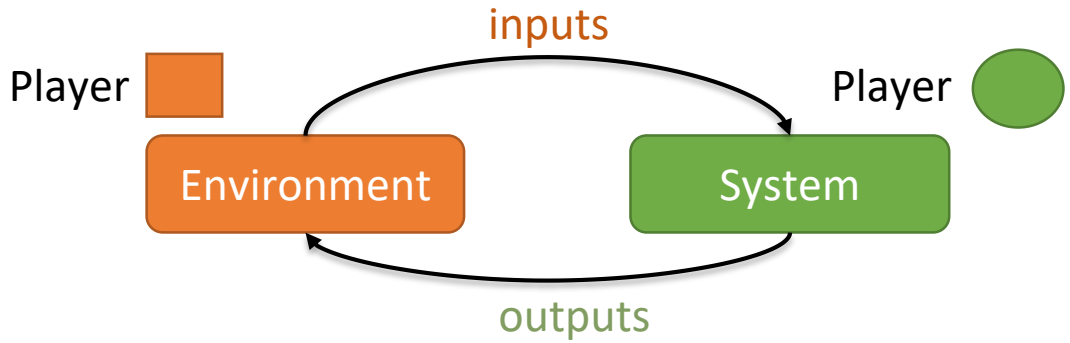


System Player wins, if  is **never** visited

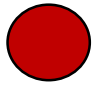
Winning Region: States from which the system can enforce that  is **never** visited

Shield Construction – Synthesis is a Game

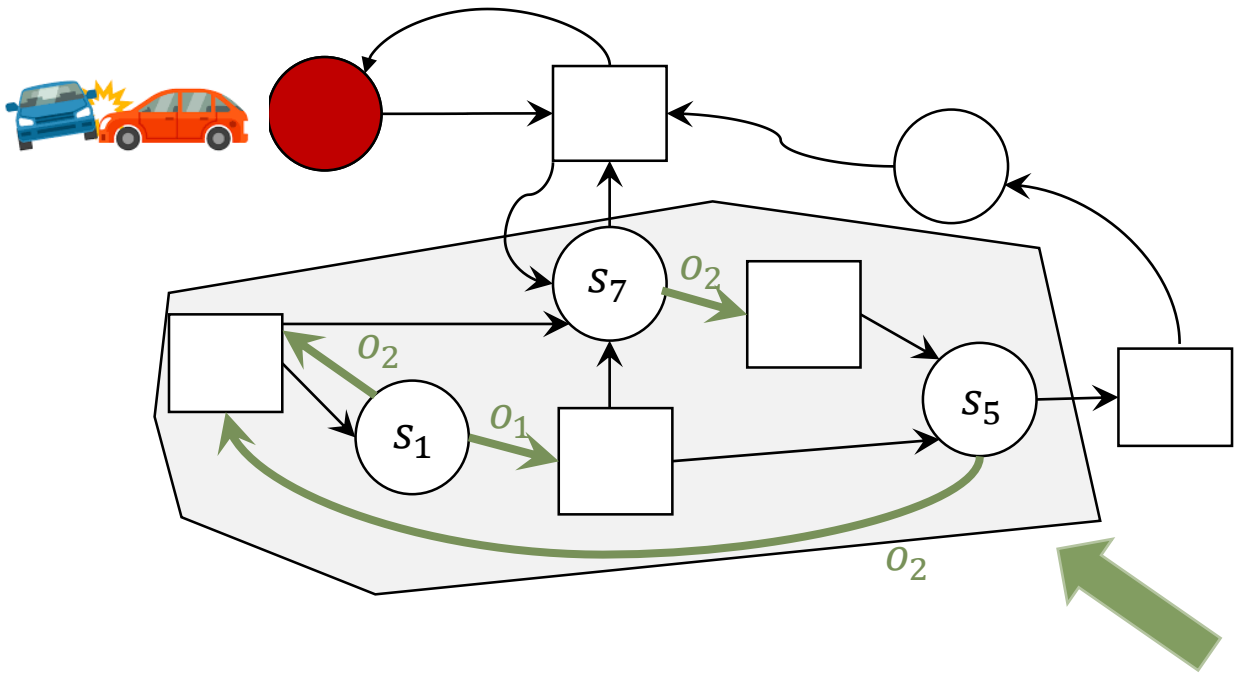




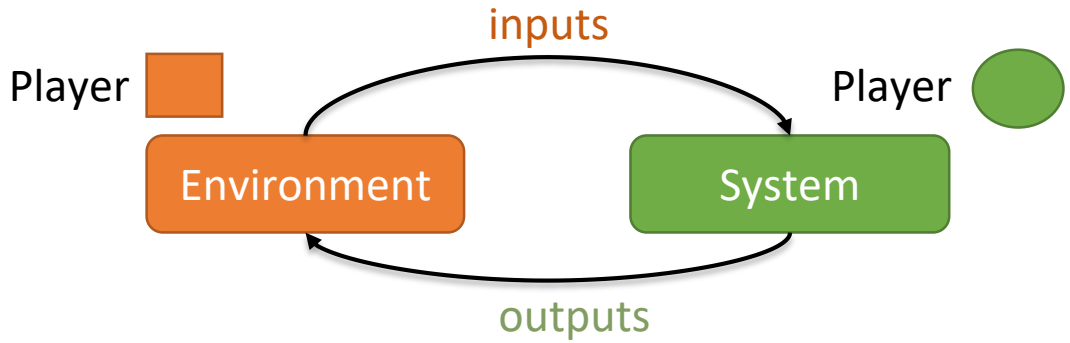
System Player wins, if  is **never** visited

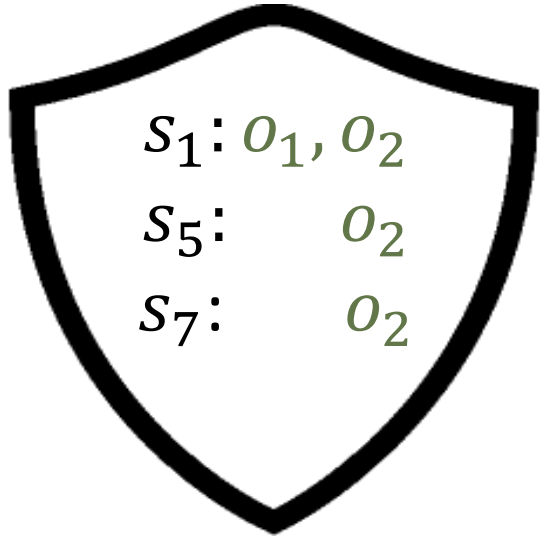
Winning Region: States from which the system can enforce that  is **never** visited

Shield Construction – Synthesis is a Game



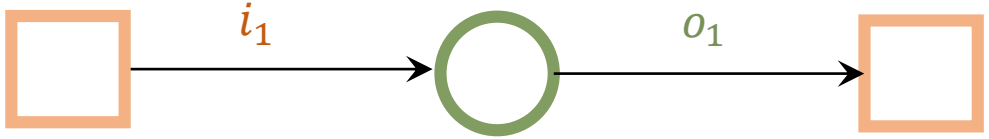
Winning Region



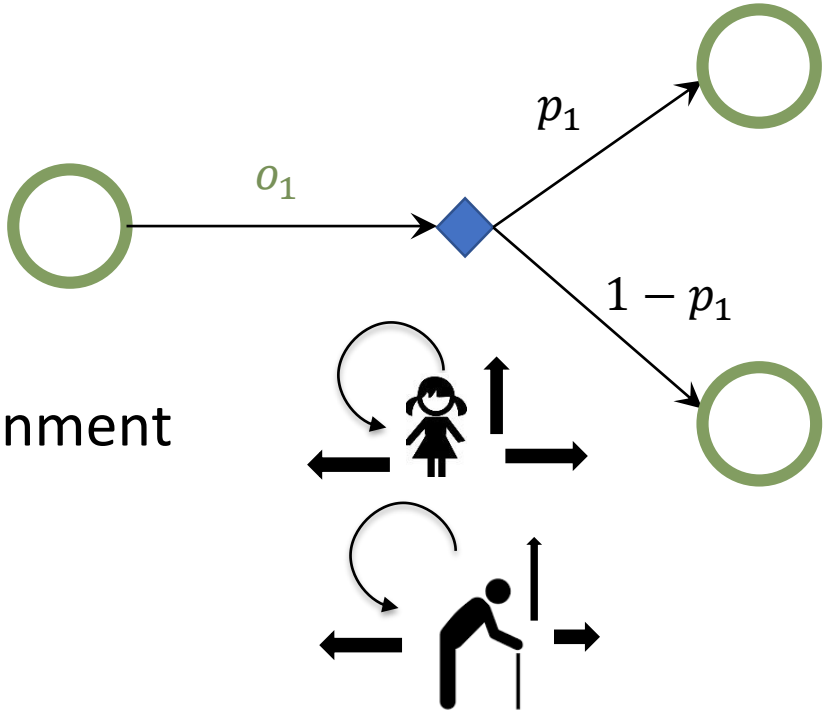


Different Types of Models

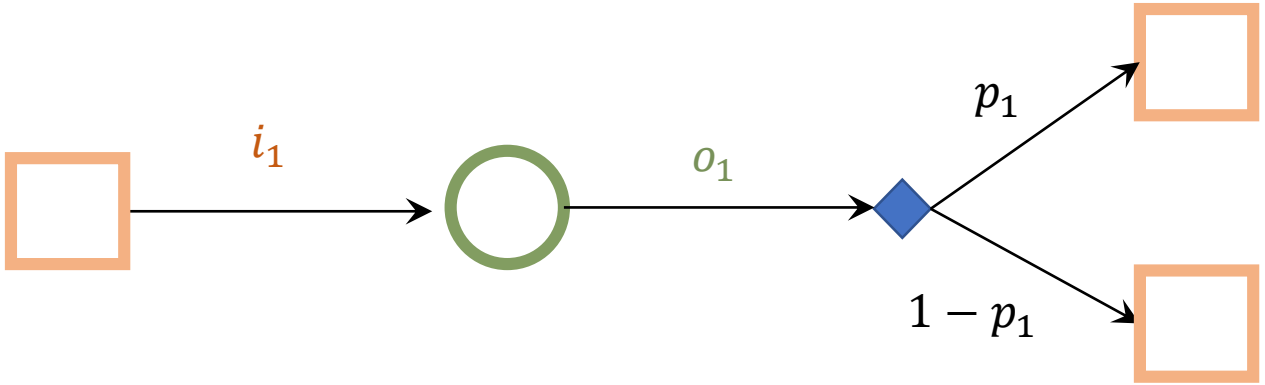
2 Player Game – adversarial environment



MDP – probabilistic environment



2 ½ Player Game – probabilistic & adversarial environment



N. Jansen, B. Könighofer, S. Junges, A. Serban, R. Bloem:
Safe Reinforcement Learning Using Probabilistic Shields. **CONCUR 2020**

Safety Shields for Probabilistic Environments

- Example: Stay safe in the next k steps
  - For all state-actions pairs: **Compute Safety-Value**:
  - $P_{max}(s,a) = P_{max}(T(s,a), G^{\leq k-1} safe)$

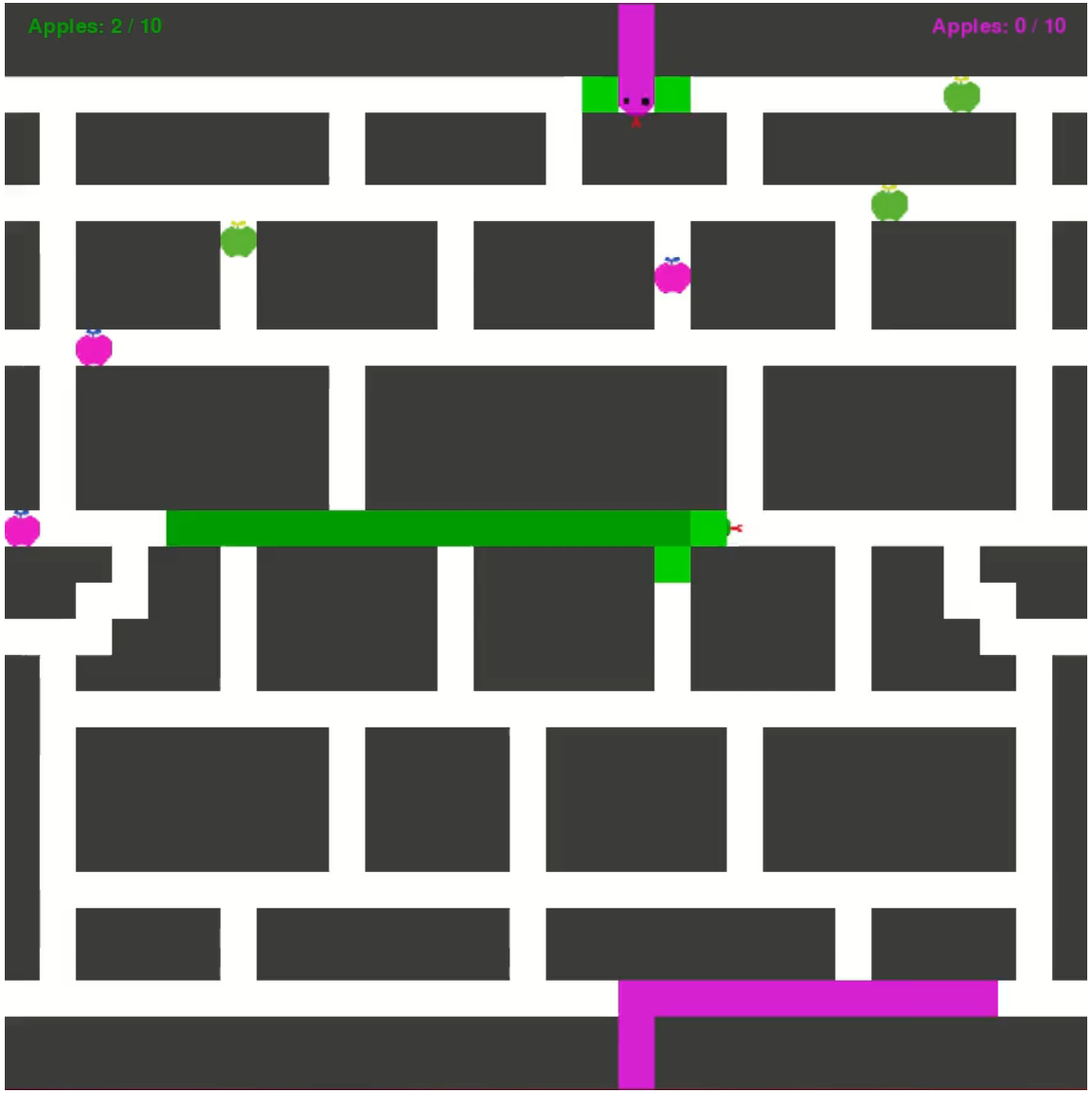
- **Absolute threshold** $\gamma \in [0,1]$
  - If $P_{max}(s,a) < \gamma \rightarrow a$ is shielded in s
  - **Not deadlock free!**

- **Relative threshold** $\lambda \in [0,1]$
  - If $P_{max}(s,a) < \lambda \cdot P_{max}(s, a_{opt}) \rightarrow a$ is shielded in s

Shielding parameters:
- Large γ *or* $\lambda \rightarrow$ strict shield
- Small γ *or* $\lambda \rightarrow$ permissive shield
- γ *and* λ can be changed on the fly

N. Jansen, B. Könighofer, S. Junges, A. Serban, R. Bloem:
Safe Reinforcement Learning Using Probabilistic Shields. **CONCUR 2020**

Video: Safety Shielding under Uncertainty



B. Könighofer, J. Rudolf, A. Palmisano, M. Tappler, R. Bloem:
Online Shielding for Stochastic Systems. **NFM 2021**

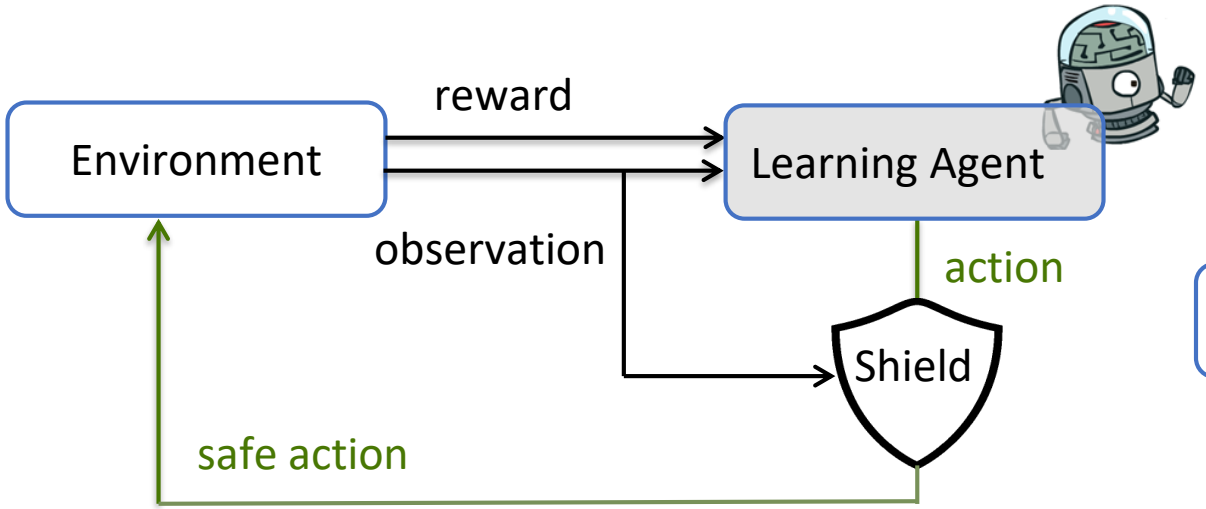
Tempest – *Shielding against the Storm*

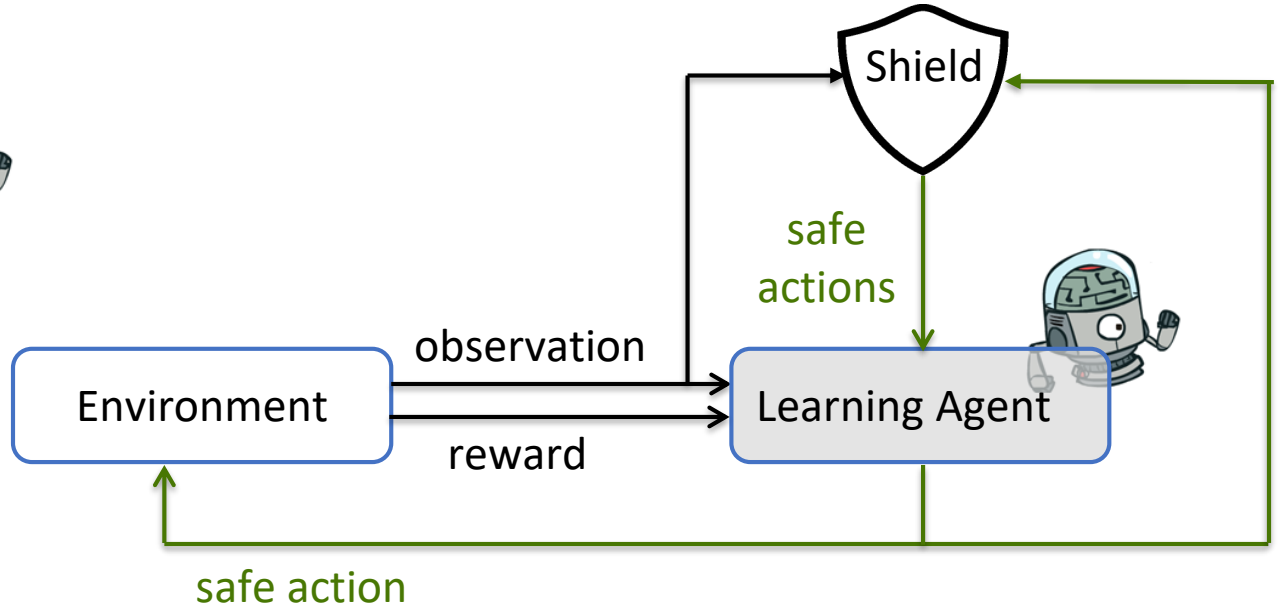
- **Synthesis tool for shields in probabilistic environments**
- **Extends model checker STORM**
  - TEMPEST is a stochastic game solver
  - Uses input language from Prism Games
- **Difference to Prism Games**
  - Solves Mean-Payoff Games without restrictions on the game graph
  - Provides most permissive strategies

https://tempest-synthesis.org/

S. Pranger, B. Könighofer, L. Posch, R. Bloem:
TEMPEST - Synthesis Tool for Reactive Systems and Shields in Probabilistic Environments. **ATVA 2021**

Pre and Post Safety Shielding





In Tempest:

$$\langle PostSafety, \gamma = 0.9 \rangle \langle shields \rangle P_{max=?}[G^{\leq 14} ! \, crash]$$

In Tempest:

$$\langle PreSafety, \lambda = 0.9 \rangle \langle shields \rangle P_{max=?}[G^{\leq 14} ! \, crash]$$

S. Pranger, B. Könighofer, L. Posch, R. Bloem:
TEMPEST - Synthesis Tool for Reactive Systems and Shields in Probabilistic Environments. **ATVA 2021**

Future Work: Explainable Shields

- Output from Tempest

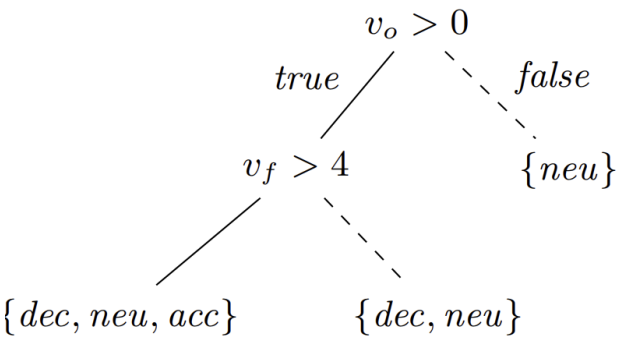
```
Post-Safety-Shield with relative comparison (lambda = 0.95):
 state_id [label]: 'forwarded actions' [<action_id> label: <forwarded_acti

0 [move=0 & x1=0 & y1=0 & x2=4  & y2=4]:  0{e}:0{e}; 1{s}:1{s}
3 [move=0 & x1=1 & y1=0 & x2=3  & y2=4]:  0{e}:2{w}; 2{w}:2{w}
4 [move=0 & x1=1 & y1=0 & x2=4  & y2=4]:  1{s}:3{n}; 3{n}:3{n}
    .....
```

Shields need to be explainable

- Represent shields as decision trees
- Use tool dtControl 🕮

v_o	v_f	d	actions
0	0	5	$\{neu\}$
2	6	10	$\{dec, neu, acc\}$
2	6	15	$\{dec, neu, acc\}$
4	4	15	$\{dec, neu\}$

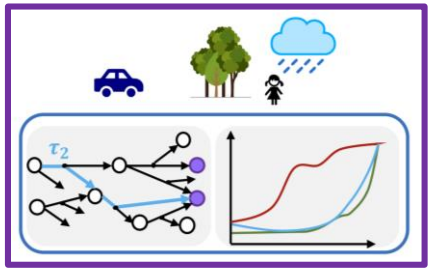


🕮 Pranav Ashok, Mathias Jackermeier, Jan Kretínský, Christoph Weinhuber, Maximilian Weininger, Mayank Yadav: dtControl 2.0: Explainable Strategy Representation via Decision Tree Learning Steered by Experts. TACAS 2021

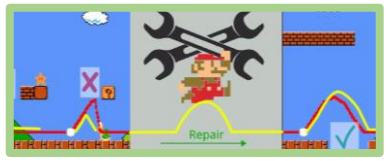
Outline

- Shielding for Safety
- Shielding for Fairness / Performance

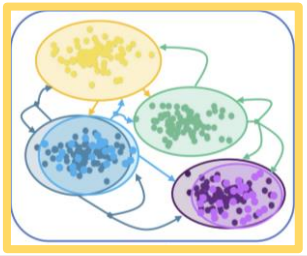
- Analyzing Evidence of Intentional Behavior



- Testing and Policy Repair



- Model Learning



Shielding for Performance/Fairness - Joint work with

University of Haifa

Radboud University Nijmegen

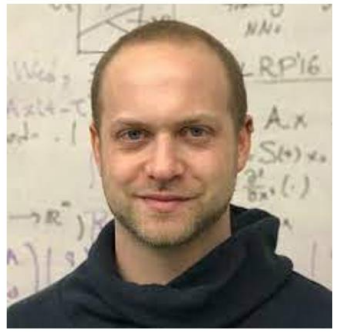
Stefan Pranger

Roderick Bloem

Martin Tappler

Guy Avni

Nils Jansen



Krishnendu Chatterjee

Thomas A. Henzinger

ISTA Institute of Science and Technology Austria

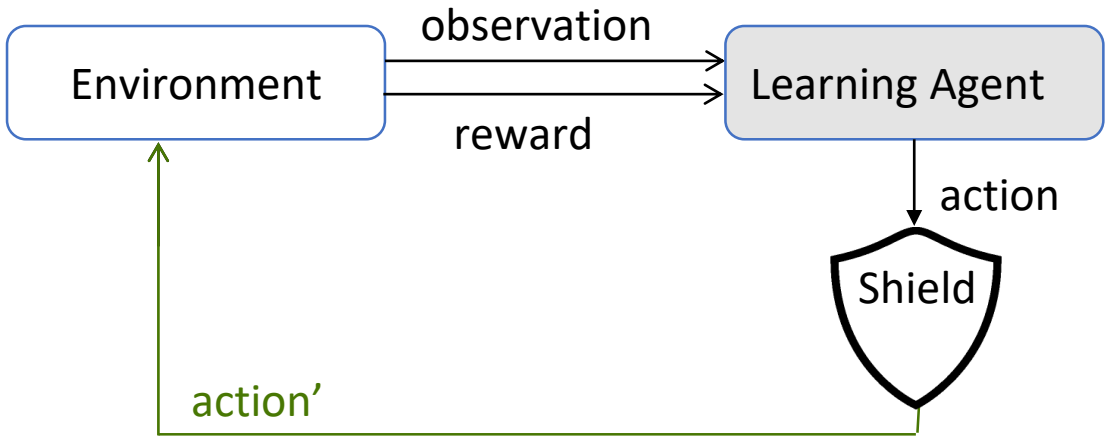
Shields for Performance / Fairness

- **Learned Controller: optimizes primary performance objective**

- **Other challenges than safety:**
  - *Optimize secondary objective / difficult to add new features*
  - *Robust performance, also on un-trained behavior*
  - *Local fairness*

G. Avni, R. Bloem, K. Chatterjee, T. A. Henzinger, B. Könighofer, S. Pranger:
Run-Time Optimization for Learned Controllers Through Quantitative Games. CAV 2019

Shields for Performance / Fairness



- **Two cost functions**
  - c_{PERF}: Performance objective of shield
  - c_{INTF}: Cost for interference

$$\lambda \cdot c_{PERF} + (1 - \lambda) \cdot c_{INTF}$$

Mean-Payoff Game, 2 Objectives

Mean-Payoff Game, 1 Objective

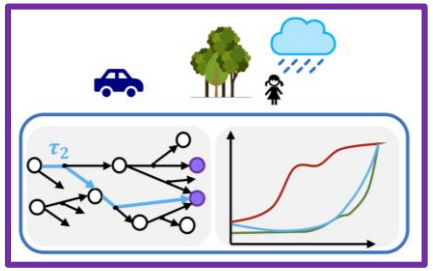
G. Avni, R. Bloem, K. Chatterjee, T. A. Henzinger, B. Könighofer, S. Pranger:
Run-Time Optimization for Learned Controllers Through Quantitative Games. CAV 2019

Video: Traffic Control

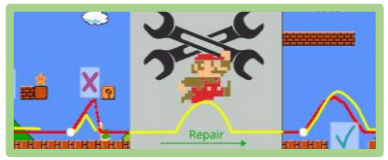
S. Pranger, B. Könighofer, M. Tappler, M. Deixelberger, N. Jansen, R. Bloem:
Adaptive Shielding under Uncertainty. ACC 2021

Outline

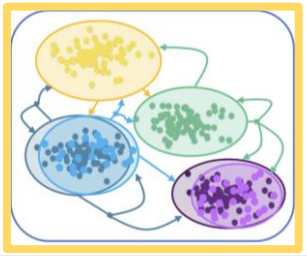
- Shielding for Safety
- Shielding for Fairness / Performance
- Analyzing Evidence of Intentional Behavior



- Testing and Policy Repair



- Model Learning



Analyzing Intentional Behavior

Ruzica Piskac

Scott Shapiro

Samuel Judson

Timos Antonopoulos

Filip Cano Cordoba

Yale University

F. Cano Córdoba, S. Judson, T. Antonopoulos, K. Bjørner, N. Shoemaker, S. J. Shapiro, R. Piskac, B. Könighofer: Analyzing Intentional Behavior in Autonomous Agents under Uncertainty**. IJCAI 2023**

Analyzing Intentional Behavior

- **Given**:
  - Model of scenario MDP M
  - Intention States S_I
  - Agent policy $\pi: S \rightarrow A$

- **Was the intention of the agent to reach S_I ?**
- Under perfect knowledge:

If the intention of the agent is to reach S_I,
then π maximizes the probability of reaching S_I.

F. Cano Córdoba, S. Judson, T. Antonopoulos, K. Bjørner, N. Shoemaker, S. J. Shapiro, R. Piskac, B. Könighofer:
Analyzing Intentional Behavior in Autonomous Agents under Uncertainty**. IJCAI 2023**

Analyzing Evidence of Intentional Behavior

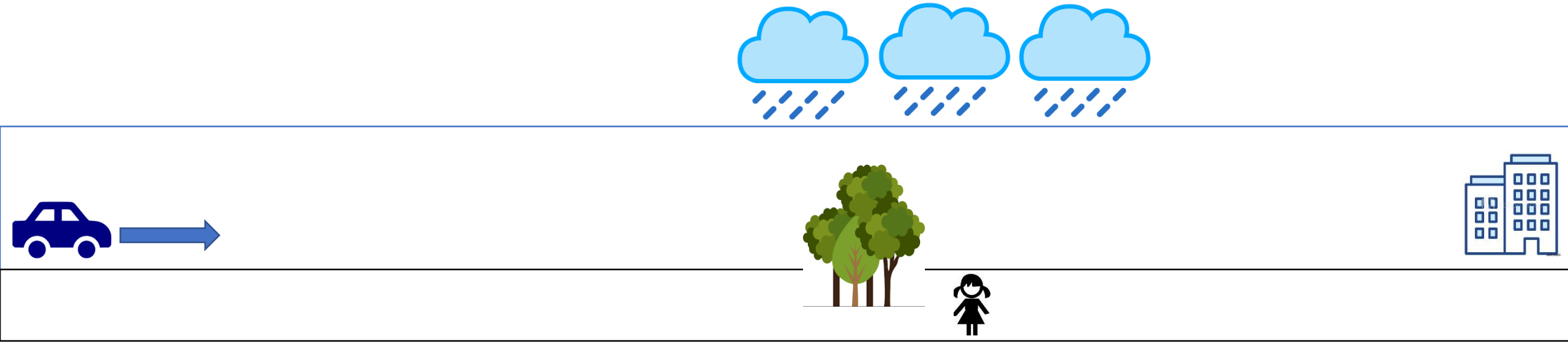
- **Given**:
  - Model of scenario (MDP M)
  - Intention (States S_I)
  - Agent (policy $\pi: S \rightarrow A$)

- **Is there evidence of intentional behavior towards reaching S_I ?**

Compare π with most-optimal und least-optimal policy for achieving S_I.

F. Cano Córdoba, S. Judson, T. Antonopoulos, K. Bjørner, N. Shoemaker, S. J. Shapiro, R. Piskac, B. Könighofer: Analyzing Intentional Behavior in Autonomous Agents under Uncertainty**. IJCAI 2023**

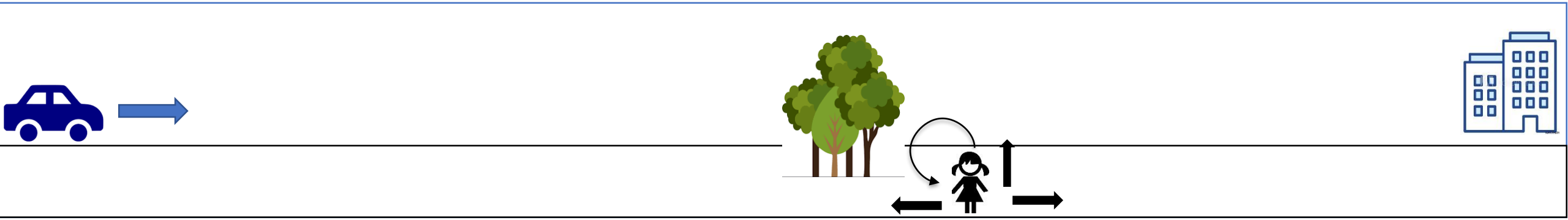
Analyzing Evidence of Intentional Behavior

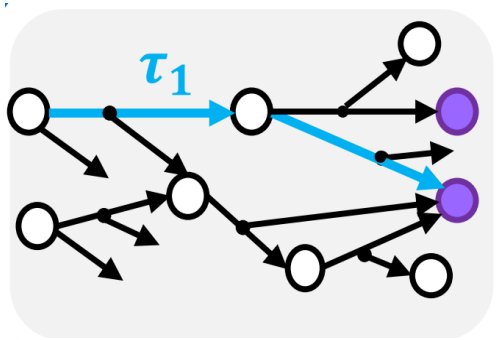


- Did the agent intentionally cause the harm?
- Analyse actions picked from the agent
- Compare with most-responsible und unsafest strategy

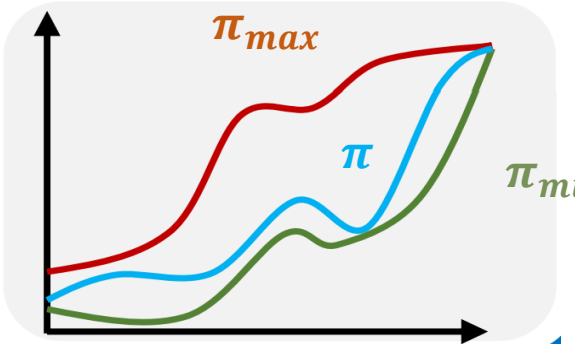
F. Cano Córdoba, S. Judson, T. Antonopoulos, K. Bjørner, N. Shoemaker, S. J. Shapiro, R. Piskac, B. Könighofer: Analyzing Intentional Behavior in Autonomous Agents under Uncertainty**. IJCAI 2023**

Analyzing Evidence of Intentional Behavior





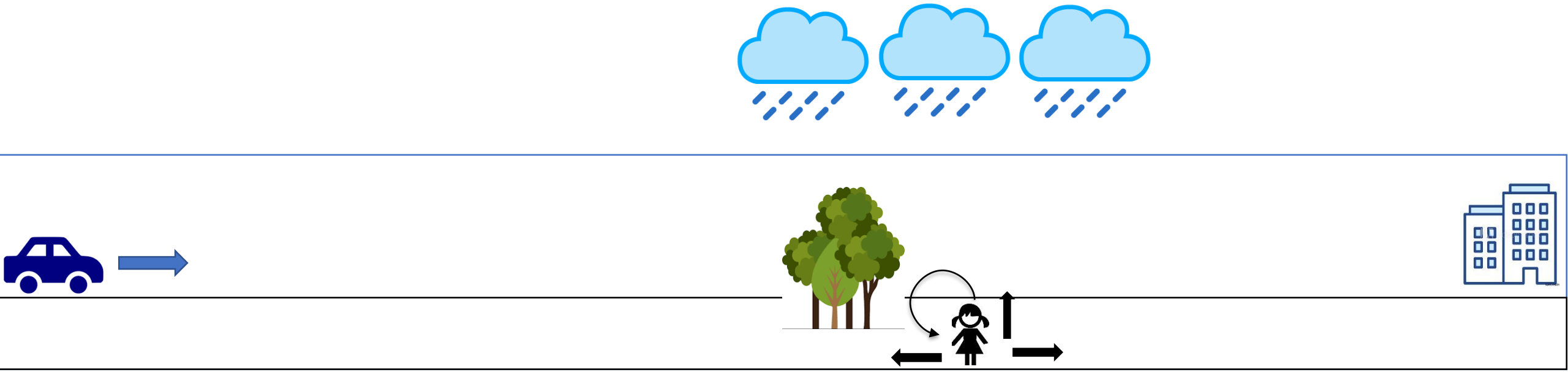
Probability of accident

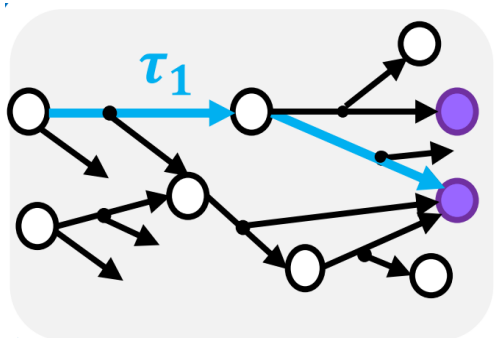


Time before accident

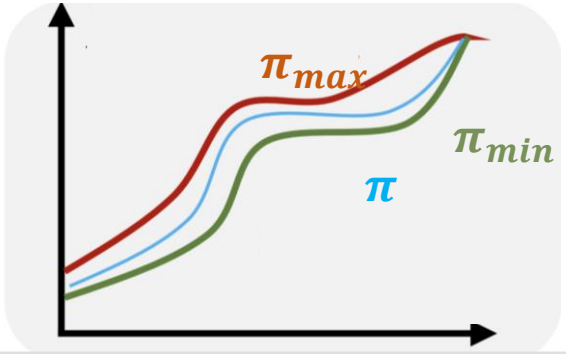
F. Cano Córdoba, S. Judson, T. Antonopoulos, K. Bjørner, N. Shoemaker, S. J. Shapiro, R. Piskac, B. Könighofer: Analyzing Intentional Behavior in Autonomous Agents under Uncertainty**. IJCAI 2023**

Analyzing Evidence of Intentional Behavior





Probability of accident

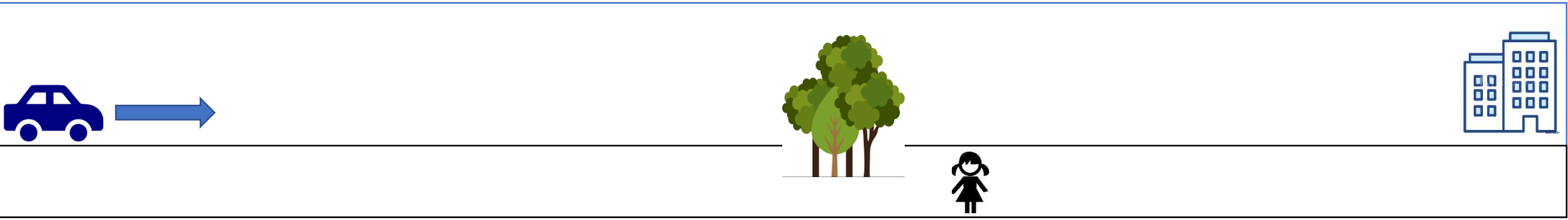


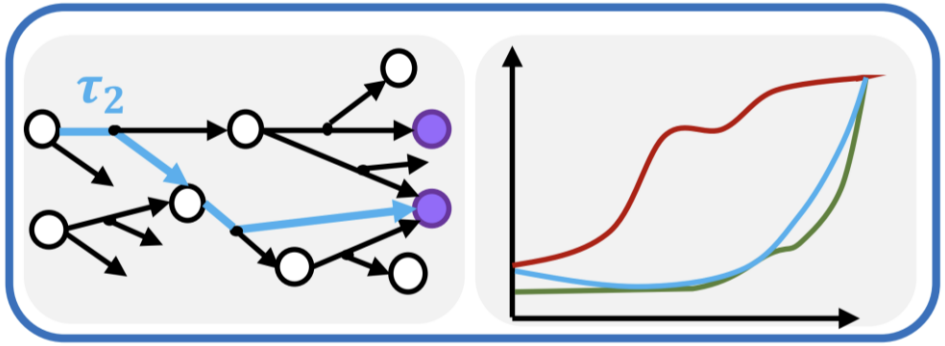
Time before accident

F. Cano Córdoba, S. Judson, T. Antonopoulos, K. Bjørner, N. Shoemaker, S. J. Shapiro, R. Piskac, B. Könighofer: Analyzing Intentional Behavior in Autonomous Agents under Uncertainty**. IJCAI 2023**

Analyze Counterfactuals

- “What if it would have been sunny?”

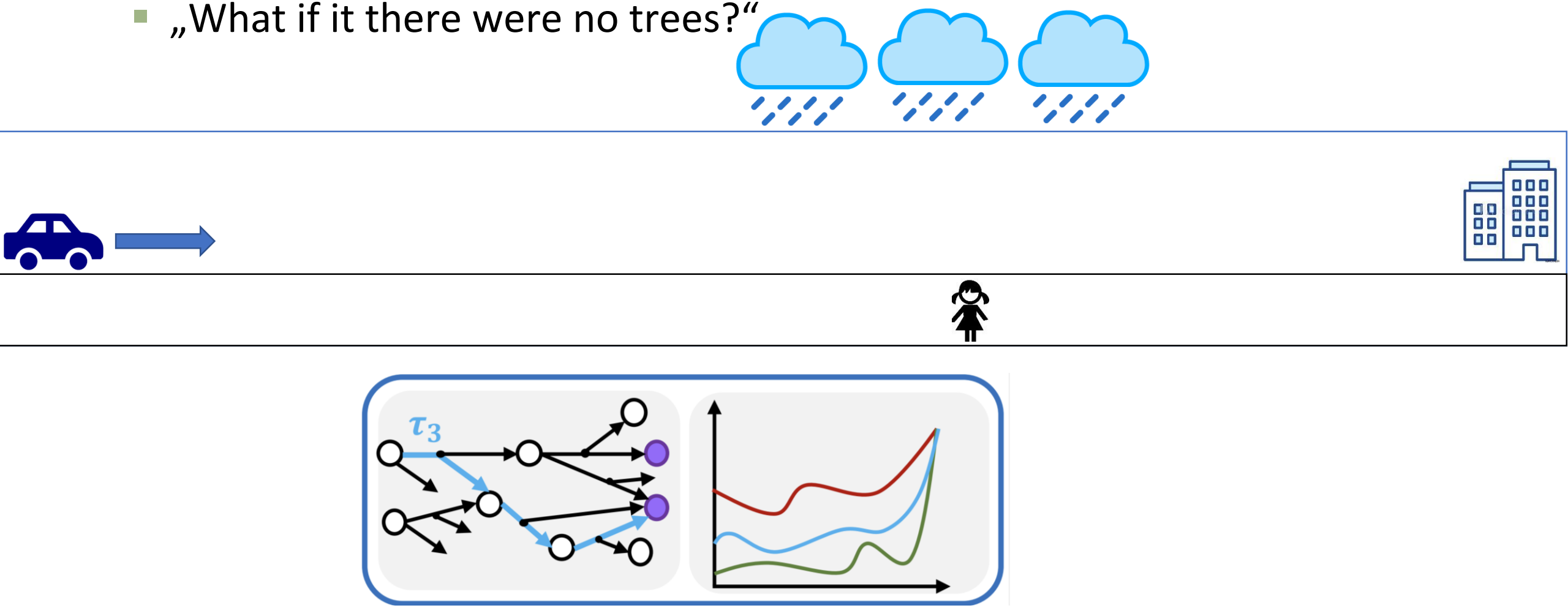




F. Cano Córdoba, S. Judson, T. Antonopoulos, K. Bjørner, N. Shoemaker, S. J. Shapiro, R. Piskac, B. Könighofer: Analyzing Intentional Behavior in Autonomous Agents under Uncertainty**. IJCAI 2023**

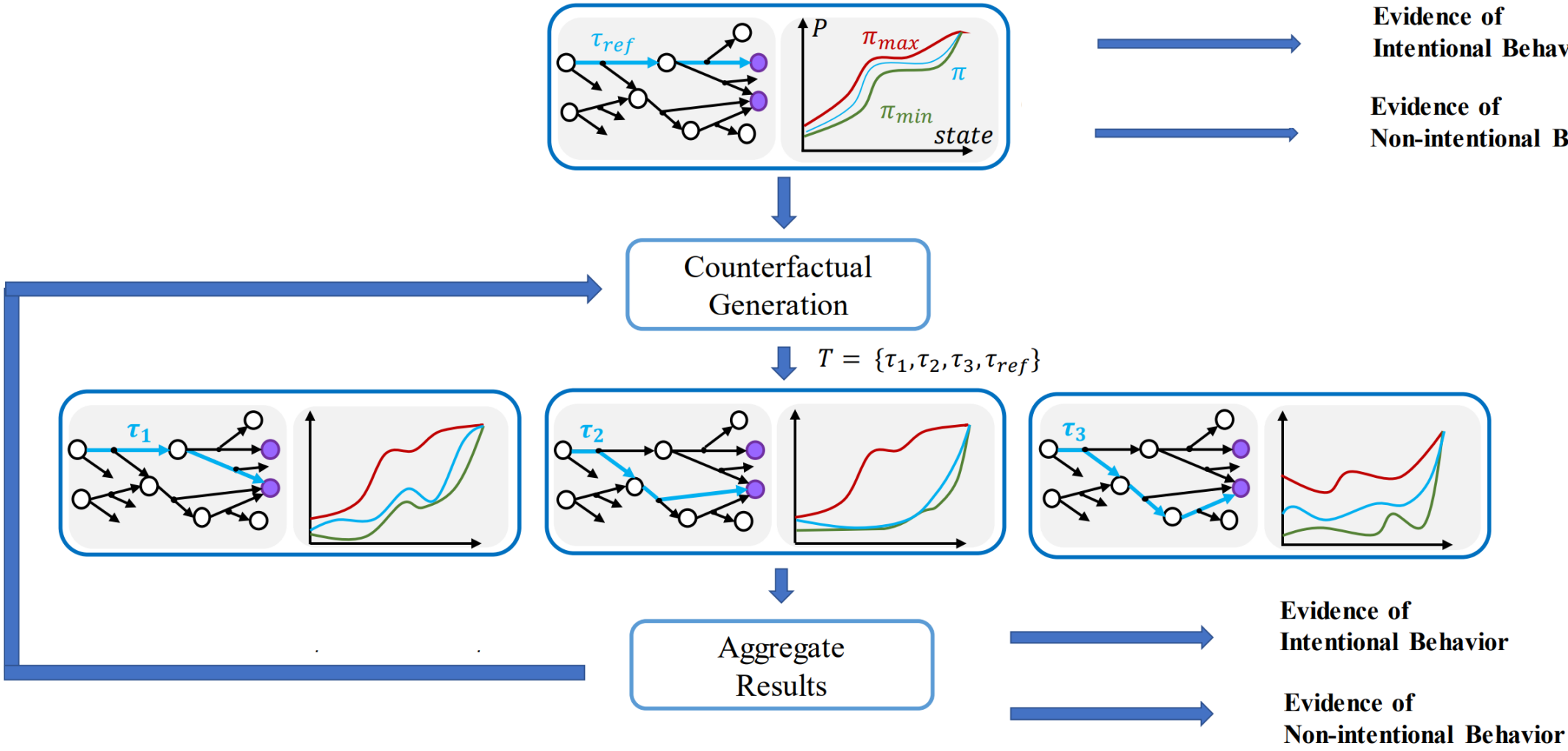
Analyze Counterfactuals

- „What if it there were no trees?“



F. Cano Córdoba, S. Judson, T. Antonopoulos, K. Bjørner, N. Shoemaker, S. J. Shapiro, R. Piskac, B. Könighofer: Analyzing Intentional Behavior in Autonomous Agents under Uncertainty**. IJCAI 2023**

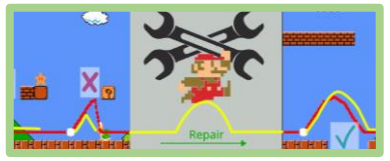
Analyzing Intentional Behvior



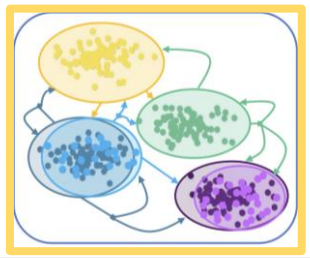
F. Cano Córdoba, S. Judson, T. Antonopoulos, K. Bjørner, N. Shoemaker, S. J. Shapiro, R. Piskac, B. Könighofer: Analyzing Intentional Behavior in Autonomous Agents under Uncertainty**. IJCAI 2023**

Outline

- Shielding for Safety
- Shielding for Fairness / Performance
- Analyzing Evidence of Intentional Behavior
- Testing and Policy Repair



- Model Learning



Learning and Repair of Deep RL Policies

Martin Tappler

Bernhard Aichernig

Andrea Pferscher

Filip Can Cordoba

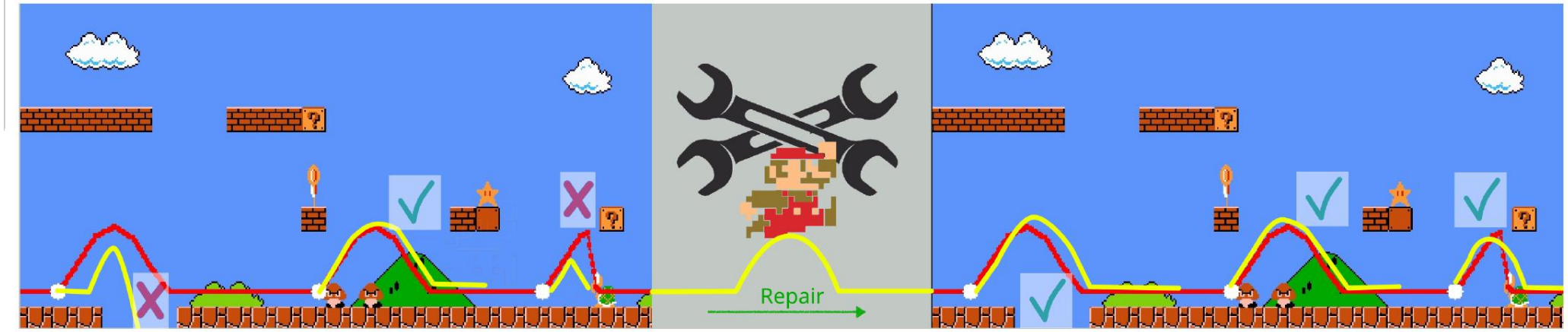
M. Tappler, A. Pferscher, B. Aichernig, B. Könighofer:
Learning and Repair of Deep Reinforcement Learning Policies from Fuzz-Testing Data. Under Submission

M. Tappler, F. Cano Córdoba, B. Aichernig, B. Könighofer:
Search-Based Testing of Reinforcement Learning. IJCAI 2022

Learning and Repair of Deep RL Policies

- **Classical Software Development**
  - Write code, testing/debugging, fix code, testing/debugging…

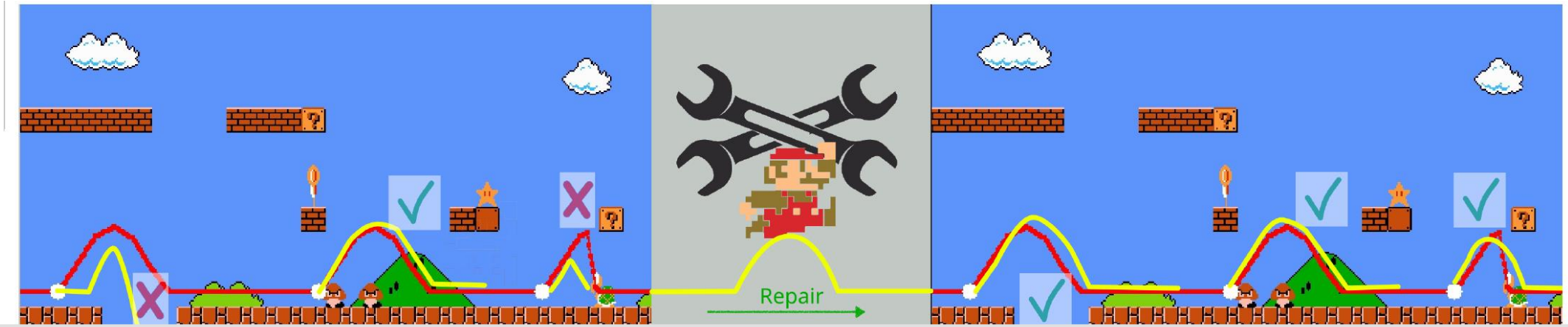
- **Classical Development of RL Agents**
  - Train it, test it, start training from scratch, test it, start training from scratch…



M. Tappler, A. Pferscher, B. Aichernig, B. Könighofer:
Learning and Repair of Deep Reinforcement Learning Policies from Fuzz-Testing Data. Under Submission

Learning and Repair of Deep RL Policies

- **Wouldn't it be better to also have a cycle?**
  - Train
  - Test
  - Repair Policy
  - Test
  - Repair Policy….



M. Tappler, A. Pferscher, B. Aichernig, B. Könighofer:
Learning and Repair of Deep Reinforcement Learning Policies from Fuzz-Testing Data. Under Submission

Development Cycle using Fuzzing

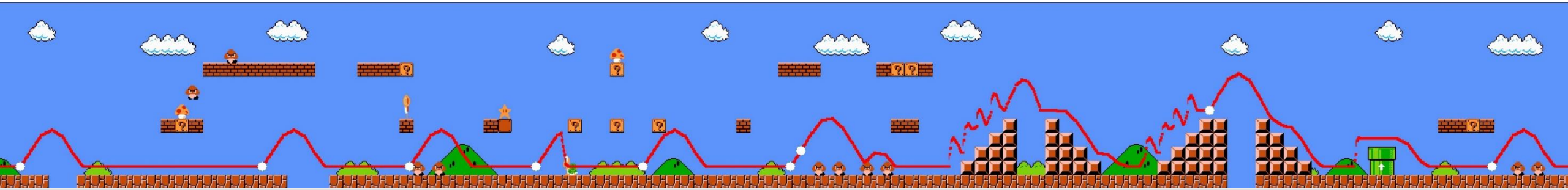
- **Step 1: Train the agent**
  - Effectively train RL agent via RLfD
  - Compute demonstrations automatically

M. Tappler, A. Pferscher, B. Aichernig, B. Könighofer:
Learning and Repair of Deep Reinforcement Learning Policies from Fuzz-Testing Data. Under Submission

Development Cycle using Fuzzing

- **Step 1: Train the agent**
  - Effectively train RL agent via RLfD
  - Compute demonstrations automatically

- (a) Search for reference demonstration (DFS)



M. Tappler, A. Pferscher, B. Aichernig, B. Könighofer:
Learning and Repair of Deep Reinforcement Learning Policies from Fuzz-Testing Data. Under Submission

Development Cycle using Fuzzing

- **Step 1: Train the agent**
  - Effectively train RL agent via RLfD
  - Compute demonstrations automatically

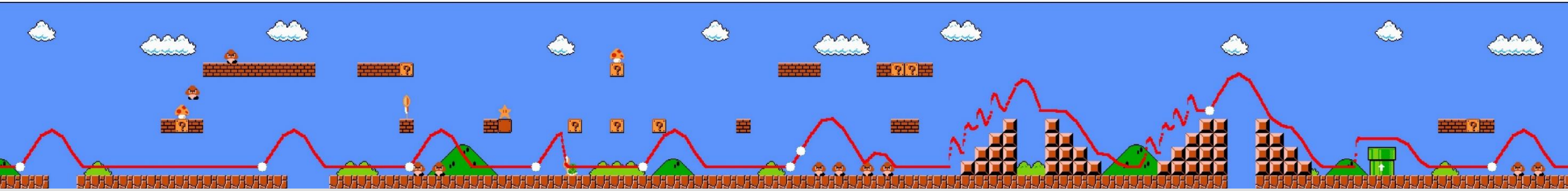
- (a) Search for reference demonstration (DFS)
- (b) Fuzz diverse set of demonstrations
- (c) Use demonstrations for RLfD

M. Tappler, A. Pferscher, B. Aichernig, B. Könighofer:
Learning and Repair of Deep Reinforcement Learning Policies from Fuzz-Testing Data. Under Submission

Development Cycle using Fuzzing

- **Step 1: Train the agent**
- **Step 2: Test the agent**

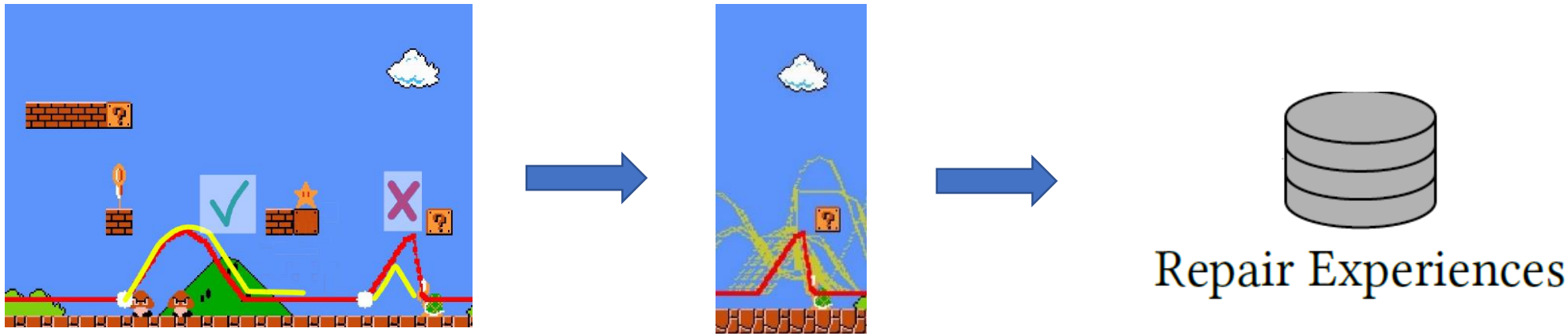
- **Search reveals critical situations**
  - DFS backtracks when reaching an unsafe state
  - Test states along reference demonstration to which the DFS backtracked



M. Tappler, F. Cano Córdoba, B. Aichernig, B. Könighofer:
Search-Based Testing of Reinforcement Learning. IJCAI 2022

Development Cycle using Fuzzing

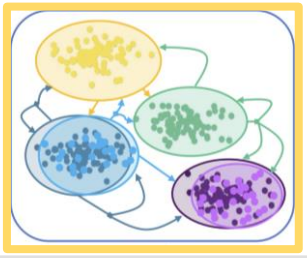
- **Step 1: Train the agent**
- **Step 2: Test the agent**
- **Step 3: Repair**
  - Collect examples of correct behavior near detected faulty states
  - Apply RLfD with repair experiences



M. Tappler, A. Pferscher, B. Aichernig, B. Könighofer:
Learning and Repair of Deep Reinforcement Learning Policies from Fuzz-Testing Data. Under Submission

Outline

- Shielding for Safety
- Shielding for Fairness / Performance
- Analyzing Evidence of Intentional Behavior
- Testing and Policy Repair
- Model Learning



Learning Environmental Models

Martin Tappler

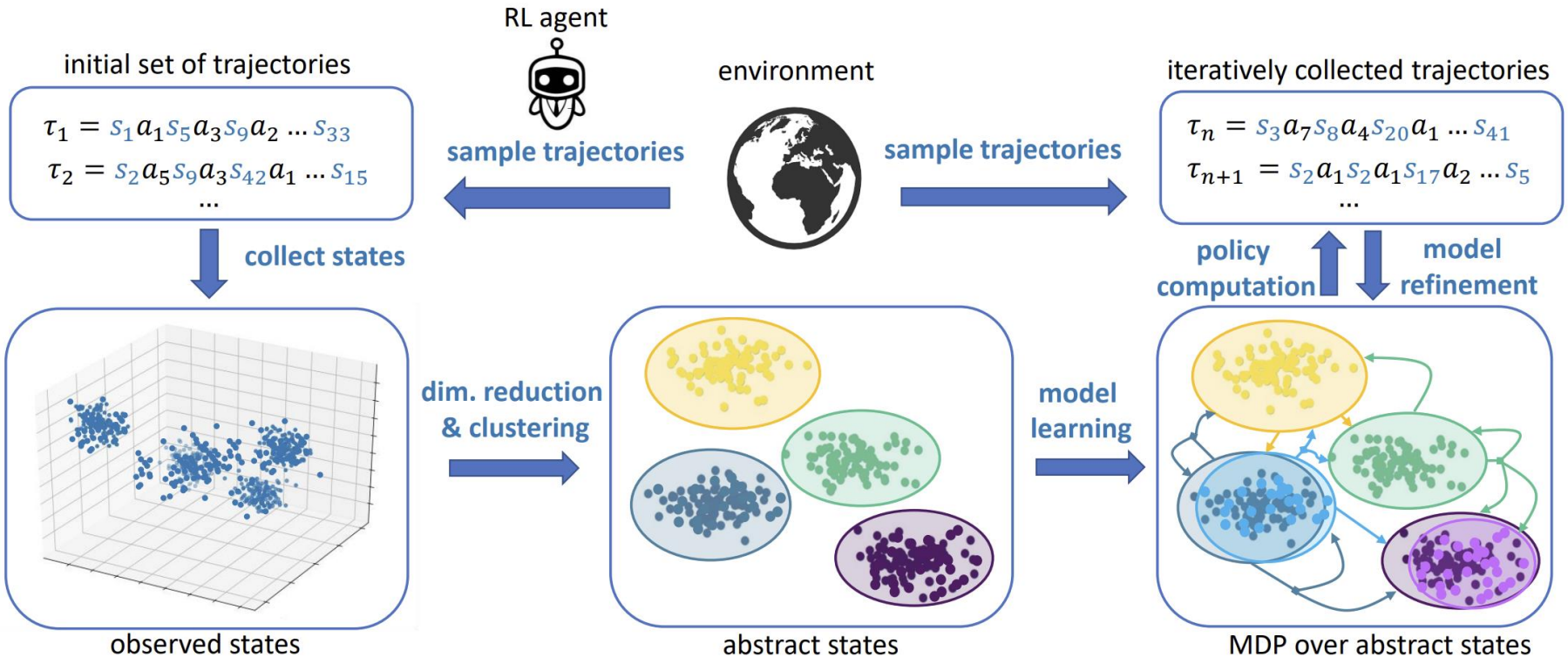
Bernhard Aichernig

Edi Muskardin

M. Tappler, E. Muskardin, B. Aichernig, B. Könighofer:
Learning Environment Models with Continuous Stochastic Dynamics. Under Submission

Learning Environmental Models

- Getting a good *model* is essential.



M. Tappler, E. Muskardin, B. Aichernig, B. Könighofer:
Learning Environment Models with Continuous Stochastic Dynamics. Under Submission

Formal methods are great for learned systems

- If you have a nice model

If you have a model, we can use it for

- Testing for robust performance and safety
- Monitoring / enforcement
- Explainability
- Accountability
- ….