Filip Cano Córdoba

Towards Responsible AI: Advances in Safety, Fairness, and Accountability of Autonomous Systems

DOCTORAL THESIS

to achieve the university degree of

Doctor of Technical Sciences

submitted to

Graz University of Technology

Assessors

Advisor and examiner
Prof. Roderick Bloem
Graz University of Technology

Examiner
Prof. Ruzica Piskac
Yale University

Graz, March 2025

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

Date, Signature

Abstract

Ensuring responsible use of artificial intelligence (AI) has become imperative as autonomous systems increasingly influence critical societal domains. However, the concept of trustworthy AI remains broad and multi-faceted. This thesis advances knowledge in the safety, fairness, transparency, and accountability of AI systems.

In safety, we extend classical deterministic shielding techniques to become resilient against delayed observations, enabling practical deployment in real-world conditions. We also implement both deterministic and probabilistic safety shields into simulated autonomous vehicles to prevent collisions with road users, validating the use of these techniques in realistic driving simulators.

We introduce fairness shields, a novel post-processing approach to enforce group fairness in sequential decision-making settings over finite and periodic time horizons. By optimizing intervention costs while strictly ensuring fairness constraints, this method efficiently balances fairness with minimal interference.

For transparency and accountability, we propose a formal framework for assessing intentional behaviour in probabilistic decision-making agents, introducing quantitative metrics of agency and intention quotient. We use these metrics to propose a retrospective analysis of intention, useful for determining responsibility when autonomous systems cause unintended harm.

Finally, we unify these contributions through the "reactive decision-making" framework, providing a general formalization that consolidates previous approaches. Collectively, the advancements presented contribute practically to the realization of safer, fairer, and more accountable AI systems, laying the foundations for future research in trustworthy AI.

Kurzfassung

Die Sicherstellung eines verantwortungsvollen Umgangs mit Künstlicher Intelligenz (KI) ist unabdingbar geworden, da autonome Systeme zunehmend kritische gesellschaftliche Bereiche beeinflussen. Dennoch bleibt das Konzept vertrauenswürdiger KI breit gefächert und facettenreich. Diese Dissertation erweitert das Wissen über Sicherheit, Fairness, Transparenz und Rechenschaftspflicht von KI-Systemen.

Im Bereich der Sicherheit erweitern wir klassische deterministische Shielding-Techniken, sodass sie auch gegenüber verzögerten Beobachtungen widerstandsfähig sind. Dadurch ermöglichen wir deren praktischen Einsatz unter realistischen Bedingungen. Zudem implementieren wir sowohl deterministische als auch probabilistische Sicherheitsschilde in simulierte autonome Fahrzeuge, um Kollisionen mit Verkehrsteilnehmern zu verhindern, und validieren so den Einsatz dieser Techniken in realitätsnahen Fahrsimulatoren.

Wir führen Fairness-Schilde ein, einen neuartigen Post-Processing-Ansatz zur Durchsetzung von Gruppenfairness in sequenziellen Entscheidungssituationen über endliche und periodische Zeithorizonte. Durch die Optimierung der Interventionskosten bei strikter Einhaltung von Fairness-Beschränkungen ermöglicht diese Methode eine effiziente Balance zwischen Fairness und minimalem Eingriff.

Für Transparenz und Rechenschaftspflicht schlagen wir einen formalen Rahmen zur Bewertung intentionalen Verhaltens bei probabilistischen Entscheidungsagenten vor und führen quantitative Maße für Handlungsfähigkeit (Agency) und Intentionsquotienten ein. Diese Maße nutzen wir für eine retrospektive Analyse der Absicht, die hilfreich ist, um Verantwortung festzustellen, wenn autonome Systeme unbeabsichtigte Schäden verursachen.

Schließlich vereinigen wir diese Beiträge im Rahmen der "reaktiven Entscheidungsfindung" und bieten so eine allgemeine Formalisierung, die bisherige Ansätze integriert. Die vorgestellten Fortschritte leisten insgesamt einen praktischen Beitrag zur Realisierung sicherer, fairer und verantwortungsvollerer KI-Systeme und bilden eine Grundlage für zukünftige Forschung zu vertrauenswürdiger KI.

Acknowledgements

This work would not have been possible without the help and support of so many people.

A very special thanks to my parents for their unconditional love and support, my grandmothers Lina and Ana, and my surrogate grandmother Isa, who raised me to be the person I am today. I also want to thank the rest of my family, Dani, Carmen, Juan, Maria, Toni, and Antonia, for being there and reminding me where I come from.

I thank all the people I had the pleasure and privilege to collaborate all these years. A special thanks to Sam, Timos, and Katrine for their patient discussions; Kaushik and Konstantin for their motivation and the long work hours; Haritz for his hard work and kind soul; and Scott, Tom, Martin, Oliver, and Ruzica for the inspiration to do great research.

Thanks to all the friends and colleagues in Graz, whose daily presence and support has been very valuable: Malte, Sonja, Masoud, Vedad, Benedikt, Johannes; and a very special appreciation to Stefan, with whom I've had the pleasure to share a working space, to work and learn (and sometimes complain about life) together. For the time I spent in Gössendorf, thanks to Ben and Niki for so many joyful moments spent together, to Alex and Irmi for the wine and the kind words, and thanks to Erika for her unending enthusiasm.

Thanks to all my friends who have had the patience to keep and feed our friendship from the distance, specially Ander, Zaira, and Irene for reminding me what is important in life; Maribel, Cristina, and Saïd for the random walks; Marc for being simply amazing; Susana for being weird together; Katia and Genís for their enormous heart; Tania for her resilience. A very special thanks to Alberto, without whom I would not have come to Graz, and without whom I may not have managed to finish. Thanks for being sunshine in spring and a lighthouse in my darkest hours.

None of this would have been possible without my teachers, who taught me everything I know and inspired me to learn more. From the high school teachers at Lestonnac and Sant Pere, who inspired my love for mathematics and showed me the kind of person I want to be, to the teachers in FME and CFIS, who showed me the passion, rigour, and beauty in mathematics and brought me to the border between mathematics and computer science, where I've spent some of my best time.

Research, for me, involves coffee, and someone had to pay for all the coffee. In my case, this work has been partially supported by the European Union's Horizon 2020 research and innovation programme under grant agreement N° 956123 - FOCETA and by the State Government of Styria, Austria - Department Zukunftsfonds Steiermark.

Last but certainly not least, I want to thank Roderick and Bettina for their guidance during these years, for hours and hours of work together, for helping me understand the inner workings of the business, for setting high standards of work, for introducing me to so many great researchers, and for inspiring me to do great work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> *Δεικνύναι δὴ δεῖ τοῖς τοιούτοις ὅτι ἔστι πᾶν τὸ πρᾶγμα οἷόν τε καὶ δι᾽ ὅσων πραγμάτων καὶ ὅσον πόνον ἔχει. ὁ γὰρ ἀκούσας, ἐὰν μὲν ὄντως ᾖ φιλόσοφος οἰκεῖός τε καὶ ἄξιος τοῦ πράγματος θεῖος ὤν, ὁδόν τε ἡγεῖται θαυμαστὴν ἀκηκοέναι συντατέον τε εἶναι νῦν καὶ οὐ βιωτὸν ἄλλως ποιοῦντι: μετὰ τοῦτο δὴ συντείνας αὐτός τε καὶ τὸν ἡγούμενον τὴν ὁδόν, οὐκ ἀνίησιν πρὶν ἂν ἢ τέλος ἐπιθῇ πᾶσιν, ἢ λάβῃ δύναμιν ὥστε αὐτὸς αὐτὸν χωρὶς τοῦ δείξοντος δυνατὸς εἶναι ποδηγεῖν.* [1]
>
> — Plato, seventh letter.

1.1 Motivation

As AI systems increasingly permeate critical domains such as healthcare, finance, mobility, and human resources; ensuring the responsible and trustworthy behaviour of these autonomous systems becomes imperative. Without proper safeguards, AI models can make decisions that are unsafe, biased, or otherwise misaligned with societal values. The need for trust in AI has gathered the attention of different stakeholders, including academic institutions, private corporations, and regulatory bodies [Eur21; Whi22].

The concept of trustworthy AI is broad, recent, and aspirational. Given its nascent stage, there is no consensus on what makes an AI system trustworthy or who has the authority to define it. Different stakeholders emphasize certain aspects over others, whether to serve their own interests or to build trust incrementally by addressing specific challenges. Meanwhile, many public and private institutions strive to be pioneers in deploying autonomous systems for critical decision-making, aligning their ambitions with the pursuit of more trustworthy AI. One of the most influential attempts to shape the meaning and

[1] "One should show such people what philosophy is in all its extent; the range of studies by which it is approached, and how much labour it involves. For the person who has heard this, if she has the true philosophic spirit and that godlike temperament which makes her a kin to philosophy and worthy of it, thinks that she has been told of a marvellous road lying before her, that she must forthwith press on with all her strength, and that life is not worth living if she does anything else."

requirements for the broad concept of *trustworthy AI* is the "Ethics guidelines" document [Com19] produced by the *High-Level Expert Group on AI*, a diverse group of experts from both academia and industry appointed by the European Commission.

In [Com19], trustworthy AI is generally defined to be lawful, ethical, and robust. The document outlines seven key requirements to achieve trustworthy AI, which can be seen as seven different fields of study that we need to collectively develop and understand. In a nutshell, these requirements are:

1. *Human agency and oversight.* AI systems should be designed ensuring oversight through human-in-the-loop, on-the-loop, and in-command mechanisms.

2. *Technical robustness and safety.* AI systems must be resilient, secure, and reliable, with fallback mechanisms to ensure safety in unexpected situations and protection against potential attacks.

3. *Privacy and data governance.* AI systems must respect privacy and data protection while ensuring data integrity, and legitimate access through robust governance mechanisms.

4. *Transparency.* AI systems must be transparent, with traceability mechanisms and clear explanations tailored to stakeholders. Users should be aware of AI interactions and understand its capabilities and limitations.

5. *Diversity, non-discrimination, and fairness.* AI systems must prevent unfair bias to avoid marginalization and discrimination while fostering diversity and accessibility.

6. *Societal and environmental well-being.* AI systems should benefit all, including future generations, by being sustainable and environmentally friendly.

7. *Accountability.* AI systems must have accountability mechanisms, including auditability for assessing algorithms, data, and design. Clear redress processes should be in place, especially for critical applications.

While all requirements are important, in this thesis, we present advances in the directions of safety, fairness, transparency, and accountability, so we will only focus on these fields. In the following Sections 1.2, 1.3, and 1.4, we introduce each field of study, presenting first a broad approximation to the main problems and debates, followed by a concrete problem inside of each field that motivates the work presented in this thesis. We start with safety, follow with fairness and end with transparency and accountability. We bundle transparency and accountability together because our contribution, while mostly motivated by the accountability requirement, is essentially a method to better understand the behaviour of an AI system, and thus fits as well in the category of transparency. A floating contribution of this thesis is a novel formalization that unifies previously existing concepts. Just as the list of requirements in [Com19] can be seen as *what* an AI system needs to be trustworthy, formal methods are a popular answer to *how* to implement these requirements. We dedicate Section 1.5 in

this chapter to present a broad motivation for the use of formal methods and summarize our novel formalization.

1.2 Safety

1.2.1 Background

Safety in AI broadly refers to ensuring that a system does not produce harmful or unintended consequences. It encompasses a range of issues, from preventing system failures to aligning AI decisions with ethical and legal standards. A key concern is technical robustness, which ensures that AI functions correctly under both normal and unexpected conditions. Examples of unexpected conditions that have been particularly studied are adversarial examples [GSS15] and distributional shifts in the input data [Wil+22].

A fundamental concept in AI safety is verification and validation, where formal methods are used to mathematically prove the correctness of an AI model's behaviour. In safety-critical applications like aviation and medical diagnostics, regulatory frameworks often require rigorous validation before deployment. Additionally, fail-safe mechanisms must be in place to handle unexpected situations gracefully, allowing the system to revert to a safe state when necessary [Cre+07; Set+98].

Ensuring AI robustness requires designing models that generalize well beyond their training phase. Techniques such as adversarial training improve resilience by exposing AI models to perturbed or adversarial examples during training, making them less susceptible to manipulation [Bai+21a]. Formal verification methods, such as model checking [BK08] and theorem proving [Har09], provide mathematical guarantees on system behaviour, ensuring that certain safety properties always hold.

Another crucial approach is certifiable AI, where models are designed to provide provable guarantees about their predictions [Fis+21]. This approach has gathered particular attention for neural networks, where verification techniques can analyze how slight variations in input data affect model outputs, helping establish bounds on safe behaviour [Alb21].

AI-driven control systems, particularly those used in robotics, autonomous vehicles, and industrial automation, require additional safety considerations [PT20].

1.2.2 Safe Reinforcement Learning

Reinforcement learning (RL) [SB18] is one of the most successful approaches to several types of problems where an agent interacts with a probabilistic environment, modelled as a Markov decision process (MDP). Notable examples beating human performance at complex games [Mni+15; Sil+16] and discovering higher order structures of proteins [Jum+21]. RL poses unique safety challenges because it learns optimal behaviour through trial and error, often by exploring unknown states. This exploration can lead to catastrophic failures if the system takes unsafe actions while learning. Safe RL methods aim to mitigate such risks by integrating safety constraints into the learning process.

One common approach in safe RL is reward shaping [NHR99], where the reward function is designed to penalize unsafe actions, guiding the agent away from hazardous behaviours. Another method is constrained RL, where policies are optimized under predefined safety constraints [SJS21; WT18]. These methods can be used not only to produce safe results but also to ensure safe exploration [Wie+23; Yan+23a].

Constraints can be implemented directly to the MDP [Alt21; GBA21; WS20], or as regularizers to the corresponding loss functions in learning schemes like constrained policy optimization [Ach+17] and trust region-based approaches [Sch+15], which steer the policy updates away from unsafe behaviours.

Another popular approach to safe RL is the use of restraining bolts [DG+19; DG+20], which steer the learning process towards safe policies by restricting unsafe behaviour. Another recent approach is to learn a controller together with a certificate that proves the controller to be safe [Cha+23].

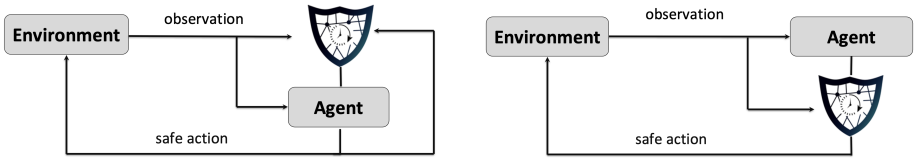
Shielded RL [Als+18; Gia+21; Car+23; Yan+23b] is another promising approach, where a safety layer acts as a filter that prevents the agent from taking dangerous actions. This can be achieved using formal verification techniques to ensure that the learned policy remains within safe bounds. Shields can be placed before the agent, serving as a mask of allowed actions (pre-shields), or after the agent, overwriting unsafe actions by safe ones (post-shields). We illustrate these two settings in Figure 1.1. While shielding methodologies are popular and collision avoidance in autonomous driving is among the common motivations, the work presented in this thesis is the first implementation of shielding techniques for collision avoidance in realistic driving simulators.

In this thesis, we present two contributions to shielding techniques for ensuring safety, that constitute Chapters 4 and 5. In Chapter 4, we develop the theory of deterministic shields resilient to delayed observations, and present experiments in a gridworld and in the CARLA driving simulator. In Chapter 5, we report on our experience in using probabilistic shielding for autonomous valet parking. Since probabilistic shielding requires a more complex model of the agent and the environment, Chapter 5 strongly focuses on how to build a realistic model.

1.2.3 Contribution: Deterministic Shielding Resilient to Delayed Observations (Chapter 4)

Deterministic shields ensure system safety by constructing a safety game from an environmental model and a formal safety specification [Kön19]. The maximally-permissive winning strategy allows all actions that won't cause safety violations over an infinite horizon. Shields allow any action allowed by the maximally-permissive winning strategy, and overwrite potentially unsafe actions by safe ones.

Real-world control systems face delays due to data collection, processing, or transmission. Ignoring delays can cause safety-critical failures. We propose delay-resilient pre- and post-shields to guarantee safety under delays in the observations. To synthesize shields, we extend the safety game to include worst-case delays, introducing imperfect state information. Both pre- and post- shields require the maximally-permissive winning strategy for the corresponding safety



(a) Pre-shield. (b) Post-shield.

Figure 1.1: Shielding scheme.

game under delayed observation. We compute it following the algorithm proposed in [Che+18].

For post-shields, we need to define which of the available correct actions the shield will use to overwrite each potentially unsafe action. To do so, we compute shields that maximize a given fitness function of states, and propose two fitness criteria: robustness and controllability. The robustness of a state measures how close it is to an unsafe state in the safety game graph. The controllability of a state is the maximal amount of delay under which that state can still be considered safe.

We tested deterministic shields resilient to delayed observations in the open source simulator CARLA [Dos+17], for avoiding collisions with pedestrians as well as car-on-car collisions in intersections. We used the default driver available in CARLA as our shielded agent.

1.2.4 Contribution: Probabilistic Shielding for Autonomous Valet Parking (Chapter 5)

Unlike deterministic shields, which enforce safety strictly, probabilistic shielding accounts for low-probability events but only intervenes when the risk of a collision exceeds a predefined threshold. This approach reduces unnecessary interventions, and uses the Markov decision process (MDP) as its underlying model, instead of the safety game.

The shield evaluates control commands by mapping sensor data and prior actions to an MDP state, then estimating the probability of avoiding collisions if the command is executed. If this probability falls below the threshold, the shield overrides the command. These probability computations rely on probabilistic model checking, requiring a well-structured MDP representation of the vehicle and its environment.

Building an appropriate MDP model is challenging—it must balance accuracy with computational feasibility. The model consists of the ego car and the pedestrians. The ego car is represented via an abstraction of the Simrod digital twin [Deb19], with discretized actions and states to handle uncertainty. The pedestrians are modeled with movement speeds following a normal distribution, varying across adults, elders, and children.

we tested probabilistic shields as part of a more complex agent developed as a shared effort in the FOCETA project [Ben+23]. In this case, the simulator used was Prescan, a proprietary tool partially developed within the project, and the

goal of the shield is to act together with an emergency brake system to avoid collisions with pedestrians.

1.3 Fairness

1.3.1 Background

Fairness in AI is essential to prevent discriminatory outcomes, ensuring that automated decisions do not reinforce or exacerbate societal biases. AI models, trained on historical data, often inherit biases present in society, leading to discriminatory outcomes that disproportionately affect marginalized groups [BHN23]. Since AI systems play an increasingly significant role in decision-making processes across domains such as hiring, lending, healthcare, and law enforcement, ensuring fairness and preventing discrimination have become critical concerns and the focus of a burgeoning field of research [ZMS23; Blu+18; Che+20; CD+17; DI19; Elz+19; Ge+21; Gra+22; SGD23; Wan+23]. Addressing these biases is essential to developing ethical AI systems that align with societal values of justice and equality.

Group fairness vs. individual fairness. Fairness in AI is typically framed in terms of two broad categories: group fairness and individual fairness. Group fairness ensures that different demographic groups (e.g., based on race, gender, or age) receive similar outcomes from an AI system. This can be formalized using constraints such as demographic parity (equal selection rates across groups) or equalized odds (equal error rates across groups). Individual fairness [GK21], on the other hand, requires that similar individuals receive similar treatment, independent of their group membership. This is typically formulated using similarity metrics that measure how closely two individuals resemble each other in relevant attributes.

Balancing these two notions is challenging, as enforcing strict group fairness constraints may sometimes lead to violations of individual fairness and vice versa. Different fairness interventions prioritize one over the other, depending on the context and ethical considerations. In this thesis, we focus on group fairness properties.

Sources of bias. AI systems can exhibit bias due to different models of the world that induce disparities. These biases can be broadly classified into two categories: intrinsic and extrinsic.

Intrinsic bias stems from biased training data that reflects historical inequalities or prejudices. For example, a hiring algorithm trained on past hiring decisions may reinforce gender disparities in hiring practices. Extrinsic bias arises from the way AI models process and generalize information. Even if the data itself is unbiased, the learning algorithms may still introduce disparities due to optimization choices, feature selection, or model architecture.

Understanding the origin of bias is crucial in determining appropriate mitigation strategies. If the bias is intrinsic, interventions may involve adjusting the data

representation, while extrinsic bias may require changes to the model's learning process.

A key discussion in fairness research is whether lower accuracy on paper equates to a more just and correct model. If the training data shows an intrinsic bias against a certain group, it stands to reason that maximizing accuracy with respect to the biased dataset does not induce the most accurate model with respect to the real underlying unbiased data. Therefore, an unbiased model, that will achieve lower accuracy with respect to the training data, is not only more fair, but arguably more accurate. However, it is not possible in many cases to determine what is the best-performing compromise. This compromise and the impossibility to find a solution that satisfies all constraints has been studied in [SG21].

Types of fairness-inducing methods. Fairness interventions can be broadly categorized into three main approaches:

- *Pre-processing methods:* These focus on modifying the training data to remove bias before model training. Techniques include re-weighting samples, adjusting labels, and generating fair representations that obfuscate sensitive attributes [KC12; Zem+13].
- *In-processing methods:* These modify the training procedure to incorporate fairness constraints directly into the learning process. Regularization techniques and adversarial training are commonly used to ensure that the model does not learn biased patterns [ZLM18; Kam+12].
- *Post-processing methods:* These adjust the model's predictions after training to equalize outcomes across demographic groups without altering the underlying model [HPS16].

Each approach has advantages and trade-offs. In summary, preprocessing ensures fairness at the data level but may not generalize well, while in-processing methods provide direct fairness guarantees but can be computationally expensive. Post-processing methods are easy to implement but tend to compromise individual fairness.

1.3.2 Fairness in Sequential Decision-Making Problems.

In sequential decision-making settings, such as loan approvals or criminal risk assessments, fairness concerns are magnified due to the compounding effects of biased decisions. Biased initial decisions can lead to feedback loops, where disadvantaged groups receive consistently lower opportunities over time, exacerbating inequalities.

Fairness interventions in sequential decision-making often involve tracking disparities over multiple time steps and designing policies that compensate for historical disadvantages.

Group fairness properties are described in terms of the joint probability distribution of the population and the outcomes. For example, demographic parity states that the probability of a favourable outcome must be independent of

group membership. In a sequential setting, these probabilities can be estimated using the relative frequencies of each outcome for each group.

For example, consider a company building a large team, with a population that we can divide into two groups, A and B, with respect to which the decisions must be fair. After $T = 1000$ candidates, n_A candidates were from group A, out of which n_A^1 were offered a job. The rest n_B candidates were from group B, and of them n_B^1 were offered a job. We can estimate the probability of a candidate from groups A and B of getting an offer as:

$$\mathbb{P}(\textit{offer} \mid A) \approx \frac{n_A^1}{n_A}, \quad \text{and} \quad \mathbb{P}(\textit{offer} \mid B) \approx \frac{n_B^1}{n_B} = \frac{n_B^1}{T - n_A}.$$

Demographic parity is formally expressed as $\mathbb{P}(\textit{offer} \mid A) = \mathbb{P}(\textit{offer} \mid B)$, and in terms of relative frequencies it would mean that

$$\lim_{T \to \infty} \left(\frac{n_A^1}{n_A} - \frac{n_B^1}{n_B} \right) = 0. \tag{1.1}$$

However, in many cases, if the convergence is too slow, it is not enough to guarantee fairness in the long run. Group fairness metrics are emerging properties, which by their own nature cannot be expected when only looking at a few decisions. After seeing $T = 6$ candidates, three from each group, and hiring one from group A and two from group B, the difference between relative frequencies is $1/3$, far from 0, but the process has just started, so it is not reasonable for demographic parity to emerge yet. If the company continues the interview process and after $T = 1000$ the acceptance ratio of group A is still $2/3$ and the acceptance ratio of group B is only $1/3$, we can argue for an underlying bias. This concept can be formalized by stating that after a certain finite horizon of T decisions, the relative frequencies may differ by no more than a certain threshold $\kappa \in [0, 1]$.

One way of looking at the study of fairness in a bounded horizon is with a monitoring perspective: if a process is biased after $T = 1000$ decisions, it is likely to be fundamentally biased, in the sense that the limit in Equation 1.1 does not converge to 0, so we raise the alarm. However, even if the relative frequencies would converge in the limit to the same value, it can be the case that the convergence is too slow. In such cases, finding a considerable disparity between relative frequencies should not be interpreted as a proxy for a bias in the limit, but as a tangible bias that is a problem in itself.

On the other hand, once a finite horizon T has been predefined, an algorithm may act with fairness until the T-th decision, and then act with bias after that, satisfying fairness in the bounded horizon, but failing in the unbounded horizon. To this end, we also study the concept of T-periodic fairness, where we require the relative frequencies to differ by no more than a certain threshold κ after $k \cdot N$ decisions, for all $k \in \{1, 2, 3, \dots\}$ [Ala+24].

1.3.3 Contribution: Fairness Shielding (Chapter 6)

In this thesis, we present *fairness shields* as a post-processing group fairness enforcement solution for bounded and periodic horizons in sequential classification problems.

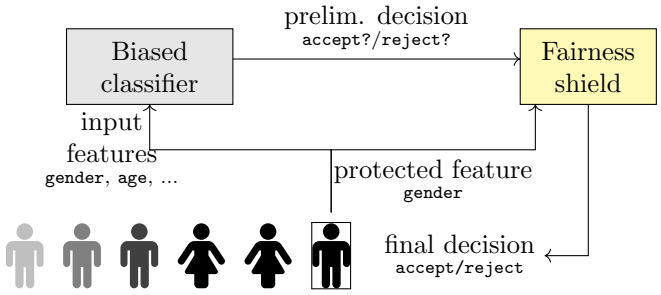


Figure 1.2: The operational diagram of fairness shields.

As we illustrate in Figure 1.2, the shield monitors the decisions of a potentially biased classifier and has the power to override them. Given a predefined fairness criterion and a time horizon or period, the shield observes individuals' protected attributes, the classifier's recommendations, and the cost of modifying decisions. It then ensures fairness while minimizing intervention costs.

To guarantee fairness in finite horizons, fairness shields are computed as bounded-horizon optimal control problems with a hard fairness constraint and a soft cost constraint. The fairness constraint ensures that empirical bias remains below a threshold, measured either at the end of the horizon or periodically. The soft cost constraint discourages excessive interventions by minimizing total expected costs.

The problem becomes harder for periodic horizons, as there are infinitely many input sequences that the shield has to potentially deal with. We conjecture that optimal shields for periodic horizons cannot be described with finite resources, and propose three "best effort" solutions that modify the computation of bounded horizon shields to obtain periodic shields. With these solutions, we lose the hard fairness guarantee for all traces. As a remedy, we study conditions on the incoming traces that ensure the shields achieve fair outputs. These solutions can be classified into the *static* approach, and the *dynamic approach*.

The static approach consists of resetting and reusing the same shield after each time period. If the shield has been computed for a finite horizon T, at step $T + 1$, the internal counters are reset to zero and the shield enforces fairness in the segment from $T + 1$ to $2T$ in the same way as it did for the segment from 1 to T. A static shield applies the same fairness criterion for each segment of decisions of length T, with the hope that the same fairness criterion applies when concatenating all segments of length T. The advantage of this approach is simplicity, both in design and computational complexity. The main drawback is that the hard guarantees hold for small subsets of traces.

The dynamic approach consists of recomputing the shield after each period, modifying the fairness condition to account for the accumulated decisions of the trace so far. The advantage of this approach is that the fairness criterion is tracked more accurately, so these shields tend to interfere less often with the classifier while ensuring fairness in a large subset of traces. The main drawback is that the synthesis algorithm has to be executed at the end of each period.

To understand the difference between the static and dynamic approach, recall the example of hiring applicants from groups A and B, trying to enforce a threshold on demographic parity no larger than $\kappa = 0.2$. After the first $T = 1000$ decisions, the acceptance rate for group A is 0.5, and the acceptance rate for group B is 0.35. Thus, this segment is biased towards group A, but not more than the threshold. A dynamic shield would allow the next segment of T decisions to have an acceptance rate for group A of 0.5 and 0.72 for group B. Even if looking at the segment from $T + 1$ to $2T$, the difference in acceptance rates is larger than the threshold, the dynamic shields knows that group B can overcompensate for the low rate in the first segment, as long as demographic parity is kept in the threshold for the longer segment of the first $2T$ decisions. On the other hand, a static shield would not allow B to overcompensate, being more restrictive than the dynamic shield.

Shields rely on a known or learned distribution of future decisions and costs. Even if the distribution is imprecise, fairness guarantees remain intact—only cost-optimality may be affected. Shields are computed via dynamic programming, optimized to run in polynomial time for a wide variety of group fairness metrics by abstracting traces to a relevant set of counters.

1.4 Transparency and Accountability

1.4.1 Background

Beyond ensuring that AI systems behave safely and fairly, they must also be explainable. Trust in AI depends not only on its performance but also on its transparency — users and stakeholders must understand why a system made a particular decision. Moreover, when AI systems cause harm, it is essential to have robust accountability mechanisms in place to determine responsibility and take corrective action. Explainability is also key for accountability: if an AI system causes harm or fails in an unexpected way, it must be possible to trace its reasoning to diagnose the issue and assign responsibility. Without explainability, AI remains a "black box", making it difficult to audit, improve, or justify its decisions in legal and ethical contexts. By integrating explainability into AI design, we can build systems that foster trust, enable human oversight, and ensure accountability in decision-making.

Explainability and accountability are closely intertwined. In human accountability processes, understanding why a person acted in a certain way is essential for assigning responsibility and determining degrees of culpability. Courts, for example, consider intent, circumstances, and explanations when assessing guilt. Similarly, for AI systems, understanding why a particular decision or action was taken is crucial in determining liability when harm occurs.

Accountability in software systems has long been a topic of interest in fields such as cybersecurity, safety-critical systems, and software engineering [FJW11; Jag+09; KTV10]. Traditional software accountability often relies on clear specifications, audit logs, and formal verification techniques to determine responsibility when a system fails or produces an unexpected outcome [Kro+17; FJW20].

However, AI-based systems, particularly those leveraging machine learning,

present unique challenges. Unlike traditional rule-based software, many AI models operate effectively as black boxes, making it difficult to trace the logic behind their decisions. This lack of transparency complicates accountability, as it becomes unclear whether failures arise from design flaws, biased training data, unforeseen interactions, or user misuse.

1.4.2 The Role of Intention in Accountability

A crucial aspect of human accountability is the notion of *intention*. Understanding whether an action was intentional, accidental, or due to negligence is key in determining degrees of responsibility. Courts, for instance, distinguish between premeditated actions and unintended mistakes, applying different legal consequences accordingly [Kno16].

For AI, the concept of intention requires further study. As a caveat, we explicitly avoid the debate on whether AI systems may have consciousness or free will [But01; CM13]. In any case, a large part of the theoretical development on the concept of intention can be applied to any rational planning agent that acts with goals and constrained resources [BIP88; Bra87]. This general definition also applies to many AI agents, regardless of the working notion of consciousness and free will. AI systems exhibit functionally intentional behaviour, such as pursuing a specified objective or optimizing for a particular reward. Understanding intentional behaviour in AI can help refine accountability frameworks by distinguishing between different sources of harm.

1.4.3 Contribution: Intentional Behaviour in Agents operating on MDPs (Chapter 7)

Interpreting the decision-making processes of modern machine-learning-based agents in probabilistic settings presents significant challenges due to the absence of explicit goals or intentions in their models. Traditionally, intention is connected to planning in both cognitive and computational reasoning [BIP88]. In this thesis, we consider intention as the "state of the world" an agent plans toward, serving as a proxy for its internal reasoning. Since modern agents, particularly those trained using reinforcement learning, do not have explicitly modeled beliefs or reasoning processes, their intentions can only be inferred probabilistically.

Our proposed framework quantitatively assesses whether an agent's behaviour exhibits evidence of intentionality. Instead of making binary assertions about intention, it provides confidence levels and quantified evidence.

We model autonomous agents as policies within a probabilistic environment as MDPs. Key to our framework are the notions of agency and intention quotient. The agency measures the agent's ability to influence outcomes, defined as the probability difference between optimally achieving or avoiding a goal. The intention quotient is a normalized value between 0 and 1 that measures how close the policy of the agent is to achieving or avoiding a goal. Intention quotient quantifies the degree of apparent intentional behaviour, with values close to 1 indicating high evidence of intentionality.

These notions can be used to study an agent preemptively by calculating agency and intention quotients towards a particular goal around the states of interest. With the mindset of serving a potential accountability process, we also propose a retrospective methodology to study concrete traces that end up in a harmful state. When the trace under study does not offer enough evidence for a confident assessment, we produce counterfactual traces and use them to increase confidence.

Our method can help understand whether an agent shows evidence of acting intentionally towards an end. While this is a necessary step for accountability processes, it is not the only one, and questions such as who is responsible or who has to pay for the harm intentionally produced by the agent are out of the scope of this work.

1.5 Formal Methods

In the field of computer science, formal methods can be broadly described as rigorous mathematical techniques used to specify, verify, and prove system properties. Rigorous formalizations not only have the advantage of providing provable guarantees, but also a deep understanding of those guarantees.

Formal methods have long been a cornerstone of trusted computing, being used in safety-critical domains like controllers in avionics and medical devices, as well as performance-critical system code like arbiters and process managers [Woo+09; Bee+24]. Because of this history, many researchers and practitioners think that formal methods for AI are destined to play a central role in the future of trustworthy AI [Li+23]. By applying logical reasoning, theorem proving, and model checking, formal methods provide strong guarantees about software and hardware systems. Unlike testing, which only checks for correctness in specific scenarios, formal verification provides mathematical certainty for all possible cases within a given model. However, the guarantees are only as good as the model, and, as the popular saying goes, all models are wrong — albeit some of them are useful. This should serve as a constant reminder throughout the thesis that all results and guarantees hold in the ideal model, and any consequences of those results on reality are mediated by how good the model is as a description of the real world.

1.5.1 Contribution: The Reactive Decision Making Framework (Chapter 3)

The frameworks used to formalize the different contributions in this thesis use different models: safety games are used for deterministic safety shielding, MDPs are used for probabilistic safety shielding and intention analysis, and sequential classification problems are used for fairness shielding.

However, these frameworks always have in common an agent interacting with an environment. In this thesis, we introduce the *reactive decision making* framework as a formal generalization of the aforementioned formal models. We also formalize the concept of shielding in the reactive decision making framework,

generalizing previously differentiated notions [Kön19] and refining the definitions to account for edge cases that had been previously mistreated.

1.6 Outline of the Thesis

Chapter 2 introduces the notation and previously established concepts that are required throughout the thesis. While this chapter, and the thesis as a whole, is self-contained, the exposition may be too succinct for an unfamiliar reader. For a deeper understanding of the background material, we give pointers to adequate reference materials. Chapter 3 introduces the reactive decision making framework and the notion of shielding. We show how the formal frameworks used in this thesis are particular cases of this general formalization. We also show how previous notions of shielding for safety properties correspond to shielding as described in the reactive decision making framework.

Chapters 4 and 5 explore shielding for safety properties with some source of uncertainty in the context of autonomous driving. Chapter 4 explores deterministic shielding for safety properties resilient to delayed observations. We show how shields can be extended to guarantee a safety specification even with imperfect information, and study different methods to choose a corrective action. In Chapter 5, we report our experience on using probabilistic shielding on an autonomous car operating in a parking lot with the objective of avoiding collisions with pedestrians. We describe how to build realistic models of a car and pedestrians in its vicinity, and report the results of a comparative test between our shields and an automatic emergency brake system.

We move our focus from safety to fairness in Chapter 6, where we introduce the notion of fairness shields and describe how to compute different types of fairness shields for finite and periodic time horizons. We validate the usefulness of fairness shields on an extensive evaluation against standard benchmarks from the literature on algorithmic fairness.

Chapter 7 moves away from runtime enforcement towards explainability and accountability. In this chapter, we present our framework for studying intentional behaviour on agents operating in MDPs using the notions of agency and intention-quotient. We present our retrospective methodology, intended for accountability processes after harm has occurred, and showcase how it would work in a toy example.

Chapter 8 rounds up the thesis with future work and concluding remarks, followed by an appendix detailing the publications associated with the completion of the PhD program.

While the landscape of AI research is currently very English-centric, its fruits shouldn't be. As a nod to diversity, each chapter is preceded by a famous quote, mostly in languages other than English. Do not take them too seriously. The last pages of this document contain a cheat sheet meant to help the reader go through the notation. Let the power of well-structured indices prevail when the power of well-written text may fail.

Chapter 2

Preliminaries

ΑΓΕΩΜΕΤΡΗΤΟΣ ΜΗΔΕΙΣ ΕΙΣΙΤΩ. [1]
— Inscription above the entrance of Plato's Academy. [2]

In this chapter, we will briefly cover the basic concepts that will be used throughout the thesis. This serves the double purpose of being a lightweight introduction to the topics and fixing the notation used throughout the thesis.

2.1 Basic Notation

Sets, numbers, and functions. We use $\mathbb{B} = \{\bot, \top\}$ to denote the Boolean domain, $\mathbb{N} = \{0, 1, \dots\}$ to denote the set of natural numbers, $\mathbb{Z}$ to denote the set of integer numbers, and $\mathbb{R}$ to denote the set of real numbers. Given $a < b \in \mathbb{R}$, we use (a, b) to denote the open interval between a and b, $[a, b]$ to denote the closed interval, and $(a, b]$ to denote the interval open at one end and closed at the other. Given a real number $a \in \mathbb{R}$, we use $\lfloor a \rfloor$ to denote the largest integer that is less or equal than a (i.e., the *floor* of a), $\lceil a \rceil$ to denote the smallest integer that is greater or equal than a (i.e., the *ceiling* of a) and $\lfloor a \rceil$ to indicate the integer that is closest to a (i.e., the result of *rounding* a). We use the standard convention that $\lfloor a \rceil = \lceil a \rceil$ when a is at the same distance of $\lfloor a \rfloor$ than $\lceil a \rceil$. In general, it is true that $a - 1 \leq \lfloor a \rfloor \leq a \leq \lceil a \rceil \leq a + 1$. Given $a, b \in \mathbb{R}$, we use the notation $a \ll b$ to indicate that a is *much smaler* than b, and $a \gg b$ to indicate that a is *much greater* than b, where how much is *much* depends on the context.

Given a finite set X, we denote its cardinality by $|X|$. Given a function $f \colon \mathcal{X} \to \mathcal{Y}$, and a subset $X \subseteq \mathcal{X}$, the image set is $f(X) = \{y \in \mathcal{Y} \,:\, \exists x \in X,\, f(x) = y\}$. Similarly, given a subset $Y \subseteq \mathcal{Y}$, the antiimage set is $f^{-1}(Y) = \{x \,:\, f(x) \in Y\}$. Given an arbitrary domain $\mathcal{X}$ and a function $f \colon X \to \mathbb{R}^n$, the *support* of f is $\mathrm{Supp}(f) = \{x \in \mathcal{X} \,:\, f(x) \neq 0\}$. Let X be a subset $X \subseteq \mathcal{X}$, we define the

[1] Let no one ignorant of geometry enter here.

[2] The existence of this inscription is a disputed fact, since the earliest known documents that mention it date about 700 years after Plato's death. It has become, however, a powerful meme.

indicator function of X in $\mathcal{X}$ as $\mathbb{1}_X \colon \mathcal{X} \to \{0,1\}$, such that $\mathbb{1}_X(x) = 1$ if and only if $x \in X$. We will use the equivalent notation $\mathbb{1}[x \in X]$. In the special case where the set is a single element, $X = \{y\}$, we will use the notation $\mathbb{1}[x = y]$. Similarly, if $X = \mathcal{X} \setminus y$, we will denote the indicator function as $\mathbb{1}[x \neq y]$.

Words and languages. An alphabet Σ is a finite set. A word (or trace) in an alphabet is a sequence of elements in the alphabet. We use Σ^i to denote the set containing words of length i in the alphabet Σ and $\Sigma^{\leq k}$ to denote the set of words of length at most k in Σ, i.e., $\Sigma^{\leq k} = \cup_{j=0}^{k} \Sigma^j$. We use Σ^* for the set of all words of finite length, i.e., $\Sigma^* = \cup_{k=0}^{\infty} \Sigma^k$. We use Σ^ω to denote the set of infinite words in Σ, and $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. We will also use 2^Σ to denote the power set of Σ. Given a word $w \in \Sigma^\infty$, we use $|w|$ to refer to its length. Given a word $w = \sigma_0 \sigma_1 \cdots \in \Sigma^\infty$, for n, m such that $n \leq m \leq |w|$, we will use the notation $w_{[n:m]} = \sigma_n \sigma_{n+1} \ldots \sigma_m$, and in the case $n = 0$ we will use the notation $w_{:m}$ to refer to $w_{[0:m]}$. Given two words $w_1 = \sigma_0 \ldots \sigma_n, w_2 = \sigma_{n+1} \ldots \sigma_{n+m}$ in Σ, the concatenation is the word $w_1 \cdot w_2 = \sigma_0 \ldots \sigma_n \sigma_{n+1} \ldots \sigma_{n+m}$. Given a finite word $w = \sigma_1 \ldots \sigma_n$, we use w^ω to denote the resulting word of concatenating w infinitely many times. A set of words $L \subseteq \Sigma^\infty$ is called a language.

2.2 Probability Theory

In this section, we define some basic notions of probability theory. We refer the reader to [Dur19, Chap. 1] for a more detailed discussion.

Let Ω be a non-empty set, a *σ-algebra* is a set $\mathcal{F} \subseteq 2^\Omega$ that satisfies: (i) $\Omega \in \mathcal{F}$, (ii) if $A \in \mathcal{F}$, then $\Omega \setminus A \in \mathcal{F}$, and (iii) if $\{A_i\}_{i=1}^{\infty}$ is a countable collection of sets in $\mathcal{F}$, then $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$. The tuple $(\Omega, \mathcal{F})$ is a *measurable space*. Given a set Ω and a subset $F \subseteq 2^\Omega$, the σ-algebra generated by F is the smallest σ-algebra $\mathcal{F}$ such that $F \subseteq \mathcal{F}$. The most commonly used measurable space is $(\mathbb{R}^n, \mathcal{B}^n)$, where $\mathcal{B}^n$ is the Borel σ-algebra on $\mathbb{R}^n$, i.e., the σ-algebra generated by the open sets of $\mathbb{R}^n$. Let $(\Omega, \mathcal{F})$ and $(\Omega', \mathcal{F}')$ be two measurable spaces, a function $f \colon \Omega \to \Omega'$ is *measurable* if for every $B \in \mathcal{F}'$, $f^{-1}(B) \in \mathcal{F}$.

A *probability space* is a tuple $(\Omega, \mathcal{F}, \mathbb{P})$, where $(\Omega, \mathcal{F})$ is a measurable space and $\mathbb{P} \colon \mathcal{F} \to [0,1]$ is a function satisfying: (i) $\mathbb{P}(\Omega) = 1$, (ii) if $A \in \mathcal{F}$, then $\mathbb{P}(\Omega \setminus A) = 1 - \mathbb{P}(A)$, and (iii) for any countable collection $\{A_i\}_{i=1}^{\infty}$ of disjoint sets in $\mathcal{F}$, we have $\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$. $\mathbb{P}$ is called a *probability measure* over $(\Omega, \mathcal{F})$. A *random variable* on Ω is a measurable function $X \colon \Omega \to \mathbb{R}$ from a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ to $(\mathbb{R}, \mathcal{B})$. We typically think of a random variable as a symbol that takes a value in $B \in \mathcal{B}$ with probability $\mathbb{P}(X^{-1}(B))$. We also denote $\mathbb{P}(X^{-1}(B))$ as $\mathbb{P}[X \in B]$. When $B = \{b\}$, we may denote $\mathbb{P}(X^{-1}(B))$ as $\mathbb{P}[X = b]$.

A random variable X induces a a probability measure μ on $(\mathbb{R}, \mathcal{B})$, called its *probability distribution* (or simply *distribution*), by setting $\mu \colon \mathcal{B} \to \mathbb{R}$, $\mu(A) = \mathbb{P}(X \in A)$. The distribution of a random variable is usually described using its *distribution function* $F \colon \mathbb{R} \to [0,1]$, defined as $F(x) = \mathbb{P}(X \in (-\infty, x))$. We say that X *follows* the distribution F, and denote it by $X \sim F$. The set of distributions over $\mathbb{R}$ is the set of probability measures over $(\mathbb{R}, \mathcal{B})$, and denoted as $\mathcal{D}(\mathbb{R})$. For a countable set Ω, a *distribution over* Ω is any function $d \colon \Omega \to \mathbb{R}$,

such that there exists a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ satisfying for all $\omega \in \Omega$ that $\{\omega\} \in \mathcal{F}$ and $\mathbb{P}(\{\omega\}) = d(\omega)$. We denote the set of distributions over Ω as $\mathcal{D}(\Omega)$.

The *expected value* of a random variable is $\mathbb{E}[X] = \int_\Omega X(\omega) d\mathbb{P}(\omega)$, where $\int$ is the standard Lebesgue integral on the measure given by $\mathbb{P}$.

2.3 Deterministic Two-Player Games

2.3.1 Games with Perfect Information

The deterministic two-player game is a widely used formalism to model deterministic interactions between an agent and an environment, where the environment is considered adversarial.

Formally, a *deterministic two-player game* is a tuple $\mathcal{G} = (S, s_0, S_{env}, S_{ag}, \mathcal{A}, \mathcal{T}, \mathrm{Acc})$, where $S = S_{env} \cup S_{ag}$ is a finite set (the *state space*), composed of the disjoint sets S_{env} (states of the environment) and S_{ag} (states of the agent). There is a special state, $s_0 \in S_{env}$, the initial state. $\mathcal{A}$ is the set of actions, $\mathcal{T} \subseteq (S_{env} \times S_{ag}) \cup (S_{ag} \times \mathcal{A} \times S_{env})$ is the transition relation and $\mathrm{Acc} \subseteq S^\omega$ is the winning condition for the agent. For convenience, we usually write $s \xrightarrow{\sigma} s'$ for $(s, \sigma, s') \in \mathcal{T}$, and $s' \xrightarrow{u} s$ for $(s', s) \in \mathcal{T}$, where u stands for "undefined". The state space is required to be deadlock-free, i.e., for all $s \in S_{env}$, there exists $s' \in S_{ag}$ such that $(s, s') \in \mathcal{T}$. Similarly, for all $s' \in S_{ag}$, there exist $\sigma \in \mathcal{A}$ and $s \in S_{env}$ such that $(s', \sigma, s) \in \mathcal{T}$. The transition relation is deterministic for the agent, i.e., for any $s \in S_{ag}$ and $\sigma \in \mathcal{A}$, if $s \xrightarrow{\sigma} s'$ and $s \xrightarrow{\sigma} s''$ then $s' = s''$. It is useful to think of $\mathcal{G}$ as a bipartite graph, where the set of nodes is partitioned into S_{ag} and S_{env}, the edges from S_{ag} to S_{env} are labeled with symbols in $\mathcal{A}$, and the edges from S_{env} to S_{ag} are unlabeled.

On some occasions, it will be useful to work with a concrete set of actions for the environment. An *environment action set* is a set $\mathcal{A}_{env}$, that uniquely labels each transition of the environment, making the transition relation deterministic for the environment. That is, for every pair of states $s \in S_{env}, s' \in S_{ag}$ such that $(s, s') \in \mathcal{T}$, there exists a unique label $x \in \mathcal{A}_{env}$ associated with the pair. We write this relation as $s \xrightarrow{x} s'$. To make the environment transitions deterministic, we require that given $s \in S_{env}, x \in \mathcal{A}_{env}$, there is at most one state $s' \in S_{ag}$ such that $s \xrightarrow{x} s'$.

Note that a set $\mathcal{A}_{env}$ has at least as many labels as the maximum out-degree of $\mathcal{T}$ for states in $\mathcal{S}_{env}$, that is

$$|\mathcal{A}_{env}| \geq \max_{s \in S_{env}} |\{s' \in S_{ag} \,:\, (s, s') \in \mathcal{T}\}| \,. \tag{2.1}$$

Furthermore, this bound is tight: since the condition on the transition is defined independently for every environment state, we can reuse labels without any issue for different environment states.

Plays. The game is played by two players: the agent and the environment. In the safety game, at every state $s \in S_{ag}$, the agent chooses a transition $s \xrightarrow{\sigma} s'$, and at every state of the environment $s \in S_{env}$, the environment chooses a transition $s \xrightarrow{u} s'$. Together, they produce a trace of states $\tau = [s_0, s_1, \dots]$,

where for all $i \in \mathbb{N}$, $s_{2i} \xrightarrow{u} s_{2i+1}$ and there exists σ_i such that $s_{2i-1} \xrightarrow{\sigma_i} s_{2i}$. In the game context, a trace of states is also called a *play*. Sometimes, it is useful to consider the concrete actions of the agent that give rise to a play. For a given play $\tau = [s_0, s_1, s_2, \dots]$, the corresponding *state-action play* is $\tau = [s_0, s_1, \sigma_1, s_2, s_3, \sigma_2, s_4]$, where for all $i \in \mathbb{N}$, $s_{2i-1} \xrightarrow{\sigma_i} s_{2i}$. Since the actions are determined by state transitions, it is equivalent to talking about plays as state sequences or state-action sequences. We will use state-action plays whenever concrete actions take an essential role. The set of all plays of a game $\mathcal{G}$ is denoted by $\Pi(\mathcal{G}) \subseteq (S_{env} \times S_{ag})^\omega$.

Strategies. A *strategy* for the agent is a function $\xi\colon (S_{env} \times S_{ag})^* \to 2^{\mathcal{A}}$, that given a trace $\tau \in S^*$, and produces a set of actions $\xi(s) \subseteq \mathcal{A}$. A strategy is *memoryless* if it only depends on the last state of the trace. In such cases, we will denote it as a function $\xi\colon S_{ag} \to 2^{\mathcal{A}}$. A strategy is *deterministic* if $\forall \tau \in S^*$, $|\xi(\tau)| = 1$. When it is clear from context that a strategy ξ is deterministic, we will denote it as a function $\xi\colon S_{env} \times S_{ag}^* \to \mathcal{A}$, and as a function $\xi\colon S_{ag} \to \mathcal{A}$ when it is also memoryless. A play $\tau = [s_0, s_1, \dots]$ *is valid under a strategy* ξ of the agent if for all $i \in \mathbb{N}$, there exists $\sigma_i \in \xi(s_{2i-1})$ such that $s_{2i-1} \xrightarrow{\sigma_i} s_{2i}$. The set of behaviours $\texttt{Beh}(\mathcal{G}, \xi)$ consists of all plays following ξ, that is:

$$\texttt{Beh}(\mathcal{G}, \xi) = \{\tau = [s_0, s_1, \dots] \in \Pi(\mathcal{G}) \,:\, \forall i \in \mathbb{N}, \exists \sigma_i, \;\; s.t.\; \sigma_i \;\in\; \xi(s_{2i-1})\}. \quad (2.2)$$

A strategy ξ is *winning* for the agent if $\texttt{Beh}(\mathcal{G}, \xi) \neq \emptyset$ and for all $\tau \in \texttt{Beh}(\mathcal{G}, \xi)$ we have $\tau \in \text{Acc}$. A winning strategy ξ is *maximally permissive* if $\texttt{Beh}(\mathcal{G}, \xi') \subseteq \texttt{Beh}(\mathcal{G}, \xi)$ for every winning strategy ξ'.

An *output-restricted* strategy is a function $\xi\colon S_{ag} \times 2^{\mathcal{A}} \to 2^{\mathcal{A}}$, satisfying that $\forall s \in S_{ag}, \forall \Sigma \in 2^{\mathcal{A}}$, $\xi(s, \Sigma) \subseteq \Sigma$. Each output-restricted strategy ξ has an equivalent unrestricted strategy $\hat{\xi}$, defined as $\hat{\xi}(s) = \xi(s, \mathcal{A})$. An output-restricted strategy is *deterministic* if $\forall s \in S_{ag}$, $\forall \Sigma \in 2^{\mathcal{A}}$, $|\xi(s, \Sigma)| = 1$. The previous definitions for winning and maximally permissive strategy apply to the unrestricted strategy.

Synthesis of winning strategies of safety games. A game $\mathcal{G}$ is a *safety game* if its winning condition is defined by the game staying in a set of safe states $\mathcal{F} \subseteq S$. That is, if $\text{Acc} = \mathcal{F}^\omega$. It is a classical result [Tho95] that the maximally permissive strategy of a safety game, if it exists, is unique, memoryless, and can be computed in $\mathcal{O}(|S| \cdot |\mathcal{A}|)$. This is done by computing the so-called winning region. The winning region is the set W of states that are part of a trace valid under a winning strategy:

$$W = \left\{ s \in S \;:\; \begin{array}{c} \exists \tau = s_0 s_1 \dots s, \text{ and } \exists \xi_\tau\colon S_{ag} \to 2^{\mathcal{A}}, \text{ such that} \\ \xi_\tau \text{ is winning and } \tau \text{ is valid under } \xi_\tau \end{array} \right\} \quad (2.3)$$

With the help of the winning region, we can define the strategy $\xi_{\texttt{max.perm.}}$ as

$$\xi_{\texttt{max.perm.}}(s) = \{\sigma \in \mathcal{A} \,:\, s \xrightarrow{\sigma} s', \;\text{ for } s' \in W\}. \quad (2.4)$$

This strategy has the property of being winning and maximally permissive [Tho95]. In safety games, we often say that a state s is *safe* when $s \in W$.

2.3.2 Games Under Delay

Delayed games follow the intuition that the system does not have access to the most recent inputs, forcing it to make decisions having only partial information on the current state. In this section, we summarize the construction of delayed safety games from [Che+21], with two differences. The first one is that we consider delays as full-step and not half-step delays. This is not a conceptual change but rather a change that lightens notation in some definitions. The second is that we extend the definitions to include an arbitrary amount of memory.

Game graph under delay. Introducing delays does not change the game graph itself, but we add two parameters, one for the amount of delay and one for the amount of memory available. Formally, a *deterministic two-player game with delay δ and memory μ* is a tuple $\mathcal{G}_{\delta,\mu} = \langle S, s_0, S_{env}, S_{ag}, \mathcal{A}, \mathcal{T}, \mathrm{Acc}, \delta, \mu\rangle$, where $\delta \in \mathbb{N}$, represents the delay in the input observation, $\mu \in \mathbb{N}$ represents the length of the register (memory) of own outputs allowed to the agent, and $\mathcal{G} = \langle S, s_0, S_{env}, S_{ag}, \mathcal{A}, \mathcal{T}, \mathrm{Acc}\rangle$ is a two-player game.

Game play under delay. A delay of δ causes the agent to be unaware of the last δ transitions produced by the environment. The agent is aware of its own actions and can store them in a register of size μ used to limit the uncertainty about the current (partially unknown) state of the game. We give the name of *observed state* of the agent to the state from which a strategy chooses which action to take. We give the name of *current state* of the agent to the state in which the action chosen by the agent is applied. In perfect information games (i.e., when $\delta = 0$), the current state and the observed state of the agent are the same. In delayed games, the *current state* of the agent is δ agent states ahead of the observed state. Therefore, a strategy under delay δ is aware that an observed state s and output register $\overline{\sigma}$ limits the possible current states, and has to choose an action to be applied, knowing it will be applied in one of the possible current states, without explicit knowledge of which one.

Strategies under delay. In a game with delay δ, the agent can only observe the first state after δ steps in the game, and has to play the first moves "blindly". In this initial transient period, the agent needs to ensure it can travel from the initial state to a state with a defined winning strategy. Therefore, we also need to define the strategy for an unobserved state. Let $S_{ag*} = S_{ag} \cup \{\varepsilon\}$, where ε represents the unobserved state.

A *strategy for the agent under delay δ with memory μ* is a function $\xi_{\delta,\mu} \colon S_{ag*} \times \mathcal{A}^{\leq\mu} \to 2^{\mathcal{A}}$. When δ and μ are clear from context or irrelevant, we will denote it simply as ξ. A state-action play $\tau = [s_0, s_1, \sigma_1, \dots]$ is valid under strategy ξ if $\forall i \in \mathbb{N}$,

$$\sigma_i \in \xi\big(s_{2(i-\delta)-1}, [\sigma_{i-\mu}, \dots, \sigma_{i-1}]\big), \tag{2.5}$$

with s_j, $\sigma_j = \varepsilon$ for any $j < 0$, to account for the transient period.

Most of the time, the strategy takes as input a state $s \in S_{ag}$ and a sequence of actions $\overline{\sigma} \in \mathcal{A}^{\mu}$ representing the last μ actions executed by the agent. The state s is the *observed* state, and the strategy has to take care that any action $\sigma' \in \xi(s, \overline{\sigma})$ will be executed in the *current* state, about which the agent has

only partial information given by s and $\overline{\sigma}$. Note that the state-action trace used in Equation 2.5 is $\tau = [s_0, s_1, \sigma_1, s_2, s_3, \sigma_2, s_4, \dots]$, where s_i is a state of the agent or the environment depending on the parity of i. More concretely, for all i, we have $s_{2i-1} \in S_{ag}$, $s_{2i} \in S_{env}$, and $s_{2i-1} \xrightarrow{\sigma_i} s_{2i}$. The strategy has to define the action σ_i to take in the (unknown) current state s_{2i-1}, using only information of the μ latest actions and the observed state $s_{2(i-\delta)-1}$ that is δ observations behind the current state.

At the beginning of the game, before there are δ environment transitions, the agent has not had time to observe any state yet, and thus the observed state is represented by the empty state ε. Similarly, if there have only been $\nu < \mu$ actions played by the agent, the action register is $\overline{\sigma} \in \mathcal{A}^{\nu}$. This transient phase is the reason why we need to define the strategy also for the domain $\{\varepsilon\} \times \mathcal{A}^{\leq \mu}$.

As in the undelayed setting, we define the behaviour $\texttt{Beh}(\mathcal{G}_{\delta,\mu}, \xi)$, as the set of traces that are valid under ξ, and a strategy ξ is winning if all traces valid under ξ are inside winning set Acc. A winning strategy ξ is *maximally permissive* if $\texttt{Beh}(\mathcal{G}_{\delta,\mu}, \xi') \subseteq \texttt{Beh}(\mathcal{G}_{\delta,\mu}, \xi)$ for every winning strategy ξ'. Chen et al. showed that a safety game $\mathcal{G}$ with delay δ and memory $\mu = \delta$ is equivalent to a safety game with no delay $\mathcal{G}'$ with a set of states $S' = \left(S \times \mathcal{A}^{\delta}\right) \cup \left(\{s'_0\} \times \mathcal{A}^{\leq \delta}\right)$ [Che+21, Lemma 2]. In particular, this result, together with [Tho95], proves that for any game that can be solved with delay δ, there exists a winning strategy with $\mu = \delta$ memory. Some games have winning strategies with less or no memory. We refer to strategies with less memory as *memory-restricted* strategies, with the special case of *memoryless* strategies when $\mu = 0$.

Synthesis of winning strategies under delay. We outline the algorithm to solve delayed safety games with delay δ and memory $\mu = \delta$, as presented in [Che+21]. A more detailed version of the algorithm is given is presented as part of Chapter 4, where we extend it to strategies with any amount of memory $\mu \leq \delta$.

The algorithm to solve a delayed safety game iteratively constructs and solves the safety game with increasing delays $d = 0, 1, \dots, \delta$ and memory size $\mu = d$. At every iteration in d, the maximally permissive strategy for the agent is computed using the strategy for the previous delay $d-1$, followed by a reduction of the game graph aiming to mitigate the exponential blow-up in the state space and the computation of the transient phase. To compute the maximally

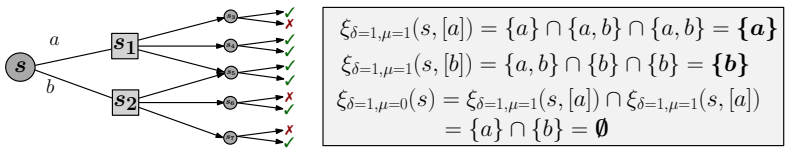


Figure 2.1: Computation of winning strategies under delay $\delta = 1$ with memory $\mu = 0, 1$. In this example, the observed state is s, and the current state is one of s_3, s_4, s_5, s_6, or s_7. From state $s \in S_{ag}$, if the last action was a, the agent knows the environment has made a transition from state s_1, making b potentially unsafe. Similarly, if the last action by the agent was b, the current state is one of s_5, s_6, or s_7, making a unsafe. If the agent cannot store its last action in memory, there is no winning strategy from s.

permissive strategy using previous delays, for each state $s \in S_{ag}$ and output register $[y_1, \ldots, y_d]$, we compute the set $\mathcal{I}_{s,y_d}$, containing all states $s'' \in S_{ag}$ that can be reached by a pair of transitions $s \xrightarrow{y_d} s' \xrightarrow{u} s''$. The intersection of the actions allowed by the maximally permissive strategy for delay $d-1$ in states of $\mathcal{I}_{s,y_d}$ corresponds then to the actions allowed with delay d in state s with output register $[y_1, \ldots, y_d]$. Figure 2.1 shows an example of computing the maximally permissive winning strategy under delay $\delta = 1$ from the one under delay $\delta = 0$, for memory $\mu \in \{0, 1\}$.

2.4 Markov Decision Process

A Markov decision process (MDP) is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$, where $\mathcal{S}$ is a countable set of states, $\mathcal{A}$ is a countable set of actions, and $\mathcal{P} \colon \mathcal{S} \times \mathcal{A} \to \mathcal{D}(\mathcal{S})$ is the probabilistic transition function. Given a state $s \in \mathcal{S}$ and an action $a \in \mathcal{A}$, $\mathcal{P}(s, a)$ is a distribution of states. It is common in the literature to denote the probability of a state $s' \in \mathcal{S}$ under the distribution $\mathcal{P}(s, a)$ as $\mathcal{P}(s, a, s')$ instead of $\mathcal{P}(s, a)(s')$. Sometimes, an MDP is described together with a special state $s_0 \in \mathcal{S}$, indicating it is the *initial state*. Sometimes, instead of a single initial state, the MDP is accompanied by a distribution of initial states $\iota \in \mathcal{D}(\mathcal{S})$. A *policy* $\pi \colon \mathcal{S} \to \mathcal{D}(\mathcal{A})$ is a function mapping each state $s \in \mathcal{S}$ to a probability distribution over the actions in $\mathcal{A}$.

A Markov chain (MC) is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{P})$, where $\mathcal{S}$ is a countable set of states and $\mathcal{P} \colon \mathcal{S} \to \mathcal{D}(\mathcal{S})$ is a transition function. Similarly to the notation used in MDPs, for $s, s' \in \mathcal{S}$, it is usual to denote the probability of s' in the distribution $\mathcal{P}(s)$ as $\mathcal{P}(s, s')$. Given an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$ and a policy $\pi \colon \mathcal{S} \to \mathcal{D}(\mathcal{A})$, the Markov chain *induced by* π is $\mathcal{M}_\pi = (\mathcal{S}, \mathcal{P}_\pi)$, where $\mathcal{P}_\pi(s, s') = \sum_{a \in \mathcal{A}} \pi(s)(a) \cdot \mathcal{P}(s, a, s')$.

An infinite *trace* (also known as *path* in the literature) in a MC $\mathcal{M}$ is a sequence $\tau = s_0 s_1 s_2 \ldots$ where $\mathcal{P}(s_i, s_{i+1}) > 0$ for all $i \geq 0$. We denote the set of all infinite traces by $\Omega^{\mathcal{M}}$.

A *distance* in an MDP is a function $d \colon \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ such that for all $x, y, z \in \mathcal{S}$, we have

- Simmetry: $d(x, y) = d(y, x)$.
- Triangular inequality: $d(x, y) \leq d(x, z) + d(z, y)$.
- Identity: $d(x, y) = 0$ if and only if $x = y$.

A *ball* of radius $r > 0$ centered at state $s \in \mathcal{S}$ is the set

$$B_r(s) = \{s' \in \mathcal{S} \,:\, d(s, s') < r\}.$$

Similary, a *ball* of radius $r > 0$ centered at a set of states $S \subseteq \mathcal{S}$ is the set

$$B_r(S) = \{s' \in \mathcal{S} \,:\, \exists s \in S.\, d(s, s') < r\}.$$

Product MDP. Given two MPDs $\mathcal{M}_1 = (\mathcal{S}_1, \mathcal{A}_1, \mathcal{P}_1)$ and $\mathcal{M}_2 = (\mathcal{S}_2, \mathcal{A}_2, \mathcal{P}_2)$, the *product MDP of* $\mathcal{M}_1$ *and* $\mathcal{M}_2$ is the MDP $\mathcal{M} = (\mathcal{S}_1 \times \mathcal{S}_2, \mathcal{A}_1 \times \mathcal{A}_2, \mathcal{P})$, where the transition function $\mathcal{P}$ is defined as follows. For each $(s_1, s_2), (s_1', s_2') \in \mathcal{S}_1 \times \mathcal{S}_2$, and for each $(a_1, a_2) \in \mathcal{A}_1 \times \mathcal{A}_2$, the transition probability is defined as

$$\mathcal{P}\Big(\big(s_1, s_2\big), \big(a_1, a_2\big), \big(s_1', s_2'\big)\Big) = \mathcal{P}_1\big(s_1, a_1, s_1'\big) \cdot \mathcal{P}\big(s_2, a_2, s_2'\big).$$

Similarly, we can define the product between an MDP and a Markov chain. Given an MDP $\mathcal{M}_D = (\mathcal{S}_D, \mathcal{A}, \mathcal{P}_D)$ and a Markov chain $\mathcal{M}_C = (\mathcal{S}_C, \mathcal{P}_C)$, the *product MDP of* $\mathcal{M}_D$ *and* $\mathcal{M}_C$ is the MDP $\mathcal{M} = (\mathcal{S}_M \times \mathcal{S}_C, \mathcal{A}, \mathcal{P})$, where the transition function is defined as follows. For each $(s_D, s_C), (s_D', s_C') \in \mathcal{S}_D \times \mathcal{S}_C$ and each $a \in \mathcal{A}$, the transition probability is defined as

$$\mathcal{P}\Big(\big(s_D, s_C\big), a, \big(s_D', s_C'\big)\Big) = \mathcal{P}_D\big(s_D, a, s_D'\big) \cdot \mathcal{P}_C\big(s_C, s_C'\big).$$

2.4.1 Cylinder Set Construction

To define a probability measure over traces, we use the *cylinder set construction*. This is a standard construction in the literature; details can be found in [BK08, Chap. 10]. Let $\mathcal{M} = (\mathcal{S}, \mathcal{P})$ be a Markov chain. For a finite trace prefix $\omega = s_0 s_1 \dots s_n$, the *cylinder set* generated by ω, denoted $\texttt{Cyl}(\omega)$, is the set of all infinite traces starting with ω. Formally: $\texttt{Cyl}(\omega) = \{\omega' \in \Omega : \omega' \text{ begins with } \omega\}$. The probability of the cylinder set $\texttt{Cyl}(\omega)$ is defined as $\mathbb{P}^{\mathcal{M}}(\texttt{Cyl}(\omega)) = \prod_{i=0}^{n-1} \mathcal{P}(s_i, s_{i+1})$. The σ-algebra associated with $\mathcal{M}$, denoted by $\mathcal{F}^{\mathcal{M}}$, is the σ-algebra generated by all $\texttt{Cyl}(\omega)$, where ω is a finite trace prefix. With this construction, $(\Omega^{\mathcal{M}}, \mathcal{F}^{\mathcal{M}}, \mathbb{P}^{\mathcal{M}})$ is a probability space that lets us measure the probabilities of finite trace prefixes (as the probability of its corresponding cylinder set), and in the limit lets us measure the probability of infinite traces.

Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$ and $\pi \colon \mathcal{S} \to \mathcal{D}(\mathcal{A})$ be a policy. We cannot define a probability space on $\mathcal{M}$. To talk about probabilities in an MDP, we need to make a cylinder set construction on the Markov chain induced by an MDP and policy. The construction for MDPs is slightly different, as we include actions as part of the trace.

An infinite *state-action* trace is a sequence $\omega = s_0 a_0 s_1 a_1 \dots$, where $\pi(s_i)(a_i) > 0$ and $\mathcal{P}(s_i, a_i, s_{i+1}) > 0$ for all $i \geq 0$. We denote the set of all infinite state-action traces as $\Omega_\pi^{\mathcal{M}}$ and make the same cylinder set construction to define the sigma algebra on $\Omega_\pi^{\mathcal{M}}$ generated by all cylinder sets of finite state-action trace prefixes, and the corresponding probability measure, where for a given trace prefix $\omega = s_0 a_0 s_1 a_1 \dots s_n$, the probability of the cylinder set associated with it is $\mathbb{P}_\pi^{\mathcal{M}}(\texttt{Cyl}(\omega)) = \prod_{i=0}^{n-1} \pi(s_i)(a_i)\mathcal{P}(s_i, s_{i+1})$. We will denote the generated probability space as $(\Omega_\pi^{\mathcal{M}}, \mathcal{F}_\pi^{\mathcal{M}}, \mathbb{P}_\pi^{\mathcal{M}})$.

2.4.2 Reachability Properties

A *reachability property* is defined as the probability of reaching a given set of target states $T \subseteq S$ from an initial state $s \in \mathcal{S}$ under a policy π after at most

$k \in \mathbb{N} \cup \{\infty\}$ transitions. Formally, we define the reachability probability of T in $\mathcal{M}$ from s_0 using the policy π in less than k transitions as:

$$\mathbb{P}_\pi^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,T)) = \mathbb{P}_\pi^{\mathcal{M}}\left(\{\omega \in \Omega_\pi^{\mathcal{M}} \,:\, s_0 = s \text{ and } \exists n \leq k,\, s_n \in T\}\right). \quad (2.6)$$

Similarly, we may be interested in computing a reachability probability after a particular action has been fixed. The probability of reaching T in $\mathcal{M}$ from s after performing action $a \in \mathcal{A}$ is defined as:

$$\mathbb{P}_\pi^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,a,T)) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') \cdot \mathbb{P}_\pi^{\mathcal{M}}\left(\{\omega \in \Omega_\pi^{\mathcal{M}} \,:\, s_0 = s' \text{ and } \exists n \leq k,\, s_n \in T\}\right). \quad (2.7)$$

Sometimes, it is of interest to know the maximum and minimum values of the reachability probability when considering the space of all policies. We define these probabilities as:

$$\begin{aligned}
\mathbb{P}_{\min}^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,T)) &= \min_{\pi\colon \mathcal{S}\to\mathcal{D}(\mathcal{A})} \mathbb{P}_\pi^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,T)),\\
\mathbb{P}_{\max}^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,T)) &= \max_{\pi\colon \mathcal{S}\to\mathcal{D}(\mathcal{A})} \mathbb{P}_\pi^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,T)),\\
\mathbb{P}_{\min}^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,a,T)) &= \min_{\pi\colon \mathcal{S}\to\mathcal{D}(\mathcal{A})} \mathbb{P}_\pi^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,a,T)),\\
\mathbb{P}_{\max}^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,a,T)) &= \max_{\pi\colon \mathcal{S}\to\mathcal{D}(\mathcal{A})} \mathbb{P}_\pi^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,a,T)).
\end{aligned}$$

It may also be interesting to consider the maximum and minimum values when restricting to a certain subset of available policies $\Pi \subseteq \{\pi\colon \mathcal{S} \to \mathcal{D}(\mathcal{A})\}$. In such cases, the definitions are analogous and we denote them as

$$\begin{aligned}
\mathbb{P}_{\min|\Pi}^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,T)) &= \min_{\pi\in\Pi} \mathbb{P}_\pi^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,T)),\\
\mathbb{P}_{\max|\Pi}^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,T)) &= \max_{\pi\in\Pi} \mathbb{P}_\pi^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,T)),\\
\mathbb{P}_{\min|\Pi}^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,a,T)) &= \min_{\pi\in\Pi} \mathbb{P}_\pi^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,a,T)),\\
\mathbb{P}_{\max|\Pi}^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,a,T)) &= \max_{\pi\in\Pi} \mathbb{P}_\pi^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,a,T)).
\end{aligned}$$

Avoidance properties. With the same spirit, we define the *avoidance* probability as the complement of the reach probability. We use the notation $\texttt{Avoid}_{\leq k}(s,T) = \neg\, \texttt{Reach}_{\leq k}(s,T)$, and define it as

$$\mathbb{P}_\pi^{\mathcal{M}}(\texttt{Avoid}_{\leq k}(s,T)) = 1 - \mathbb{P}_\pi^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,T)). \quad (2.8)$$

We also consider the minimum and maximum probabilities as described before for avoidance probabilities. Since the avoidance property is the complement of the corresponding reachability property, the policy that maximizes one minimizes the other, and viceversa. That is:

$$\begin{aligned}
\mathbb{P}_{\min}^{M}(\texttt{Avoid}_{\leq k}(s,T)) &= 1 - \mathbb{P}_{\max}^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,T)), \qquad \text{and}\\
\mathbb{P}_{\max}^{M}(\texttt{Avoid}_{\leq k}(s,T)) &= 1 - \mathbb{P}_{\min}^{\mathcal{M}}(\texttt{Reach}_{\leq k}(s,T)).
\end{aligned}$$

The same reasoning applies in the case that an action has already been decided — $\mathbb{P}_{\min\,/\,\max}(\texttt{Avoid}_{\leq k}(s,a,T))$ — and the case when there is a restriction on the set of policies — $\mathbb{P}_{\min\,/\,\max|\Pi}(\texttt{Avoid}_{\leq k}(s,T))$.

Bounded and unbounded properties. When $k \in \mathbb{N}$, we say that these are *bounded reachability/avoidance* properties. When $k = \infty$, we say that these are *unbounded reachability/avoidance* properties. In such cases, we may drop the explicit reference to k in our notation, writing $\mathbb{P}^{\mathcal{M}}_{\pi}(\texttt{Reach}(s, T))$ instead of $\mathbb{P}^{\mathcal{M}}_{\pi}(\texttt{Reach}_{\leq\infty}(s, T))$ for unbounded reachability.

The probabilities for bounded and unbounded reachability can be computed using probabilistic model checking algorithms [Kat16].

2.4.3 Reinforcement Learning

Reinforcement learning (RL) [SB18] is a category of machine learning where an agent learns to select actions from observations through trial and error, with the goal of maximising the long-term returns defined by a reward function. A reinforcement learning problem is formalized with an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$.

In RL problems, the MDP is accompanied by a reward function $\mathcal{R} \colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. An RL agent executes a policy $\pi \colon \mathcal{S} \to \mathcal{D}(\mathcal{A})$ in the MDP. A *state-action trace* is a sequence of states, actions and rewards $\tau = s_0 a_0 r_1, s_1 a_1 r_2 \ldots$, where $s_0 s_1 \ldots$ is a trace in the MDP induced by π and $r_{i+1} = \mathcal{R}(s_i, a_i, s_{i+1})$.

The interaction between the environment and the agent generates state-action-reward traces as follows. At each step, the agent observes the current state $s_i \in \mathcal{S}$, selects an action $a_i \in \mathcal{A}$, the environment transitions to a next state s_{i+1}, sampled from the probability distribution $\mathcal{P}(s_i, a_i)$, and the agent receives a reward $\mathcal{R}(s_i, a_i, s_{i+1})$. Note that adding the rewards is just a formalism, since a state-action trace already determines the corresponding state-action-reward trace. The *discounted return* for a state-action-reward trace τ is $G(\tau) = \sum_{k=0}^{\infty} \gamma^t r_t$, where $\gamma \in [0, 1)$ is the *discount factor*. Note that if $\mathcal{R}$ is a bounded function, $\gamma < 1$ guarantees that $G(\tau)$ is finite for any τ. Since τ can be seen as an element of $\Omega^{\mathcal{M}}_{\pi}$, G is a random variable on $\Omega^{\mathcal{M}}_{\pi}$, so we can consider its expectation $\mathbb{E}(G)$. When it becomes important to state which policy π is being used to induce an MC and generate a probability space, we will write $\mathbb{E}_{\tau\sim\pi}(G(\tau))$.

The goal of the agent is to find a policy that maximises the expected discounted return. Formally, the goal of the agent is to find $\pi^* \colon \mathcal{S} \to \mathcal{D}(\mathcal{A})$ such that

$$\pi^* \in \underset{\pi \colon \mathcal{S} \to \mathcal{D}(\mathcal{A})}{\arg\max}\ \mathbb{E}_{\tau\sim\pi}[G(\tau)].$$

There are many algorithms to approximate the optimal policy from available traces; see [SPC23] for a recent survey. We will use in this thesis the Q-learning algorithm [WD92], which is one of the most classic approaches to the problem.

2.5 Classification Problems and Fairness

Classification problems are a standard setting in supervised machine learning where there is an input space $\mathcal{X} \subseteq \mathbb{R}^n$, a discrete set of labels $\mathcal{Y} \subseteq \mathbb{N}$ and a ground truth distribution $\theta \in \mathcal{D}(\mathcal{X} \times \mathcal{Y})$. An element $x = (x_1, \ldots, x_n) \in \mathcal{X}$ is an *instance*, and each of the x_i's are the different *features*.

The *classification problem* consists on finding $f\colon \mathcal{X} \to \mathcal{Y}$ that minimizes the expected loss, defined as $\mathcal{L}(f) = \mathbb{E}_{(x,y)\sim\theta}\big[\mathbb{1}[y \neq f(x)]\big]$, when given a set of samples $(x_0, y_0), \ldots, (x_N, y_N)$ sampled from θ.

In some problems, there are concrete features of the instances that are considered *protected* or *sensitive* features, and it is of utmost importance to protect against bias with respect to those features. For example, in the problem of screening applications for a job, one of the protected features may be the applicant's gender. There are many metrics used to determine whether a classifier is biased with respect to sensitive features. When talking about fairness with respect to a given feature, it is useful to partition the input space into $\mathcal{X} = \mathcal{F} \times \mathcal{G}$, where $\mathcal{G}$ represents the sensitive feature, and $\mathcal{F}$ represents the rest of the features. It is also useful to think of X as a random variable on $\mathcal{X}$ that follows the input part of the ground truth distribution θ, and $f(X)$ as a random variable on the label space $\mathcal{Y}$. Similarly, $X = (F, G)$, where F is a random variable on $\mathcal{F}$ and G is a random variable on $\mathcal{G}$. In this thesis, we will use fairness metrics based on demographic parity and equal opportunity that assume the sensitive feature can only take a finite set of values, i.e., $\mathcal{G} = \{g_1, \ldots, g_k\}$. A classifier $f\colon \mathcal{F} \times \mathcal{G} \to \mathcal{Y}$ satisfies *demographic parity* (DP) if for all $i \in \{1, \ldots, k\}$, we have $\mathbb{E}[f(X) \mid G = g_i] = \mathbb{E}[f(X)]$. When $\mathcal{Y} = \{0, 1\}$, a classifier $f\colon \mathcal{F} \times \mathcal{G} \to \mathcal{Y}$ satisfies *equal opportunity* (EqOpp) if for all $i \in \{1, \ldots, k\}$, we have $\mathbb{E}[f(X) \mid G = g_i,\, y = 1] = \mathbb{E}[f(X) \mid y = 1]$.

The literature on enforcing fairness properties in classification problems is vast and rich (see [BHN23] for a recent account of the state of the art).

Chapter 3

Reactive Decision Making Framework

> *If I seem to wander, if I seem to stray, remember that true stories seldom take the straightest way.* — Patrick Rothfuss, The Name of the Wind.

3.1 Motivation and Outline

This thesis presents different lines of work with the common motivation of advancing trust in autonomous systems, using different formal models. While each chapter can be seen as a standalone contribution towards this goal, the formal models and methods used in each chapter can be encompassed as part of a general framework.

In this chapter, we introduce the reactive decision-making framework, which generalizes the many models used throughout the thesis. Following the general definition, we justify how safety games, MDPs, and classification problems can be expressed as particular cases in this framework.

We also introduce a generalized definition of shielding, a method that is used in most of the chapters in this thesis. We present a unified definition and justify how this adapts to different notions of shielding in the literature.

Outline. In Section 3.2, we introduce the reactive decision-making framework and show how other formalizations used in the paper can be viewed as particular cases of it. In Section 3.3 we introduce a generalized definition of shielding and in Section 3.4 we show how classical notions of shielding can be expressed in this generalized framework.

Declaration of sources. This chapter is the original work of the author of this thesis and, at the time of writing, remains unpublished.

3.2 Reactive Decision-Making

The *reactive decision-making* framework is an abstract framework that models the different problems presented in this thesis. In a reactive decision-making framework, an agent interacts with an environment, depicted in Figure 3.1. There is a set of observations $\mathcal{O}$, controlled by the environment, and a set of actions $\mathcal{A}$, controlled by the agent. An *environment* is a tuple $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$, where $\mathscr{T} \colon (\mathcal{O} \times \mathcal{A})^* \to \mathcal{D}(\mathcal{O})$. We call $\mathscr{T}$ the *environment transition function* . An *agent* is a tuple $Ag = (\mathcal{O}, \mathcal{A}, \pi)$, where $\pi \colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \to \mathcal{D}(\mathcal{A})$. We call π the *agent policy function.* Given $\mathcal{O}$ and $\mathcal{A}$, the set of all policies is $\texttt{Pol}(\mathcal{O}, \mathcal{A})$. An environment and an agent are *compatible* if they share the same set of observations and actions.

An environment $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$ is *deterministic* if for all input $\tau \in (\mathcal{O} \times \mathcal{A})^*$, the support of the environment transition function, $\text{Supp}(\mathscr{T}(\tau))$, has only a single element. In such cases, we would write the environment transition function as $\mathscr{T} \colon (\mathcal{O} \times \mathcal{A})^* \to \mathcal{O}$. An agent $Ag = (\mathcal{O}, \mathcal{A}, \pi)$ is *deterministic* if for all input $(\tau, o) \in (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O}$, $\text{Supp}(\pi(\tau, o))$ has only a single element. In such cases, we would write the policy function as $\pi \colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \to \mathcal{A}$. Given a policy $\pi \colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \to \mathcal{D}(\mathcal{A})$, a *determinization of* π is any deterministic policy $\pi_{det} \colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \to \mathcal{A}$, such that for all $(\tau, o) \in (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O}$, we have that $\pi_{det}(\tau, o) \in \text{Supp}\,(\pi(\tau, o))$.

An environment $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$ is *memoryless* if the transition function depends only on the last pair of action and observation, that is, if for all $(o, a) \in \mathcal{O} \times \mathcal{A}$ and for all $\tau, \tau' \in (\mathcal{O} \times \mathcal{A})^*$, we have $\mathscr{T}(\tau \cdot (o, a)) = \mathscr{T}(\tau' \cdot (o, a))$. Similarly, an agent $Ag = (\mathcal{O}, \mathcal{A}, \pi)$ is *memoryless* if the policy function depends only on the last observation. Formally, if for all $o \in \mathcal{O}$ and for all $\tau, \tau' \in (\mathcal{O} \times \mathcal{A})^*$, $\pi(\tau, o) = \pi(\tau', o)$.

Note that the definitions we give of agents and environments are strictly functional, so we do not consider how the elements are internally designed. For example, an agent may be designed as an automaton, with internal states and internal transition functions. For the general theory presented in this chapter, we do not model such details of the inner structure; we just study agents and environments as abstract functions that produce an output when given an input.

An *observation-action trace* is a (finite or infinite) sequence of observations and actions $\tau = (o_0 a_0), (o_1 a_1), \cdots \in (\mathcal{O} \times \mathcal{A})^\infty$. Given an environment $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$ and an agent $Ag = (\mathcal{O}, \mathcal{A}, \pi)$, an observation-action trace is *valid* if for all $k < |\tau|$, we have that $o_{k+1} \in \text{Supp}(\mathscr{T}(\tau_{:k}))$ and $a_{k+1} \in \text{Supp}(\pi(\tau_{:k}, o_{k+1}))$. In other words, a trace is valid if it could have been produced by the pair agent-environment.

An *observation trace* is a (finite or infinite) sequence of observations $\tau_O = o_0, o_1, \ldots \in \mathcal{O}^\infty$. An *action trace* is a (finite or infinite) sequence of actions $\tau_A = a_0 a_1, \ldots \mathcal{A}^\infty$. Given an observation trace τ_O and an action trace τ_A, we can produce an observation-action trace $\tau = (o_0 a_0), (o_1 a_1), \cdots \in (\mathcal{O} \times \mathcal{A})^\infty$ by interlacing them. We will denote this by $\tau_O || \tau_A$.

Given an environment $\mathscr{E}$, an observation trace τ_O is *valid* if there exists an agent Ag and an action trace τ_A such that $\tau_O || \tau_A$ is valid for $\mathscr{E}$ and Ag. Given an

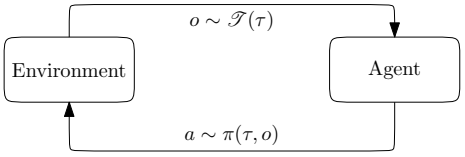


Figure 3.1: Reactive decision-making framework.

agent Ag, an action trace τ_A is *valid* if there exists an environment $\mathscr{E}$ and an observation trace such that $\tau_O||\tau_A$ is valid for $\mathscr{E}$ and Ag.

Probability of traces

Given an environment $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$ and an agent $Ag = (\mathcal{O}, \mathcal{A}, \pi)$, we can make a similar cylinder set construction as in Section 2.4.1 to define a probability measure on the space of finite and infinite traces.

Let $\Omega^{\mathscr{E},Ag}$ be the set of all observation-action traces. Let $\omega = o_1 a_1, \ldots, o_n a_n$ be a finite prefix, the cylinder set generated by ω is

$$\texttt{Cyl}(\omega) = \{\omega' \in \Omega^{\mathscr{E},Ag} \ : \ \omega' \text{ begins with } \omega\}.$$

The probability of the cylinder set $\texttt{Cyl}(\omega)$ is

$$\mathbb{P}^{\mathscr{E},Ag}(\texttt{Cyl}(\omega)) = \prod_{i=0}^{n} \mathscr{T}\left(\omega_{[:i-1]}\right)(o_i) \cdot \pi\left(\omega_{[:i-1]}, o_i\right)(a_i).$$

Let $\mathcal{F}^{\mathscr{E},Ag}$ be the σ-algebra generated by the cylinder sets of all finite traces ω. The space $(\Omega^{\mathscr{E},Ag}, \mathcal{F}^{\mathscr{E},Ag}, \mathbb{P}^{\mathscr{E},Ag})$ is a probability space.

Sometimes it is useful to talk about traces of a given length and assign probabilities to them, instead of thinking of infinite traces. Let $\Omega_k^{\mathscr{E},Ag}$ be the set of traces of length equal to k. Since $\Omega_k^{\mathscr{E},Ag}$ is countable, we can use $\mathcal{F}_k^{\mathscr{E},Ag} = 2^{\Omega_k^{\mathscr{E},Ag}}$ as our σ-algebra. Let $\omega \in \Omega_k^{\mathscr{E},Ag}$, its probability is defined as $\mathbb{P}_k^{\mathscr{E},Ag}(\omega) = \mathbb{P}^{\mathscr{E},Ag}(\texttt{Cyl}(\omega))$. Then $(\Omega_k^{\mathscr{E},Ag}, \mathcal{F}_k^{\mathscr{E},Ag}, \mathbb{P}_k^{\mathscr{E},Ag})$ is a probability space.

Note that in $\Omega_k^{\mathscr{E},Ag}$, a trace ω is valid if and only if $\mathbb{P}_k^{\mathscr{E},Ag}(\omega) \neq 0$. This is not true for infinite traces.

3.2.1 Deterministic Two-player Games

In this section, we show how a deterministic two-player game corresponds to a memoryless environment in the reactive decision-making framework, where the sets of observations and actions are both finite, and all probability distributions over observations and actions are uniform over their support.

A deterministic two-player game is formalized as a tuple $\mathcal{G} = (S, s_0, S_{env}, S_{ag}, \mathcal{A}, \mathcal{T}, \text{Acc})$ (Section 2.3). The set S_{ag} corresponds to the set of observations $\mathcal{O}$ in the reactive decision-making framework, and the set S_{env} corresponds to $\mathcal{O} \times \mathcal{A}$. The set of actions $\mathcal{A}$ is the same for both formalisms. Given $o \in S_{ag}$, and $\sigma \in \mathcal{A}$, the

transition is trivially $o \xrightarrow{\sigma} (o, \sigma)$. Given $(o, \sigma) \in S_{env}$, the allowed transitions $(o, \sigma) \xrightarrow{u} o'$ are those for which $o' \in \mathscr{T}(o, \sigma)$.

Note that a play in the safety game, $\tau = [s_0, s_1, s_2, \dots]$, always has the form $\tau = [(o_0, \sigma_0), o_1, (o_1, \sigma_1), o_2, (o_2, \sigma_2), \dots]$, where $s_{2i} = (o_i, \sigma_i)$ and $s_{2i+1} = o_{i+1}$, so it naturally corresponds to an observation-action trace.

In a deterministic two-player game, the concrete value of the probability of a given observation or action is unimportant, it is only relevant whether the probability is non-zero. Therefore, the transition functions can be expressed as $\mathscr{T} \colon \mathcal{O} \times \mathcal{A} \to 2^{\mathcal{O}}$ and $\pi \colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \to 2^{\mathcal{A}}$, instead of $\mathscr{T} \colon \mathcal{O} \times \mathcal{A} \to \mathcal{D}(2^{\mathcal{O}})$ and $\pi \colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \to \mathcal{D}(2^{\mathcal{A}})$. Note that this is an *interpretation* of the model. All allowed transitions are considered to have the same probability because the only thing that matters is whether the probability is non-zero. In the game setting, the environment is considered adversarial, so any action with a non-zero probability (no matter how low), needs to be considered when computing strategies for the agent.

3.2.2 Markov Decision Processes

In the reactive decision-making framework, a Markov decision process corresponds to a memoryless environment. In typical MDP notation, as introduced in Section 2.4, observations are called *states*, denoted as $\mathcal{S}$ (instead of $\mathcal{O}$). Since the environment transition function is memoryless, it is written with $(\mathcal{S} \times \mathcal{A})$ as its domain – instead of $(\mathcal{S} \times \mathcal{A})^*$ –, denoted by $\mathcal{P} \colon \mathcal{S} \times \mathcal{A} \to \mathcal{D}(\mathcal{S})$, and called the *probabilistic transition function*.

3.2.3 Classification Problems

In classification problems, the observation space is the input space of the problem, i.e., following the notation of Section 2.5, $\mathcal{O} = \mathcal{X}$. It is standard to assume that there is a single distribution from which problem instances are sampled. This would correspond with an environment transition function that does not depend in any way on the current trace. This is a stronger condition as being memoryless since we are imposing that for all $\tau, \tau' \in (\mathcal{O} \times \mathcal{A})^*$, $\mathscr{T}(\tau) = \mathscr{T}(\tau')$. Therefore, we can simply represent the environment as a distribution $\Theta_{\mathcal{X}} \in \mathcal{D}(\mathcal{O})$. There is also literature studying classification problems where the data distribution changes according to the actions (accepts or rejects) given by a classifier. This phenomenon has been studied as it relates to fairness in [D'A+20]. Our framework is well adapted to such cases, as it would mean just keeping a full environment transition function.

Another characteristic of classification problems is that the set of actions represents the labels available for classification, i.e. $\mathcal{A} = \mathcal{Y}$.

3.2.4 Delayed Observations

In some use cases, it is interesting to consider agents that work with uncertain observations. We are particularly interested in the case of agents that receive information about observations produced by the environment delayed by a certain number of steps, in the same spirit as safety games with delayed input

presented in Section 2.3.2. These delays are a common challenge when dealing with asynchronous control architectures. In this section, we provide two formalizations of the delayed framework and show that they are equivalent.

The first formalization is intended to be an intuitive one, while the second formalization is an operational one, that we will use to develop the theory of shielding resilient to delayed observations. We will show that both formulations are, in fact, equivalent in Section 3.2.4.3.

3.2.4.1 Delayed Observations through Modified Agents

We can formalize delays as part of the reactive decision-making framework by considering a variation of the actions available to the agent, as depicted in Figure 3.2. The environment is the same, and samples are the next observation from the full observation-action trace. On the other side, the agent does not have access to the last δ observations produced by the environment. In the initial transient phase, the agent only has access to its own action record. After the transient phase, after δ steps, the agent starts receiving the first observations, entering the steady observation phase.

Given an environment $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$, an agent under delay δ has a policy function $\pi \colon \texttt{Delayed}_\delta(\mathcal{O}, \mathcal{A}) \to \mathcal{D}(\mathcal{A})$, where

$$\texttt{Delayed}_\delta(\mathcal{O}, \mathcal{A}) = \mathcal{A}^{\leq \delta} \cup \left\{ (o_1, a_1, \ldots, o_n, a_n, a_{n+1}, \ldots, a_{n+\delta-1}) \; : \; o_i \in \mathcal{O},\, a_i \in \mathcal{A},\, n \geq 0 \right\}. \quad (3.1)$$

In other words, the agent has either access to a trace longer than δ with the δ last observations removed (steady observation phase), or it has access to a trace shorter than δ composed of only actions (transient phase). This is similar to how we defined strategies in games under delay in Equation (2.5).

3.2.4.2 Delayed Observations through Restriction to Agnostic Agents

While Figure 3.2 is a more intuitive formulation, we will use an equivalent formulation that is more operational, in terms of considering only agents restricted to a certain domain.

Definition 3.1 (Delayed-observation agent)**.** Let $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$ be an environment and $\delta \geq 0$. An agent $Ag = (\mathcal{O}, \mathcal{A}, \pi)$ *works with observations delayed*

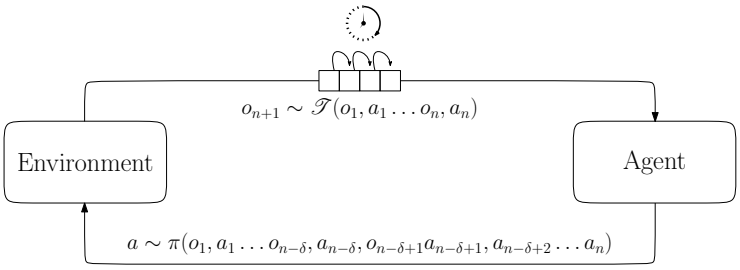


Figure 3.2: Reactive decision-making framework with delayed observations.

by δ if it is agnostic to the last δ observations. That is, if for all $\tau \in (\mathcal{O} \times \mathcal{A})^*$, all $(a_1, \ldots, a_\delta) \in \mathcal{A}^\delta$, and all $(o_1, \ldots, o_\delta), (o'_1, \ldots, o'_\delta) \in \mathcal{O}^\delta$

$$\pi\Big(\big(\tau \cdot (o_1, a_1), \ldots, (o_{\delta-1}, a_{\delta-1})\big), o_\delta\Big) = \pi\Big(\big(\tau \cdot (o'_1, a_1), \ldots, (o'_{\delta-1}, a_{\delta-1})\big), o'_\delta\Big) \tag{3.2}$$

We denote the set of agents working with observations delayed by δ as $\Pi^{\circledcirc}_\delta$.

3.2.4.3 Equivalence

The equivalence between Definition 3.1 and an agent with domain $\texttt{Delayed}_\delta(\mathcal{O}, \mathcal{A})$ as in Equation (3.1) stems from the fact that if an agent π with domain $(\mathcal{O} \times \mathcal{A})^* \times \mathcal{O}$ is agnostic to the last δ observations, it is fully determined by the agent $\pi' \colon \texttt{Delayed}_\delta(\mathcal{O}, \mathcal{A}) \to \mathcal{D}(\mathcal{A})$, and vice versa.

For every $\tau \in (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O}$, we can factor it as $\tau = \big((o_1, a_1), \ldots, (o_n, a_n), \ldots, (o_{n+\delta-1}, a_{n+\delta-1}), o_\delta\big)$, and then define $\pi(\tau)$ as

$$\pi(\tau) = \pi'\big((o_1, a_1), \ldots, (o_n, a_n), a_{n+1}, \ldots, a_{n+\delta-1}\big).$$

For the backwards direction, given $\tau \in \texttt{Delayed}_\delta(\mathcal{O}, \mathcal{A})$, we can determine $\pi'(\tau)$ as follows. The element $\tau \in \texttt{Delayed}_\delta(\mathcal{O}, \mathcal{A})$ is of the form $\tau = (o_1, a_1, \ldots, o_n, a_n, a_{n+1}, \ldots, a_{n+\delta-1})$, for some $n \in \mathbb{N}$, $o_i \in \mathcal{O}$ and $a_i \in \mathcal{A}$. We choose δ arbitrary observations $(o_{n+1}, \ldots, o_{n+\delta}) \in \mathcal{O}^\delta$ and define $\pi'(\tau)$ as

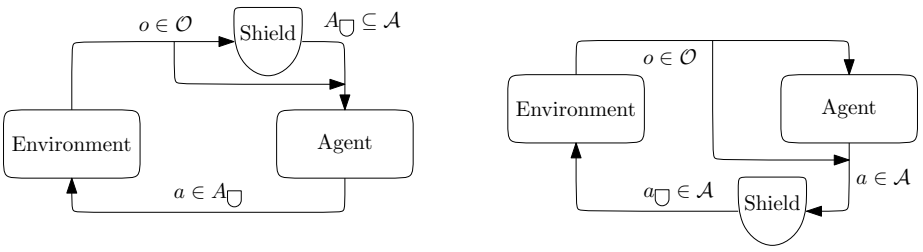
$$\pi'(\tau) = \pi\Big(\big((o_1, a_1), \ldots, (o_{n+\delta-1}, a_{n+\delta-1})\big), o_{n+\delta}\Big).$$

The distribution corresponding to $\pi'(\tau)$ is well defined, i.e., does not depend on the choice of observations $o_{n+1}, \ldots, o_{n+\delta}$, by virtue of Equation (3.2).

3.3 Shielding

3.3.1 Definitions

A *shield* is an element that modifies the behaviour of an agent, filtering actions either before (pre-shield) or after (post-shield) the agent decides on them. Figure 3.3a illustrates the introduction of a pre-shield in a reactive decision-making pair, while Figure 3.3b illustrates it if for a post-shield. We use the symbol $\bigcup$, intended to be read as "shield".



(a) Pre-shield. (b) Post-shield.

Figure 3.3: Shielded reactive decision-making framework.

Definition 3.2 (pre-shield-ready agent)**.** A *pre-shield-ready agent* is a function $Ag_{pre}\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}\times 2^{\mathcal{A}} \to \mathcal{D}(\mathcal{A})$, such that for any input $(\tau, o, A) \in (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}\times 2^{\mathcal{A}}$, the agent only proposes actions in A, i.e., $\mathrm{Supp}(Ag_{pre}(\tau, o, A)) \subseteq A$.

Definition 3.3 (Pre-shield)**.** A *pre-shield* is a function $\bigcup\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O} \to 2^{\mathcal{A}}$, that, given a trace and an observation, produces a set of allowed actions. A pre-shield together with a pre-shield-ready agent form a new agent $Ag_{\bigcup}\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O} \to \mathcal{D}(\mathcal{A})$, defined as $Ag_{\bigcup}(\tau, o) = Ag_{pre}(\tau, o, \bigcup(\tau, o))$.

Given a pre-shield-ready agent $Ag\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}\times 2^{\mathcal{A}} \to \mathcal{D}(\mathcal{A})$, it induces a regular agent $Ag_{reg}\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O} \to \mathcal{D}(\mathcal{A})$, by considering only unrestricted actions. That is, for any input $(\tau, o) \in (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}$, Ag_{reg} is defined as $Ag_{reg}(\tau, o) = Ag(\tau, o, \mathcal{A})$. We call this the *regular agent induced by* Ag.

Definition 3.4 (Post-shield)**.** A *post-shield* is a function $\bigcup\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}\times\mathcal{A} \to \mathcal{A}$ that given a trace, an observation and an action, produces a new distribution of allowed actions. A shield together with an agent Ag_{pos} form a new agent $Ag_{\bigcup}\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O} \to \mathcal{D}(\mathcal{A})$, defined as $Ag_{\bigcup}(\tau, o) = \bigcup(\tau, o, Ag_{pos}(\tau, o))$.

Note that the probabilistic nature of a post-shielded agent $Ag_{\bigcup}$ does not come from the shield itself, but from the fact that given (τ, o), $Ag_{pos}(\tau, o)$ is a distribution of actions, which induces a distribution of shielded actions $\bigcup(\tau, o, Ag_{pos}(\tau, o))$.

As mentioned for the environment-agent construction, our definitions of shields are strictly functional, so we develop the general theory of shielding without discussing how these functions are to be computed or implemented. The discussions on implementation are specific for each type of shielding and are a substantial part of the content of Section 3.4 and Chapter 6.

3.3.2 Shielding Induced by Agents

A trivial way of building shields is through the *induced shield construction*.

Definition 3.5 (Induced pre-shield)**.** Let $Ag\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O} \to \mathcal{D}(\mathcal{A})$ be an agent. The *pre-shield induced by* Ag is the pre-shield $\bigcup^{pre}_{Ag}$, defined for any $(\tau, o) \in (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}$ as

$$\bigcup^{pre}_{Ag}(\tau, o) = \mathrm{Supp}\left(Ag(\tau, o)\right).$$

To build the induced post-shield, we also need a determinization of the agent.

Definition 3.6 (Induced post-shield)**.** Let $Ag\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O} \to \mathcal{D}(\mathcal{A})$ be an agent, and let Ag_{det}, a determinization of Ag. The *post-shield induced by* Ag *and* Ag_{det} is the post-shield $\bigcup^{pos}_{Ag}$, defined for any $(\tau, o, a) \in (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}\times\mathcal{A}$ as

$$\bigcup^{pos}_{Ag, Ag_{det}}(\tau, o, a) = \begin{cases} a & \text{if } a \in \mathrm{Supp}\left(Ag(\tau, o)\right) \\ Ag_{det}(\tau, o) & \text{otherwise.} \end{cases}$$

Any agent shielded with these shields will produce only traces that are valid for Ag.

With a similar spirit, we can also build agents from given shields. However, a shield may be compatible with many agents. We formalize this concept with the following definitions.

Definition 3.7 (Agents associated with a pre-shield)**.** Let $\shield\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}\to 2^{\mathcal{A}}$ be a pre-shield. An agent $Ag\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}\to\mathcal{D}(\mathcal{A})$ is *associated with* $\shield$ if $\shield^{pre}_{Ag}=\shield$.

Definition 3.8 (Agents associated with a post-shield)**.** Let $\shield\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}\times\mathcal{A}\to\mathcal{A}$. An agent $Ag\colon (\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}\to\mathcal{D}(\mathcal{A})$ is *associated with* $\shield$ if there exists a determinization of Ag, named Ag_{det}, such that $\shield^{pos}_{Ag,Ag_{det}}=\shield$.

For both pre- and post-shields, the set of agents associated with a shield $\shield$ is denoted by $\Pi_{\shield}$.

Given a set of agents Π, the set of shields associated with Π is Σ_Π defined as

$$\Sigma_\Pi=\left\{\shield\,:\,\Pi_{\shield}\subseteq\Pi\right\}. \tag{3.3}$$

We can characterize certain aspects of a shield by its set of associated agents. For example, we have the following technical result, that will be used to characterize shields in a delayed observation setting as shields whose set of associated agents is inside $\Pi^{\oslash}_{\delta}$ for some $\delta\in\mathbb{N}$.

Lemma 3.1. *Let $\mathscr{E}=(\mathcal{O},\mathcal{A},\mathscr{T})$ be an environment, $\delta\in\mathbb{N}$, and let $\shield$ be a shield such that $\Pi_{\shield}\subseteq\Pi^{\oslash}_{\delta}$. Then $\shield$ is also agnostic to the last δ observations. That is, for all $\tau\in(\mathcal{O}\times\mathcal{A})^*$, all $(a_1,\dots,a_{\delta-1})\in\mathcal{A}^\delta$, and all $(o_1,\dots,o_\delta),(o'_1,\dots,o'_\delta)\in\mathcal{O}^\delta$ we have:*

1. *If $\shield$ is a pre-shield, i.e., $\shield\colon(\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}\to 2^{\mathcal{A}}$, then*

$$\shield\Big(\big(\tau\cdot(o_1,a_1),\dots,(o_{\delta-1},a_{\delta-1}),\big),o_\delta\Big)=\shield\Big(\big(\tau\cdot(o'_1,a_1),\dots,(o'_{\delta-1},a_{\delta-1}),\big),o'_\delta\Big). \tag{3.4}$$

2. *If $\shield$ is a post-shield, i.e., $\shield\colon(\mathcal{O}\times\mathcal{A})^*\times\mathcal{O}\times\mathcal{A}\to\mathcal{A}$, then for all action $a_\delta\in\mathcal{A}$*

$$\shield\Big(\big(\tau\cdot(o_1,a_1),\dots,(o_{\delta-1},a_{\delta-1}),\big),o_\delta,a_\delta\Big)=\shield\Big(\big(\tau\cdot(o'_1,a_1),\dots,(o'_{\delta-1},a_{\delta-1}),\big),o'_\delta,a_\delta\Big). \tag{3.5}$$

Proof. This results follows directly from the definitions. Suppose $\shield$ is a pre-shield and let $Ag\in\Pi_{\shield}$. Since Ag is associated with $\shield$, we have that for any trace $\tau\in(\mathcal{O}\times\mathcal{A})^*$ and any observation $o\in\mathcal{O}$, $\shield(\tau,o)=\mathrm{Supp}(Ag(\tau,o))$ (by Definition 3.5). Using the hypothesis that $\Sigma_{\shield}\subseteq\Pi^{\oslash}_{\delta}$ we have that $Ag\in\Pi^{\oslash}_{\delta}$, so the condition in Equation (3.4) is satisfied by virtue of Definition 3.1. The proof is analogous for the case of $\shield$ being a post-shield. □

3.3.3 Correctness

The idea behind shielding is that a shielded agent should satisfy a certain desirable property, while being as close as possible to the original agent. In its

most general form, a property can be defined as a language of correct traces $\mathcal{L} \subseteq (\mathcal{O} \times \mathcal{A})^\infty$.

Definition 3.9 (Correctness)**.** Let $\mathcal{L} \subseteq (\mathcal{O} \times \mathcal{A})^\infty$. An agent is *correct with respect to* $\mathcal{L}$ if any trace τ that is valid under Ag is in $\mathcal{L}$. A pre-shield-ready policy $Ag\colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \times 2^{\mathcal{A}} \to \mathcal{D}(\mathcal{A})$ is correct if the regular policy induced by Ag is correct. A (pre- or post-) shield is *correct* with respect to $\mathcal{L}$ if any shielded policy $Ag_{\bigcup}$ is correct with respect to $\mathcal{L}$.

A correct shield always exists as long as a correct agent exists, since we can build a shield that strictly follows a correct agent, using the *induced shield* construction from the previous section. Therefore, if we pick Ag to be correct with respect to a specification $\mathcal{L} \subseteq (\mathcal{O} \times \mathcal{A})^\infty$, then any agent shielded by $\bigcup_{Ag}^{pre}$ or $\bigcup_{Ag, Ag_{det}}^{pos}$ will also be correct with respect to $\mathcal{L}$.

While it is good to know that a correct shield always exists, it is not very useful to build a shield that is trivially correct by not caring about the underlying agent.

3.3.4 Interference

Another desirable property in shields is that they minimize their interference with the agent being shielded. Intuitively, a post-shield interferes with its agent every time that it overwrites the agent's action. For pre-shields the intuition is a bit different. A pre-shield interferes with an agent every time that the action the agent would take if it had no restrictions is not allowed by the shield. We formalize these ideas in the following definition.

Definition 3.10 (Interference set of a shield)**.** Let $\bigcup\colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \to 2^{\mathcal{A}}$ be a pre-shield, and $Ag_{pre}\colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \times 2^{\mathcal{A}} \to \mathcal{D}(\mathcal{A})$ be a pre-shield-ready agent. The *interference set* of $\bigcup$ applied to Ag_{pre} is

$$\mathbb{J}_{\bigcup}(Ag_{pre}) = \left\{(\tau, o) \in (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \;:\; \exists a \in \text{Supp}\left(Ag_{pre}(\tau, o, \mathcal{A})\right),\, a \notin \bigcup(\tau, o)\right\}.$$

Let $\bigcup\colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \times \mathcal{A} \to \mathcal{A}$ be a post-shield, and $Ag_{pos}\colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \to \mathcal{D}(\mathcal{A})$ be an agent. The *interference set* of $\bigcup$ applied to Ag_{pos} is

$$\mathbb{J}_{\bigcup}(Ag_{pos}) = \left\{(\tau, o) \in (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \;:\; \exists a \in \text{Supp}\left(Ag_{pos}(\tau, o)\right),\, a \neq \bigcup(\tau, o, a)\right\}.$$

A non-interfering shield always exists, independent of whether there exist correct agents or not, we call it the *transparent shield*. It is built as follows. The transparent post-shield is $\bigcup_{trans}^{pos}$ defined for any $(\tau, o, a) \in (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \times \mathcal{A}$ as

$$\bigcup_{trans}^{pos}(\tau, o, a) = a.$$

The transparent pre-shield is $\bigcup_{trans}^{pre}$ defined for any $(\tau, o) \in (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O}$ as

$$\bigcup_{trans}^{pre}(\tau, o) = \mathcal{A}.$$

As in the case of the trivially correct shield, this transparent shield is an interesting theoretical construct, but it is not of much utility, because it is generally not correct.

Definition 3.11 (Equivalence modulo interference)**.** Let $\bigcirc$, and $\bigcirc'$ be two post-shields. We say that $\bigcirc$ and $\bigcirc'$ are *equivalent modulo interferences*, denoted it by $\bigcirc \equiv_i \bigcirc'$, if for all $(\tau, o, a) \in (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \times \mathcal{A}$, we have that $\bigcirc(\tau, o, a) = a$ if and only if $\bigcirc'(\tau, o, a) = a$.

Interference sets are useful because a shield is determined by its interference sets, and the interference set of a shield induced by an agent Ag on that same agent Ag is always empty. We formalize these properties in the following result.

Lemma 3.2. *Let $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$ be an environment and Π be a set of agents. The following are true.*

1. *Let $\bigcirc, \bigcirc'$ be two pre-shields, $\bigcirc, \bigcirc' \subseteq \Sigma_\Pi$, and such that for any pre-shield-ready agent $Ag \in \Pi$, we have $\mathbb{J}_{\bigcirc}(Ag) = \mathbb{J}_{\bigcirc'}(Ag)$. Then $\bigcirc = \bigcirc'$.*

2. *Let $\bigcirc, \bigcirc'$ be two post-shields, $\bigcirc, \bigcirc' \subseteq \Sigma_\Pi$, and such that for any agent $Ag \in \Pi$, we have $\mathbb{J}_{\bigcirc}(Ag) = \mathbb{J}_{\bigcirc'}(Ag)$. Then $\bigcirc \equiv_i \bigcirc'$.*

3. *Let $Ag \colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \times 2^{\mathcal{A}} \to \mathcal{D}(\mathcal{A})$ be a pre-shield-ready agent, Ag_{reg} be the regular agent induced by Ag and $\bigcirc^{pre}_{Ag}$ be the shield induced by Ag_{reg}. Then $\mathbb{J}_{\bigcirc^{pre}_{Ag}}(Ag) = \emptyset$.*

4. *Let $Ag \colon (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O} \to \mathcal{D}(\mathcal{A})$ be an agent and $\bigcirc^{pos}_{Ag, Ag_{det}}$ be any shield induced by Ag. Then $\mathbb{J}_{\bigcirc^{pos}_{Ag, Ag_{det}}}(Ag) = \emptyset$.*

Proof. We argue all cases by contradiction.

1. Suppose $\bigcirc \neq \bigcirc'$, and let (τ, o) be an input such that $\bigcirc(\tau, o) \neq \bigcirc'(\tau, o)$. Without loss of generality, we may assume there exists $a \in \mathcal{A}$ such that $a \in \bigcirc(\tau, o)$ and $a \notin \bigcirc'(\tau, o)$. Consider a pre-shield-ready agent Ag such that $\mathrm{Supp}\,(Ag(\tau, o)) = \{a\}$. Since $a \in \mathrm{Supp}\,(Ag(\tau, o)) = \{a\}$, but $a \notin \bigcirc'(\tau, o)$, we have that $(\tau, o) \in \mathbb{J}_{\bigcirc'}(Ag)$. However, since $\mathrm{Supp}\,(Ag(\tau, o)) \subseteq \bigcirc(\tau, o)$, we have that $(\tau, o) \notin \mathbb{J}_{\bigcirc}(Ag)$, contradicting $\mathbb{J}_{\bigcirc}(Ag) = \mathbb{J}_{\bigcirc'}(Ag)$.

2. Suppose $\bigcirc \not\equiv_i \bigcirc'$, and let (τ, o, a) be an input making them so. Without loss of generality, we may assume that $\bigcirc'(\tau, o, a) \neq a$, and $\bigcirc(\tau, o, a) = a$. Consider an agent Ag such that $Ag(\tau, o) = a$. As in the previous point, this implies that $(\tau, o) \in \mathbb{J}_{\bigcirc'}(Ag)$, but on the other hand $(\tau, o) \notin \mathbb{J}_{\bigcirc}(Ag)$, contradicting $\mathbb{J}_{\bigcirc}(Ag) = \mathbb{J}_{\bigcirc'}(Ag)$.

3. Suppose $(\tau, o) \in \mathbb{J}_{\bigcirc^{pre}_{Ag}}(Ag)$. Then there would exist $a \in \mathrm{Supp}\,(Ag(\tau, o, \mathcal{A}))$, such that $a \notin \bigcirc^{pre}_{Ag}$. However, by definition, $\bigcirc^{pre}_{Ag}(\tau, o) = \mathrm{Supp}\left(Ag_{reg}(\tau, o)\right)$, and $Ag_{reg}(\tau, o) = Ag(\tau, o, \mathcal{A})$. Therefore, such a cannot exist, contradicting $(\tau, o) \in \mathbb{J}_{\bigcirc^{pre}_{Ag}}(Ag)$.

4. Suppose $(\tau, o) \in \mathbb{J}_{\bigcirc^{pos}_{Ag, Ag_{det}}}(Ag)$. Then there would exist $a \in \mathrm{Supp}\,(Ag(\tau, o))$, such that $a \neq \bigcirc^{pos}_{Ag}(\tau, o, a)$. However, by definition, $\bigcirc^{pos}_{Ag, Ag_{det}}(\tau, o, a) = a$ for all $a \in \mathrm{Supp}\,(Ag(\tau, o))$. Therefore, such a cannot exist, finishing the proof.

□

3.3.5 Minimal Correctness

The goals of generating correct traces and minimizing interference often conflict. As we have seen, one can construct a correct shield from a correct policy by simply overwriting any action proposed by the agent with the corresponding action from the correct policy. However, this approach results in significant interference for many agents. At the other extreme, a fully transparent shield imposes no interference at all, leaving interference sets empty. An ideal shield balances these objectives, ensuring correctness while keeping interference sets as small as possible. Formally, we define this as follows.

Definition 3.12 (Minimal correctness)**.** Let $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$ be an environment and Π be a set of agents. A shield $\shield \in \Sigma_\Pi$ is *minimally correct* restricted to Π if it is correct and for all agent $Ag \in \Pi$ (pre-shield-ready in case of pre-shield, regular in case of post-shield), and for all correct shield $\shield' \in \Sigma_\Pi$, we have $\mathbb{J}_{\shield}(Ag) \subseteq \mathbb{J}_{\shield'}(Ag)$.

Theorem 3.1. *Let $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$ be an environment, $\mathcal{L} \subseteq (\mathcal{O} \times \mathcal{A})^\infty$ be a specification, Π be a set of agents, and $\shield$ be a minimally correct shield. Then:*

1. *If $\shield$ is a pre-shield, $\shield$ is unique.*
2. *If $\shield$ is a post-shield, $\shield$ is unique modulo interferences.*
3. *For any correct agent $Ag \in \Pi$ (pre-shield-ready in case of pre-shield, regular in case of post-shield), we have $\mathbb{J}_{\shield}(Ag) = \emptyset$.*

Proof. Uniqueness is a consequence of Lemma 3.2. Suppose $\shield$ and $\shield'$ are minimally correct. Since $\shield$ is minimally correct and $\shield'$ is correct, we have for any agent Ag that $\mathbb{J}_{\shield}(Ag) \subseteq \mathbb{J}_{\shield'}(Ag)$. And using that $\shield'$ is minimally correct and $\shield$ is correct, we have that $\mathbb{J}_{\shield'}(Ag) \subseteq \mathbb{J}_{\shield}(Ag)$, leading to $\mathbb{J}_{\shield}(Ag) = \mathbb{J}_{\shield'}(Ag)$. By Lemma 3.2, this implies $\shield = \shield'$ for pre-shields and $\shield \equiv_i \shield'$ for post-shields.

Since $\shield$ is minimally interfering, for any correct shield $\shield'$, we have that $\mathbb{J}_{\shield}(Ag) \subseteq \mathbb{J}_{\shield'}(Ag)$. In particular, since Ag is correct, we can choose $\shield'$ to be the shield induced by Ag. By Lemma 3.2, $\mathbb{J}_{\shield'}(Ag) = \emptyset$. Therefore, $\mathbb{J}_{\shield'}(Ag)$ is a subset of the empty set, so it can only be that $\mathbb{J}_{\shield'}(Ag) = \emptyset$. □

While this will be our operational definition of a useful shield for most problems, we want to note that it is not without drawbacks. We explore now its two main drawbacks: non-existence and cost-independence.

Existence of minimally correct shields. The first drawback of this definition is that there are environments and specifications for which a minimally correct shield does not exist, even if correct agents do exist. Consider an environment with a single possible observation $\mathcal{O} = \{o\}$, two actions $\mathcal{A} = \{a, b\}$, and a specification $\mathcal{L} = \{w \ : \ w \text{ contains at least one } a\}$. Given this specification, the transparent shield cannot be correct, since there exists the agent that only outputs b, which is not correct. Suppose $\shield$ is a correct post-shield. Then there exists $(\tau, o) \in (\mathcal{O} \times \mathcal{A})^* \times \mathcal{O}$ such that $\shield(\tau, o, b) = \{a\}$. Let $k = |\tau|$. Consider the agent Ag such that $Ag(\tau', o) = b$ if $|\tau'| \leq k$ and $Ag(\tau', o) = a$ if $|\tau'| > k$.

Clearly Ag is a correct agent, however, $\mathbb{J}_{\bigcirc}(Ag) \neq \emptyset$. By Theorem 3.1, $\bigcirc$ cannot be minimally correct. An analogous construction can be made for pre-shields. In Section 3.4, we explore some concrete cases where the existence of minimally correct shields is guaranteed.

Uniform cost. The second drawback is that all interferences are given the same importance. Depending on the problem, it may be useful to assign a numerical cost to each intervention and ask which is the shield that guarantees certain correctness properties while having minimal cost. We explore this in the context of fairness shields in Chapter 6.

3.4 Classical Shielding through the Lens of the Reactive Decision-Making Framework

Shielding has been successfully used for specifications of the safety type in deterministic two-player games and in MDPs. In this section, we explain how these methods can be seen in our framework.

3.4.1 Shielding in Safety Games with Perfect Information

Shielding was first introduced in safety games [Blo+15]. Recall (Section 3.2.1) how safety games can be regarded in the reactive decision-making framework as memoryless environments with a uniform transition function. Also, when considering the perfect information regime, i.e., no delayed observations, memoryless strategies are enough to define any winning strategy.

Following the general definition (Definition 3.3), and adapting to the fact that transitions in a safety game are memoryless, a pre-shield is a function $\bigcirc\colon S_{ag} \to 2^{\mathcal{A}}$. Given a strategy of the safety game $\xi\colon S_{ag} \to 2^{\mathcal{A}}$, we call $\bigcirc^{pre}_{\xi} = \xi$ the pre-shield induced by the strategy ξ.

Given a strategy $\xi\colon S_{ag} \to 2^{\mathcal{A}}$ and a deterministic strategy $\chi\colon S_{ag} \to \mathcal{A}$, we call $\bigcirc^{pos}_{\xi,\chi}$ the post-shield induced by the pair of strategies (ξ, χ), defined as:

$$\bigcirc^{pos}_{\xi,\chi}(s,a) = \begin{cases} a & \text{if } a \in \xi(s) \\ \chi(s) & \text{otherwise.} \end{cases}$$

A pre-shield can be added before an output-restricted strategy $\xi\colon S_{ag} \times 2^{\mathcal{A}} \to 2^{\mathcal{A}}$ to generate a regular strategy $\xi_{\bigcirc}\colon S_{ag} \to 2^{\mathcal{A}}$. Similarly, a post-shield is a function $\bigcirc\colon S_{ag} \times \mathcal{A} \to \mathcal{A}$. A post-shield can be added after a regular strategy $\xi\colon S_{ag} \to 2^{\mathcal{A}}$ to generate a new regular strategy $\xi_{\bigcirc}\colon S_{ag} \to 2^{\mathcal{A}}$.

Note that, by definition, if ξ is a winning strategy of ξ, then $\bigcirc^{pre}_{\xi}$ is a correct pre-shield. Similarly, if ξ is a winning strategy and χ is a determinization of ξ, then $\bigcirc^{pos}_{\xi,\chi}$ is a correct post-shield.

Theorem 3.2. *Let $\mathcal{G} = (S, s_0, S_{env}, S_{ag}, \mathcal{A}, \mathcal{T}, \mathcal{F})$ be a safety game with a winning strategy. Let ξ be the maximally permissive winning strategy of $\mathcal{G}$. Then:*

1. *The minimally correct pre-shield exists and is $\bigcirc^{pre}_{\xi}$.*

2. *For any deterministic winning strategy χ, the shield $\shield^{pos}_{\xi,\chi}$ is minimally correct.*

Proof. **(1.)** We argue the first point by contradiction. Since ξ is a winning strategy, $\shield^{pre}_{\xi}$ is correct by construction, so we only have to argue the minimality property. Suppose that $\shield^{pre}_{\xi}$ is not minimally correct. Then there exists a pre-shield-ready agent $Ag\colon S_{ag} \times 2^{\mathcal{A}} \to 2^{\mathcal{A}}$, a correct pre-shield $\shield\colon S_{ag} \to 2^{\mathcal{A}}$, and $s \in S_{ag}$ such that $s \in \mathbb{J}_{\shield^{pre}_{\xi}}(Ag)$, but $s \notin \mathbb{J}_{\shield}(Ag)$. Since $s \notin \mathbb{J}_{\shield}(Ag)$, then $Ag(s, \mathcal{A}) \subseteq \shield(s)$. Since $s \in \mathbb{J}_{\shield^{pre}_{\xi}}(Ag)$, then there exists $a \in \mathcal{A}$ such that $a \in Ag(s, \mathcal{A}) \subseteq \shield(s)$, but $a \notin \shield^{pre}_{\xi}(s)$. Since $\shield^{pre}_{\xi}$ is implemented with the maximally permissive winning strategy, by definition (recall Equation (2.4)) we have $s \xrightarrow{a} s'$, with $s' \notin W$, where W is the winning region of $\mathcal{G}$.

On the other hand, consider an pre-shield-ready agent Ag' such that $Ag'(s, \shield(s)) = \{a\}$, which exists since $a \in \shield(s)$. Let $Ag'_{\shield}$ be the agent resulting from applying $\shield$ on Ag'. This agent is correct because $\shield$ is correct, and by construction, $Ag'_{\shield}(s) = a$. But then s' would be part of a valid trace under a winning strategy, so $s' \in W$. This is a contradiction, as we had previously established that $s' \notin W$.

(2.) As for the second point, the shield is correct since ξ is winning and χ is a determinization of ξ, and we will use a similar argument to prove minimality by contradiction. Suppose $\shield^{pos}_{\xi,\chi}$ is not minimally correct. Then there exists an agent $Ag\colon S_{ag} \to 2^{\mathcal{A}}$, a correct post-shield $\shield\colon S_{ag} \times \mathcal{A} \to \mathcal{A}$ and $s \in S_{ag}$ such that $s \in \mathbb{J}_{\shield^{pos}_{\xi,\chi}}(Ag)$, but $s \notin \mathbb{J}_{\shield}(Ag)$.

Since $s \notin \mathbb{J}_{\shield}(Ag)$, then for all $a \in Ag(s)$, we have $\shield(s, a) = a$. Since $s \in \mathbb{J}_{\shield^{pre}_{\xi}}(Ag)$, then there exists $a \in Ag(s)$ such that $\shield^{pos}_{\xi,\chi}(s, a) \neq a$. Since $\shield^{pos}_{\xi,\chi}$ is implemented with the maximally permissive winning strategy ξ, and a determinization of it χ, it means that $a \notin \xi(s)$, and thus s' defined by $s \xrightarrow{a} s'$ is not in the winning region W.

On the other hand, consider an agent $Ag'\colon S_{ag} \to 2^{\mathcal{A}}$ satisfying $Ag(s) = a$, and consider $Ag'_{\shield}$, that same agent shielded with $\shield$. We know that $Ag'_{\shield}$ is correct, because $\shield$ is correct. Since $\shield(s, a) = a$, we have that $Ag'_{\shield}(s) = a$. The proof finishes by noting that in such case $s' \in W$, contradicting $s' \notin W$. □

While the definition of minimally correct shields with interference sets is original to this work, the construction of shields using the maximally permissive strategy of the underlying safety game is the same as has been described in the original shielding literature [Blo+15; Kön+17; Als+18; Kön19].

The definition presented in this work captures corner cases that previous definitions did not, as we show in the following example.

Example 3.1. *Consider the safety game $\mathcal{G} = (S, s_0, S_{env}, S_{ag}, \mathcal{A}, \mathcal{T}, \mathcal{F})$ illustrated in Figure 3.4, where the $S_{env} = \{s_0, s_2, s_3, s_7\}$, $S_{ag} = \{s_1, s_4, s_5, s_6\}$, $\mathcal{A} = \{a, b\}$, $\mathcal{F} = S \setminus \{s_6, s_7\}$, and $\mathcal{T}$ is as described in the figure. The winning region of this safety game is $W = \{s_0, s_1, s_3, s_4, s_5\}$, and therefore no trace*

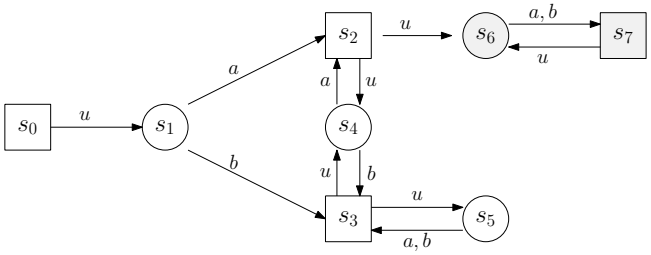


Figure 3.4: Safety game illustrating Example 3.1.

containing s_2 would be allowed by a shielded agent. However, according to the definition of shield as minimally interfering in previous work [Kön+17, Def. 1], a trace $\tau = s_0 s_1 s_2 s_4 (s_3 s_5)^\omega$ should be allowed by a shielded agent since it does leave $\mathcal{F}$ at any time.

3.4.2 Shielding in Safety Games with Delayed Observations

When considering games under delay, we need to be aware that memoryless strategies are not enough, as discussed in Section 2.3.2.

From a reactive decision-making framework point of view, the correspondence between a game $\mathcal{G}_{\delta,\mu} = \langle S, s_0, S_{env}, S_{ag}, \mathcal{A}, \mathcal{T}, \mathrm{Acc}, \delta, \mu \rangle$, and an environment $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$ is the same as the correspondence described in the previous section for games with $\delta = \mu = 0$.

The only relevant difference is that we consider only agents restricted to the set of agents agnostic to the last δ observations, and with a restricted memory μ. Therefore, the shields we consider are going to have the same restrictions.

In Section 3.2.4 we have defined, for a given environment $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$, the set of agents $\Pi_\delta^{\circledcirc}$ as those agnostic to the last δ observations, and we have explained how restricting to this set of agents is equivalent to considering agents that work with delayed observations.

A characteristic of safety games is that the transition relation $\mathcal{T}$ depends only on the state – and not the trace leading to that state. Therefore, the agents relevant for solving safety games under delay δ are only not agnostic to the $\delta{+}1$-th observation, counting from the tail. Formally, mirroring Definition 3.1, an agent $Ag = (\mathcal{O}, \mathcal{A}, \pi)$ works only with the last $\delta{+}1$-th observation if for all $\tau, \tau' \in (\mathcal{O} \times \mathcal{A})^*$, all $o \in \mathcal{O}$, all $(a_0, a_1, \ldots, a_\delta) \in \mathcal{A}^\delta$ and all $(o_1, \ldots, o_\delta), (o'_1, \ldots, o'_\delta) \in \mathcal{O}^\delta$, we have

$$\pi\Big(\big(\tau{\cdot}(o, a_0), (o_1, a_1), \ldots, (o_{\delta-1}, a_{\delta-1})\big), o_\delta\Big) = \pi\Big(\big(\tau'{\cdot}(o, a_0), (o'_1, a_1), \ldots, (o'_{\delta-1}, a_{\delta-1})\big), o'_\delta\Big). \tag{3.6}$$

Following the same equivalence as in Section 3.2.4.3, the agents satisfying Equation (3.6) are equivalent to agents working on the domain $\mathcal{A}^{\leq\delta} \cup (\mathcal{O} \times \mathcal{A}^\delta)$. Introducing a restriction on memory, as described for safety games in Section 2.3.2 is also modelled as a restriction to agents agnostic to certain parts of the input.

Concretely, the part of the input that corresponds to actions that overflow the memory.

Putting these two concepts together, we have the following definition.

Definition 3.13. Let $\mathcal{G}_{\delta,\mu} = \langle S, s_0, S_{env}, S_{ag}, \mathcal{A}, \mathcal{T}, \mathrm{Acc}\rangle$ be a two player game and let $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$ be its corresponding environment. Let $\delta \in \mathbb{N}$ and $\mu \leq \delta$ be two integers representing delay and memory. An agent $Ag = (\mathcal{O}, \mathcal{A}, \pi)$ *works in the safety game with memory μ and observations delayed by δ* if for all $\tau, \tau' \in (\mathcal{O} \times \mathcal{A})^*$, all $(a_{\delta-\mu}, \ldots, a_{\delta-1}) \in \mathcal{A}^\mu$, all $(a_0, \ldots, a_{\delta-\mu-1}), (a'_0, \ldots, a'_{\delta-\mu-1}) \in \mathcal{A}^{\delta-\mu}$, all $o \in \mathcal{O}$, and all $(o_1, \ldots, o_\delta), (o'_1, \ldots, o'_\delta) \in \mathcal{O}^\delta$, we have:

$$\pi\Big(\big(\tau \cdot (o, a_0), (o_1, a_1), \ldots, (o_{\delta-\mu-1}, a_{\delta-\mu-1}), (o_{\delta-\mu}, a_{\delta-\mu}), \ldots, (o_{\delta-1}, a_{\delta-1})\big), o_\delta\Big) = \pi\Big(\big(\tau' \cdot (o, a'_0), (o'_1, a'_1), \ldots, (o'_{\delta-\mu-1}, a'_{\delta-\mu-1}), (o'_{\delta-\mu}, a_{\delta-\mu}), \ldots, (o'_{\delta-1}, a_{\delta-1})\big), o'_\delta\Big) \tag{3.7}$$

The set of agents in this regime is denoted as $\Pi^{\circledcirc}_{\delta,\mu}$.

Again, following the same equivalence as in Section 3.2.4.3, the agents in $\Pi^{\circledcirc}_{\delta,\mu}$ can be characterized as agents of the form $\pi\colon \mathcal{A}^\mu \cup (\mathcal{O} \times \mathcal{A}^\mu) \to 2^{\mathcal{A}}$, which is the same form as that of strategies in games with delay δ and memory μ (Section 2.3.2).

This serves as the basis for the result analogous to Theorem 3.2 for games with delayed observations.

Theorem 3.3. *Let $\mathcal{G} = (S, s_0, S_{env}, S_{ag}, \mathcal{A}, \mathcal{T}, \mathcal{F}, \delta, \mu)$ be a safety game under delayed observation δ with a winning strategy with memory μ. Let $\xi\colon S_{ag*} \times \mathcal{A}^{\leq\mu} \to 2^{\mathcal{A}}$ be the maximally permissive winning strategy of $\mathcal{G}$. Then, restricting to the set of agents $\Pi^{\circledcirc}_{\delta,\mu}$, we have that:*

1. *The minimally correct pre-shield exists and is $\bigcirc^{pre}_{\xi} \in \Sigma_{\Pi^{\circledcirc}_{\delta,\mu}}$.*
2. *For any deterministic winning strategy χ, the shield $\bigcirc^{pos}_{\xi,\chi} \in \Sigma_{\Pi^{\circledcirc}_{\delta,\mu}}$ is minimally correct.*

The proof of Theorem 3.3 follows the same argument as the proof of the analogous theorem for regular safety games (Theorem 3.2), since games under delay are equivalent to regular games with exponentially many states [Che+21, Lemma 2]. In any case, we include it here for the sake of completeness.

Proof. **(1.)** We argue the first point by contradiction. Since ξ is a winning strategy, $\bigcirc^{pre}_{\xi}$ is correct by construction, so we only have to argue the minimality property. Suppose that $\bigcirc^{pre}_{\xi}$ is not minimally correct. Then there exists a pre-shield-ready agent $Ag\colon S_{ag*} \times \mathcal{A}^{\leq\mu} \times 2^{\mathcal{A}} \to 2^{\mathcal{A}}$, a correct pre-shield $\bigcirc\colon S_{ag*} \times \mathcal{A}^{\leq\mu} \to 2^{\mathcal{A}}$, and $(s, \overline{\sigma}) \in S_{ag*} \times \mathcal{A}^{\leq\mu}$ such that $(s, \overline{\sigma}) \in \mathbb{J}_{\bigcirc^{pre}_{\xi}}(Ag)$, but $(s, \overline{\sigma}) \notin \mathbb{J}_{\bigcirc}(Ag)$. Assuming [1] $s \neq \varepsilon$, we have $\overline{\sigma} \in \mathcal{A}^\mu$, thus it is of the form $\overline{\sigma} = (\sigma_{\delta-\mu+1}, \ldots, \sigma_\delta)$. Since $(s, \overline{\sigma}) \notin \mathbb{J}_{\bigcirc}(Ag)$, then $Ag(s, \overline{\sigma}, \mathcal{A}) \subseteq \bigcirc(s)$. Since $(s, \overline{\sigma}) \in$

[1] We show at the end of the proof how to treat the case $s = \varepsilon$.

$\mathbb{J}_{\bigcirc^{pre}_{\xi}}(Ag)$, then there exists $a \in \mathcal{A}$ such that $a \in Ag(s, \overline{\sigma}, \mathcal{A}) \subseteq \bigcirc(s)$, but $a \notin \bigcirc^{pre}_{\xi}(s, \overline{\sigma})$. Since $\bigcirc^{pre}_{\xi}$ is implemented with the maximally permissive winning strategy, this means that there exists $s'_1, \ldots, s'_{2\delta+1} \in S$, and $(\sigma_1, \ldots, \sigma_{\delta-\mu}) \in \mathcal{A}^{\delta-\mu}$, such that

$$s \xrightarrow{\sigma_1} s'_1 \xrightarrow{u} s'_2 \xrightarrow{\sigma_2} \ldots \xrightarrow{\sigma_\delta} s'_{2\delta-1} \xrightarrow{u} s'_{2\delta} \xrightarrow{a} s'_{2\delta+1}, \tag{3.8}$$

and such that $(s'_2, \overline{\sigma}')$ is not part of any winning strategy, where $\overline{\sigma}' = (\sigma_{\delta-\mu+2}, \ldots, \sigma_\delta, a) \in \mathcal{A}^\mu$.

On the other hand, consider a pre-shield-ready agent Ag' such that $Ag'(s, \overline{\sigma}, \bigcirc(s, \overline{\sigma})) = \{a\}$, which exists since $a \in \bigcirc(s, \overline{\sigma})$. Let $Ag'_{\bigcirc}$ be the agent resulting from applying $\bigcirc$ on Ag'. This agent is correct because $\bigcirc$ is correct, and by construction, $Ag'_{\bigcirc}(s, \overline{\sigma}) = a$. But then $s'_2, \overline{\sigma}'$ would be part of a valid trace under a winning strategy. This is a contradiction, as we had previously established that $(s'_2, \overline{\sigma}')$ cannot be part of any winning strategy.

(2.) The shield is correct since ξ is winning and χ is a determinization of ξ, and we use a similar argument to prove minimality by contradiction. Suppose $\bigcirc^{pos}_{\xi,\chi}$ is not minimally correct. Then there exists an agent $Ag\colon S_{ag*} \times \mathcal{A}^{\leq\mu} \to 2^{\mathcal{A}}$, a correct post-shield $\bigcirc$: $S_{ag*} \times \mathcal{A}^{\leq\mu} \times \mathcal{A} \to \mathcal{A}$ and $(s, \overline{\sigma}) \in S_{ag*} \times \mathcal{A}^{\leq\mu}$ such that $(s, \overline{\sigma}) \in \mathbb{J}_{\bigcirc^{pos}_{\xi,\chi}}(Ag)$, but $(s, \overline{\sigma}) \notin \mathbb{J}_{\bigcirc}(Ag)$.

Since $(s, \overline{\sigma}) \notin \mathbb{J}_{\bigcirc}(Ag)$, then for all $a \in Ag(s, \overline{\sigma})$, we have $\bigcirc(s, \overline{\sigma}, a) = a$. Since $(s, \overline{\sigma}) \in \mathbb{J}_{\bigcirc^{pre}_{\xi}}(Ag)$, then there exists $a \in Ag(s, \overline{\sigma})$ such that $\bigcirc^{pos}_{\xi,\chi}(s, \overline{\sigma}, a) \neq a$. Since $\bigcirc^{pos}_{\xi,\chi}$ is implemented with the maximally permissive winning strategy ξ, and a determinization of it χ, it means that $a \notin \xi(s, \overline{\sigma})$, and thus $(s'_2, \overline{\sigma}')$ defined by the same procedure as Equation (3.8) cannot be part of any winning strategy. To build $(s'_2, \overline{\sigma})'$ we need to assume again that $s \neq \varepsilon$.

On the other hand, consider an agent $Ag'\colon S_{ag*} \times \mathcal{A}^{\leq\mu} \to 2^{\mathcal{A}}$ satisfying $Ag(s, \overline{\sigma}) = a$, and consider $Ag'_{\bigcirc}$, that same agent shielded with $\bigcirc$. We know that $Ag'_{\bigcirc}$ is correct, because $\bigcirc$ is correct. Since $\bigcirc(s, \overline{\sigma}, a) = a$, we have that $Ag'_{\bigcirc}(s, \overline{\sigma}) = a$. The proof finishes by noting that in such case $(s'_2, \overline{\sigma}')$ as obtained in Equation (3.8) is part of a valid trace under a winning strategy, contradicting the previously established point.

Initial phase. In both proofs we have assumed that $s \neq \varepsilon$, and therefore $\sigma \in \mathcal{A}^\mu$ in order to obtain the construction in Equation (3.8). The case for $s = \varepsilon$ follows the same argument, only considering that $\sigma = (\sigma_1, \ldots, \sigma_\nu)$ for some $\nu \leq \mu$ and thus fewer transitions in Equation (3.8). □

3.4.3 Probabilistic Shielding in Markov Decision Processes

Probabilistic safety shielding in MDPs was introduced in [Jan+20]. We present probabilistic safety shielding adapting the definitions in [Jan+20] to our framework. The main difference of shielding in MDPs is that the safety specification has a probabilistic nature. A probabilistic shield blocks an action when the probability of the action causing harm is larger than some threshold λ.

Property specification.

Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$ be an MDP, $T \subseteq \mathcal{S}$ be a set of "unsafe" states to avoid, $k \in \mathbb{N} \cup \{\infty\}$ a step horizon, and $\lambda \in [0, 1]$. The language specifying correct traces will be denoted $\mathcal{L}_{T,\lambda,k}$. A trace $\tau = (s_0, a_0, s_1, a_1, \dots)$ is in $\mathcal{L}_{T,\lambda,k}$ if for every $i \geq 0$, we have

$$\mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_k(s_i, a_i, T)\right) \geq \lambda \cdot \mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_k(s_i, T)\right). \tag{3.9}$$

This means that for an action a_i to be safe at state s_i, the probability of not reaching a bad state if the agent behaves "optimally" has to be at least λ times the probability of an agent that is optimal in avoiding bad states.

For example, if $\lambda = 1/2$, and the policy that best avoids T reaches it with a 10% probability, any action from which T can be avoided with a 20% probability is considered to be "safe enough". In general, the larger the value of λ, the more restrictive or cautious the shield is. In the extremes, when $\lambda = 0$, any action is allowed, and when $\lambda = 1$, only the safest actions are allowed, i.e., the actions for which $\mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, a_i, T)\right) = \mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, T)\right)$.

An agent $\pi \colon \mathcal{S} \to \mathcal{D}(\mathcal{A})$ is correct with respect to a specificaiton $\mathcal{L}_{T,\lambda,k}$ if any valid trace of π is in $\mathcal{L}_{T,\lambda,k}$. Following the same notation convention as in Section 2.4.2, when we are considering unbounded properties, i.e., when $k = \infty$, we may drop the k from our notation, writing the specification as $\mathcal{L}_{T,\lambda}$ instead of $\mathcal{L}_{T,\lambda,\infty}$.

Shield synthesis.

Since the environment and the safety specification are defined in a memoryless manner, we can also consider shields as memoryless. We define the shields induced by a specification $\mathcal{L}_{T,\lambda,k}$ as one would expect.

A pre-shield in an MDP is a function $\bigcup \colon \mathcal{S} \to 2^{\mathcal{A}}$ A pre-shield-ready agent in an MDP is a function $\pi \colon \mathcal{S} \times 2^{\mathcal{A}} \to \mathcal{D}(\mathcal{A})$. A post-shield in an MDP is a function $\bigcup \colon \mathcal{S} \times \mathcal{A} \to \mathcal{A}$.

Let $T \subseteq \mathcal{S}$, $k \in \mathbb{N} \cup \{\infty\}$, and $\lambda \in [0, 1]$ defining a probabilistic safety specification $\mathcal{L}_{T,\lambda,k}$. The pre-shield induced by T, k, and λ is $\bigcup^{pre}_{T,\lambda}$ defined as:

$$\bigcup^{pre}_{T,\lambda,k}(s) = \left\{a \in \mathcal{A} \,:\, \mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, a_i, T)\right) \geq \lambda \cdot \mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, T)\right)\right\}.$$

Similarly, let $\pi \colon \mathcal{S} \to \mathcal{D}(\mathcal{A})$ be a correct agent. The post-shield induced by T, λ, and π is $\bigcup^{pos}_{T,\lambda,\pi}$ defined as

$$\bigcup^{pos}_{T,\lambda,k,\pi}(s, a) = \begin{cases} a & \text{if} \quad \mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, a_i, T)\right) \geq \lambda \cdot \mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, T)\right) \\ \pi(s) & \text{otherwise.} \end{cases}$$

Theorem 3.4. *Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$ be an MDP. Let $T \subseteq \mathcal{S}$, $k \in \mathbb{N} \cup \{\infty\}$, and $\lambda \in [0, 1]$ forming a probabilistic safety specification $\mathcal{L}_{T,\lambda,k}$. Then:*

1. *The minimally correct pre-shield exists and is $\bigcup^{pre}_{T,\lambda,k}$.*

2. For any correct agent π, the shield $\shield^{pos}_{T,\lambda,k,\pi}$ is minimally correct.

Proof. Correctness should be clear by construction in both cases, so we only need to argue for minimality. The arguments to prove minimality are similar to the arguments used to prove Theorem 3.2.

(1.) We argue the first point by contradiction. Assume that $\shield^{pre}_{T,\lambda,k}$ is not minimally correct. Then there exists a pre-shield-ready agent $Ag\colon \mathcal{S} \times 2^{\mathcal{A}} \to \mathcal{D}(\mathcal{A})$ and a correct pre-shield $\shield$ such that $\mathbb{J}_{\shield^{pre}_{T,\lambda,k}}(Ag) \not\subseteq \mathbb{J}_{\shield}(Ag)$. Therefore, there exists $s \in \mathcal{S}$ such that $s \in \mathbb{J}_{\shield^{pre}_{T,\lambda,k}}(Ag)$ but $s \notin \mathbb{J}_{\shield}(Ag)$. Since $s \in \mathbb{J}_{\shield^{pre}_{T,\lambda}}$, there exists $a \in \mathrm{Supp}(Ag(s,\mathcal{A}))$ with $a \notin \shield^{pre}_{T,\lambda,k}(s)$. By the definition of $\shield^{pre}_{T,\lambda,k}$, the action a is not in the allowed actions of the shield only if

$$\mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, a_i, T)\right) < \lambda \cdot \mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, T)\right). \tag{3.10}$$

On the other hand, since $s \notin \mathbb{J}_{\shield}(Ag)$, it means that $\mathrm{Supp}(Ag(s,\mathcal{A})) \subseteq \shield(s)$. In particular, $a \in \shield(s)$. Consider a pre-shield-ready agent Ag' such that $Ag'(s,\shield(s)) = \{a\}$. Let $Ag'_{\shield}$ be the shielded agent resulting from applying $\shield$ to Ag'. This is a correct agent such that $Ag'(s) = a$. But then the fragment (s,a) would be part of a correct trace, meaning that

$$\mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, a_i, T)\right) \geq \lambda \cdot \mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, T)\right),$$

which contradicts Equation (3.10).

(2.) We also argue the second point by contradiction. Suppose there exists $\pi\colon \mathcal{S} \to \mathcal{D}(\mathcal{A})$ correct such that $\shield^{pos}_{T,\lambda,k,\pi}$ is not minimally correct. Then there exists an agent $Ag\colon \mathcal{S} \to \mathcal{D}(\mathcal{A})$ and a correct post-shield $\shield$ such that $\mathbb{J}_{\shield^{pos}_{T,\lambda,k,\pi}}(Ag) \not\subseteq \mathbb{J}_{\shield}(Ag)$. Therefore, there exists $s \in \mathcal{S}$ such that $s \in \mathbb{J}_{\shield^{pos}_{T,\lambda,k,\pi}}(Ag)$ but $s \notin \mathbb{J}_{\shield}(Ag)$.

Since $s \notin \mathbb{J}_{\shield}(Ag)$, it means that for all $a \in \mathrm{Supp}(Ag(s))$, we have $\shield(s,a) = a$. Since $s \in \mathbb{J}_{\shield^{pos}_{T,\lambda,k,\pi}}$, there exists $a \in \mathrm{Supp}(Ag(s))$ with $a \neq \shield^{pos}_{T,\lambda,k,\pi}(s,a)$. In particular, since $a \in \mathrm{Supp}(Ag(s))$, we have $\shield(s,a) = a$. This means that

$$\mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, a_i, T)\right) < \lambda \cdot \mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, T)\right). \tag{3.11}$$

On the other hand, consider an agent $Ag'\colon \mathcal{S} \to \mathcal{D}(\mathcal{A})$ such that $Ag'(s)(a) > \varepsilon$ for some $\varepsilon > 0$, i.e., an agent that from s outputs a with a positive probability. The shielded version of Ag', denoted by $Ag'_{\shield}$ is a correct agent, and $Ag'_{\shield}(s) = a$ with probability ε. Therefore, (s,a) is a fragment contained in valid traces of a correct agent, and thus satisfies

$$\mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, a_i, T)\right) \geq \lambda \cdot \mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, T)\right),$$

which contradicts Equation (3.11). □

While this is our operational definition of a probabilistic shield and the one we will be using in Chapter 5, there are certain variations that have been proposed in the literature

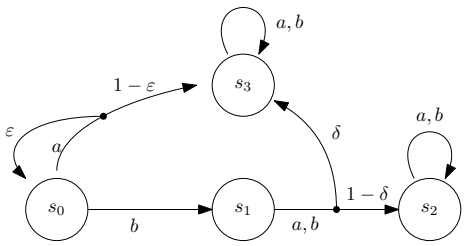


Figure 3.5: MDP described in Example 3.2.

Absolute threshold

The value of λ in Equation (3.9) is considered a *relative threshold*, as it states what the minimum probability of reaching an unsafe state can be relative to the best choice of action. An alternative that has been studied in the literature [Pra+21a] is to use λ as an absolute minimum threshold on the probability of not reaching a bad state. In such shields, the condition to let an action pass is

$$\mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s_i, a_i, T)\right) \geq \lambda. \tag{3.12}$$

Note that if an action satisfies Equation (3.12), then it also satisfies Equation 3.9, making shields with absolute threshold more restrictive than shields with relative threshold.

This choice has the advantage of being properly restrictive in the more "critical" states. For example, consider a threshold $\lambda = 0.6$. In a state $s \in \mathcal{S}$ with $\mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s, a, T)\right) = 0.75$, the shield with an absolute threshold would only let actions pass with an optimal probability of avoiding an unsafe state between 0.6 and 0.75. On the other hand, the shield with λ as a relative threshold would allow actions with the optimal probability of avoiding an unsafe state as low as 0.45.

The main downside of the "absolute" approach is that in some states there may not be any "allowed" action. In such states, one should just resort to the optimal actions (that would always be allowed with a relative threshold approach).

Global guarantees

The property defined in Equation (3.9) is local for every decision. A desirable property would be that a shield $\bigcup$ synthesized for a specification $\mathcal{L}_{T,\lambda,k}$ would satisfy for every agent Ag and any initial state of the MDP s, that we have $\mathbb{P}^{\mathcal{M}}_{Ag_{\bigcup}}(\texttt{Avoid}_{\leq k}(s, T)) \geq \lambda$.

This is not the case. In fact, the following example shows that we can make $\mathbb{P}^{\mathcal{M}}_{Ag_{\bigcup}}(\texttt{Avoid}_{\leq k}(s, T))$ arbitrarily small while keeping λ arbitrarily large.

Example 3.2. *Let $\varepsilon, \delta \in (0, 1)$. Consider the MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$ illustrated in Figure 3.5, with $\mathcal{S} = \{s_0, s_1, s_2, s_3\}$, $\mathcal{A} = \{a, b\}$ and $\mathcal{P}$ as described in the figure, where any transition that is drawn and has no number on it has probability 1. Consider $T = \{s_3\}$ and $k = \infty$. The only state where the agent's decision*

matters is s_0, so an agent can be described as P_a, the probability of taking action a in s_0. If an agent chooses a, it reaches s_3 with probability $1-\varepsilon$ and goes back to s_0 with probability ε. If an agent chooses b, it reaches s_3 with probability δ and avoids s_3 altogether with probability $1-\delta$. The optimal strategy to avoid s_3 is clearly $p_a = 0$, and in such case $\mathbb{P}_{\min}(\texttt{Reach}(s_0, \{s_3\})) = \delta$. For a general agent p_a, we have

$$\begin{aligned}
&\mathbb{P}_{p_a}\left(\texttt{Reach}(s_0, a, \{s_3\})\right) = 1 - \varepsilon + \varepsilon \cdot \mathbb{P}_{p_a}\left(\texttt{Reach}(s_0, \{s_3\})\right), \\
&\mathbb{P}_{p_a}\left(\texttt{Reach}(s_0, b, \{s_3\})\right) = \delta, \\
&\quad \mathbb{P}_{p_a}\left(\texttt{Reach}(s_0, \{s_3\})\right) = p_a \cdot [1 - \varepsilon + \varepsilon \cdot \mathbb{P}_{p_a}\left(\texttt{Reach}(s_0, \{s_3\})\right)] + (1 - p_a)\delta.
\end{aligned} \tag{3.13}$$

Therefore, with a threshold λ, action a is allowed by the shield if

$$1 - (1 - \varepsilon + \varepsilon\delta) \geq \lambda(1 - \delta),$$

which is equivalent to $\varepsilon \geq \lambda$. On the other hand, isolating from Equation 3.13 we get:

$$\mathbb{P}_{p_a}\left(\texttt{Reach}(s_0, \{s_3\})\right) = \frac{p_a(1-\varepsilon) + (1-p_a)\delta}{1 - p_a\varepsilon} = 1 - \frac{(1-p_a)(1-\delta)}{1 - p_a\varepsilon}. \tag{3.14}$$

Once ε is fixed, we can make the value in Equation 3.14 arbitrarily close to 1 by modifying δ and p_a.

A potential solution for such cases would be to define the set of correct agents as those agents that satisfy

$$\mathbb{P}^{\mathcal{M}}_{Ag}(\texttt{Avoid}(s, T)) \geq \lambda. \tag{3.15}$$

This would require us to change the concepts used to define shields, as the property described in Equation 3.15 cannot be described in terms of traces.

Chapter 4

Safety Shielding Resilient to Delayed Observation

No oblideu mai que si ens llevem ben d'hora, però ben d'hora ben d'hora, i no hi ha retrets ni hi ha excuses, i ens posem a pencar, som un país imparable. [1] — Josep Guardiola i Sala.

4.1 Motivation and Outline

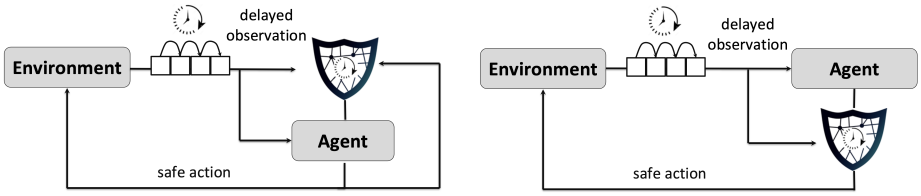
Incorporating delays into safety computations is essential for nearly all real-world control problems. These delays, arising from data collection, processing, or transmission, are ubiquitous in systems operating within complex environments [HL72; Bal92; Nil98; Tri04; Ber+08; Che+16; HFM17]. When not properly addressed, such delays can become the root cause of critical safety issues.

Example 4.1. *Consider a scenario where a car detects a pedestrian at position (x, y) and accounts for a known time delay δ between sensing and acting. The vehicle must plan its actions to ensure safety for any possible position of the pedestrian within the interval $(x \pm \varepsilon, y \pm \varepsilon)$, where ε is determined based on assumptions about the pedestrian's velocity and the delay δ.*

Safety shielding is often used to guarantee safe execution of an agent in an environment that a safety game can model. However, traditional safety shields assume no delay between sensing and acting, which limits their applicability in real-world scenarios.

In this chapter, we introduce synthesis algorithms for *delay-resilient* safety shields, i.e., shields specifically designed to maintain safety even when input delays are present. These algorithms account for the uncertainties introduced by delays, enabling robust performance in dynamic environments. Figure 4.1 illustrates the shielding setup under delayed conditions.

[1] Never forget that if we get up very early, but very early, very early, and there are no reproaches or excuses, and we get down to business, we are unstoppable.



(a) Pre-shield. (b) Post-shield.

Figure 4.1: Delay-resilient shielding scheme.

To synthesise delay-resilient shields, we incorporate a worst-case delay in the safety game, which induces imperfect state information, and use the algorithm proposed in [Che+18; Che+21] to compute the maximally permissive winning strategy. The delay-resilient pre-shields are then computed from the maximally permissive winning strategy in the delayed safety game. For post-shields, in addition to the maximally permissive winning strategy, we need a deterministic winning strategy that will be used to obtain a fixed replacement action for any unsafe action. To do so, we can define a property over the state space and set the action maximising such property as the one fixed by the shield. We study two such properties: controllability and robustness. The *controllability value* assigns to any state s the *maximal delay* on the input under which s stays safe. The *robustness value* of a state s is the length of the minimal path from s to any unsafe state. We discuss how to maximise a state property under the uncertainty introduced by the delayed input.

Finally, we evaluate delay-resilient shields in two case studies. The first one is a gridworld in which we implemented all proposed types of delay-resilient shields and compare computation cost and interference rates. We show that delay-resilient post-shields that choose corrective actions maximising either robustness or controllability tend to stabilise the execution, requiring fewer interferences by the shield. In the second case study, we integrate shielding under delay in the driving simulator CARLA [Dos+17] to enforce collision avoidance for autonomous driving agents at intersections with cars and pedestrians under delayed observations. Our results show the effects of delays on the safety analysis and that our method is scalable enough to be applied in complex application domains. The source code and scripts to reproduce the experiments, along with *videos* from our experiments in CARLA, are available on the accompaning repository[2].

Contribution. The work presented in this chapter can be summarized in the following contributions.

- We formalize the concept of pre-shield and post-shield resilient to delayed observation, showing how to compute them with the maximally permissive strategy of the corresponding safety game.
- We describe in detail the algorithm to compute the maximally permissive strategy of a safety game under delay with restricted memory, extending

[2] `https://github.com/filipcano/safety-shields-delayed`

the algorithm presented in [Che+21].

- We introduce the concepts of *robustness* and *controllability* and how to build post-shields maximising each one.
- We provide theoretical insight about the differences and similarities of the *controllability* and the *robustness* criterion when choosing a corrective action in post-shielding.
- We validate our approach in two use cases: a gridworld and a realistic driving scenario. As far as we know, we present the first integration of shields in a realistic driving simulator.

Outline. We use in this chapter the formalism of two-player safety games with delayed inputs as defined in Section 2.3. In Section 4.2 we present the concept of shields resilient to delays and explain the algorithm required to computed them by finding the maximally permissive winning strategy of the underlying safety game undel delayed information. In Section 4.3 we present the two properties proposed to guide the synthesis of post-shields and how to synthesise shields, maximising them. In Section 4.4, we explore the relation between robustness and controllability, proving that they can be arbitrarily different. Finally, in Section 4.5 we present the results of our experimental evaluation on two use cases and in Section 4.6 we discuss limitations and related work.

Declaration of sources. This chapter is partially based and reuses material from the following source previously published by the author of this thesis:

[CC+23b] Filip Cano Córdoba, Alexander Palmisano, Martin Fränzle, Roderick Bloem, and Bettina Könighofer. "Safety Shielding under Delayed Observation". In: *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)* 33.1 (2023), pp. 80–85.

4.2 Delay Resilient Shields as Strategies in Safety Games

As we have described in Sections 3.2.1 and 3.4.1, a safety game can be seen as a particular case of the reactive decision-making framework, where minimally correct shields (Definition 3.12) correspond to maximally permissive winning strategies of the corresponding safety game.

Following the construction outlined in Section 2.3.2, given a safety game, $\mathcal{G} = \langle S, s_0, S_{env}, S_{ag}, \mathcal{A}, \mathcal{T}, \mathcal{F} \rangle$, we consider the corresponding game played with delay δ and memory μ, for given values of $\delta, \mu \in \mathbb{N}$, $\mu \leq \delta$.

As described in Section 3.4.2, specifically stated in Theorem 3.3, computing minimally correct shields in safety games under delay δ and memory μ corresponds to computing the maximally-permissive strategy ξ of the corresponding game. Then, using the notation in Section 3.4.2, given the maximally permissive winning strategy ξ, the minimally correct pre-shield is $\bigcirc_{\xi}^{pre}$, and minimally correct post-shields are computed as $\bigcirc_{\xi,\chi}^{pos}$, where χ is a determinization of ξ.

The following section describes the algorithm used to compute such strategies. As mentioned before, this is a natural extension of the algorithm presented in [Che+21] for games where the amount of memory and the delay are the same.

4.2.1 Computation of Maximally Permissive Winning Strategies in Safety Games under Delay

The algorithm is given in pseudocode in Algorithm 1. The method to solve a delayed safety game consists of iteratively constructing and solving the safety game with increasing delays $d = 0, 1, \ldots, \delta$ and memory size $m = \min(d, \mu)$, starting with $d = m = 0$, which corresponds to the case without delays, as presented in Section 2.3. At every iteration in d, the maximally permissive strategy for the agent is computed using the strategy for the previous delay $d-1$ (lines 4,5 or 9,10 depending on the value of m), followed by a reduction of the game graph aiming to mitigate the exponential blow-up in the state space (line 11) and the computation of the transient phase (line 12). Note that, following the convention in Equation 2.5, the action register $[y_1, \ldots, y_m]$ is in reversed order, i.e., the last action performed by the agent is y_m.

The method to compute the maximally permissive strategy using the previous delays is slightly different for the case of full memory ($m = d$) and the case of restricted memory $m < d$.

Algorithm 1: Maximally Permissive Strategy under Delay, memory $\mu \leq \delta$ (adapted and extended from [Che+18, Algorithm 1]).

```
   input : Safety Game 𝒢, maximum delay δ, memory μ
1  ξ_0 ← StrategyPerfectInfo(𝒢);
2  for d = 1, ..., δ do
3      m ← min(d, μ);
4      for s ∈ S_ag, [y_1, ..., y_m] ∈ 𝒜^m do
5          if m = d then
6              ℐ_{s,y_m} ← {s'' ∈ S_1 : s --y_m--> s' --u--> s''};
7              ξ_{d,m}(s, [y_1, ..., y_m]) ← ⋂_{s''∈ℐ_{s,y_m}} ξ_{d-1,m-1}(s'', [y_1, ..., y_{m-1}]);
8          else
9              ℐ_s ← {s'' ∈ S_1 : s --y--> s' --u--> s'', y ∈ 𝒜};
10             ξ_{d,m}(s, [y_1, ..., y_m]) ← ⋂_{s''∈ℐ_s} ξ_{d-1,m}(s'', [y_1, ..., y_m]);
11     Shrink(ξ_{d,m});
12     InitialMoves(ξ_{d,m});
13 return ξ_{δ,μ}
```

- Case $m = d$. To compute the maximally permissive strategy using previous delays, we compute I_{s,y_m} (line 6), corresponding to the set of states that the agent can get as the next observation when the current observation is state s and the chosen action is y_m. From states where the agent has already decided upon an output, it is equivalent to playing with delay $d-1$. Therefore, the strategy allows the actions that would be safe for delay $d-1$ on all possible next observations, eliminating the last executed action, y_m, from the action register (line 7).

- Case $m < d$. To compute the maximally permissive strategy using previous delays, we compute I_s (line 9), corresponding to the set of states that the agent can get as the next observation when the current observation is state s and chosen action is any $y \in \mathcal{A}$. In this case, y is undetermined because of the restricted memory: the output that is provided just next to the observed state has already been forgotten by the system. From states where the agent has already decided upon an output, it is equivalent to playing with delay $d-1$. Therefore, the strategy allows the actions that would be safe for delay $d-1$ in all possible next observations, maintaining, in this case, the same memory $[y_1, \ldots, y_m]$ (line 10).

The method `StrategyPerfectInfo` (line 1) computes the maximally permissive strategy for the game with perfect information [Tho95]. The method `Shrink` (line 11) ensures that in case the intersection in lines 5 or 10 is empty, the maximally permissive strategy in a state $s'' \in \mathcal{I}_{s,y_m}$ or $s'' \in \mathcal{I}_s$ does not contain the output y_m [Che+21, Algorithm 3]. The method `InitialMoves` (line 12) computes the strategy for the transient period, before the agent can get any observed state, see Algorithm 2.

Algorithm 2: `InitialMoves`: Strategy for the transient period (adapted from [Che+18, Algorithm 1]).

input : Safety game $\mathcal{G}$, ongoing maximally permissive strategy $\xi_{d,m}$

$\mathcal{J} = \{s : s_0 \xrightarrow{u} s\}$;
for $[y_1, \ldots, y_{m-1}] \in \mathcal{A}^{m-1}$ **do**
  $\xi_{d,m}(\varepsilon, [y_1, \ldots, y_{m-1}]) \leftarrow \{y_d : \bigcup_{s \in \mathcal{J}} \xi_d(s, [y, y_1, \ldots, y_{m-1}]) \neq \emptyset\}$;
for $k = m-2, \ldots, 0$ **do**
  for $[y_1, \ldots, y_k] \in \mathcal{A}^k$ **do**
    $\xi_{d,m}(\varepsilon, [y_1, \ldots, y_k]) \leftarrow \{y_0 : \xi_{d,m}(\varepsilon, [y_0, y_1, \ldots, y_k]) \neq \emptyset\}$;
return $\xi_{d,m}$

Complexity analysis. Algorithm 1 computes each strategy $\xi_{d,m}$ for increasing values of $d = 1, \ldots, \delta$ (main loop, lines 2-12). For each strategy, the algorithm goes over all states in S_{ag} and registers in $\mathcal{A}^m$ (loop in lines 4-10), and at each iteration, computes an intersection of $\mathcal{A}_{env}$ elements. The cost of `Shrink`($\xi_{\mathtt{d},\mathtt{m}}$) and `InitialMoves`($\xi_{\mathtt{d},\mathtt{m}}$) is negligible in comparison.

Therefore, the cost of computing the strategy $\xi_{d,m}$ is $\mathcal{O}(S_{ag} \cdot |\mathcal{A}|^m \cdot |\mathcal{A}_{env}|)$, and the total cost of computing $\xi_{\delta,\mu}$ is $\mathcal{O}\left(|S_{ag}| \cdot |\mathcal{A}_{env}| \cdot (\mu \cdot |\mathcal{A}|^\mu + (\delta - \mu)\mathcal{A}^\mu)\right)$, simplified to

$$\mathcal{O}\left(\delta \cdot |S_{ag}| \cdot |\mathcal{A}_{env}| \cdot \mathcal{A}^\mu\right). \tag{4.1}$$

Recall that $\mathcal{A}_{env}$ represents a set of actions for the environment. As discussed in Equation (2.1), without loss of generality, we can assume $|\mathcal{A}_{env}|$ is the maximum out-degree of the environment transitions.

4.3 Determinization of Strategies Resilient to Delays

The synthesis procedure for a delay-resilient post-shield relies on the construction of a deterministic winning strategy, denoted by $\chi_{\delta,\mu}$. This section outlines the process for deriving such a strategy.

In Section 4.3.1, we describe the method for computing the deterministic strategy that maximises a given fitness value, considering both memory and delay.

We present two examples of fitness functions in Sections 4.3.2 and 4.3.3. These are specifically designed to minimise the number of instances where the post-shield must interfere due to delays in the input.

4.3.1 Determinisation of Delayed Strategies Maximising a Fitness Function

When deciding which action to use as a corrective action for post-shields, we want to decide on an action that maximizes a certain criterion. In this section, we assume the existence of a fitness function $\varphi\colon S \to \mathbb{R}$ that assigns a fitness value to each state. We will show how to find the actions that maximize an abstract fitness value, and in the following sections, we will apply this method to concrete fitness functions designed to prevent unsafe transitions due to delayed inputs.

Our goal is to choose at each state the action that maximises this fitness function among all actions allowed by $\xi_{\delta,\mu}$. However, because of the uncertainty of the transitions of the environment, it is not clear what it means to maximise the fitness function, since the agent has no complete control over the state of the safety game after each of the agent's actions.

For this computation, we will take the implicit assumption that from a given state $s \in S_{env}$, and a given number of transitions n, all traces from s with n transitions are equally probable. In this sense, we say that the strategy we compute maximises the *expected fitness value* – with the implicit understanding that the expectation is taken under the assumption of a uniform probability environment. This assumption can be refined to include more accurate representations of the probabilistic nature of the environment whenever such models are available. We leave this extension for future work.

For our computation, we need to define the *k-forward multiset of states* $F_k(s, \overline{\sigma})$, which captures the states reachable from s within k steps respecting a given memory $\overline{\sigma} \in \mathcal{A}^\mu$, i.e. the last μ actions of the agent.

Definition 4.1 (k-Forward Multiset of States)**.** Let $\overline{\sigma} = [z_1 \dots z_\mu] \in \mathcal{A}^\mu$ be a register of actions. For a state $s \in S_{ag}$, the *k-forward multiset* is

$$F_k(s,\overline{\sigma}) = \left\{ s_{2k} \quad : \exists s_1,\dots,s_{2k-1} \in S, \text{ and } \exists y_1,\dots,y_k \in \mathcal{A} \text{ such that} \right.$$
$$(1)\, \forall i = 1 \dots \mu,\ y_{k-\mu+i} = z_i, \text{ and}$$
$$\left. (2)\, s \xrightarrow{y_1} s_1 \xrightarrow{u} s_2 \xrightarrow{y_2} s_3 \xrightarrow{u} \dots \xrightarrow{u} s_{2k-2} \xrightarrow{y_k} s_{2k-1} \xrightarrow{u} s_{2k} \right\},$$

where each state s_{2k} is counted as many times as there are distinct sequences $s_1,\dots,s_{2k-1} \in S$ and $y_1,\dots,y_k \in \mathcal{A}$ satisfying conditions (1) and (2).

The expected fitness value is computed over the k-forwarded multiset of states, thus, each state adds to the value as many times as it appears in the multiset.

Definition 4.2 (Expected Fitness Value)**.** Let $s \in S$ be a state, $\overline{\sigma} \in \mathcal{A}^\mu$ a register of actions and $\varphi : S \to \mathbb{R}$ a fitness function. For a given delay δ, the *expected fitness value* $\mathbb{E}_\varphi(s, \overline{\sigma})$ is defined as the *average* of the fitness values of all states s' in $F_\delta(s, \sigma)$,

$$\mathbb{E}_\varphi(s, \overline{\sigma}) = \frac{1}{|F_\delta(s, \overline{\sigma})|} \sum_{s' \in F_\delta(s, \overline{\sigma})} \varphi(s').$$

The strategy $\chi_{\delta,\mu} \colon S_{ag} \times \mathcal{A}^\mu \to \mathcal{A}$ that maximises the expected value of φ is:

$$\chi_{\delta,\mu}\big(s, [z_1 \dots z_\mu]\big) = \underset{y \in \xi_{\delta,\mu}(s, [z_1 \dots z_\mu])}{\arg\max} \mathbb{E}_\varphi\big(s, [y, z_1 \dots z_\mu]\big).$$

Complexity of strategy determinisation. The deterministic strategy is computed for $|S_{ag}| \cdot |\mathcal{A}|^\mu$ states. Each forward multiset contains at most $|\mathcal{A}_{env}|^\delta \cdot |\mathcal{A}|^{\delta-\mu}$ states, where $\mathcal{A}_{env}$ is a set of actions for the environment. For each of these states, the fitness value φ is computed. Assuming $c(\varphi)$ is the computational cost of computing φ for one state, and φ is stored in a lookup table, the total complexity adds up to

$$\mathcal{O}\left(|S_{ag}| \cdot \left(|\mathcal{A}_{env} \times \mathcal{A}|^\delta + c(\varphi)\right)\right). \tag{4.2}$$

4.3.2 Post-Shields that Maximise Controllability

In this section, we define and compute a fitness function called the controllability value. that assigns to each state the maximum delay for which a safe output exists. For any state $s \in S_{ag}$, the controllability value $\varphi_c : S_{ag} \to \mathbb{R}$ is the largest delay for which a register of actions exists that makes s safe. To formally define the controllability value, we use the notion of controllable states.

Definition 4.3 (Controllable State)**.** A state $s \in S_{ag}$ is *controllable* under delay δ and memory μ if there exists $\overline{\sigma} \in \mathcal{A}^\mu$ such that $\xi_{\delta,\mu}(s, \overline{\sigma}) \neq \emptyset$, and *uncontrollable* otherwise. A state $s \in S_{env}$ is *controllable* under delay δ and memory μ if all states $s' \in S_{ag}$ such that $s \xrightarrow{u} s'$ are controllable under delay δ and memory μ.

Definition 4.4 (Controllability Value)**.** The *controllability value* with memory μ of a state $s \in S$ is the maximum delay δ for which s is controllable with delay δ and memory μ. We denote it as $\varphi_c(s)$.

To unpack this definition, for an agent state $s \in S_{ag}$, we say that $\varphi_c(s) = \delta$ if there exists $\overline{\sigma} \in \mathcal{A}^\mu$ such that $\xi_{\delta,\mu}(s, \overline{\sigma}) \neq \emptyset$ and for all $\overline{\sigma}' \in \mathcal{A}^\mu$, we have $\xi_{\delta+1,\mu}(s, \overline{\sigma}') = \emptyset$.

In Definition 4.3, guaranteeing only the existence of $\overline{\sigma} \in \mathcal{A}^\mu$ such that $\xi_{\delta,\mu}(s, \overline{\sigma}) \neq \emptyset$ might seem too weak because one does not know in advance what the action register may be when observing a state. Note that, however, Algorithm 1 guarantees that an agent following $\xi_{\delta,\mu}$ can only go through pairs $(s, \overline{\sigma}) \in S_{ag*} \times \mathcal{A}^\mu$ such that $\xi_{\delta,\mu}(s, \overline{\sigma})$ is non-empty. This is one of the main consequences of the

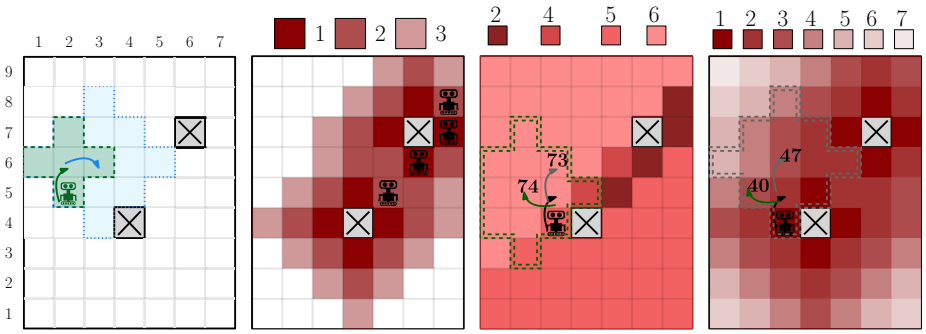


Figure 4.2: **(a)** Effects of delay on state observation. **(b)** Gridworld depicting the least delay-resilient states. **(c)** Gridworld with φ_{dr} values for all states. **(d)** Gridworld with φ_{rs} values for all states.

`Shrink` method in Algorithm 1, and is extensively discussed in [Che+21, Algorithm 3].

If a state s is inside the winning region W of the safety game without delay, it has a controllability value greater or equal to 0. Furthermore, note that as a consequence of the iterative computation of winning strategies for safety games under delay, if a state s is controllable for delay $\delta > 0$, it is also controllable for delay $\delta - 1$. By convention, if $s \notin W$, i.e., there is no delay δ that makes it controllable, we say that $\varphi_c(s) = -1$.

A shield that maximizes controllability will tend to steer the agent towards states that can be safe even with large delays. In a setting with variable delay, the shield always operates with the worst-case delay in mind, but the agent may make a more refined use of the variable delay, so steering the agent towards high-controllability states translates to more freedom for which actions to choose in the future, as there are fewer paths leading to uncontrollable states.

Since the maximal delay possible can be very large and the state space of the corresponding safety game grows exponentially with the delay, we introduce a cutoff value $\delta_{\max}$ and compute the maximally-permissive winning strategy until $\delta_{\max}$.

Example 4.2. *We showcase the computation of φ_c on a simple 7×9 gridworld, depicted in Fig. 4.2(a). Initially, a robot is placed at $(1, 9)$. The environment and the agent can move the robot by one field in alternating turns. The safety specification requires that the robot never visits the fields $(4, 4)$ nor $(6, 7)$. Encoding the model and the specification leads to the safety game $\mathcal{G} = \langle S, S_{ag}, S_{env}, \mathcal{A}, \mathcal{T}, \mathcal{F} \rangle$:*

- *$S = X \times Y \times \mathbb{B}$, where $X = \{1, \ldots, 7\}, Y = \{1, \ldots, 9\}$ represent the robot's position and $\mathbb{B} = \{\top, \bot\}$ indicates whether it is the turn of the agent ($\top$) or the environment ($\bot$) to move the robot. The states of the environment are $S_{env} = X \times Y \times \{\bot\}$, and of the agent are $S_{ag} = X \times Y \times \{\top\}$.*
- *The unsafe states are $S \setminus \mathcal{F} = \{(4, 4), (6, 7)\} \times \mathbb{B}$.*

- *The agent's actions are* $\mathcal{A} = \{\mathtt{U}, \mathtt{D}, \mathtt{R}, \mathtt{L}, \mathtt{N}\}$ *to move the robot one field up, down, right, left, or to hold. Formally:* $(x, y, \top) \xrightarrow{\mathtt{U}} (x, y+1, \bot)$, $(x, y, \top) \xrightarrow{\mathtt{D}} (x, y-1, \bot)$, $(x, y, \top) \xrightarrow{\mathtt{R}} (x+1, y, \bot)$, $(x, y, \top) \xrightarrow{\mathtt{L}} (x-1, y, \bot)$, $(x, y, \top) \xrightarrow{\mathtt{N}} (x, y, \bot)$.
- *The actions of the environment player to move the robot are* $\mathcal{A}_{env} = \{\mathtt{U}', \mathtt{D}', \mathtt{R}', \mathtt{L}', \mathtt{N}'\}$, *with a meaning analogue to those of the agent's actions.*

Moves that would lead the robot outside of the game's boundary are replaced by N. *For this game* $\mathcal{G}$, *we now compute the controllability values* $\varphi_c(s)$ *for all states* $s \in S_{ag}$.

First, we illustrate in Fig. 4.2(a) the effects of delays on the state information of the play. In the example, we have a delay $\delta = 1$ *and memory* $\mu = 1$, *and the observed state of the game is* $s = (2, 5, \top)$ *(green robot) with memory* $\overline{\sigma} = [\mathtt{U}]$. *The set of possible current states is* $F_1(s, \overline{\sigma})$ *(marked green). To check whether a next action* $y =$ R *is safe, we compute* $F_2(s, [\mathtt{R}, \mathtt{U}])$ *(marked blue or green). Since* $F_2(s, [\mathtt{R}, \mathtt{U}]) \subseteq \mathcal{F}$, R *is a safe action from* $(s, [\mathtt{U}])$.

Next, we exemplify the computation of the controllability values for the states $(5, 5), (6, 6), (7, 7)$ *and* $(7, 8)$. *In Fig. 4.2 (b), each field of the grid is coloured according to the smallest distance to one of the unsafe states for distances 1, 2 and 3. With this colour coding, the state* $(x, y, \top)$ *is controllable with delay* δ *if there exists sequence of actions* $\overline{\sigma}$ *of size* δ, *that takes the robot outside of the region coloured with* δ. *The reader can see that all states are controllable for delay* $\delta = 1, 2$, *but for* $\delta = 3$, *states* $(5, 5), (6, 6), (7, 7)$ *and* $(7, 8)$ *are uncontrollable (marked with a black robot).*

Fig. 4.2 (c) illustrates the controllability value $\varphi_c(s)$ *for all states. Each field of the grid is coloured according to its controllability value. Next, we exemplify how to compute a deterministic strategy* $\chi_{\delta=1,\mu=1}(s, \overline{\sigma})$ *that maximizes the average of* φ_c *over all possible current states. Consider a state* $s = (3, 4, \top)$ *(black robot) with memory* $\overline{\sigma} = [\mathtt{U}]$. *The only two outputs allowed by* $\xi_{1,1}(s, \overline{\sigma_1})$ *are* U *and* L *since any other output would lead the robot to a state at a distance two or less from an unsafe state. The forward multiset* $F_2(s, [\mathtt{L}, \mathtt{U}])$ *is marked with a dashed green line and results in the expected value* $\mathbb{E}_{\varphi_c}(s, [\mathtt{L}, \mathtt{U}]) = 74/26$. *The expected value* $\mathbb{E}_{\varphi_c}(s, [\mathtt{U}, \mathtt{U}]) = 73/26$ *is computed analogously. A delay-resilient post-shield that maximises the controllability value corrects the outputs* R, D, *and* N *in state* $((3, 4, \top), [\mathtt{U}])$ *to the output* L.

Complexity of computing controllability values. Computing the controllability value as a fitness function only requires computing the maximally permissive winning strategy for the delay $\delta_{\max}$ chosen as the cutoff value — see Equation (4.1).

4.3.3 Post-Shields that Maximise Robustness

In this section, we define an alternative fitness function. The *robustness value* $\varphi_r : S \to \mathbb{R}$ assigns to every state the shortest distance to any unsafe state in the game graph. Intuitively, a large robustness value suggests that the system is in a state that "easily" satisfies the specification, while values near zero suggest

that the system is close to violating it. A shield that maximises robustness potentially requires fewer corrections in the near future.

Definition 4.5 (Robustness Value)**.** Let $\mathcal{G} = \langle S, s_0, S_{env}, S_{ag}, \mathcal{A}, \mathcal{T}, \mathcal{F} \rangle$ be a safety game with winning region W – as defined in Equation (2.3). For any state $s \in S_{ag}$, the robustness value $\varphi_r(s)$ is defined as the smallest k such that there exists $\overline{\sigma} \in \mathcal{A}^k$ such that $F_k(s, \overline{\sigma}) \not\subseteq W$.

Note that in our definition, we are counting distance only in states of the agent. That is, when a state has a robustness value of k, there exists a trace with $2k$ states – half of them of the agent, half of them of the environment – that leads to a state outside of the winning region.

Example 4.3 (Continuation of Example 4.2)**.** *We exemplify the computation of $\varphi_r(s)$ on the gridworld of Fig. 4.2 (d). Each field of the gridworld is colored with $\varphi_r(s)$ of its corresponding agent state s. For any s, the fitness function $\varphi_r(s)$ is computed as the distance to the closest unsafe state. From $s = (3, 4, \top)$ with $\overline{\sigma} = [\mathtt{U}]$ at delay $\delta = 1$ and memory $\mu = 1$, $\xi_{1,1}(s, \overline{\sigma})$ allows the actions* `U` *and* `L`*. Since the expected robustness value $\mathbb{E}_r(s, [\mathtt{U}, \mathtt{U}])$ is greater than $\mathbb{E}_r(s, [\mathtt{R}, \mathtt{U}])$, the deterministic strategy $\chi_{1,1}(s, [\mathtt{U}])$ that maximizes φ_r would choose* `U` *as corrective output.*

A shield that maximizes robustness will steer the agent towards states that are as far away as possible in the game graph from unsafe states. While this is a useful heuristic, note that distance in the safety game may not translate to real safety under delayed observations, as there may be states that are far away from the unsafe region, but with well-defined paths that the environment can force the agent to take toward unsafe states. We explore some of these examples in the following section.

Complexity of computing robustness values. The fitness function φ_r can be computed as a breadth-first search on the states, so all robustness values can be computed in $\mathcal{O}(|S|)$ time and memory.

4.4 Relation between Robustness and Controllability

Once we have shields that maximize robustness and controllability, we would like to find results that guarantee certain safety properties when using these shields. By construction, the post-shields defined in the previous sections maximize their corresponding fitness function.

By increasing the value of δ, we can make post-shields that guarantee a certain controllability value. Similarly, we can add a buffer zone to the winning region W, to force that only states at a distance at least d from unsafe states are ever visited.

A more interesting guarantee would be some result that guarantees robustness values in terms of controllability, and vice-versa. In this section, we study the relationship between robustness and controllability values.

Although the intuition behind robustness and controllability is very similar, and in our experiments, we found them to be equal most of the time, we show that only very basic relations hold in general. We show that there are example games where robustness is arbitrarily higher than controllability (Theorem 4.2), and vice-versa (Theorem 4.3). These examples would break any result of guaranteeing controllability when maximizing robustness, or robustness when maximizing controllability.

4.4.1 Relation between Robustness and Controllability for Memory-Restricted Strategies.

For strategies with a restricted memory. i.e., with $\mu \leq \delta$, we show a single result and explore its consequences for the edge cases.

Theorem 4.1. *Let $\mathcal{G}$ be a safety game with delay δ and memory size μ. For any controllable state $s \in S_{ag}$ it holds that*

$$\varphi_r(s) \geq \delta - \mu + 1. \tag{4.3}$$

Proof. We prove the result by contradiction. Let $s \in S_{ag}$ be controllable, i.e. with $\xi_{\delta,\mu}(s, \overline{\sigma}) \neq \emptyset$, for some $\overline{\sigma} \in \mathcal{A}^{\mu}$. Assume that $\varphi_r(s) < \delta - \mu + 1$, or equivalently, $\varphi_r(s) \leq \delta - \mu$.

Then, there exists a trace of length $\delta - \mu$ leading outside of the winning region W. Let $\tau = s, s_1, \ldots, s_{2(\delta-\mu)}$ be such trace, where $s_{2(\delta-\mu)} \notin W$ and $s \xrightarrow{y_1} s_1 \xrightarrow{u} s_2 \xrightarrow{y_2} s_3 \xrightarrow{u} \ldots \xrightarrow{u} s_{2(\delta-\mu-1)} \xrightarrow{y_{\delta-\mu}} s_{2(\delta-\mu)-1} \xrightarrow{u} s_{2(\delta-\mu)}$, for some actions $y_1, \ldots, y_{\delta-\mu} \in \mathcal{A}$.

Since the memory of the agent is limited to μ, when the observed state is s, the agent only knows that the current state is s' at a distance δ from s, with a trace where the last μ actions are known. However, this would already be too late: whatever the last μ actions are, any trace starting with τ – of which the agent has no control – will pass through $s_{2(\delta-\mu)} \notin W$. Therefore, s cannot be controllable with delay δ and memory μ. This proves the result. □

A corollary of Theorem 4.1 is that $\varphi_r(s) \geq \varphi_c(s) - \mu + 1$, since the controllability value is a valid delay for which a state s is controllable.

This result gives us information about the minimum amount of memory required for a winning strategy. The argument is as follows.

A safety game $\mathcal{G}$ admits a winning strategy under delay δ if any state $s \in S_{ag}$ with $s_0 \xrightarrow{u} s$ is controllable under delay δ. Therefore, the minimum amount of memory required for a winning strategy is

$$\mu \geq \delta - \varphi_r(s_0) + 1. \tag{4.4}$$

This implies that for a fixed game $\mathcal{G}$ and increasing delay δ, the amount of memory needed to have a winning strategy increases after a certain threshold. At some point, the delay is so high that no memoryless strategies exist: if we set $\mu = 0$ in Equation (4.4), we get $\delta \leq \varphi_r(s_0) - 1$.

On the other extreme case, if we set $\mu = \delta$ in Equation (4.4), we get $\varphi_r(s_0) \geq \delta - \delta + 1 = 1$, which just means $s_0 \in W$. More generally, if we set $\mu = \delta$ in Equation (4.3), we get that any state s that is controllable satisfies $\varphi_r(s) \geq 1$, which just means that s is in the winning region. Therefore, Theorem 4.1 provides no bound for φ_c in terms of φ_r.

4.4.2 Relation between Robustness and Controllability for Strategies with Full Memory.

In this section, we will prove two results that give counterexamples to any possible bound of φ_c in terms of φ_r and vice-versa, for strategies with full memory, i.e., with $\mu = \delta$.

Theorem 4.2. *For all delay $\delta > 0$, and all $k \geq 0$, there exists a safety game $\mathcal{G}^k_{\delta,\mu=\delta} = \langle S, s_0, S_{ag}, S_{env}, \mathcal{A}, \mathcal{T}, \mathcal{F} \rangle$ with one state $s \in S$ satisfying*

$$\varphi_c(s) < \delta \quad \text{and} \quad \varphi_r(s) \geq \delta + k + 1.$$

Proof. We will do the proof by induction on k for any delay δ.

Base Case. For $k = 0$, we need to construct a safety game $\mathcal{G}^k_\delta$ containing a state s, that is uncontrollable for delay δ, but at least $\delta + 1$ steps are needed to get to an unsafe state. A game with a section as depicted in Figure 4.3 serves as an example, with action set $\mathcal{A} = \{x, y\}$.

For any given delay δ, the dotted pattern in the middle of the figure repeats $\delta - 3$ times. In this case, we will prove that state s is not controllable for delay δ. The environment has a choice in s_e for the next state to be $\hat{s}$ or $\tilde{s}$. When the observed state is s, the current state is either $\hat{s}'$ or $\tilde{s}'$. Any action register will consist of a sequence of x's and y's, and the only information relevant in the register is the parity of y's: an even number of y actions takes the state of the game from $\hat{s}$ to $\hat{s}'$ or from $\tilde{s}$ to $\tilde{s}'$, while an odd number of y actions takes the state of the game from $\hat{s}$ to $\tilde{s}'$ or from $\tilde{s}$ to $\hat{s}'$.

Therefore, in this game graph, without knowing the first choice of the environment – state $\hat{s}$ or $\tilde{s}$ –, the agent cannot know whether the current state is $\hat{s}'$ or $\tilde{s}'$. Since a safe action in $\hat{s}'$ leads to the unsafe state $s_\times$ when taken from $\tilde{s}'$ and vice versa, the state s is uncontrollable for delay δ.

Induction step. For a general k, we use the property for $k - 1$. Consider the game graph $\mathcal{G}^{k-1}_\delta$ satisfying the hypothesis.

This graph contains a state s uncontrollable for delay δ and with a robustness value of at least $\delta + (k-1) + 1 = \delta + k$. Without loss of generality, we assume that the robustness value is exactly $\varphi_r(s) = \delta + k$. If $\varphi_r(s)$ was larger, $\mathcal{G}^{k-1}_\delta$ at state s would serve already as $\mathcal{G}^k_\delta$ and the induction step would be finished.

State s being uncontrollable means that for any action register $\overline{\sigma} = (y_1 \dots y_\delta)$, the forward multiset $F_\delta(s, \overline{\sigma})$ contains at least another new uncontrollable state s' — which may be inside or outside W. The same argument can be applied to the newly introduced uncontrollable states. Therefore, for any action register $\overline{\sigma}$, following repeatedly transitions from uncontrollable state to uncontrollable state, eventually leads to a state outside of the winning region W.

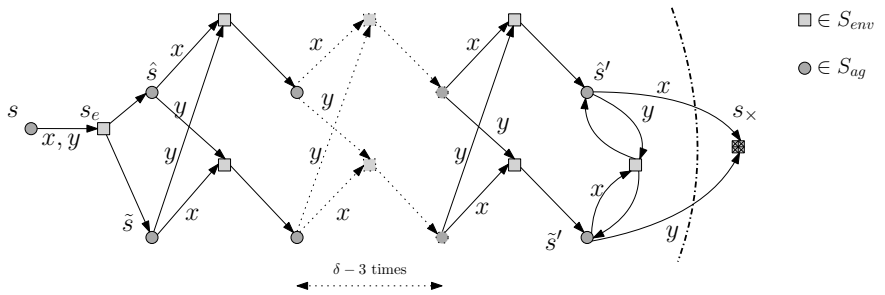


Figure 4.3: Construction for base case of Theorem 4.2. Square nodes represent states of the environment, circle nodes represent states of the agent. The dash-dotted line represents the divide between the winning region and the rest of the game. Except for states s_e and $s_\times$, only states and transitions of the agent are labelled.

We define $\Pi_U(s)$ as the set of all paths starting from s that end outside of the winning region in exactly $\delta + k$ transitions. That is, paths of the form $\tau = s, s_1, \ldots, s_{2(\delta+k)}$, with $s_{2(\delta+k)} \notin W$. This set is non-empty because $\varphi_r(s) = \delta + k$. We enumerate paths in $\Pi_U(s) = \{\tau_i \ : \ i \in I\}$ with some appropriate index set I.

While these paths are longer than δ, each of them contains at least one state that is uncontrollable because it leads directly outside the winning region W. For each $i \in I$, the path τ_i contains a state s^i that is uncontrollable for delay δ because it leads directly outside the winning region (and not to a state uncontrollable but inside the winning region).

Since s^i leads directly outside the winning region and is uncontrollable, we have the following construction, illustrated in Figure 4.4.

From state s^i, there is at least a path of length δ that ends in a state $s^b \in S_{ag}$, for which one transition with label x leads to a state outside of the winning region $s_d \notin W$. Since $s_b \in W$, there is another transition y leading to a safe state s^c. Since s^i is uncontrollable for delay δ, there is at least another state s'_b with a transition to an unsafe state s'_d labeled by y, and a transition to a safe state s'_c labelled by another action z. This builds a tuple of states and actions (s_b, s_d, x), as illustrated in Figure 4.4. We do not keep track of the rest of the states and actions defined but keep in mind that they exist.

Note that $s_b \neq s'_b$, $s \neq s^i$ and $x \neq y \neq z$. All the other states and actions could be the same. In particular, Fig. 4.5 is drawn assuming $s_d = s'_d$.

There may be other tuples of states and outputs bordering with the unsafe region. We enumerate them as $(s_b, s_d, x)^{i,j}$, where i is the index of s^i and $j \in \{1, \ldots, n_i\}$, where n_i is the number of different tuples when fixed $i \in I$. When it is convenient to distinguish concrete states and actions for each index (i, j), we use the equivalent notation $(s_b^{i,j}, s_d^{i,j}, x^{i,j})$ instead of $(s_b, s_d, x)^{i,j}$.

For each i, we make the following construction. Consider all the tuples $(s_b^{i,j}, s_d^{i,j}, x^{i,j})$ as previously described for $j \in \{1, \ldots, n_i\}$. We construct a new game $\mathcal{G}_\delta^{k-1}$ from

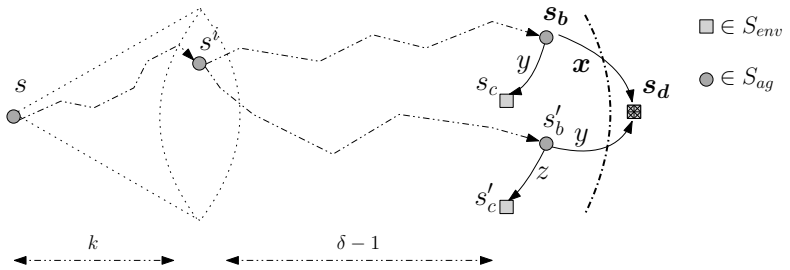


Figure 4.4: Construction of a tuple (s_b, s_d, x) on the border of the winning region as described in the text. Most i superscripts are omitted to make the image cleaner. Marked in bold are the elements that are part of the tuple.

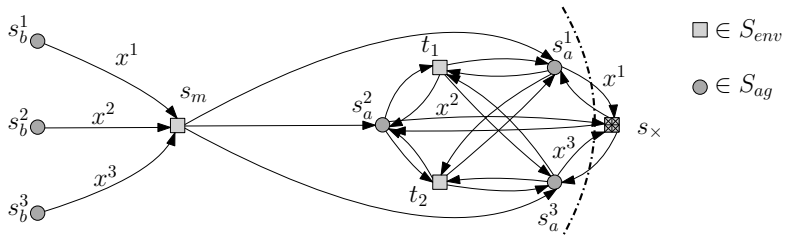


Figure 4.5: Construction of the bipartite complete graph described in the proof, for a single index i and $n_i = 3$. For the sake of simplicity all all three unsafe states $s_d^{i,j}$ for $j \in \{1, 2, 3\}$ are collapsed into a single unsafe state $s_\times \notin W$.

$\mathcal{G}_\delta^k$ where we add a state of $s_m^i \in S_{env}$ such that $s_b^{i,j} \xrightarrow{x^{i,j}} s_m^i$ for all i, j. Then we add new states $s_a^{i,j} \in S_{ag}$, one for each tuple. We connect these states as follows, for all $i \in I$ and all $j \in \{1, \ldots, n_i\}$:

$$s_m^i \xrightarrow{u} s_a^{i,j}, \qquad \text{and} \qquad s_a^{i,j} \xrightarrow{x^{i,j}} s_d^{i,j}.$$

We also add another family of states, indexed by k, denoted $t_k^i \in S_{env}$. We add as many of them to make it such that for all i, j and all action $x \neq x^{i,j}$, there exists k such that

$$s_a^{i,j} \xrightarrow{x} t_k^i.$$

We also connect all t_k with all the newly added agent states, that is

$$t_k^i \xrightarrow{u} s_a^{i,j}, \qquad \text{for all } i, j, k.$$

The unsafe states in $\mathcal{G}_\delta^k$ are inherited from $\mathcal{G}_\delta^{k-1}$. In particular, recall that the states $s_d^{i,j}$ are unsafe by construction for all i and j.

In Figure 4.5, we illustrate this construction for the case of three tuples ($n_i = 3$) on a single index i and an action set comprised of three actions ($\mathcal{A} = \{x^1, x^2, x^3\}$). For each value of i, there would correspond a similar separate construction. For larger values of n_i, the corresponding complete bipartite graph would become larger.

The first observation is that the states $s_a^{i,j}$ are only controllable for delay $\delta = 0$. This is because for any register of action $\overline{\sigma} = [x]$, $s_a^{i,j} \xrightarrow{x} t_k$ for some k, and then

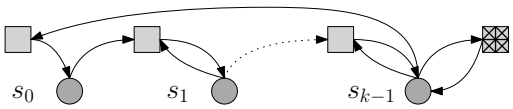


Figure 4.6: Game graph where $\varphi_c(s_0)$ is arbitrarily large, and $\varphi_r(s_0) = k$.

$t_k \xrightarrow{u} s_a^{i',j'}$ for all i', j'. So when the observed state is $s_a^{i,j}$, the current state (with delay $\delta = 1$), can be any of the states $s_a^{i',j'}$. By construction, any possible action x will be $x = x^{i',j'}$ for some i', j', and would lead to the state $s_d^{i',j'} \notin W$.

With this construction, the state s_i is still uncontrollable for delay δ, because any strategy that made it uncontrollable before leads now to s_m^i, which is uncontrollable for any delay larger or equal to one, as we have explained in the previous paragraph.

The existence of at least one of the t_k^i for each $i \in I$ is enough to ensure that states $s_a^{i,j}$ are safe, i.e., $s_a^{i,j} \in W$ for all i, j. By adding enough states t_k^i we ensure that each state newly added to S_{ag} has a defined transition for each action in $\mathcal{A}$.

Since the states $s_a^{i,j}$ are safe, the path τ_i needs to be extended by length 2 to arrive at an unsafe state, which would be one of the $s_d^{i,j}$.

Repeating this construction for each path τ_i of length $\delta+k$, we make all previous paths of length $\delta+k$ go through a construction as illustrated in Figure 4.5 before reaching any unsafe state, making it take at least one more action to reach any unsafe state. Thus, the robustness value of s is increased to $\delta+k+1$ in the new game graph $\mathcal{G}_\delta^k$, while the controllability value of s stays the same as it was in $\mathcal{G}_\delta^{k-1}$.

□

Theorem 4.3. *For all delay $\delta > 0$, and all $k > 0$, there exists a safety game $\mathcal{G}_{\delta,\mu=\delta}^k = \langle S, s_0, S_{ag}, S_{env}, \mathcal{A}, \mathcal{T}, \mathcal{F} \rangle$ with one state $s \in S$ satisfying*

$$\varphi_r(s) \leq \delta \quad \textit{and} \quad \varphi_c(s) \geq \delta + k.$$

Proof. Consider a safety game where the environment has only one choice in each state. In these kind of games, a player with memory can know exactly where it is making the next move, so each safe state is controllable. With this idea in mind, we construct a family of safety games $\mathcal{G}^k$ for which the initial state s_0 satisfies $\varphi_r(s_0) = k$ and is controllable with memory for any delay. Figure 4.6 illustrates this family of games.

□

4.5 Experimental Evaluation

For our experimental evaluation, we evaluate different types of shields resilient to delays with full memory on two use cases: a simple gridworld and a more complex scenario based on a realistic driving simulation.

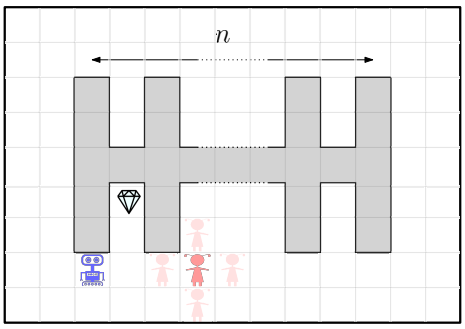


Figure 4.7: Gridworld with possible states after delay $\delta = 1$.

4.5.1 Shielding in a Gridworld

Setting. Our first case study is an extension of the one from [Che+21]. Figure 4.7 illustrates a grid world of size $3n+4\times 9$, where the width is parameterised by the number of pairs of dead-ends n. There are two actors that operate in the grid world: a robot (controlled by the agent), and a kid (controlled by the environment). The safety specification requires the robot to avoid any collision with the kid.

Game graph. The game graph encoding the relevant safety dynamics for the grid world is $\mathcal{G} = \langle S, s_0, S_{ag}, S_{env}, \mathcal{A}, \mathcal{T}, \mathcal{F}\rangle$, defined as follows.

- $S = X_{env} \times Y_{env} \times X_{ag} \times Y_{ag} \times \mathbb{B} \setminus P \times P \times \mathbb{B}$, where $X_{ag/env} = \{1, \ldots, 2n+5\}$ and $Y_{ag/env} = \{1, \ldots, 9\}$ represent the (x, y) position of the robot (agent) and the kid (environment), respectively. $\mathbb{B}$ indicates whether it is the turn of the robot or the kid, and P represents the illegal positions, marked in grey in Figure 4.7. Formally, $P = \{((x, 5), (2k + 1, y) \ : \ x \in \{3, \ldots, 2n + 3\},\ y \in \{3, \ldots, 7\},\ k \in \{1, \ldots, n + 1\}\}$. The initial state is $s_0 = (0, 0, 2n+5, 9, \bot)$, indicating that the robot is in the lower left corner, the kid is in the upper right corner and it is the kid's turn to move.
- The unsafe states are

$$S \setminus \mathcal{F} = \{(x_{env}, y_{env}, x_{ag}, y_{ag}, b) \ : \ (x_{env} = x_{ag}) \ \wedge \ (y_{env} = y_{ag})\}.$$

- The moves of the kid are defined by an action set $\mathcal{A}_{env} = \{\mathtt{U}', \mathtt{D}', \mathtt{R}', \mathtt{L}'\}$, with the usual meanings of up, down, right, left. We define a richer action set for the robot to compensate for the existence of delays in the input. The action set is $\mathcal{A} = \{\mathtt{N}, \mathtt{U}, \mathtt{D}, \mathtt{R}, \mathtt{L}, \mathtt{UU}, \mathtt{DD}, \mathtt{RR}, \mathtt{LL}, \mathtt{UR}, \mathtt{RU}, \mathtt{UL}, \mathtt{LU}, \mathtt{DR}, \mathtt{RD}, \mathtt{DL}, \mathtt{LD}, \mathtt{UUR}, \mathtt{UUL}, \mathtt{DDR}, \mathtt{DDL}, \mathtt{RRU}, \mathtt{RRD}, \mathtt{LLU}, \mathtt{LLD}\}$. In summary, the robot can move zero, one or two steps in each direction, and can also perform three-step L-shaped moves.
- The transitions work as expected. Environment transitions modify the position of the kid (x_{env}, y_{env}), while the agent's actions modify the position of the robot (x_{ag}, y_{ag}). Any illegal transition (those that would go out of boundaries or clash with the grey region depicted in Figure 4.7) is changed to $\mathtt{N}$ ("no move").

Delay (steps)		0	1	2	3
Score	Pre-shield	50.1	36.0	34.6	30.2
	Robustness	42.5	34.3	31.5	26.8
	Controllability	41.3	33.9	31.8	27.5
Interventions	Pre-shield	117.4	150.4	160.1	182.2
	Robustness	90.9	107.5	114.1	122.0
	Controllability	85.0	95.9	106.9	122.7

Table 4.1: Performance of different shielding strategies.

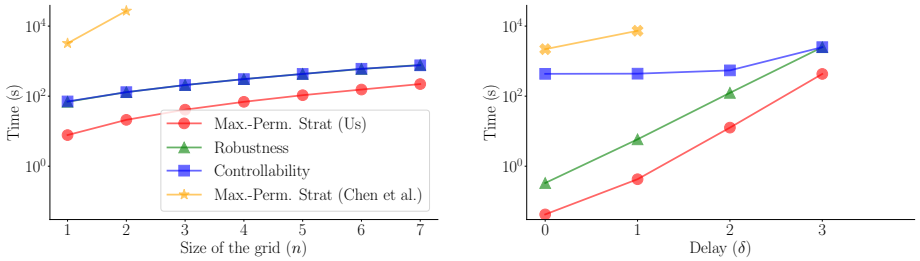
Results: interference rates. To evaluate the interference of the shields during runtime, we implemented a robot with the goal of collecting treasures that are placed at random positions in a grid world with 4 dead ends. At any time step, there is one treasure placed in the grid world. As soon as this treasure is collected, the next treasure spawns at a random location. Collecting a treasure rewards the agent with +1 score points. The kid is implemented such that it chases the robot in a stochastic way.

The interference results are presented in Table 4.1. In the table, the first three rows show the score obtained by the robot, and the last three rows show the number of times the shield intervenes. Both score and number of interventions correspond to the amount accumulated over a game of 2000 steps. We compare pre-shields with post-shields that maximise either robustness or controllability. Since both the robot and the kid are implemented with stochastic behaviour, each data point in the table is the average of 100 plays.

The results show that the agent's score decreases with the delay, as expected. Since the shield has more uncertainty about the current position of the kid, it enforces a larger distance between the current position of the robot and the last observed position of the kid. For the same reason, the shields need to interfere more frequently with increasing delays. In general, pre-shields compare to post-shields show a better performance in terms of score and a worse performance in terms of number of interventions, as it was expected. Additionally, we compared the corrective actions chosen by post-shields that maximise controllability with the actions chosen by shields that maximise robustness. We noticed that in most states, both shields pick the same corrective action, which is reflected in the similar results obtained.

Results: synthesis times. We compute all types of presented shields. The synthesis times are presented in Figure 4.8, where Figure 4.8a corresponds to a fixed delay of $\delta = 2$, and Figure 4.8b corresponds to a fixed-size grid with four dead-ends, i.e., $n = 2$.

In the figure we compare the synthesis times for the synthesis of shields, maximising robustness (▲) and controllability (■). We also include the cost of computing the maximally permissive winning strategy, which is required for all shields and is the only cost associated with synthesising pre-shields. To compare with a baseline, we show the cost of computing the maximally-permissive



(a) Fixed delay $\delta = 2$, increasing size of the grid.

(b) Fixed size of the grid $n = 2$, increasing delay.

Figure 4.8: Shield synthesis times for the grid world experiments.

strategy in the delayed safety game for our implementation (●) and the implementation of [Che+21] (✖).

The improvement of our method compared to the baseline results from a faster implementation in C++, with only minor algorithmic reasons. The cutoff value for controllability is set to $\delta_{\max} = 3$. Since the cost for computing shields grows exponentially with δ, the synthesis times for shields maximising robustness grow exponentially. This effect does not show for shields maximising controllability, as they always compute the maximally permissive strategy until delay $\delta_{\max}$ irrespective of the particular delay δ.

4.5.2 Shielded Driving in Carla

We implemented our delayed shields in the driving simulator Carla [Dos+17]. In all scenarios, the default autonomous driver agent in Carla is used with adequate modifications to make it a more reckless driver. To capture the continuous dynamics of Carla using discrete models, we designed the safety game with overly conservative transitions, i.e., accelerations are overestimated, and braking power is underestimated. In both scenarios, we use delay-resilient shields, maximising robustness.

4.5.2.1 Shielding against Collisions with Cars

We consider a scenario in which two cars (one of them controlled by the driver agent) approach an uncontrolled intersection. The shield has to guarantee collision avoidance for any braking and acceleration behaviour of the uncontrolled car while the observation of the uncontrolled car is delayed. A screenshot of the Carla simulation is given in Figure 4.9a.

Game graph. To compute delay-resilient shields, the scenario is encoded as a safety game $\mathcal{G} = \langle S, s_0, S_{ag}, S_{env}, \mathcal{A}, \mathcal{T}, \mathcal{F} \rangle$, defined as follows.

The set of states is defined as $S = P_{\text{ag}} \times P_{\text{env}} \times V_{\text{ag}} \times V_{\text{env}}$, where P_{ag} and P_{env} represent, respectively, the distances of the agent's car and the environment's car to the crossing, and V_{ag} and V_{env} represent the velocity of the agent's car and the environment's car, respectively. The range of modeled distances is

(a) Car intersection. (b) Pedestrians at a crosswalk.

Figure 4.9: Screenshots of the CARLA simulator.

$P_{\text{ag}} = P_{\text{env}} = \{0, 2, 4, \ldots, 100\}$ m. The range of modelled velocities is $V_{\text{agent}} = V_{\text{env}} = \{0, 1, 2, \ldots, 20\}$ m/s.

Each time step in the game corresponds to $\Delta t = 0.5$ s in the simulation. Each car can perform three actions: `a` (accelerate), `b` (brake) or `c` (coast, touch no pedal). Therefore, the set of environment actions is $\mathcal{A}_{env} = \{\mathtt{a}_{\text{env}}, \mathtt{b}_{\text{env}}, \mathtt{c}_{\text{env}}\}$ and the set of actions of the agent is $\mathcal{A} = \{\mathtt{a}_{\text{ag}}, \mathtt{b}_{\text{ag}}, \mathtt{c}_{\text{ag}}\}$. In our model, braking and throttling have the effect of applying a constant acceleration of $a = \pm 2$ m/s^2. Therefore, the position p_t and the velocity v_t at time step t is updated as

$$p_{t+\Delta t} = p_t - v_t \Delta t - \tfrac{1}{2} a \Delta t^2, \qquad v_{t+\Delta t} = v_t + a \Delta t. \tag{4.5}$$

Unsafe states represent collisions, therefore $\mathcal{S}_{\text{unsafe}} = \{(p_{\text{agent}}, v_{\text{agent}}, p_{\text{env}}, v_{\text{env}}) : p_{\text{agent}} = p_{\text{env}}\}$. From the safety game, we compute delay-resilient shields that maximise the expected robustness. Note that in the transitions in Equation (4.5) the velocity is applied as negative because the car gets closer to the intersection at every step.

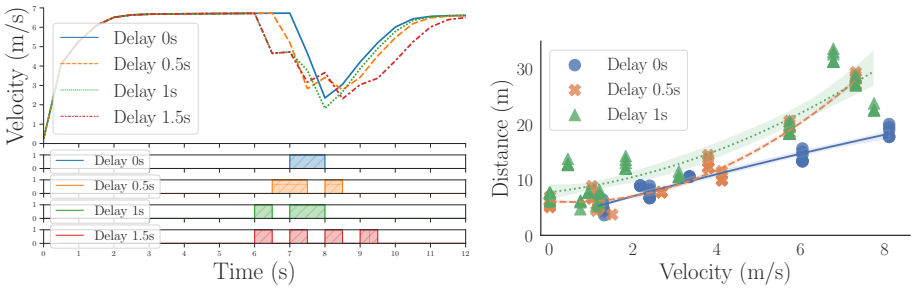
In this use case, we implemented post-shields that always correct to the most conservative safe action, with the understanding that `c` (coast) is more conservative than `a` (accelerate) and that `b` (brake) is more conservative than both `a` and `c`.

Results. In Figure 4.10a, we present the speed of the agent's car over time, alongside the occurrences of shield interventions, represented as coloured bars, for various delays measured in increments of $\Delta t = 0.5$ s. As anticipated, the duration of shield interference increases with larger delays.

For a delay of 0, the agent's car brakes continuously until it exits the danger zone. However, as the delay increases, the shield intervenes earlier, ensuring the car accounts for the worst-case behaviour of the other vehicle. The shield always assumes the most adverse environmental conditions, even when these conditions fail to materialise. This conservative approach explains the frequent

Delay (in steps)		0	1	2	3
Synthesis times (in s)	Car example	1.5	13	48	167
	Pedestrian example	0.8	9	34	119

Table 4.2: Shield synthesis times (in seconds).



(a) Activation times results. (b) Activation speeds and distances.

Figure 4.10: Experimental results on the driving simulator. In these experiments, we measure when and how often the shields get activated in each scenario.

switching between active and inactive shield states within the same execution, particularly for larger delays.

We evaluated the shields across multiple safety-critical scenarios by varying initial positions and velocities. In all cases, the shields successfully prevented collisions, demonstrating their robustness. Table 4.2 provides the synthesis times required to compute the shields. Each delay step listed in Table 4.2 corresponds to an increment of $\Delta t = 0.5$ s.

4.5.2.2 Shielding against Collisions with Pedestrians

In the second experiment, we compute shields for collision avoidance with pedestrians. Similar to before, the shields guarantee safety under delay, even under the worst possible behaviour of the pedestrians. A screenshot of the CARLA simulation is given in Figure 4.9b.

Shield computation. The car, which is controlled by the driver agent, is modelled in the same manner as before. Pedestrians are controlled by the environment and only have their position as state variables. In our model, we assume that a pedestrian can move 1 m in any direction within one timestep of $\Delta t = 0.5$ s. We consider a state to be unsafe whenever the ego car moves fast while being close to a pedestrian and the pedestrian is closer to the crosswalk than the car. Formally

$$\mathcal{S}_{\text{unsafe}} = \{(p_{\text{ag}}, v_{\text{ag}}, p_{\text{ped}}) : (v_{\text{ag}} > 2 \text{ m/s} \wedge |p_{\text{ag}} - p_{\text{ped}}| < 5 \text{ m} \wedge p_{\text{ped}} < p_{\text{ag}})\}$$

Results. In Figure 4.10b, we illustrate the shield's interference points by plotting the distance to the pedestrian and the car's speed at the moment of each shield intervention. Because pedestrians are modelled to potentially move toward the car, the shield must account for closer pedestrian positions than those directly observed, as delays in sensing introduce uncertainty.

With larger delays, this uncertainty grows, requiring the shield to initiate braking earlier to ensure safety. This conservative approach ensures that the shield

compensates for any positional ambiguity introduced by the delay. The synthesis times required for the shield computation are provided in Table 4.2.

In our experiments, we occasionally observed the system entering states with no available strategy due to discretisation errors. However, despite these occurrences, the safety specification was never violated. Our findings suggest that by using a sufficiently fine-grained model, these discretisation errors can be minimised to the point of being negligible, ensuring the system operates reliably under all tested conditions.

4.6 Discussion

4.6.1 Limitations

We have demonstrated with our experiments that shielding can be a useful tool to ensure safety specifications in an application so complex as autonomous driving. However, this is still a methodology that is not ready to be implemented in today's technology. In this section, we discuss the main limitations that hinder the applicability of our method. Further research and development is needed to address these challenges.

Requiring a deterministic model. One of the foundational assumptions of our method is the availability of a deterministic model of the system and its environment. While we tackle one of the sources of uncertainty in this chapter by proposing shields resilient to delayed information, this assumption may still be unrealistic. Many real-world systems operate in inherently stochastic environments, where uncertainties arise due to sensor noise, unpredictable human behaviour, or dynamic external factors. Attempting to model such systems deterministically may lead to oversimplifications, resulting in shields that fail to capture the full complexity of the environment. In our driving simulator experiments, we circumvent this limitation by finding a rather conservative model that reflects the reality most of the time, and observed that this model was good enough in our experiments to enforce the safety specification. This is an imperfect solution and requires fine-tuning the model for each application, making the implementation of shielding more labour-intensive.

Overly conservative strategies Safety shields are designed to handle worst-case scenarios, ensuring that safety is maintained regardless of how adverse conditions may become. While this is the only way to get the strong safety guarantees that shielding provides, it can lead to overly conservative strategies that limit the agent's performance and utility excessively. Furthermore, the handling of delayed observations only accentuates the shield's conservativism. Over-conservatism can also erode user trust, as the system may appear unnecessarily cautious or suboptimal in typical operational conditions. This trade-off between safety and performance poses a significant challenge and calls into question whether deterministic shielding can be a viable solution.

Handling divergences from the model A fundamental limitation of safety shielding lies in its reliance on a predefined safety game model. In practice, the

real world may diverge from these models due to inaccuracies in modeling or changes in the environment over time. When such divergences occur, the shield is forced to react to situations that were not considered reachable in the original safety game. In our current approach, the shield's behaviour is undefined in these situations.

Overcoming these challenges will be essential for deploying safety shielding techniques in increasingly complex and dynamic real-world systems.

4.6.2 Related Work

Runtime enforcement is a technique in which a monitor modifies the execution of a system to comply with a specified property [FP19; Ren+19]. Shields for discrete systems were introduced in [Blo+15] and several extensions and applications have already been published [Als+18; Els+21; Pra+21b; Jan+20; Yan+23b].

Chen et al. [Che+18; Che+21] first investigated the synthesis problem for time-delay discrete systems by the reduction to solving two-player safety games. We base our shields on the proposed algorithm for solving delayed safety game. Note that the delayed games discussed by Zimmermann et al. [KZ15a; KZ15b; Zim17; WZ20] follow a concept different from the delayed safety games considered in this paper. In their setting, *a delay is a lookahead* that grants an advantage to the delayed player: the delayed agent player P1 lags behind the environment player P0 in that P1 has to produce the i-th action when $i + j$ environment actions are available. In contrast to Zimmermann et al., we do not grant a lookahead into future inputs but consider delays in the input data. It was shown in [Che+18] that the different concepts of delays could not be exchanged for each other by a swap of roles i.e., by exchanging the players and then giving a lookahead of j to the input player in order to simulate a delay of j for the output player.

The notion of delay employed in this paper is different from that in timed games [Beh+07]. In timed games, delay refers to the possibility of deliberately delay the next single action. However, both players have full and up-to-date information in timed games. In [Dav+13] a general framework on using UPPAAL-TIGA with partial observability was presented. Combining both approaches to synthesise delay-resilient shields from timed automata specifications is potential future work.

For runtime enforcement in continuous dynamical and hybrid systems, control barrier functions [Ame+19] are used to verify and to enforce safety properties. Prajna and Jadbabaie extended the notion of barrier certificates to time-delay systems [PJ05]. Bai et al. [Bai+21b] introduced a new model of hybrid systems, called delay hybrid automata, to capture the continuous dynamics of dynamical systems with delays. However, this work does not address the fact that state observation in embedded systems is *de facto* in discrete time and that a continuous-time shielding mechanism, therefore would require adequate interpolation between sampling points, which could be an interesting future endeavour.

Chapter 5

Probabilistic Shielding for Autonomous Valet Parking

Qui no s'arrisca no pisca. [1] — Catalan popular saying.

5.1 Motivation and Outline

In this chapter, we present work done in the framework of the FOCETA [Ben+23] project. One of the two use cases of the project consists of building an autonomous driving car capable of operating safely and effectively in a parking lot. To achieve this goal, various partners incorporate different components for perception, planning, and movement execution. The work presented in this chapter is the theoretical conception and experimental evaluation of a safety element in the form of a probabilistic shield designed to prevent car collisions with pedestrians in the parking lot.

We have already demonstrated the usability of deterministic shields in autonomous driving use cases in Chapter 4. In this chapter, we present an alternative approach using probabilistic shielding. By assuming a probabilistic model of the environment, the shield can consider low probability events as possible but only react to them when the probability of a harmful event goes over a particular threshold value.

A *probabilistic shield* (see Section 3.4.3) is an enforcer that overwrites control commands when the probability of violating a safety specification is larger than some state-dependent thresholds. While probabilistic shielding does not enforce safety with full reliability, a probabilistic safety guarantee makes the use of costly measures like emergency braking less intrusive during execution. In our use case, the agent being shielded is an RL-based controller trained to follow a precomputed trajectory along the parking lot. Probabilistic shielding is especially well-suited for RL-based controllers, since, in both cases, the underlying model of the system is a Markov Decision Process (MDP).

[1] The one who does not risk, does not gain.

When given a control command a, the shield maps the information available (from sensors, previous actions, etc.) to a state in the MDP and checks that the maximum probability of avoiding a collision with a pedestrian after executing a in the MPD is large enough. In case it is not, the shield overwrites the control command appropriately. This probability is computed using probabilistic model checking techniques [Kat16]. To do so, we need an explicit description of the MDP.

Constructing an appropriate MDP model is a challenging task. The MDP must faithfully represent the agent and its environment while remaining compact enough to enable feasible model-checking computations. However, our model only needs to capture the safety-relevant dynamics of the system. Our approach factors the model into two components: the ego car, controlled by our agent, and the pedestrians. For the ego car, we build an abstraction based on the digital twin model of the Simrod vehicle [Deb19]. To derive a tractable MDP from the digital twin [Jon+20; Sin+21], we discretise actions and states, incorporating uncertainty into transitions to account for discretisation errors. Pedestrian behaviour is modelled with movement speeds following a normal distribution, varying the parameters for three different types of pedestrians: adults, elders, and children.

Finally, we evaluate our shielding strategy in several scenarios of the car interacting with moving pedestrians. These scenarios are implemented using the proprietary driving simulator `Prescan`[2]. The goal of our experiments is to show that shielding provides a more gentle and efficient safety layer than the coarser approach of an automatic emergency brake (AEB) based only on the expected time to collision.

Contribution. The work presented in this chapter constitutes the first instance of implementing a probabilistic shielding approach in a realistic driving simulation environment. To make it possible, we had to:

- Design a suitable MDP structure for the car and the pedestrian, and populate it by mimicking a digital twin model of the car and suitable behavioural models of the pedestrians.
- Implement our shielding pipeline and integrate it into an existing agent controlling an autonomous vehicle in a simulated environment.
- Validate experimentally the fitness of the model as well as the effectiveness of shielding as a safety measure.

Outline. In Section 5.2 we explain how we build the models required for shielding, as well as the integration of shielding into the existing controller-simulation framework. In Section 5.3, we show the results of our experimental evaluation. Finally, in Section 5.4, we discuss limitations and related work.

Declaration of sources. This chapter is based on work performed by the author of this thesis in the framework of the FOCETA project [Ben+23], and it reuses material from currently unpublished deliverables of said project.

[2] `https://plm.sw.siemens.com/en-US/simcenter/autonomous-vehicle-solutions/prescan`

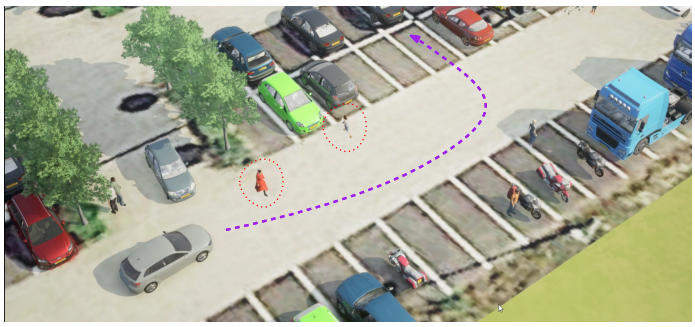


Figure 5.1: Screenshot of the `Prescan` simulation. The path of the ego vehicle is marked with a dashed purple line. The two pedestrians that are currently being shielded, the ones that are close enough, are marked with red dotted circles.

5.2 Methodology

In this section, we discuss the methodology used to adapt the theoretical concept of shielding to our realistic use case. We discuss the shielding setting, the constructions of MDP models for the ego car and the pedestrians, the computation of shields for the models and the integration of the shielding module on the whole autonomous driving controller.

5.2.1 Modeling Scenarios as Markov Decision Processes

In Figure 5.1, we illustrate the use case with a car being controlled by our shielded agent and several pedestrians that move around the parking lot. The global car controller is broadly composed of a path-planning module and an RL agent that controls the pedals and steering wheel. Given an initial position and orientation of the ego car, a target position and orientation, and a map with the static elements of the parking lot, the path-planning module computes the path to follow, and the RL agent controls the pedal and steering commands to best follow said path. While the goal of the RL agent is to follow the predefined path, the goal of the shield is to ensure that the car does not collide with any of the pedestrians while following the path.

Instead of having a unique shield that models interactions with all other pedestrians, we develop shields that model the interaction between the ego car and a single pedestrian. For each pedestrian detected in the vicinity of the ego car, we instantiate a shield that ensures collision avoidance against that one pedestrian. All instantiated shields work then cooperatively: each shield computes the set of actions that are safe according to its own safety specification. Finally, the action proposed by the agent is checked against the intersection of all safe actions and overwritten if needed.

This approach lets us use models that are less complex, with fewer states and transitions, by not modelling pedestrian-to-pedestrian interactions. The reduced complexity of the models permits using accurate models with low computational cost, as well as producing models that are easier to develop, test, and understand.

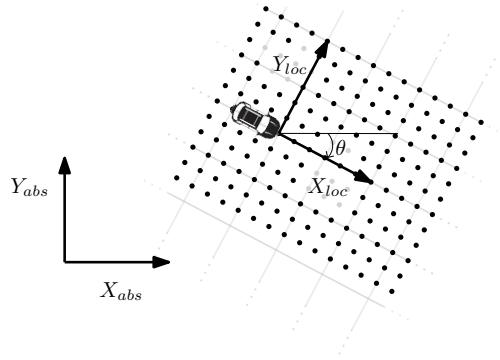


Figure 5.2: Representation of absolute and local coordinate systems. The dotted grid represents the local-discrete system.

The drawback is that we have no theoretical guarantee that the intersection of all safe actions is non-empty. It is theoretically possible to have a scenario with two pedestrians in which the only way to avoid colliding with the first pedestrian is accelerating, while the only way to avoid colliding with the second pedestrian is braking. In such cases, the behaviour of the shield is undefined. While this is theoretically possible, we have not encountered such cases in our experiments, since braking at maximum strength is typically a safe action at relatively low speeds, regardless of the positions of the pedestrians.

5.2.2 MDP Structure and State Discretisation

We need to model the relevant dynamics of the ego car and the behaviour of pedestrians with an MDP with finite sets of states and actions. The states are built from sensor readings, with relevant magnitudes such as positions and velocities being continuous. We need a discrete representation of those readings. Similarly, the set actions proposed by the agent are continuous, so we will need to find a suitable discretisation for the action space.

As stated before, we instantiate an individual shield for each pedestrian. Under the assumption that the behaviour of the pedestrian and the ego vehicle are independent, we can build the model for a shield as a product MDP $\mathcal{M} = \mathcal{M}_{\text{car}} \times \mathcal{M}_{\text{ped}}$, where $\mathcal{M}_{\text{car}} = (\mathcal{S}_{\text{car}}, \mathcal{A}, \mathcal{P}_{\text{car}})$ is an MDP that encodes the dynamics of the ego vehicle and $\mathcal{M}_{\text{ped}} = (\mathcal{S}_{\text{ped}}, \mathcal{P}_{\text{ped}})$ is a Markov chain that encodes the behaviour of the pedestrian. For both $\mathcal{M}_{\text{car}}$ and $\mathcal{M}_{\text{ped}}$, the transition-probability function models transitions of a fixed timestep Δt.

In the following sections, we describe $\mathcal{M}_{\text{car}}$ that models the ego vehicle's dynamics, and $\mathcal{M}_{\text{ped}}$ that models the behaviour of a pedestrian. But before that, we need to introduce the three coordinate systems we will be using for building the shields.

5.2.2.1 Coordinate Systems.

Both sets of states $\mathcal{S}_{\text{car}}$ and $\mathcal{S}_{\text{ped}}$ are defined in terms of positions and velocities of the ego vehicle and the pedestrian, respectively. To define positions and

velocities, we work with three coordinate systems, depicted in Figure 5.2. We call them *absolute*, *local-continuous* and *local-discrete*.

- **Absolute**. The absolute coordinate system has its origin at an arbitrary reference point O, and two orthogonal axes X and Y. This coordinate system does not move during an execution, and the input from sensors is assumed to be given in absolute coordinates. Coordinates can be any real numbers, and values are in units of metre (m) and metre per second ($\mathrm{m\,s^{-1}}$).

- **Local-continuous**. The local-continuous coordinate system is a reference frame that moves with the ego vehicle and is oriented in such a way that the X axis is always parallel to the ego vehicle's velocity. In local-continuous coordinates, the position of the ego vehicle is always $(0, 0)$, and the velocity is $\left(\sqrt{v_x^2 + v_y^2}, 0\right)$, where $v = (v_x, v_y)$ is the velocity of the ego car in absolute coordinates. Coordinates can be any real numbers, and values are in units of m and $\mathrm{m\,s^{-1}}$. The difference between the absolute and local-continuous systems is a translation and a rotation. Therefore, magnitudes stay constant, and the formula to change from local to absolute coordinates is
$$\begin{bmatrix} x \\ y \end{bmatrix}_{abs} = \begin{bmatrix} \mathrm{car}_x \\ \mathrm{car}_y \end{bmatrix}_{abs} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix}_{loc},$$
where θ is the angle between the absolute X-axis and the velocity of the ego car, as depicted in Figure 5.2. The rotation matrix can be inverted to change from absolute to local coordinates.

- **Local-discrete**. The local-discrete coordinate system is a discretization of the local-continuous system. This is the coordinate system used internally in the shield. We define two multipliers $\mu_{\mathrm{pos}}, \mu_{\mathrm{vel}} \in \mathrm{R}$ for the magnitudes of distance and velocity, respectively. Magnitudes in local-discrete coordinates are therefore given in units of $\frac{1}{\mu_{\mathrm{pos}}}$ m for positions and $\frac{1}{\mu_{\mathrm{vel}}}$ $\mathrm{m\,s^{-1}}$ for velocities. Given a distance magnitude x in local-continuous coordinates, it is transformed to local-discrete as $X = \lfloor \mu \cdot x \rceil$, where μ is the corresponding multiplier for the type of magnitude x and $\lfloor z \rceil$ is the result of rounding z to its closest integer.

Since both local-continuous and local-discrete coordinate systems share the same X-axis and Y-axis, we will call them in the following the local X-axis and local Y-axis, without specifying discrete or continuous.

Finding adequate multipliers μ_{pos} and μ_{vel} is a matter of finding a suitable compromise. Larger values provide coarser discretisations, so the models are smaller and easier to work with. However, coarse discretisation incurs larger modelling errors. After some trial and error, we found $\mu_{\mathrm{pos}} = \mu_{\mathrm{vel}} = 1/2$ to be a good value for our experiments.

The absolute reference is the reference frame used by the simulator. Global variables monitored in the experiments, such as vehicle and pedestrian positions and velocities, are logged in this reference system. The local continuous is the system used internally by the car sensors. The information that arrives to the shield, such as the relative positions of pedestrians to the ego car is obtained

in this reference frame. Finally, the local-discrete is the internal system that the shield uses to represent states in the MDP and to decide on the safety of actions depending on the results given by the model checker. During execution, the conversion from local-continuous to local-discrete and back is necessary for every action taken, as the information needs to arrive to the shield and be given back to the car controller. In contrast, conversion from absolute to local-continuous is only required to interpret the results of the experiments.

5.2.3 Model of the Car

The model of the car is based on the digital twin model of the Simrod vehicle [Deb19]. The digital twin model was provided to us as a functional mock-up unit (FMU) compatible with both C++ and Simulink. We are only allowed to interact with this unit as a black box: we can initialise certain variables such as positions and velocities, set an action profile, and let the digital twin run for a given span of time Δt. At the end, we read the new values of the variables of interest. The digital twin only models the dynamics of the car on an empty road, so any interactions of the ego car with other road users have to be modelled on top of it.

5.2.3.1 State Space

We consider the position and velocity along the X-axis of the local-discrete coordinate system. Since we only shield for throttle and brake, we can assume that the direction of the ego vehicle stays locally unchanged.

Moreover, the MDP model used by the shield is constrained to positions no greater than a predefined limit, $x_{\max}$, and velocities not exceeding a threshold, $v_{\max}$. The value of $v_{\max}$ is set slightly above the maximum speed permitted on the road, ensuring realistic constraints. In contrast, $x_{\max}$ is a more flexible parameter, chosen to be sufficiently large to encompass all possible consequences of the ego vehicle's decisions made at the origin of the local-continuous coordinate system.

Formally, we consider $\mathcal{S}_{\text{car}} = X_{\text{car}} \times V_{\text{car}}$, where $X_{\text{car}} = [0, \dots, x_{\max}]$ represents the position of the ego car along the X-axis and $V_{\text{car}} = [0, \dots, v_{\max}]$ represents the velocity of the ego car. The values in X_{car} are integers and represent positions in units of $\frac{1}{\mu_{\text{pos}}}$ m, while values in V_{car} are integers representing velocities in units of $\frac{1}{\mu_{\text{vel}}}\ \mathrm{m\,s^{-1}}$.

5.2.3.2 Action Space

Following the convention found in the Simrod model, we consider actions in the range $[-1, 1]$. For an action $a \in [-1, 1]$, the sign indicates which pedal to press (brake for $a < 0$, throttle for $a > 0$), and $|a|$ indicates how much the pedal is pressed. Since the MDP needs to work with a discrete set of actions, we define a set of representative actions $\mathcal{A} = \{\alpha_1, \dots, \alpha_n\}$, satisfying $-1 \leq \alpha_1 < \cdots < \alpha_n \leq 1$.

For both space and action spaces, the input received by sensors and controller is rounded to the closest value in the MDP discretisation.

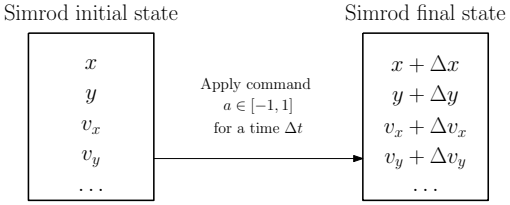


Figure 5.3: Overview of an experiment on the Simrod model.

For the sake of simplicity, the shield can only overwrite the throttle and brake commands and leaves the steering command untouched.

5.2.3.3 Transition Probabilities

We obtain the transition probabilities for our model by probing the Simrod digital twin model at concrete values, as we will describe in this section. These experiments are best thought of within the local-continuous coordinate system.

Given an initial position (x, y) and initial velocity (v_x, v_y) for the ego car, a driving command $a \in [-1, 1]$ and a timestep Δt, the Simrod digital twin provides us with a new position $(x+\Delta x, y+\Delta y)$ and velocity $(v_x+\Delta v_x, v_y+\Delta v_y)$, as the result of applying the driving command c for a duration of Δt. Figure 5.3 illustrates such an experiment. In an ideal scenario, we would execute the experiment in Figure 5.3 for each combination of initial positions, velocities and driving commands. However, this is unfeasible because of the complexity of the digital twin model, and the need to account for a discretised MDP model. Therefore, we build the transition probabilities by adjusting them to experimental data obtained by the Simrod model. We design the following experiment:

- We define m reference velocities, $0 < v_1 < \cdots < v_m < v_{\max}$. For each reference velocity v_j, the interval $rv_j = [(v_{j-1}+v_j)/2, (v_j+v_{j+1})/2)$ is the range of velocities that have v_j as its reference velocity, i.e. for all $v \in rv_j$, v_j is the closest velocity to v among the set of reference velocities. For the first and last ones, we include the minimum and maximum velocities, respectively, i.e. $rv_0 = [0, (v_1 + v_2)/2)$ and $rv_m = [(v_{m-1} + v_m)/2, v_{\max}]$.

- Similarly, for each value in the action space α_i, we define the action range $r\alpha_i = [(\alpha_{i-1} + \alpha_i)/2, (\alpha_i + \alpha_{i+1})/2)$, with the special cases of $r\alpha_0 = [0, (\alpha_1 + \alpha_2)/2)$ and $r\alpha_n = [(\alpha_{n-1} + \alpha_n)/2, \alpha_{\max}]$. In Figure 5.4, we provide a visual representation of these ranges for actions. The ranges for velocities are analogous.

- We choose a value N of the number of samples used for each reference action and velocity. For each action range $r\alpha_i$ and velocity range rv_j, we sample N pairs (a_i^k, u_j^k), for $k \in \{1, \ldots, N\}$ uniformly at random from $r\alpha_i \times rv_j$.

- For each pair (a_i^k, u_j^k), we perform the experiment described in Figure 5.3, setting the initial velocity in the X-direction to $v_x = u_j^k$, the remaining initial parameters to zero (i.e. $x = y = v_y = 0$) and the driving command

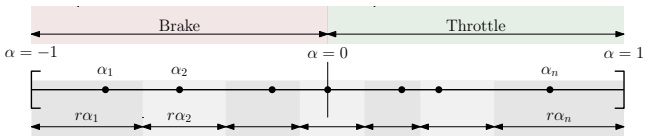


Figure 5.4: Scheme of reference actions and action ranges.

to a_i^k. We label the position and velocity increases as $\Delta x = \Delta x_{i,j}^k$ and $\Delta v_x = \Delta v_{i,j}^k$.

To build the transition probabilities of $\mathcal{P}_{\text{car}}$ from the available data, we adopt the following two assumptions:

- *Assumption 1.* The increase in velocity, $\Delta v_{i,j}^k$, primarily depends on the reference action (α_i) and the velocity at which it is applied ($u_{i,j}^k$), and does not depend significantly on other variables such as position, previous accelerations, or the concrete applied action (a_i^k).
- *Assumption 2.* The increase in position, $\Delta x_{i,j}^k$ is proportional to the initial velocity ($u_{i,j}^k$), and the proportionality factor primarily depends on the applied action (a_i^k).

Assumption 1 implies that transition probabilities should be derived as statistical measures based on the experimental data described earlier, with a single probability distribution for each reference action α_i. Assumption 2 suggests that for each reference velocity v_j and each reference action α_i, there exists a value $\gamma_{i,j}$ such that $\Delta x_{i,j}^k \approx \gamma_{i,j}^k u_{i,j}^k$ provides a good approximation. To determine $\gamma_{i,j}$, we minimise the error using the following optimisation criterion. For each $i \in [1, \ldots, n]$ and $j \in [1, \ldots, m]$:

$$\gamma_{i,j} = \arg\min_{\gamma \in \mathbb{R}} \sum_{k=1}^{N} \left(\Delta x_{i,j} - \gamma u_{i,j}^k\right)^2. \tag{5.1}$$

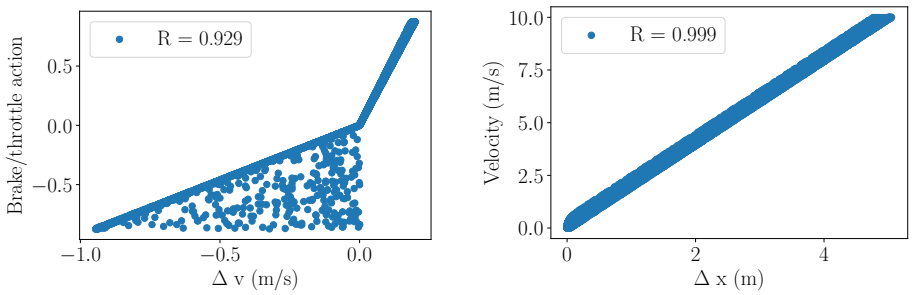
With all this data, we compute the transition probabilities as follows. For each $s = (X, V) \in \mathcal{S}_{\text{car}}$ and action $\alpha_i \in \mathcal{A}$, let v_j be the corresponding reference velocity to V, i.e., the only j for which $V/\mu_{\text{vel}} \in rv_j$. Then for each $s' = (X + \Delta X, V + \Delta V)$ such that $\Delta X = \lfloor \gamma_{i,j} V \rceil$:

$$\mathcal{P}_{\text{car}}(s, \alpha_i, s') = \frac{1}{N} \#\left\{k \in [1, \ldots, m] \,:\, \lfloor \Delta v_{i,j}^k \cdot \mu_{\text{vel}} \rceil = \Delta V\right\}. \tag{5.2}$$

For any s' such that $\Delta X \neq \lfloor \gamma_{i,j} V \rceil$, the transition probability is $\mathcal{P}_{\text{car}}(s, \alpha_i, s') = 0$. In Equation (5.2), the transition probability for a given velocity increase ΔV is defined as the relative frequency of the increase in velocity that gets discretized into ΔV.

5.2.4 Model of the Pedestrian

The behaviour of the pedestrian is modelled by a Markov chain $\mathcal{M}_{\text{ped}} = (\mathcal{S}_{\text{ped}}, \mathcal{P}_{\text{ped}})$, in which the states $\mathcal{S}_{\text{ped}} = X_{\text{ped}} \times Y_{\text{ped}}$ represent the position of the pedestrian



(a) Scatter plot of action (α) vs. Δv. (b) Scatter plot of velocity (v_x) vs. Δx.

Figure 5.5: Scatter plots to validate $\mathcal{M}_{\text{car}}$. The high correlation factor (R) in both cases validates the assumptions on probabilities for the ego car model. In particular, the high correlation between the chosen action and the increase in velocity (a) validates Assumption 1, while the high correlation between velocity and Δx (b) validates Assumption 2.

relative to the same coordinate origin as the car in the local-discrete system. The probability transition function $\mathcal{P}_{\text{ped}}$ determines how the pedestrian behaves in a stochastic manner.

At each timestep, the pedestrian moves in the X-axis and the Y-axis following two independent Gaussian distributions. The distributions are centred at 0 m/s so that the pedestrian is equally likely to move in any direction. The standard deviation σ_{ped} indicates how erratic the movement of the pedestrian tends to be. We set $\sigma_{\text{ped}} = 2$ m/s for adults, $\sigma_{\text{ped}} = 1$ m/s for elders and $\sigma_{\text{ped}} = 3$ m/s for children. With this model, the average speed of a pedestrian is $\sqrt{2}\sigma_{\text{ped}}$.

A limitation of this model is that pedestrians, as we model them, are non-inertial, i.e., their velocity at one step does not influence their velocity at the next step. While inertial pedestrians would certainly be more realistic, the issue is mitigated by the fact that the pedestrian's velocities are small.

5.2.5 Shield Computation

A safety specification is given in the form of a set of states $S_{\text{crash}} \subseteq \mathcal{S}_{\text{car}} \times \mathcal{S}_{\text{ped}}$ representing collisions, a safety threshold $\lambda \in (0, 1)$, and a bounded horizon $k \in \mathbb{N}$. Given the models $\mathcal{M}_{\text{car}}$ and $\mathcal{M}_{\text{ped}}$ and a safety specification, we use Tempest [Pra+21a] to compute a shield that enforces the safety specification, as described in Equation (3.9). For each tuple $(s_{\text{car}}, s_{\text{ped}}, a) \in \mathcal{S}_{\text{car}} \times \mathcal{S}_{\text{ped}} \times \mathcal{A}$, Tempest determines whether executing action a at $(s_{\text{car}}, s_{\text{ped}})$ is safe according to the specification. An action a is considered safe from state s if

$$\mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s, a, S_{\text{crash}})\right) \geq \lambda \cdot \mathbb{P}^{\mathcal{M}}_{\max}\left(\texttt{Avoid}_{\leq k}(s, S_{\text{crash}})\right).$$

Tempest produces a lookup table that specifies, for each state, a safe alternative action to replace any unsafe action. This lookup table constitutes the shield.

5.3 Experimental Evaluation

Our experimental evaluation pursues two primary objectives. First, we empirically validate the assumptions underlying the construction of the car MDP derived from the digital twin FMU model. Next, we assess whether the integrated shields can effectively prevent collisions with pedestrians and determine if their approach outperforms the automated emergency braking system.

5.3.1 Validation of the Car Model

In Section 5.2.3 we describe a method to obtain experimental data from the Simrod model in a structured way, that we then use to build our MDP model of the car. The transition probabilities are built from the data, taking two assumptions on the dependency of the increases in velocity and position (Δv and Δx) on the actions applied and the current velocity.

We can validate experimentally the assumptions made to build this model by checking the correlations in the data obtained from the experiment described. In Figure 5.5 we show that both key correlations between Δv and α (Assumption 1), and between Δx and v_x (Assumption 2) are very high. The data comes from performing the experiment for reference velocities $v = [0, 1, 2, \ldots, 10]$ m/s, action set $\mathcal{A} = [-0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75]$, and a number of samples $N = 100$. The high correlation factor (R) in both cases validates the assumptions on probabilities for the ego car model.

In Figure 5.5a there are some data points that look a bit odd for brake actions ($\alpha < 0$), where the decrease in velocity $-\Delta v$ is smaller than expected. These data points correspond to individual experiments where the initial velocity is already very small, so that the brake action applied is more than enough to fully stop the car, even with a deceleration smaller than typical for such action. We also observe a steeper curve in the throttle range than in the brake range. This indicates that, for this car, the deceleration produced by the brake pedal is more potent than the acceleration produced by the throttle pedal at the same level. This is a standard safety feature in automobiles.

5.3.2 Safety Shielding vs. Automatic Emergency Brake

To evaluate the performance of the shield, we integrate it as part of the agent controlling the ego vehicle developed by several partners in the FOCETA project.

In evaluating the performance of the safety shield, we focus on how effective the shield is in enforcing the safety specification. We also assess whether the shield is an improvement with respect to the AEB in terms of efficiency, perceived safety by other road users, and comfort of the passengers. Figure 5.6 illustrates a case where the shield mitigates collision risks more efficiently and earlier than the standard emergency brake, resulting in lower usage of the brake pedal and more gentle deceleration. To evaluate the performance of the safety shield in a systematic way, we created a scene in the AVP scenario with fixed initial and goal positions and added several pedestrians. We produced 20 configurations of the pedestrians' initial positions and moving patterns by random sampling the pedestrians' initial positions and velocities. We executed the scene for a

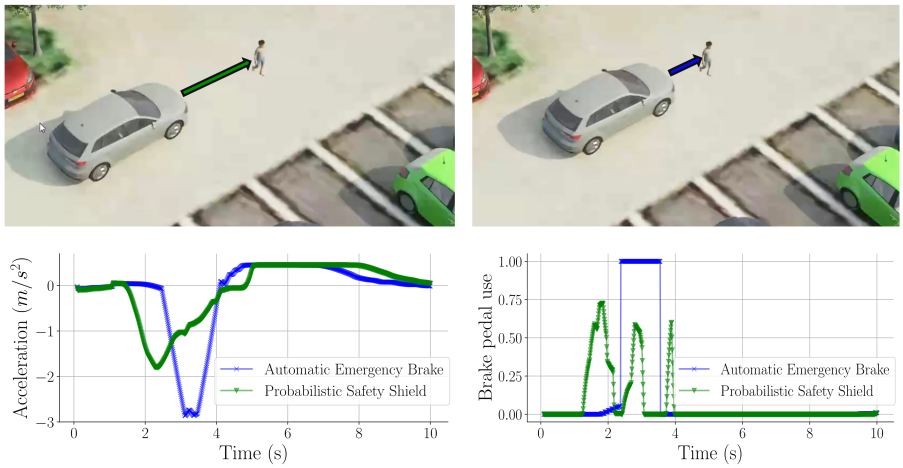


Figure 5.6: Example of the advantage of shielding with respect to an automated emergency brake, avoiding the collision in a smoother and more efficient way.

fixed timespan of 20 seconds, having (a) only the shield as a safety enforcer, (b) only the emergency brake as a safety enforcer and (c) both the shield and the emergency brake together. In the latter case, whenever both systems propose an enforcement action, the emergency brake takes priority over the shield. We tested the following metrics.

- **Effectiveness in avoiding collisions.**
  - *Distance to pedestrian.* The average distance from the front of the car to the pedestrian. Larger values indicate increased safety.
  - *AEB activation.* The percentage of time in the 20-second experiment where the automated emergency brake is active.
- **Efficiency in driving.**
  - *Brake pedal.* The average value of the brake pedal signal. Recall that both pedals have a signal from 0 – not activated – to 1 – fully activated.
  - *Throttle pedal.* Use of the throttle pedal, analogous to the brake pedal. A higher value for either pedal metric suggests a driving style that may accelerate the wear and tear on the vehicle.
  - *Distance to goal.* Average distance to the goal position during the experiment. Lower values indicate increased efficiency in reaching the goal.
- **Comfort.**
  - *Acceleration.* Average value of the acceleration of the ego car. The average is taken in absolute value. Lower values indicate increased efficiency and comfort.

Test	Only AEB	Both	Only shield
Distance to pedestrian (m)	$\mathbf{4.20} \pm 1.00$	3.95 ± 0.86	4.07 ± 0.96
AEB activation (%)	19 ± 11	$\mathbf{8} \pm 8$	—
Brake pedal (avg. use, 0 to 1)	0.39 ± 0.22	0.49 ± 0.26	$\mathbf{0.29} \pm 0.27$
Throttle pedal (avg. use, 0 to 1)	0.51 ± 0.21	0.46 ± 0.24	$\mathbf{0.40} \pm 0.18$
Distance to goal (m)	13.2 ± 4.9	14.2 ± 4.4	$\mathbf{12.8} \pm 5.2$
Acceleration (m/s^2)	0.49 ± 0.21	0.44 ± 0.18	$\mathbf{0.36} \pm 0.20$
Jerk (m/s^3)	1.00 ± 0.80	0.93 ± 0.86	$\mathbf{0.65} \pm 0.65$
Time to collision (s)	4.2 ± 1.8	$\mathbf{4.4} \pm 1.6$	4.0 ± 2.0

Table 5.1: Quantitative analysis of probabilistic shielding. Marked in boldface the best result for each metric on average.

- *Jerk.* Average value of the jerk felt by the ego car, that is, the variation in acceleration. This is a standard measure of comfort.
- *Time to collision.* The hypothetical time that it would take to collide with the closest pedestrian if the car would maintain its current speed and trajectory. It is a measure of perceived safety by other road users.

In Table 5.1, we present the results we obtained. For each metric being measured, we provide mean and standard deviation across all our experiments. We do not include the number of collisions as the scenarios are designed in such a way that there would be a collision, but the safety mechanism (be it the shield, the AEB, or both) has to act to prevent it.

In terms of effectiveness, we can see that the three methods show a similar performance in terms of maintaining a safe distance with respect to the closest pedestrian, and we see that when the shield is active, the use of the emergency brake is down by half. In terms of driving efficiency, we see that the shield tends to produce less use of both the brake and throttle pedals while maintaining a low distance to the goal. In our results, however, we do observe that having both enforcing systems together produces higher use of the brake pedal and higher distance to the goal, suggesting that the resulting controller may be overly conservative. In terms of comfort, we observe mainly a notable difference in jerk, where shielding significantly reduces the discomfort to the passengers due to high jerk, associated with a harsh use of the brake and throttle pedals. The data from acceleration supports this claim as well, albeit in a lower magnitude. Time to collision proves to be very similar across the board, with the shielded controller showing a slight reduction.

5.4 Discussion

5.4.1 Limitations

Probabilistic safety guarantees. The approach towards safety using probabilistic shielding mitigates two of the main concerns discussed in the previous

chapter with regard to deterministic shields, namely the requirement of a deterministic model of what are sometimes inherently stochastic phenomena, and the worst-case scenario guarantees generating overly conservative controllers. This step up is made available at the price of relaxing the safety guarantees. This relaxed specification is also somewhat unintuitive, as we have seen in Example 3.2, which can work towards eroding the trust of the user in the runtime enforcement method.

Model size and control variables. Discretising both the observation and action spaces inevitably introduces errors in the model. However, this trade-off is necessary to keep the model size manageable, enabling the use of probabilistic model-checking methods with reasonable resource consumption. This affects, in our case, both the car and the pedestrian models. For the pedestrian, a richer model of their behaviour, distinguishing behaviour modes depending on their context or having a more fine-grained account of their velocities would make the model significantly more useful. For the car, allowing a larger model would allow us to introduce steering as a control variable to be shielded. While steering is not required to enforce safety in our use case, some pedestrians can be more efficiently avoided by a gentle steer than the use of the brake pedal.

5.4.2 Related Work

Probabilistic shielding in MDPs was first introduced in [Jan+20], and has been extended to partially observable MDPs [Car+23]. To the best of our knowledge, probabilistic shielding on MDPs has not been previously used for realistic self-driving use cases.

Probabilistic model checking tools. Several tools implement probabilistic model checking to verify the safety of RL agents. COOL-MC [Gro+22b] takes a Gymnasium [Tow+24] environment and an MDP model as inputs, querying the agent's decisions across all MDP states and using the resulting Markov chain to verify a user-defined property. MoGym [Gro+22a] converts a user-provided MDP into a Gymnasium-compatible environment, enabling RL policy training and statistical model checking through policy queries. Unlike these verification-based approaches, shielding does not verify the agent's policy but ensures its correct execution at runtime. The most widely used tools for probabilistic model checking include Storm [Hen+22], Prism [KNP11], Modest [HH14], and PET [MW24]. Our work utilises Tempest [Pra+21a], a fork of Storm specifically designed to synthesise shields for Markov decision processes and stochastic multiplayer games.

Shielding methods for autonomous driving. RL has been one of the main methods to develop autonomous driving agents [Kir+21; Pan+17].

There is further work on probabilistic safe RL methods that fit in the shielding framework, even though they do not use the same formalism of model checking on MDPs. Most of this work focuses on collision avoidance and safe driving [Bou+19; KWA20; Lin+24; He+23; Sax+19]. Shielding methods for RL have been used to optimise the navigation path of a self-driving car [HWL24; VDL24] or a platoon of self-driving cars [BLS24]. Moreover, they have also

been used for vehicle trajectory tracking control tasks [XZL22], as well as to optimise self-driving car fuel consumption during traffic congestion [Che+19]. Most of the work that integrates any type of shielding for self-driving car applications has been tested in simulation [Che+19; BLS24; HWL24; XZL22; VDL24]. There is also recent work implementing autonomous driving capabilities in car scale prototypes such as the F1Tenth [O'k+20] to validate the proposed solutions [Koc+23].

Chapter 6

Fairness Shields: Enforcing Fairness Properties for Bounded and Periodic Horizons

Jeder nach seinen Fähigkeiten, jedem nach seinen Bedürfnissen.[1]
— Popular socialist slogan. [2]

6.1 Motivation and Outline

With the rise of machine learning (ML) in human-centric decisions, such as banking and college admissions, concerns about bias based on protected attributes like gender and race have grown [DF18; Obe+19; SPB19; Liu+18; Ber+21]. Mitigating such biases is a crucial and active research area in AI.

Most bias prevention methods rely on *design-time* interventions, such as preprocessing training data [KC12; CŽ13], modifying loss functions [Aga+18; Ber+17] — known as *in-processing methods* —, or post-processing decisions with calibrated output functions [HPS16; CH20]. We introduce *fairness shielding*, the first *run-time* intervention method to safeguard fairness in deployed decision-makers.

Fairness shields address fairness in *sequential* decision-making, where observations come one after the other, and decisions have to be made without knowledge of the following observations. While fairness has traditionally been studied in a history-independent way, the sequential setting better models real-world decisions [ZL21]. Prior work mostly focuses on fairness over the long run in

[1] From each according to their ability, to each according to their needs.

[2] While it was Karl Marx who most popularized this saying, it was a common slogan within the socialist movement of the XIX century, and its origin is still a disputed fact.

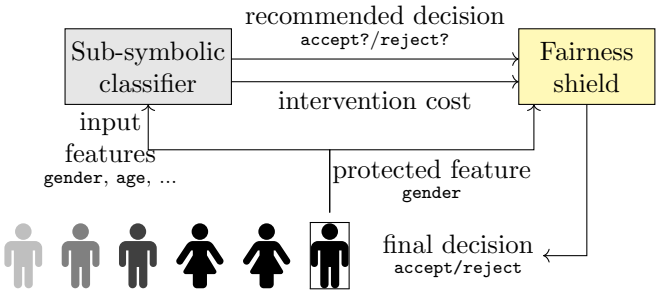


Figure 6.1: The operational diagram of fairness shields.

unbounded horizons [HZ22], but *finite-horizon* and *periodic* fairness — evaluating fairness over fixed timeframes — better align with real-world regulatory assessments [Ala+24]. Our fairness shields enforce these fairness guarantees by monitoring decisions and intervening only when necessary.

Figure 6.1 illustrates fairness shielding. Given a predefined fairness criterion and time horizon, the shield observes the protected attribute, classifier (agent) recommendation, and the cost of altering that recommendation. The final decision ensures fairness while minimizing intervention costs, with costs either specified by the decision-maker or assumed constant.

Example 6.1 (Running example - Bilingual team)**.** *Consider the process of assembling a customer service team for a company in a bilingual country, where language A and B hold both official status. Because of the nature of the task, it is essential to maintain a balanced representation of native speakers of both languages. To achieve this, the company enforces a policy requiring that the difference between the number of employees proficient in each language must not exceed* 20% *of the total team size. The hiring process operates within a bounded time horizon, with a fixed number of T candidates to be screened. Candidates apply sequentially, and decisions about each applicant must be made before considering future candidates. Suppose the company uses an ML model to screen candidates, which is designed without considering linguistic balance, as it is irrelevant in other regions where the company operates, and is biased towards language A candidates. Relying solely on this ML model's recommendations could lead to an unbalanced team composition. The recruitment team could follow the ML models' recommendation until achieving a balanced team becomes impossible, and then hire some language B candidates. This would create a situation in which many qualified language A candidates might have to be rejected. Furthermore, it could also prolong the process of finding suitable language B candidates. Even if the ML model aims for long-term workforce balance, an influx of strong candidates from one language group could still skew the team's composition.*

A fairness shield can be deployed, which will monitor and intervene in the decisions at runtime to guarantee that the final team is linguistically balanced as required while keeping the deviations from the decision-maker's at a minimum.

Computation of fairness shields. Fairness shields are computed by solving bounded-horizon optimal control problems, which incorporate a *hard fairness constraint* and a *soft cost constraint* designed to discourage interventions. For the hard fairness constraint, we consider the empirical variants of standard group fairness properties, like demographic parity and equal opportunity. We require that the *empirical bias remains below a given threshold* with the bias being measured either at the end of the horizon or periodically. This hard constraint corresponds to the shield being "correct" with respect to a given specification (Definition 3.9).

For the soft cost constraint, we assume that the shield receives a separate cost penalty for each decision modification. The shield is then required to minimize the total expected future cost, either over the entire horizon or within each period. The definition of cost may vary by application. In general, the cost should be associated to the confidence in the classification, with high-confidence recommendations requiring high costs.

For shield computation, we assume that the distribution over future decisions (of the agent) and costs are known, either from the knowledge of the model or *learned* from queries. Fairness shields are computed through dynamic programming. While the straightforward approach would require exponential time and memory, we present an efficient abstraction for the dynamic programming algorithm that reduces the complexity.

Types of fairness shields. We propose four types of shields: (i) FinHzn, (ii) Static-Fair, (iii) Static-BW, and (iv) Dynamic shields. FinHzn is specific to the bounded-horizon problem, ensuring fairness in every run while being cost-effective. The other three are suited for the periodic setting, guaranteeing fairness under diverse assumptions on how often individuals from each group will appear in a period. Static-Fair and Static-BW reuse a statically computed FinHzn shield for each period, while Dynamic shields require online re-computation of shields at the start of each period.

Experiments. We empirically demonstrate the effectiveness of fairness shielding on various ML classifiers trained on well-known datasets. While unshielded classifiers often show biases, their shielded counterparts are fair in *every* run in the bounded-horizon setting and in most runs in the periodic setting. In most cases, the shielded classifiers exhibit a slightly lower classification accuracy as their unshielded counterparts. This discrepancy is more pronounced under stricter fairness conditions and less pronounced, if the classifier was already trained to be fair.

Contributions. The contributions presented in this work can be summarized as follows.

- We formalize the concept of fairness shields, the first runtime intervention procedure for safeguarding the fairness of already deployed decision-makers.
- We propose an efficient algorithm for synthesizing fairness shields for finite horizons and explain how it can be extended to fairness shields in a

periodic setting.

- We study the problem of safeguarding for periodic fairness and propose three solutions formalized in three types of shields: static-fair, static with bounded welfare, and dynamic. For each of the proposed solutions, we study under which assumptions they guarantee periodic fairness.
- We evaluate our shields with extensive experiments on several benchmark datasets, shielding ML agents trained with state of the art in-processing fairness learning methods. In our experiments, we show the effectiveness of our shields, validating our theoretical results and evidencing the gap between theoretical and practical guarantees.

Outline. In Section 6.2 we present the formal setting and how it fits within the general reactive decision-making framework presented in Chapter 3. In Section 6.3 we present our main algorithm to synthesize fairness shields for the finite horizon, which is later re-used for the periodic fairness setting. We present a general algorithm and a more efficient version for typical fairness properties. In Section 6.4 we present diverse approaches to extend finite horizon shields to an unbounded horizon in a periodic manner. In the periodic setting we loose the strong fairness guarantees of the finite horizon setting, so we focus most of the section on results studying under which conditions fairness can be guaranteed. We present our experimental evaluation in Section 6.5, and finish the chapter in Section 6.6 discussing edge cases, limitations, and related work.

Declaration of sources. This chapter is partially based and reuses material from the following source previously published by the author of this thesis:

[Can+25a] Filip Cano, Thomas A. Henzinger, Bettina Könighofer, Konstantin Kueffner, and Kaushik Mallik. "Fairness Shields: Safeguarding against Biased Decision Makers". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2025,

[Can+24b] Filip Cano, Thomas A. Henzinger, Bettina Könighofer, Konstantin Kueffner, and Kaushik Mallik. *Fairness Shields: Safeguarding against Biased Decision Makers (extended version)*. 2024. arXiv: 2412.11994.

6.2 Fairness Shielding Setting

In this section, we present the setting and notation elements that will be used throughout this chapter. As illustrated in Figure 6.1, the problem of fair classification in this chapter can also be interpreted in the general reactive decision-making framework. As we have done in previous chapters, we will use a slightly adapted notation, focusing on the relevant elements of the work presented in this chapter. In particular, in this chapter we use for the first time shields that are not minimally correct, but are rather synthesized minimizing a certain cost function. We continue using the bilingual team-building problem as our running example to illustrate the notation elements being introduced. We reserve Section 6.2.3 to connect the formalization of this chapter with the general

framework presented in Chapter 3.

6.2.1 Environment and Shielding Setting

Data-driven classifier. We are given a population of individuals, described by features. Among them, we consider one binary feature to be *protected* or *sensitive*. Typical protected features are race, gender, language, etc. Without loss of generality, the protected feature takes values in the set $\mathcal{G} = \{a, b\}$, and the population can be therefore partitioned into groups a and b, according to the value of the protected feature. We consider a data-driven classifier that at each step samples one individual from the population, and outputs a *recommended decision* from the set $\mathbb{B} = \{1, 0\}$ along with an intervention cost from the finite set $\mathbb{C} \subset \mathbb{R}_{\geq 0}$. As convention, decisions "1" and "0" will correspond to "accept" and "reject," respectively. We assume that the sampling and classification process gives rise to a given *input distribution* $\theta \in \mathcal{D}(\mathcal{X})$, where the set $\mathcal{X} \coloneqq (\mathcal{G} \times \mathbb{B} \times \mathbb{C})$ is called the *input space*. The non-protected features of individuals are hidden from the input space because they are irrelevant for shielding. We will assume that θ is given, i.e., the shields are computed using knowledge about θ. When doing experiments, we estimate an approximation of θ from the available data, as we detail in Section 6.5.1.

Example 6.2 (Continuation of Example 6.1)**.** *In the bilingual team example, an individual is represented by a tuple $(g, z) \in \mathcal{G} \times \mathcal{Z}$, where $\mathcal{G} = \{a, b\}$ denotes the language in which the candidate is proficient, and $\mathcal{Z}$ encompasses all non-protected features relevant to evaluating a candidate's suitability for the job, such as years of experience, relevant education, and so on. For simplicity, we assume that a candidate is proficient in only one of the two languages.*

The company uses a classifier $f \colon \mathcal{G} \times \mathcal{Z} \to \mathbb{B} \times \mathbb{C}$, which outputs a preliminary decision for each candidate (accept or reject) along with a cost associated with altering that decision. The cost reflects the classifier's confidence: candidates who are clearly good or bad incur a high cost for decision changes, while borderline candidates can have their decisions reversed at a lower cost.

Shields. A *shield* is a symbolic decision-maker that selects the *final decision* from the *output space* $\mathcal{Y} \coloneqq \mathbb{B}$ after observing a given input from $\mathcal{X}$, and possibly accounting for past inputs and outputs.

Formally, a shield is a function $\pi \colon (\mathcal{X} \times \mathcal{Y})^* \times \mathcal{X} \to \mathcal{Y}$, and its bounded-horizon variants are functions of the form $(\mathcal{X} \times \mathcal{Y})^{\leq t} \times \mathcal{X} \to \mathcal{Y}$, for a given t. Following the notions introduced in Chapter 3, a fairness shield is a particular case of a *post-shield* (Definition 3.4). In particular, its output is a concrete `accept/reject` decision, and not a set of allowed decisions. Note that the input of a shield is (τ, x), where $\tau \in (\mathcal{X} \times \mathcal{Y})^*$ a sequence of previous inputs and outputs, and $x \in \mathcal{X}$ is a tuple $x = (g, b, c)$ representing the last individual, for which a final decision has not yet been made, where $g \in \mathcal{G}$ is the group membership, b is the `accept/reject` recommendation of the classifier and $c \in \mathbb{C}$ is the cost of overwritting the classifier's recommendation. We will write Π and Π^t to respectively denote the set of all shields and the set of bounded-horizon shields

with horizon t [3]. The *concatenation* of a sequence of shields $\pi_1, \pi_2, \ldots \in \Pi^t$ is a shield π, such that for every trace τ, if τ can be decomposed as $\tau\tau'$ with $|\tau| = jt$ for some j and $\tau' < t$, then $\pi(\tau, x) \coloneqq \pi_{j+1}(\tau', x)$.

Sequential decision making setting. We consider the sequential setting where inputs are sampled from θ one at a time, and the shield π needs to produce an output without seeing the inputs from the future. Formally, at every time $i = 1, 2, \ldots$, we sample an input $x_i = (g_i, r_i, c_i)$ from θ. The probability of getting input x_i is $\theta(x_i) > 0$. The shield's output at time i is $y_i = \pi([(x_1, y_1), \ldots, (x_{i-1}, y_{i-1})], x_i)$. After applying this process of sampling input and getting the corresponding shield output for t time-steps, the resulting finite sequence $\tau = (x_1, y_1), \ldots, (x_t, y_t)$ is called a *trace* induced by θ and π, and the integer t is called the *length* of the trace, denoted as $|\tau|$. We use $\mathtt{FT}^t_{\theta,\pi}$ to denote the set of every such trace. For every t, the probability distribution θ and the shield π induce a probability distribution $\mathbb{P}(\cdot; \theta, \pi)$ over the set $(\mathcal{X} \times \mathcal{Y})^t$ as follows. For every trace $\tau \in (\mathcal{X} \times \mathcal{Y})^t$,

$$\mathbb{P}(\tau; \theta, \pi) \coloneqq \begin{cases} \prod_{i=1}^{t} \theta(x_i) & \text{if } \tau \in \mathtt{FT}^t_{\theta,\pi}. \\ 0 & \text{otherwise.} \end{cases} \tag{6.1}$$

The notation $\mathbb{P}(\tau; \theta, \pi)$ is to be read as "the probability of obtaining the trace τ when sampling inputs from the input distribution θ, and applying shield outputs from π". Note that θ and π are *parameters* of the distribution, i.e., $\mathbb{P}(\cdot; \theta, \pi)$ is a probability distribution, while τ is the element in the sample space (denoted Ω in Section 2.2) that has a certain probability to be sampled. Given a prefix τ, the probability of observing the trace $\tau \cdot \tau'$, for some $\tau' \in (\mathcal{X} \times \mathcal{Y})^*$, is $\mathbb{P}(\tau' \mid \tau; \theta, \pi) = \mathbb{P}(\tau \cdot \tau'; \theta, \pi)/\mathbb{P}(\tau; \theta, \pi)$. Note that the statistical dependence of τ' on τ is due to π's history-dependence.

Cost. Let $\tau = (x_1, y_1), \ldots, (x_t, y_t)$ be a trace of length t, where $x_i = (g_i, r_i, c_i)$. At time i, the shield pays the cost c_i if its output y_i is different from the recommended decision r_i. The *total* (intervention) *cost incurred by the shield* on τ up to a given time $s \leq t$ is

$$\mathit{cost}(\tau; s) \coloneqq \sum_{i=1}^{s} c_i \cdot \mathbb{1}\left[r_i \neq y_i\right]. \tag{6.2}$$

The cost incurred up to time t (the length of τ) is simply written as $\mathit{cost}(\tau)$, instead of $\mathit{cost}(\tau; t)$. For a given time horizon t, we define the expected value of cost after time t as

$$\mathbb{E}[\mathit{cost}; \theta, \pi, t] \coloneqq \sum_{\tau \in (\mathcal{X} \times \mathcal{Y})^t} \mathit{cost}(\tau) \cdot \mathbb{P}(\tau; \theta, \pi), \tag{6.3}$$

and if additionally a prefix τ is given, the conditional expected cost after time t (from the end of τ) is

$$\mathbb{E}[\mathit{cost} \mid \tau; \theta, \pi, t] \coloneqq \sum_{\tau' \in (\mathcal{X} \times \mathcal{Y})^t} \mathit{cost}(\tau') \cdot \mathbb{P}(\tau' \mid \tau; \theta, \pi). \tag{6.4}$$

[3] We define bounded shields and the set Π^t to emphasize that our synthesis algorithm only defines the behaviour for traces τ with a length at most t.

Note that the difference between Equations (6.3) and (6.4) is that in the second one, the prefix τ is given as part of the trace to the shield. One can see Eq. (6.3) as a particular case of Eq. (6.4) when the prefix is the empty trace $\tau = \varepsilon$. If τ is "very fair", i.e., $\varphi(\tau) \ll \kappa$, the cost of enforcing fairness in the next t steps will be generally lower than the case where τ is on the limit of being fair, i.e., $\varphi(\tau) \approx \kappa$. This effect is amplified with longer prefixes.

The shield π is an element external to the classifier. It takes the protected feature of the candidate and the classifier's recommendation as inputs and has the authority to issue a final `accept/reject` decision. If the shield's decision differs from the classifier's, the incurred cost is as specified by the classifier. The shield's inputs are the features of candidates, the classifier's decisions, and the costs, and the input distribution is assumed to be known in advance.

Note that, from the shield's perspective, the distribution of non-protected features is unimportant, as these features are already processed by the data-driven classifier and summarized into a single cost value. By sampling individuals from the candidate pool and processing them through both f and π, we obtain a trace τ that records the individuals and their decisions. In the case of our running example, this trace encapsulates the results of the hiring process, including the linguistic distribution of hired candidates and the total cost incurred by the shield.

6.2.2 Fairness Enforcement with Minimal Cost

Fairness. We model (group) *fairness properties* as functions that map every finite trace to a real-valued *bias* level through intermediate statistics. A *statistic* μ maps each finite trace τ to the values of a finite set of counters, represented as a vector in $\mathbb{N}^p$, where p is the number of counters. The *welfare* for group $g \in \{a, b\}$ is a function $\mathtt{WF}^g \colon \mathbb{N}^p \to \mathbb{R}$. When μ is irrelevant or clear from the context, we will write $\mathtt{WF}^g(\tau)$ instead of $\mathtt{WF}^g(\mu(\tau))$. A fairness property φ is an aggregation function mapping $(\mathtt{WF}^a(\tau), \mathtt{WF}^b(\tau))$ to a real-valued *bias*. Table 6.1 summarize how existing fairness properties, namely demographic parity (DP) [Dwo+12], disparate impact (DI) [Fel+15], and equal opportunity (EqOpp) [HPS16] can be cast into this form.

Estimating EqOpp requires the ground truth labels of the individuals be revealed after the shield has made its decisions on them. To accommodate ground truth, we introduce the set $\mathcal{Z} = \{0, 1\}$, such that traces are of the form $\tau = (x_1, y_1, z_1), \ldots, (x_t, y_t, z_t) \in (\mathcal{X} \times \mathcal{Y} \times \mathcal{Z})^*$, where each z_i is the ground truth label of the i-th individual. The shield is adapted to $(\mathcal{X} \times \mathcal{Y} \times \mathcal{Z})^* \times \mathcal{X} \to \mathcal{Y}$, where the set $\mathcal{Z}$ is treated as another input space and the probability distribution $\mathbb{P}(\mathcal{Z} = z_i \mid \mathcal{X} = x_i)$ is assumed to be available.

Example 6.3 (Continuation of Example 6.2)**.** *In the bilingual team example, the welfare of a linguistic group g is defined as the fraction of the team proficient in language g. A more nuanced interpretation considers the welfare of group g as the fraction of accepted candidates among those proficient in language g, which is the empirical variant of demographic parity (DP). This measure accounts for the possibility that the linguistic distribution of the population may not be evenly split. If one language is more prevalent in the target population, the hired*

Name	Counters	$\texttt{WF}^g$	φ
Demographic parity (DP)	n_a, n_{a1}, n_b, n_{b1}	n_{g1}/n_g	$\lvert\texttt{WF}^a(\tau) - \texttt{WF}^b(\tau)\rvert$
Disparate impact (DI)	n_a, n_{a1}, n_b, n_{b1}	n_{g1}/n_g	$\left\lvert\texttt{WF}^a(\tau) \div \texttt{WF}^b(\tau)\right\rvert$
Equal opportunity (EqOpp)	$n'_a, n'_{a1}, n'_b, n'_{b1}$	n'_{g1}/n'_g	$\lvert\texttt{WF}^a(\tau) - \texttt{WF}^b(\tau)\rvert$

Table 6.1: Empirical variants of fairness properties: For $g \in \{a, b\}$, the counters n_g and n_{g1} represent the total numbers of individuals from group g who appeared and were accepted, respectively. Counters n'_g and n'_{g1} denote the total numbers of appeared and accepted individuals whose ground truth labels are "1." If a welfare value is undefined due to a null denominator, we set $\varphi = 0$.

team should proportionally include more members proficient in that language. To obtain an empirical variant of equal opportunity (EqOpp), we would need to assume the existence of a ground truth on whether a candidate is actually a good employee for this job. This can typically only be assessed after the candidate actually works for some time with the team, so it is not easy to estimate a priori. In such case, the welfare of a group would be the fractions of hired candidates with respect to the actually good *candidates for each linguistic group.*

Bounded-horizon fairness shields. From now on, we use the convention that θ is the input distribution, φ is the fairness property, and κ is the *bias threshold*. Let T be a given time horizon. The set $\Pi^{\theta,T}_{\texttt{fair}}$ of *fairness shields over time* T is the set of every shield that fulfills $\varphi(\cdot) \leq \kappa$ after time T, i.e.,

$$\Pi^{\theta,T}_{\texttt{fair}} \coloneqq \left\{\pi \in \Pi^T \mid \forall \tau \in \texttt{FT}^T_{\theta,\pi} \,.\, \varphi(\tau) \leq \kappa\right\}. \tag{6.5}$$

We now define optimal bounded-horizon fairness shields as below.

Definition 6.1 (Finite Horizon Shields)**.** Let $T > 0$ be the time horizon. A *finite horizon shield* (usually abbreviated to FinHzn) is the one that solves:

$$\pi^* \coloneqq \underset{\pi \in \Pi^{\theta,T}_{\texttt{fair}}}{\arg\min}\, \mathbb{E}[\mathit{cost}; \theta, \pi, T]. \tag{6.6}$$

Note that even if the input distribution θ is learned and imprecise, as long as it shares the same support as the true distribution, the fairness guarantees provided by the shield remain unaffected; only the cost-optimality may be compromised.

Periodic fairness shields. FinHzn shields stipulate that fairness be satisfied at the end of the given horizon. However, in many situations, it may be desirable to ensure fairness not only at the end of the horizon but also at intermediate points occurring at regular intervals. For instance, a human resources department that is required to maintain a fair distribution of employees over the course of a quarter might also need to ensure a similar property for their yearly revision, after four quarters. This type of fairness is referred to as *periodic fairness* in the literature [Ala+24]. For this class of fairness properties, we define the set of T-periodic fairness shields as

$$\Pi_{\texttt{fair-per}} \coloneqq \left\{\pi \in \Pi \mid \forall m \in \mathbb{N} \,.\, \forall \tau \in \texttt{FT}^{mT}_{\theta,\pi} \,.\, \varphi(\tau) \leq \kappa\right\}. \tag{6.7}$$

Note that $\Pi_{\texttt{fair-per}}$ does not force every subtrace of length T, or mT for some $m \in \mathbb{N}$, to satisfy a certain fairness constraint. The reader may think of the multiples of the period $T, 2T, 3T, \ldots$ as "examination dates": the trace will be inspected at time mT, and by then it has to be correct. Therefore, subtraces of any length that do not end in an examination date may have bias values slightly over the threshold.

Definition 6.2 (Optimal T-periodic fairness shield)**.** Let $T > 0$ be the time period. An *optimal T-periodic fairness shield* is given by:

$$\pi^* \coloneqq \operatorname*{arg\,min}_{\pi \in \Pi_{\texttt{fair-per}}} \sup_{\substack{m \in \mathbb{N} \\ \tau \in \texttt{FT}_{\theta,\pi}^{mT}}} \mathbb{E}[\mathit{cost} \mid \tau; \theta, \pi, T]. \tag{6.8}$$

Equation (6.8) requires fairness at each mT-th time (measured from the beginning), and minimizes the maximum expected cost over each period. The existence of this minimum remains an open question. In Section 6.4, we propose three "best-effort" approaches to compute periodically fair shields (under mild assumptions) that are as cost-optimal as possible.

6.2.3 Fairness Shielding within the Reactive Decision Making Framework

The sequential input can be modelled by an environment $\mathscr{E} = (\mathcal{O}, \mathcal{A}, \mathscr{T})$, with $\mathcal{O} = \mathcal{X}$ (what we called the input space, $\mathcal{X} = \mathcal{G} \times \mathbb{B} \times \mathbb{C}$), $\mathcal{A} = \mathcal{Y}$, and a transition function $\mathscr{T}$ characterized by a single input distribution. That is, for all trace $\tau \in (\mathcal{O} \times \mathcal{A})^*$, there is a unique distribution $\theta \in \mathcal{D}(\mathcal{X})$ such that $\mathscr{T}(\tau) = \theta$.

The main difference between this shielding setting and those presented in Chapters 4 and 5 is that correctness is not established by avoiding concrete observations states, but rather correctness depends on the whole trace via a series of counters. In fact, every observation has a probability that is independent from the behaviour of the agent or the shield. Another notable difference in this case is that we are not interested in the notion of minimal interference as expressed in Definitions 3.10 and 3.12. In contrast, we assign a specific weight, or cost, to each interference, and build the shield that minimizes interferences in expectation.

6.3 Algorithm for Finite Horizon Shield Synthesis

We present our algorithm for synthesizing FinHzn shields as defined in Definition 6.1. A FinHzn shield π^* computes an output $y = \pi^*(\tau, x)$ for every trace $\tau \in (\mathcal{X} \times \mathcal{Y})^{\leq T}$ and every input $x \in \mathcal{X}$. Our synthesis algorithm builds π^* recursively for traces of increasing length, using an auxiliary *value function* $v(\tau)$ that represents the minimal expected cost conditioned on traces with prefix τ. To define $v(\tau)$, we generalize fairness shields with the condition that a certain trace has already occurred. Given a time horizon t and a trace τ, whose length can differ from t, the set of *fairness shields over time t after τ* is defined as

$$\Pi_{\texttt{fair}}^{\theta,t|\tau} \coloneqq \left\{ \pi \in \Pi^t \mid \forall \tau' \in (\mathcal{X} \times \mathcal{Y})^t \,.\, \tau\tau' \in \texttt{FT}_{\theta,\pi}^{|\tau|+t} \implies \varphi(\tau\tau') \leq \kappa \right\}.$$

Then $v(\tau)$ is given by:

$$v(\tau) \coloneqq \min_{\pi \in \Pi_{\text{fair}}^{\theta,(T-|\tau|)|\tau}} \mathbb{E}[\mathit{cost} \mid \tau; \theta, \pi, T - |\tau|]. \tag{6.9}$$

For every trace τ and every input $x \in \mathcal{X}$, the optimal value of the shield is $\pi^*(\tau, x) = \arg\min_{y \in \mathcal{Y}} v(\tau, (x, y))$.

In Section 6.3.1, we present a recursive dynamic programming algorithm for computing $v(\tau)$, whose complexity grows exponentially with the length of τ. In Section 6.3.2, we present show how the algorithm proposed in Section 6.3.1 can actually be adapted to use only the p counters defining the fairness property, thus solving the synthesis problem more efficiently, with polynomial complexity for the vast majority of fairness properties.

6.3.1 Recursive Computation of the Value Function

We compute the value function recursively, defining a trivial value for traces of length $|\tau| = T$, and showing how the value function for traces of lenght $|\tau| < T$ can be computed by simulating a single optimization step by the shield and using the value function for traces of length $|\tau| + 1$.

Base case. Let T be the time horizon and τ be a trace of length T. Since the horizon has already been reached, if $\varphi(\tau) \leq \kappa$, then the expected cost is zero because fairness is already satisfied and no more cost needs to be incurre. On the other hand, if $\varphi(\tau) > \kappa$, the expected cost is infinite, because, no matter what cost is paid, fairness can no longer be achieved. Formally,

$$v(\tau) = \begin{cases} 0 & \varphi(\tau) \leq \kappa, \\ \infty & \text{otherwise.} \end{cases} \tag{6.10}$$

Recursive case. Let τ be a trace of length smaller than T. The probability of the next input being $x = (g, r, c)$ is $\theta(x)$, and the shield decides to output y that either agrees with the recommendation r (the case $y = r$) or differs from it (the case $y \neq r$)—whichever minimizes the expected cost. When $y = r$, the trace becomes $(\tau \cdot (x, y = r))$. Therefore, no cost is incurred and the total cost is the same as $v(\tau \cdot (x, y = r))$. When $y \neq r$, the trace becomes $(\tau \cdot (x, y \neq r))$. Thus, the incurred cost is c and the new total cost becomes $c + v(\tau \cdot (x, y = r))$. Therefore

$$v(\tau) = \sum_{x=(g,r,c) \in \mathcal{X}} \theta(x) \cdot \min \begin{Bmatrix} v(\tau \cdot (x, y = r)), \\ v(\tau \cdot (x, y \neq r)) + c \end{Bmatrix}. \tag{6.11}$$

Equations (6.10) and (6.11) can be used to recursively compute $v(\tau)$ for every τ of length up to T, and the time and space complexity of this procedure is $\mathcal{O}(|\mathcal{X} \times \mathcal{Y}|^T)$. The correctness of Equation (6.11) is formally proven in Lemma 6.1. Before formalizing the argument, let us see an example of its inner workings.

Example 6.4 (Continuation of Example 6.3)**.** *Consider the task of hiring a linguistically balanced team with a horizon of $T = 50$ candidates and a target*

demographic parity property φ with a threshold $\kappa = 0.2$, i.e., 20%. By the end of the process, a trace τ of $|\tau| = 50$ candidates must satisfy $\varphi(\tau) < \kappa$.

Consider the following situation. Suppose τ' be the trace obtained after observing the first 48 candidates, i.e., $|\tau'| = 48$, and just two more candidates are going to be observed before the horizon ends. In τ', 24 candidates have been observed for each language proficiency group among A and B, and among them 12 from group A and 17 from group B have been accepted, resulting in $\varphi(\tau') = |12/24 - 17/24| = 0.208 > \kappa$. This temporary violation of DP is allowed since the process is ongoing. Suppose a new candidate $x = (g, r, c)$ appears, with $g = B$. The classifier tentatively accepts x and informs the shield that reversing this decision would incur a cost c. If the shield accepts x, the shield will be forced to reject the next candidate proficient in B or accept the next candidate proficient in A, regardless of the cost. Conversely, if the shield rejects x, it incurs an immediate cost of c but balances the languages to a point where intervention will not be required for the next decision. The shield must therefore weigh its options: either incur a known cost c now by rejecting x or risk an unknown future cost c' by accepting x. If the candidate is exceptionally qualified, i.e., the classifier recommends acceptance with a high cost of modifying its decision, the shield might choose to accept x, accepting the potential risk of rejecting another well-qualified candidate proficient in B in the next round. On the other hand, when the shield is considering a borderline candidate, it may be better to pay a small price with the current candidate and ensuring that the agent's decision will be respected for the next candidate, whatever the decision is.

Lemma 6.1. *Let $\theta \in \mathcal{D}(\mathcal{X})$ be a given joint distribution of sampling individuals and the output of the agent, let $\kappa > 0$ be a given threshold for a fairness property φ, and let $T > 0$ be a time horizon. For a trace $\tau \in (\mathcal{X} \times \mathcal{Y})^{\leq T}$, let $v(\tau)$ be the minimum expected cost after τ, formally defined as*

$$v(\tau) \coloneqq \min_{\pi \in \Pi_{\texttt{fair}}^{\theta, (T-|\tau|)|\tau}} \mathbb{E}[\mathit{cost} \mid \tau; \theta, \pi, T - |\tau|].$$

Then for τ with length $|\tau| = T$

$$v(\tau) = \begin{cases} 0 & \varphi(\tau) \leq \kappa, \\ \infty & \mathit{otherwise}, \end{cases} \tag{6.12}$$

for τ with $|\tau| < T$

$$v(\tau) = \sum_{x=(g,r,c) \in \mathcal{X}} \theta(x) \cdot \min \begin{Bmatrix} v(\tau \cdot (x, y = r)), \\ v(\tau \cdot (x, y \neq r)) + c \end{Bmatrix}, \tag{6.13}$$

and the shield defined as $\pi^*(\tau, x) \coloneqq \arg\min_{y \in \mathcal{Y}} v(\tau, (x, y))$ is an optimal fairness bounded horizon fairness shield, i.e.,

$$\pi^* = \arg\min_{\pi \in \Pi_{\texttt{fair}}^{\theta, T}} \mathbb{E}[\mathit{cost}; \theta, \pi, T]. \tag{6.14}$$

Proof. Consider the term to be minimized:

$$\mathbb{E}[\mathit{cost} \mid \tau; \theta, \pi, T - |\tau|] = \sum_{\tau' \in (\mathcal{X} \times \mathcal{Y})^{T - |\tau|}} \mathit{cost}(\tau') \cdot \mathbb{P}(\tau' \mid \tau; \theta, \pi). \tag{6.15}$$

The sum over traces $\tau' \in (\mathcal{X} \times \mathcal{Y})^{T-|\tau|}$ can be partitioned into a sum over inputs $x \in \mathcal{X}$ and traces $\tau'' \in (\mathcal{X} \times \mathcal{Y})^{T-|\tau|-1}$, by taking $\tau' = x\pi(\tau, x)\tau''$. The cost term is then

$$cost(\tau') = cost(x\pi(\tau, x)\tau'') = cost(\tau'') + cost(x\pi(\tau, x)).$$

If $x = (g, r, c)$, then

$$cost(x\pi(\tau, x)) = \begin{cases} 0 & \text{if } r = \pi(\tau, x) \\ c & \text{otherwise.} \end{cases}$$

The probability term is then:

$$\mathbb{P}(\tau' \mid \tau; \theta, \pi) = \mathbb{P}(x\pi(\tau, x) \mid \tau; \theta, \pi) \cdot \mathbb{P}(\tau'' \mid \tau x\pi(\tau, x); \theta, \pi)$$

The value in Equation (6.15) can be written as

$$\sum_{\tau' \in (\mathcal{X} \times \mathcal{Y})^{T-|\tau|}} cost(\tau') \cdot \mathbb{P}(\tau' \mid \tau; \theta, \pi) = A + B, \tag{6.16}$$

where

$$A = \sum_{x \in \mathcal{X}} \sum_{\tau'' \in (\mathcal{X} \times \mathcal{Y})^{T-|\tau|-1}} cost(\tau'') \cdot \mathbb{P}(x\pi(\tau, x) \mid \tau; \theta, \pi) \cdot \mathbb{P}(\tau'' \mid \tau x\pi(\tau, x); \theta, \pi), \tag{6.17}$$

and

$$B = \sum_{x \in \mathcal{X}} \sum_{\tau'' \in (\mathcal{X} \times \mathcal{Y})^{T-|\tau|-1}} cost(x\pi(\tau, x)) \cdot \mathbb{P}(x\pi(\tau, x) \mid \tau; \theta, \pi) \cdot \mathbb{P}(\tau'' \mid \tau x\pi(\tau, x); \theta, \pi). \tag{6.18}$$

Note that the term $\mathbb{P}(x\pi(\tau, x) \mid \tau; \theta, \pi)$ appears several times. This is the probability of getting a trace $x\pi(\tau, x)$ after having seen a trace τ. This is, by definition $\mathbb{P}(x\pi(\tau, x) \mid \tau; \theta, \pi) = \theta(x)$.

Since $\theta(x)$ does not depend on τ'', the sum in A can be rearranged as

$$A = \sum_{x \in \mathcal{X}} \theta(x) \cdot \sum_{\tau'' \in (\mathcal{X} \times \mathcal{Y})^{T-|\tau|-1}} cost(\tau'') \cdot \mathbb{P}(\tau'' \mid \tau x\pi(\tau, x); \theta, \pi), \tag{6.19}$$

and therefore

$$A = \sum_{x \in \mathcal{X}} \theta(x) \cdot \mathbb{E}[cost \mid \tau x\pi(\tau, x); \theta, \pi, T - |\tau| - 1]. \tag{6.20}$$

The term B can be similarly rearranged, taking into consideration that in this case $cost(x\pi(\tau, x))$ is also independent of τ'':

$$B = \sum_{x \in \mathcal{X}} cost(x\pi(\tau, x)) \cdot \theta(x) \cdot \sum_{\tau'' \in (\mathcal{X} \times \mathcal{Y})^{T-|\tau|-1}} \mathbb{P}\left(\tau'' \mid \tau x\pi(\tau, x); \theta, \pi\right). \tag{6.21}$$

The hanging term is the sum of probabilities, so by definition adds up to 1:

$$\sum_{\tau'' \in (\mathcal{X} \times \mathcal{Y})^{T-|\tau|-1}} \mathbb{P}\left(\tau'' \mid \tau x\pi(\tau, x); \theta, \pi\right) = 1.$$

Therefore

$$B = \sum_{x \in \mathcal{X}} \theta(x) \cdot cost(x\pi(\tau, x)) \tag{6.22}$$

Putting A and B together we get:

$$\begin{aligned} &\mathbb{E}[cost \mid \tau; \theta, \pi, T - |\tau|] = \\ &= \sum_{x \in \mathcal{X}} \theta(x) \cdot \Big(cost(x\pi(\tau, x)) + \mathbb{E}[cost \mid \tau x \pi(\tau, x); \theta, \pi, T - |\tau| - 1] \Big). \end{aligned} \tag{6.23}$$

This partitions the value of $\mathbb{E}[cost \mid \tau; \theta, \pi, T - |\tau|]$ into a sum of cost of current decision ($cost(x\pi(\tau, x))$) and expected cost in the rest of the trace. For every x, the optimal value of the shield $\pi(\tau, x)$ is the one that minimizes

$$cost(x\pi(\tau, x)) + \mathbb{E}[cost \mid \tau x \pi(\tau, x); \theta, \pi, T - |\tau| - 1].$$

This is precisely, the recursive property that we want to prove.

Finally, to prove Equation (6.14), just note that for the empty trace $\tau = \varepsilon$, we have $\Pi^{\theta,(T-|\tau|)|\tau}_{\texttt{fair}} = \Pi^{\theta,T}_{\texttt{fair}}$, which is precisely the set of shields set as minimization domain in Equation (6.14). □

6.3.2 Efficient Value Function Computation through Trace Abstraction

We now present an efficient recursive procedure for computing FinHzn shields that runs in polynomial time and space. The key observation is that φ is a fairness property that depends on τ through a statistic that uses p counters, as defined in Section 6.2.2. Consequently, the value function $v(\tau)$ in Equation (6.10) and Equation (6.11) depends only on counter values, not on exact traces. This allows us to define our dynamic programming algorithm over the set of counter values taken by the statistic μ. Let $R_{\mu,T} \subseteq \mathbb{N}^p$ be the set of values the statistic μ can take from traces of length at most T. We have the following complexity result.

Theorem 6.1. *Solving the bounded-horizon shield-synthesis problem requires $\mathcal{O}(|R_{\mu,T}| \cdot |\mathcal{X}|)$-time and $\mathcal{O}(|R_{\mu,T}| \cdot |\mathcal{X}|)$-space.*

Proof. In this section we have described a dynamic programming approach to synthesize the shield by recursively computing $v(\tau)$ for all possible traces $\tau \in (\mathcal{X} \times \mathcal{Y})^{\leq T}$. As explained before, these computations do not depend directly on τ, but rather on the statistic μ, that depends on p counters, taking values in the set $R_{\mu,T}$. We need to build a table with the shield values for every pair of counter values and input. Therefore, the table occupies a space $\mathcal{O}(|R_{\mu,T}| \cdot |\mathcal{X}|)$. Every element of the table has to be computed only once, and it is done as a sum over all elements of x, thus the cost in time is $\mathcal{O}(|R_{\mu,T}| \cdot |\mathcal{X}|)$. □

Most fairness properties, e.g., DP and EqOpp, have a range of $R_{\mu,T} = [0, T]^p$, where p is the number of counters ($p = 4$ for DP, and $p = 5$ for EqOpp), making the complexity polynomial in the length of the time horizon.

6.4 Algorithms for Periodic Shield Synthesis

Until now, we have described a method to synthesize finite-horizon shields, that is, shields that ensure fairness after a finite horizon T (Definition 6.1). In this section we explore the problem of synthesizing T-periodic shields, which guarantee fairness for unbounded traces at every T decisions (Definition 6.2). As previously noted, we leave the question of computing optimal T-periodic shields open, and present three "best-effort" solutions to the problem, each with different costs and guarantees.

We present algorithms for computing periodic fairness shields for a broad subclass of group fairness properties, which we call *difference of ratios* (DoR) properties. A statistic μ is *single-counter* if it maps every trace τ to a single counter value, i.e., $\mu(\tau) \in \mathbb{N}$, and *additive* if $\mu(\tau \cdot \tau') = \mu(\tau) + \mu(\tau')$ for any traces τ and τ'. A group fairness property φ is DoR if

(a) for each group g, $\texttt{WF}^g(\tau) = \texttt{num}^g(\tau)/\texttt{den}^g(\tau)$, where $\texttt{num}^g(\tau)$ and $\texttt{den}^g(\tau)$ are additive single-counter statistics, and

(b) $\varphi(\tau) = |\texttt{WF}^a(\tau) - \texttt{WF}^b(\tau)|$.

Many fairness properties, including DP and EqOpp, are DoR. See, for example, [BHN23, Table 3.5] for a non-exhaustive list. In this case, DI is an exception because it violates the condition (b).

For DoR fairness properties, we propose two approaches for constructing periodic fairness shields: *static* and *dynamic*, and we explore their respective strengths and weaknesses.

6.4.1 Periodic Shielding: The Static Approach

In the static approach, a periodic shield is obtained by *concatenating infinitely many identical copies of a statically computed bounded-horizon shield* π, synthesized with the time period T as the horizon. We present two ways of computing π so that its infinite concatenation is T-periodic fair.

6.4.1.1 Approach I: Static-Fair Shields.

Definition 6.3 (Static-Fair shields)**.** A shield is called Static-Fair if it is the concatenation of infinite copies of a FinHzn shield (from Definition 6.1).

Unfortunately, Static-Fair shields do not always satisfy periodic fairness. Consider a trace $\tau = \tau_1 \dots \tau_m$ for an arbitrary $m > 0$, generated by a Static-Fair shield, such that each segment τ_i is of length T. It follows from the property of FinHzn shields that $\varphi(\tau_i) \leq \kappa$ for each individual i. However, T-periodic fairness may be violated because $\varphi(\tau)$ need not be bounded by κ.

Example 6.5. *Consider DP with* $0 < \kappa < 1 - 2/T$*. Suppose* τ_1 *and* τ_2 *are traces of length* T*, defined as follows. The first trace* τ_1 *contains* 1 *candidate from group* A*,* $T - 1$ *candidates from group* B*, and none were accepted. The second trace* τ_2 *contains* $T - 1$ *candidates from group* A*,* 1 *candidate from group* B*, and all were accepted. Both traces are fair, since* $\varphi(\tau_1) = \varphi(\tau_2) = 0$*. However, when concatenating the two traces together, the resulting trace* $\tau_1 \cdot \tau_2$ *is very biased,*

since it contains T candidates from both group A and B, but only one accepted candidate from group B, while having $T-1$ candidates accepted from group A. Concretely, $\varphi(\tau_1\tau_2) = |(T-1)/T - 1/T| = 1 - 2/T > \kappa$ (biased). This example is summarized in Table 6.2.

An important feature of these counter-examples is the excessive skewness of appearance rates across the two groups. We further explore this phenomenon in Section 6.6.1.2. We show that Static-Fair shields are T-periodic fair if the appearance rates of the two groups are constant across every period.

Theorem 6.2 (Conditional correctness of Static-Fair shields)**.** *Let φ be a DoR fairness property. Consider a Static-Fair shield π, and let $\tau = \tau_1 \dots \tau_m \in \mathtt{FT}_{\theta,\pi}^{mT}$ be a trace such that $|\tau_i| = T$ for all $i \leq m$. If $\mathtt{den}^g(\tau_i) = \mathtt{den}^g(\tau_j)$ for every $i, j \leq m$ and $g \in \{a, b\}$, then the fairness property $\varphi(\tau) \leq \kappa$ is guaranteed.*

Proof. Given the condition, we can name $\mathtt{den}^a$ and $\mathtt{den}^b$ to the unique values of $\mathtt{den}^a(\tau_i)$ and $\mathtt{den}^b(\tau_i)$. For each $i \leq m$, we have the condition that

$$\left| \frac{\mathtt{num}^a(\tau_i)}{\mathtt{den}^a} - \frac{\mathtt{num}^b(\tau_i)}{\mathtt{den}^b} \right| \leq \kappa. \tag{6.24}$$

We want to prove a fairness condition for the trace $\tau_1 \dots \tau_m$, that is expressed as

$$\left| \frac{\sum_{i=1}^m \mathtt{num}^a(\tau_i)}{m \cdot \mathtt{den}^a} - \frac{\sum_{i=1}^m \mathtt{num}^b(\tau_i)}{m \cdot \mathtt{den}^b} \right| \leq \kappa. \tag{6.25}$$

Because of the denominators being the same across all traces, we can reorder the left-hand-side of Equation (6.25) as

$$\frac{1}{m} \left| \sum_{i=1}^m \left(\frac{\mathtt{num}^a(\tau_i)}{\mathtt{den}^a} - \frac{\mathtt{num}^b(\tau_i)}{\mathtt{den}^b} \right) \right|. \tag{6.26}$$

Applying the triangular inequality to Equation (6.26) and the condition in Equation (6.24), we get

$$\begin{aligned} \frac{1}{m} \left| \sum_{i=1}^m \left(\frac{\mathtt{num}^a(\tau_i)}{\mathtt{den}^a} - \frac{\mathtt{num}^b(\tau_i)}{\mathtt{den}^b} \right) \right| &\leq \frac{1}{m} \sum_{i=1}^m \left| \frac{\mathtt{num}^a(\tau_i)}{\mathtt{den}^a} - \frac{\mathtt{num}^b(\tau_i)}{\mathtt{den}^b} \right| \leq \\ &\leq \frac{1}{m} \cdot m\kappa = \kappa. \end{aligned}$$

□

	n_a	n_{a1}	n_b	n_{b1}	DP (φ)	$\varphi \leq \kappa$?
τ_1	1	0	$T-1$	0	0	✓
τ_2	$T-1$	$T-1$	1	1	0	✓
$\tau_1\tau_2$	T	$T-1$	T	1	$1-2/T$	✗

Table 6.2: Counterexample showing that Static-Fair shields may not be periodically fair for DP. Suppose the bias threshold is $0 < \kappa < 1 - 2/T$. The traces τ_1, τ_2 fulfill DP but their concatenation does not.

While the condition in Theorem 6.2 appears conservative, we show in Section 6.6.1.2 (Theorem 6.5) that it is in fact tight. The tightness result is expressed in terms of *balanced traces*, which is a concept that will appear also in the following section.

Definition 6.4 (Balanced traces)**.** Let $\mu^a, \mu^b \colon (\mathcal{X} \times \mathcal{Y})^* \to \mathbb{N}$ be a pair of single-counter statistics, $T > 0$ be a given time horizon, and $N \leq T/2$ be a given integer. A trace τ of length T is *N-balanced with respect to μ^a* and μ^b if both $\mu^a(\tau) \geq N$ and $\mu^b(\tau) \geq N$. We denote the set of all N-balanced traces of length t as $\mathtt{BT}^T(\mu^a, \mu^b, N)$.

A particular case of Theorem 6.2 is that fairness is guaranteed when all traces are $(T/2)$-balanced with respect to the denominators. In Theorem 6.5, we show, for the case of demographic parity, with $\mu^a, \mu^b = \mathtt{den}^a, \mathtt{den}^b$, for every κ, there exist m and $\lfloor (T-1)/2 \rfloor$-balanced traces $\tau_1, \ldots, \tau_m$ such that $\varphi_{\mathtt{DP}}(\tau_i) \leq \kappa$ for each i, but $\varphi_{\mathtt{DP}}(\tau_1 \ldots \tau_m) > \kappa$. However, these are worst-case scenarios and are uninteresting from a practical point of view. In our experiments, Static-Fair shields fulfill periodic fairness in a majority of cases even if the traces violate the condition in Theorem 6.2.

6.4.1.2 Approach II: Static-BW Shields.

When the condition of Theorem 6.2 is violated, Static-Fair shields cannot guarantee fairness as the bound on the bias is not closed under concatenation of traces (see Example 6.5). A stronger property that is indeed closed under concatenation is when a bound is imposed on each group's welfare. Let l, u be constants with $0 \leq l < u \leq 1$. A trace τ has *bounded welfare* (BW) if for each group $g \in \mathcal{G}$, $\mathtt{WF}^g(\tau) = \mathtt{num}^g(\tau)/\mathtt{den}^g(\tau)$ belongs to $[l, u]$. The pair (l, u) will be called *welfare bounds*. We show that BW is closed under trace concatenations, which depends on the additive property of $\mathtt{num}^g$ and $\mathtt{den}^g$.

Lemma 6.2. *Let (l, u) be given welfare bounds, and $\mathtt{WF}^g(\cdot) \equiv \mathtt{num}^g(\cdot)/\mathtt{den}^g(\cdot)$ for additive $\mathtt{num}^g, \mathtt{den}^g$. For a trace $\tau = \tau_1 \ldots \tau_m$, if for each i, $\mathtt{WF}^g(\tau_i) \in [l, u]$, then $\mathtt{WF}^g(\tau) \in [l, u]$.*

To prove Lemma 6.2, we first need to prove the following auxiliary result.

Lemma 6.3. *Let $a_1, \ldots, a_m$, $b_1, \ldots, b_m$ be positive real numbers. Then*

$$\min_{i \in \{1 \ldots m\}} \frac{a_i}{b_i} \leq \frac{\sum_{i=1}^m a_i}{\sum_{i=1}^m b_i} \leq \max_{i \in \{1 \ldots m\}} \frac{a_i}{b_i}. \tag{6.27}$$

Proof. This is an extension of the following known inequality: given positive numbers w, x, y, z, if $w/x < y/z$, then $\frac{w}{x} \leq \frac{w+y}{x+z} \leq \frac{y}{z}$. We can restate it as:

$$\min\left(\frac{w}{x}, \frac{y}{z}\right) \leq \frac{w+y}{x+z} \leq \max\left(\frac{w}{x}, \frac{y}{z}\right). \tag{6.28}$$

We prove this result by induction on m. The base case for $m = 1$ is trivial.

For a general m, we start applying inequality (6.28) with $w = \sum_{i=1}^{m-1} a_i$, $x = \sum_{i=1}^{m-1} b_i$, $y = a_m$, and $z = b_m$, to obtain:

$$\frac{\sum_{i=1}^{m} a_i}{\sum_{i=1}^{m} b_i} \leq \max\left(\frac{\sum_{i=1}^{m-1} a_i}{\sum_{i=1}^{m-1} b_i}, \frac{a_m}{b_m}\right).$$

Applying the induction hypothesis we have that

$$\frac{\sum_{i=1}^{m-1} a_i}{\sum_{i=1}^{m-1} b_i} \leq \max_{i\in\{1\ldots m-1\}} \frac{a_i}{b_i}, \tag{6.29}$$

and therefore:

$$\frac{\sum_{i=1}^{m} a_i}{\sum_{i=1}^{m} b_i} \leq \max\left(\max_{i\in\{1\ldots m-1\}} \frac{a_i}{b_i}, \frac{a_m}{b_m}\right) = \max_{i\in\{1\ldots m\}} \frac{a_i}{b_i}.$$

This proves the right-side inequality of Equation (6.27). The left-side is analogous. □

Proof (Of Lemma 6.2). Let $n_i^a = \texttt{den}^a(\tau_i)$, $n_i^{a1} = \texttt{num}^a(\tau_i)$, $n_i^b = \texttt{den}^b(\tau_i)$, and $n_i^{b1} = \texttt{den}^b(\tau_i)$. Applying Lemma 6.3, we have for all $g \in \mathcal{G}$ that

$$\min_{i\in\{1\ldots n\}} \frac{n_i^{g1}}{n_i^g} \leq \frac{\sum_{i=1}^{n} n_i^{g1}}{\sum_{i=1}^{n} n_i^g} \leq \max_{i\in\{1\ldots n\}} \frac{n_i^{g1}}{n_i^g}. \tag{6.30}$$

And we also know that all welfare values are bounded by l and u. That is, for all $i \in \{1\ldots n\}$ and all $g \in \mathcal{G}$

$$l \leq \frac{n_i^{g1}}{n_i^g} \leq u \tag{6.31}$$

In particular, Equation (6.31) applies to the maximum and minimum welfare values. This, together with Equation (6.30) finishes the proof. □

For DoR properties, BW implies fairness when $u - l \leq \kappa$. Combining this with Lemma 6.2, we infer that if π is a bounded-horizon shield that fulfills BW on every trace τ of length T for welfare bounds (l, u) with $u - l \leq \kappa$, then the concatenation of infinite copies of π would be a T-periodic fairness shield. The natural course of action for computing shields that fulfill BW is to mimic Definition 6.1, replacing the condition on φ with a condition on welfare. However, if we define the set of BW-fulfilling shields as

$$\Pi_{\texttt{BW}}^{\theta,T} \coloneqq \left\{\pi \in \Pi \mid \forall \tau \in \texttt{FT}_{\theta,\pi}^{T} \,.\, \forall g \in \{a, b\} \;.\, l \leq \texttt{WF}^g(\tau) \leq u\right\},$$

the set $\Pi_{\texttt{BW}}^{\theta,T}$ can be empty for some T, l, u. Following is an example.

Example 6.6. *Suppose* $\texttt{WF}^g(\tau) = n_{g1}/n_g$*, where* n_{g1} *and* n_g *are the total numbers of accepted and appeared individuals from group* g *(as in DP). Suppose* $T = 2, l = 0.2, u = 0.4$*. It is easy to see that no matter what the shield does, for every* τ *of length* 2, $\texttt{WF}^g(\tau) \in \{0, 0.5, 1\}$*. Therefore,* $\Pi_{[0.2,0.4]}^{2} = \emptyset$*.*

The emptiness of $\Pi_{\texttt{BW}}^{\theta,T}$ is due to a large disparity between the appearance rates of individuals from the two groups, which occurs for shorter time horizons and for datasets where one group has significantly lesser representation than the other group. To circumvent this technical inconvenience, we make the following assumption on observed traces.

Assumption 6.1. *Let l, u be welfare bounds, and $\tau = \tau_1 \dots \tau_m \in \texttt{FT}_{\theta,\pi}^{mT}$ be a trace with $|\tau_i| = T$ for each i. Every τ_i is N-balanced w.r.t. $\texttt{den}^a$ and $\texttt{den}^b$ for $N = \lceil 1/(u-l) \rceil$.*

Assumption 6.1 may be reasonable depending on l, u, T, and the input distribution θ. Intuitively, for a larger T and a smaller skew of appearance probabilities for individuals between the two groups, the probability of fulfilling Assumption 6.1 is larger (for a given finite m). At the end of this section (Equation (6.36)) we quantify it as the probability of a sample from a binomial distribution lying between N and $T - N$.

Definition 6.5 (Static-BW shields)**.** Let l, u be given welfare bounds, and T be a given time period. A Static-BW shield is the concatenation of infinite copies of the shield π^* solving

$$\pi^* = \underset{\pi \in \Pi_{\texttt{BW}}^{\theta,T,N}}{\arg\min}\, \mathbb{E}[\mathit{cost}; \theta, \pi, T], \tag{6.32}$$

where $N = \lceil 1/(u-l) \rceil$, and

$$\Pi_{\texttt{BW}}^{\theta,T,N} \coloneqq \left\{ \pi \in \Pi \mid \forall \tau \in \texttt{FT}_{\theta,\pi}^{T} \cap \texttt{BT}_{N}^{T}. \forall g \in \{a, b\} \ . \ l \le \texttt{WF}^g(\tau) \le u \right\}.$$

With the following technical result prove that $\Pi_{\texttt{BW}}^{\theta,T,N}$ is indeed non-empty when Assumption 6.1 is fulfilled. We do so by constructing the shield that keeps $\texttt{WF}^g(\tau)$ just above l and showing that it also guarantees $\texttt{WF}^g(\tau) \le u$ when the trace is sufficiently balanced.

Lemma 6.4. *Let φ be a DoR property with $\varphi(\tau) = |\texttt{WF}^a(\tau) - \texttt{WF}^b(\tau)|$, and $\texttt{WF}^g(\tau) = \texttt{num}^g(\tau)/\texttt{den}^g(\tau)$. Let $0 \le l < u \le 1$ be a pair of welfare bounds. The set of shields*

$$\Pi_{\texttt{BW}}^{T,N} \coloneqq \left\{ \pi \in \Pi \mid \forall \tau \in \texttt{FT}_{\theta,\pi}^{t} \cap \texttt{BT}_{N}^{T} \ . \ \forall g \in \{a, b\} \ . \ l \le \texttt{WF}^g(\tau) \le u \right\}$$

is not empty for $N \ge \left\lceil \frac{1}{u-l} \right\rceil$.

Proof. For a shield to exist that can enforce bounds $[l, u]$ on the welfare, there must exist, for every value of $\texttt{den}^g(\tau)$, at least one way of deciding for increasing or not $\texttt{num}^g(\tau)$ that maintains the welfare in the desired bounds. Since we do not know *a priori* the value of $\texttt{den}^g(\tau)$, this decision must be incremental, and be such that the welfare is maintained for any value of $\texttt{den}^g(\tau)$.

To express this, there needs to exist a sequence $(x_n) \subseteq \mathbb{N}$ for all $n \ge N$ such that

$$l \le \frac{x_n}{n} \le u, \quad \text{and} \quad x_{n+1} - x_n \in \{0, 1\}. \tag{6.33}$$

Given l, and u, if $\texttt{den}^g(\tau)$ is at least N for a given group g, the shield can force $\texttt{num}^g(\tau)$ to be exactly x_n to ensure the bound on welfare is met.

The condition in Equation (6.33) can be reformulated as $ln \leq x_n \leq un$, and since x_n needs to be an integer, we can tighten it to

$$\lceil ln \rceil \leq x_n \leq \lfloor un \rfloor . \tag{6.34}$$

One option is to try $x_n = \lceil ln \rceil$. We have to prove that this choice satisfies two conditions: (i) $x_{n+1} - x_n \in \{0, 1\}$, and (ii) Equation (6.34).

(i) This is true for any sequence x_n built as the integer part of nl, where $l \in [0, 1]$. For any number x, it is known that $x = \lceil x \rceil - \{x\}$, where $0 \leq \{x\} < 1$. Applying this inequality twice, we get

$$x_{n+1} - x_n = \lceil l(n+1) \rceil - \lceil ln \rceil < l(n+1) - \lceil ln \rceil \leq l(n+1) - ln = 1 + l < 2.$$

Since $\lceil l(n+1) \rceil - \lceil ln \rceil$ is an integer strictly smaller than 2, it is smaller or equal than 1. It is also clearly non-negative, so it has to be either 0 or 1.

(ii) By construction, $ln \leq \lceil ln \rceil$. Now we have to see that $\lceil ln \rceil \leq un$. If $\lceil ln \rceil = ln$, then for any $n \geq 1$, we have $x_n \leq un$ on account of $l < u$. If $\lceil ln \rceil = ln + 1$, we need $ln + 1 \leq un$, which is equivalent to $n \geq \frac{1}{u-l}$. Since n needs to be an integer, selecting $N = \left\lceil \frac{1}{u-l} \right\rceil$ ensures this condition is satisfied for all $n \geq N$.

□

This result guarantees that the optimization problem in (6.32) is feasible, and thus Static-BW shields are well-defined. Intuitively, we obtain a "best-effort" solution for π^*: when a trace satisfies Assumption 6.1, π^* guarantees that τ satisfies BW with minimum expected cost. Otherwise, π^* has no BW requirement, and thus for traces that violate Assumption 6.1, the shield will incur zero cost by never intervening, voiding any potential fairness guarantee.

Synthesis of Static-BW shields follows the same approach as in Section 6.3 with Equation (6.10) replaced by:

$$v(\tau) = \begin{cases} 0 & \text{if } \tau \notin \texttt{BT}_N^T \vee \bigwedge_{g \in \{a,b\}} \texttt{WF}^a(\tau) \in [l, u], \\ \infty & \text{otherwise.} \end{cases} \tag{6.35}$$

We summarize the fairness guarantee below.

Theorem 6.3 (Conditional correctness of Static-BW shields)**.** *Let φ be a DoR fairness property. Let l, u be welfare bounds such that $u - l \leq \kappa$. For a given Static-BW shield π, let $\tau = \tau_1 \dots \tau_m \in \texttt{FT}_{\theta,\pi}^{mT}$ be a trace with $|\tau_i| = T$ for each $i \leq m$. If Assumption 6.1 holds, then the fairness property $\varphi(\tau) \leq \kappa$ is guaranteed.*

Proof. This is a direct consequence of Lemmas 6.2 and 6.4. If Assumption 6.1 holds, Lemma 6.4 ensures that the set of shields is non-empty. Furthermore, any such shield satisfies the fairness condition $\varphi(\tau) \leq \kappa$ for any trace in $\tau \in \texttt{FT}^{mT}$ by Lemma 6.2. □

Existence of Static-BW shields. The feasiblity of the condition $N \geq \left\lceil \frac{1}{u-l} \right\rceil$ in real cases depends on the values of l and u to enforce, as well as the incoming probability distribution. This condition, formulated as Assumption 6.1, is the key to guarantee the existence of Static-BW shields in Theorem 6.4. In its most simplified form, if we just care about the group membership of any incoming candidate, the distribution of incoming candidates follows a Bernoulli distribution $B(p)$, where p is the probability to receive a candidate of group A. After a time horizon T, the number of incoming candidates of group A follows a binomial distribution $Bin(T, p)$, and the probability to see at least N candidates of each group is the probability of the binomial being between N and $T - N$, which is

$$\sum_{k=N}^{T-N} \binom{T}{k} p^k (1-p)^{T-k}. \tag{6.36}$$

In practice, this corresponds to the probability that our shield will encounter a trace where demographic parity with the given bound on acceptance rates can be enforced. It is up to the user to evaluate whether this guarantee is enough for a given application.

6.4.2 Periodic Shielding: The Dynamic Approach

While the static approaches repeatedly use one statically computed bounded-horizon shield, the dynamic approach recomputes a new bounded-horizon shield at the beginning of each period, and thereby adjusts its future decisions based on the past biases. We formalize this below.

Definition 6.6 (Dynamic shields)**.** Suppose we are given a parameterized set of *available* shields $\Pi'(\tau) \subseteq \Pi$ where the parameter τ ranges over all finite traces. A Dynamic shield π is the concatenation of a sequence of shields $\pi_1, \pi_2, \ldots$ such that for every trace $\tau \in \mathtt{FT}_{\theta,\pi}^{mT}$ with $m \geq 0$, for every $\tau' \in (\mathcal{X} \times \mathcal{Y})^{<T}$, and for every input $x \in \mathcal{X}$, we have $\pi(\tau \cdot \tau', x) = \pi_{m+1}(\tau', x)$, where

$$\pi_{m+1} = \underset{\pi' \in \Pi'(\tau)}{\arg\min}\, \mathbb{E}[\mathit{cost} \mid \tau; \theta, \pi', T]. \tag{6.37}$$

The set $\Pi'(\tau)$ restricts the available set of shields that can be used for the next period for the given history τ. A naïve attempt for $\Pi'(\tau)$ would be to choose $\Pi'(\tau) = \Pi_{\mathtt{fair}}^{\theta,T|\tau}$ for every τ, so that fairness is guaranteed at the end of the current period. However, there exist histories for which $\Pi_{\mathtt{fair}}^{\theta,T|\tau}$ would be empty, implying that Equation (6.37) would not have a feasible solution for some τ, and the Dynamic shield would exhibit undefined behaviors.

This happens because there may be traces of length jT that satisfy a certain fairness constraint, but no shield can guarantee the next trace will satisfy the same constraint.

Example 6.7. *Consider* $\varphi = \mathtt{DP}$, $\kappa = 0.1$, $T = 100$, *and a trace* τ *such that* $n_a(\tau) = 2$, $n_{a1}(\tau) = 1$, $n_b(\tau) = 98$, *and* $n_{b1}(\tau) = 49$. *The trace* τ *satisfies* $\mathtt{DP}(\tau) = |1/2 - 49/98| = 0$. *Now assume we build a shield for the next fragment, and in generating the next trace* τ', *only individuals from group b have appeared for the first 99 samples. Let* $\tau'_{[1:99]}$ *denote this trace, and let* $\mathtt{Acc}_b$ *denote*

$\mathtt{WF}^b\tau \cdot \tau'_{[1:99]}$. *Then* $\mathtt{DP}(\tau \cdot \tau'_{[1:99]}) = |1/2 - \mathtt{Acc}_b|$. *If the last individual of* τ' *happens to be from group* a*, the acceptance rate of group* a *moves from* $1/2$ *to either* $1/3$ *(if it gets rejected) or* $2/3$ *(if it gets accepted). There is no possible value of* $\mathtt{Acc}_b$ *that simultaneously guarantees* $|1/3 - \mathtt{Acc}_b| \leq \kappa$ *and* $|2/3 - \mathtt{Acc}_b| \leq \kappa$.

To circumvent this technical inconvenience, we make the following mild assumption on the set of allowed histories, requiring $\Pi'(\tau)$ to fulfill fairness only if τ fulfills this assumption.

Assumption 6.2. *For a given trace* $\tau \in \mathtt{FT}^{jT}_{\theta,\pi}$ *with* $j > 0$*, every valid suffix* τ' *of length* t*, i.e.,* $\tau' \in \left\{\tau'' \in (\mathcal{X} \times \mathcal{Y})^T \mid \tau\tau'' \in \mathtt{FT}^{(j+1)T}_{\theta,\pi}\right\}$*, fulfills:*

$$\frac{1}{\mathtt{den}^a(\tau\tau')} + \frac{1}{\mathtt{den}^b(\tau\tau')} \leq \kappa + \varphi(\tau).$$

The set of shields $\Pi'(\cdot)$ available to the Dynamic shield in Definition 6.6 is then defined as:

$$\Pi'(\tau) = \Pi^{\theta,T}_{\texttt{fair-dyn}}(\tau) \coloneqq \begin{cases} \Pi^{\theta,T|\tau}_{\texttt{fair}} & \tau \text{ fulfills Assumption 6.2,} \\ \Pi & \text{otherwise.} \end{cases} \tag{6.38}$$

With the following technical result, we prove that $\Pi^{\theta,T|\tau}_{\texttt{fair}}$ is non-empty whenever τ fulfills Assumption 6.2, implying that $\Pi^{\theta,T}_{\texttt{fair-dyn}}(\tau)$ is non-empty for every τ. This is the analogous result to Lemma 6.4 for dynamic shields.

Lemma 6.5. *Let* φ *be a DoR fairness property with* $\varphi(\tau) = |\mathtt{WF}^a(\tau) - \mathtt{WF}^b(\tau)|$*, and* $\mathtt{WF}^g(\tau) = \mathtt{num}^g(\tau)/\mathtt{den}^g(\tau)$*. Let* τ_1 *be a trace and* $\kappa \geq 0$*. There exists a shield* $\pi \in \Pi$ *such that every trace* $\tau_2 \in \mathtt{FT}^T_{\theta,\pi} \cap S$ *satisfies* $\varphi(\tau_1 \cdot \tau_2) \leq \kappa$*, where*

$$S = \left\{\tau_2 \in (\mathcal{X} \times \mathcal{Y})^T \;:\; \frac{1}{\mathtt{den}^a(\tau_1\tau_2)} + \frac{1}{\mathtt{den}^b(\tau_1\tau_2)} \leq \kappa + \varphi(\tau_1)\right\}.$$

Proof. The proof of this result is analogous to that of Lemma 6.4, with a slightly more convoluted argument.

Let $n^1_a = \mathtt{den}^a(\tau_1)$, $n^1_{a1} = \mathtt{num}^a(\tau_1)$, $n^1_b = \mathtt{den}^b(\tau_1)$, and $n^1_{b1} = \mathtt{num}^b(\tau_1)$. Without loss of generality, we can assume that $n^1_{a1}/n^1_a - n^1_{b1}/n^1_b \geq 0$. The alternative case is analogous.

For a shield to exist that can enforce $\varphi(\tau_1\tau_2) \leq \kappa$ there must exist, for every value of $\mathtt{den}^g(\tau_1\tau_2)$ (in demographic parity, the amount of individuals of a group), at least one way of deciding acceptance and rejection (value of $\mathtt{num}^g(\tau_1\tau_2)$) that maintains the fairness property in the target bound. Since we do not know *a priori* how many individuals of each group will appear, this decision must be incremental, and be such that the fairness property is maintained for any number of individuals.

If we name $n_a = \mathtt{den}^a(\tau_2) \geq N_a$ and $n_b = \mathtt{den}^b(\tau_2) \geq N_b$, a new trace τ_2 is enforceable if we can choose N_a and N_b satisfying the following condition: there

exist two sequences $(x_{n_a}), (y_{n_b}) \subseteq \mathbb{N}$ such that for all $n^a \geq N_a$ and $n^b \geq N_b$

$$\left| \frac{n^1_{a1} + x_{n_a}}{n^1_a + n_a} - \frac{n^1_{b1} + y_{n_b}}{n^1_b + n_b} \right| \leq \kappa, \tag{6.39}$$

and for both sequences $x_{n_a+1} - x_{n_a} \in \{0, 1\}$ and $y_{n_a+1} - y_{n_a} \in \{0, 1\}$.

With the spirit of maintaining the welfare bounds as a proxy to maintaining fairness, we try

$$x_{n_a} \coloneqq \left\lfloor \frac{n^1_{a1}}{n^1_a} n_a \right\rfloor, \quad \text{and} \quad y_{n_a} \coloneqq \left\lceil \frac{n^1_{b1}}{n^1_b} n_b \right\rceil.$$

Using the same argument as in the proof of Theorem 6.4, point (i), the conditions on x_{n_a} and y_{n_b} incrementing by 0 or 1 are met by the fact that $n^1_{a1} \leq n^1_a$ and $n^1_{b1} \leq n^1_b$.

By definition of the floor function and ceiling functions

$$\frac{n^1_{a1} + x_{n_a}}{n^1_a + n_a} \leq \frac{n^1_{a1} + \frac{n^1_{a1}}{n^1_a} n_a}{n^1_a + n_a} = \frac{n^1_{a1}}{n^1_a},$$
$$\frac{n^1_{b1} + y_{n_b}}{n^1_b + n_b} \geq \frac{n^1_{b1} + \frac{n^1_{b1}}{n^1_b} n_b}{n^1_b + n_b} = \frac{n^1_{b1}}{n^1_b}.$$

Therefore

$$\frac{n^1_{a1} + x_{n_a}}{n^1_a + n_a} - \frac{n^1_{b1} + y_{n_b}}{n^1_b + n_b} \leq \frac{n^1_{a1}}{n^1_a} - \frac{n^1_{b1}}{n^1_b} = \varphi(\tau_1) \leq \kappa \tag{6.40}$$

To prove Equation (6.39), we still have to prove that

$$\frac{n^1_{a1} + x_{n_a}}{n^1_a + n_a} - \frac{n^1_{b1} + y_{n_b}}{n^1_b + n_b} \geq -\kappa. \tag{6.41}$$

By the definition of the floor function

$$\frac{n^1_{a1} + x_{n_a}}{n^1_a + n_a} \geq \frac{n^1_{a1} + \frac{n^1_{a1}}{n^1_a} n_a - 1}{n^1_a + n_a} = \frac{n^1_{a1}}{n^1_a} - \frac{1}{n^1_a + n_a},$$

and by the definition of the ceiling function

$$\frac{n^1_{b1} + y_{n_b}}{n^1_b + n_b} \leq \frac{n^1_{b1} + \frac{n^1_{b1}}{n^1_b} n_b + 1}{n^1_b + n_b} = \frac{n^1_{b1}}{n^1_b} + \frac{1}{n^1_b + n_b}.$$

Putting the previous two inequalities together, we have

$$\frac{n^1_{a1} + x_{n_a}}{n^1_a + n_a} - \frac{n^1_{b1} + y_{n_b}}{n^1_b + n_b} \geq \varphi(\tau_1) - \left(\frac{1}{n^1_a + n_a} + \frac{1}{n^1_b + n_b} \right).$$

To ensure that Equation (6.41) holds, it is sufficient to ensure that

$$\varphi(\tau_1) - \left(\frac{1}{n^1_a + n_a} + \frac{1}{n^1_b + n_b} \right) \geq -\kappa.$$

Rewriting the previous inequality we arrive to

$$\left(\frac{1}{n_a^1 + n_a} + \frac{1}{n_b^1 + n_b}\right) \leq \kappa + \varphi(\tau_1), \tag{6.42}$$

which is the condition defining the set S. Therefore the proposed sequences (x_{n_a}) and (y_{n_b}) satisfy Equation (6.39) for traces in S. □

Technically, this guarantees that the optimization problem in (6.37) is feasible and π_{m+1} always exists, making Dynamic shields are well-defined (Definition 6.6). Intuitively, we obtain a "best-effort" solution: If Assumption 6.2 is fulfilled then π_{m+1} is in $\Pi_{\texttt{fair}}^{\theta,T|\tau}$ and achieves fairness for the minimum expected cost. Otherwise, π_{m+1} can be any shield in Π that only optimizes for the expected cost; in particular, π_{m+1} will be the trivial shield that never intervenes (has zero cost).

Synthesis of Dynamic shields involves computing the sequence of shields $\pi_1, \pi_2, \ldots$, which are to be concatenated. We outline the algorithm below.

1. Generate a FinHzn shield (Definition 6.1) π for the property φ and the horizon T. Set $\pi_1 \coloneqq \pi$.

2. For $i \geq 1$, let π be the concatenation of the shields $\pi_1, \ldots, \pi_i$, and let $\tau \in \texttt{FT}_{\theta,\pi}^{iT}$ be the generated trace. Compute π_{i+1} that uses the same approach as in Section 6.3 with Equation (6.10) being replaced by:

$$v(\tau') = \begin{cases} 0 & \varphi(\tau\tau') \leq \kappa, \\ \infty & \text{otherwise.} \end{cases} \tag{6.43}$$

We summarize the fairness guarantee below.

Theorem 6.4 (Conditional correctness of Dynamic shields)**.** *Let φ be a DoR fairness property. Let π be a Dynamic shield that uses $\Pi_{\texttt{fair-dyn}}^{\theta,T}(\cdot)$ as the set of available shields. Let $\tau = \tau_1 \ldots \tau_m \in \texttt{FT}_{\theta,\pi}^{mT}$ be a trace with $|\tau_i| = T$ for each $i \leq m$. Suppose for every $i \leq m$, $\tau_1 \ldots \tau_i$ fulfills Assumption 6.2. Then the fairness property $\varphi(\tau) \leq \kappa$ is guaranteed.*

6.5 Experimental Evaluation

6.5.1 Experimental Setup

We demonstrate the effectiveness of fairness shields by testing them in the task of shielding several ML classifiers in tasks that are standard benchmarks in the fairness literature. We consider several state-of-the-art learning algorithms from literature of in-processing fairness, i.e., methods that enforce fairness *during* the learning process, typically by means of adding fairness-inducing regularizers to the loss functions. To preserve consistency among our experiments, we use the same neural architecture for every dataset and learning algorithm.

Dataset	Task	Sensitive Attribute	Instances	Features Num./Cat.	y_0 %	y_1 %	g_a %	g_b %	Stat. Par.
`Adult`	income	race	43131	5 / 7	75	25	10	90	0.14
`Adult`	income	gender	45222	5 / 7	75	25	33	68	0.20
`Bank`	credit	age	41188	9 / 10	89	11	2.6	97	0.13
`Compas`	recid.	gender	6172	5 / 4	54	46	19	81	0.13
`Compas`	recid.	race	6172	5 / 4	54	46	66	34	0.10
`German`	credit	gender	1000	6 / 13	30	70	31	69	0.07
`German`	credit	age	1000	6 / 13	30	70	19	81	0.15

Table 6.3: Datasets characteristics. The columns $y_{0/1}$ represent the percentage of instances where the ground truth is "accept" (1) and "reject" (0), respectively. The columns $g_{a/b}$ represent the percentage of instances that belong to group a and b, respectively. The last column is the statistical parity, representing the inherent bias in the dataset.

Computing infrastructure. All experiments were performed with a workstation with AMD Ryzen 9 5900x CPU, Nvidia GeForce RTX 3070Ti GPU, 32GB of RAM, running Ubuntu 20.04.

Datasets. We used four tabular datasets in our experiments, all common benchmarks in the fairness community: Adult [BK96], COMPAS [Kir+16], German Credit [Hof94] and Bank Marketing [MCR12]. Details on the task, sensitive attributes, size of the dataset, number of numerical and categorical features, as well as existing bias can be found in Table 6.3.

Training ML classifiers. To train our ML models, we adapted the implementation provided by the FFB benchmark [Han+24], using the same neural network, train-test splits, and most training hyper-parameters set as default in their implementation, tuning only the hyper-parameters related to fairness. The classifiers receive the full set of features as their input, including the protected feature, which is marked as a special feature. This is appropriate, since the learning algorithms that enforce fairness are of the in-processing type, so they use the protected feature as part of their input, and typically include a term in the loss function that depends especially on the protected feature.

We use fixed architecture multi-layer perception (MLP) with three hidden layers with sizes 512, 256, and 64 in all our experiments. In each case, the model is trained for 150 epochs with batches of 1024 instances, with the exception of the `German` dataset, which we trained with batches of 128, as the dataset has only 1000 instances. We use the Adam optimizer [KB15], with a learning rate of 0.01.

Learning algorithms. To train our classifiers, we used the following methods from the in-processing fairness literature:

- Differential Demographic Parity (DiffDP) is a gap regularization method for demographic parity. DiffDP introduces a term in the loss function that

	acc	ap	auc	f1	DP	EqOpp
ERM	91	62	94	**57**	11	**3.7**
DiffDP	90	57	93	40	3.4	26
HSIC	**91**	**63**	**94**	57	7.0	6.6
LAFTR	91	60	94	40	6.0	4.4
PR	91	59	93	49	**3.3**	34

Table 6.4: Performance of the ML models. Dataset: Bank.

	race						gender					
	acc	ap	auc	f1	DP	EqOpp	acc	ap	auc	f1	DP	EqOpp
ERM	**85**	**79**	**91**	**66**	11	6.1	**85**	**79**	**91**	**66**	16	9.6
DiffDP	84	76	90	62	**5.4**	4.0	83	71	87	54	0.2	33
HSIC	85	79	91	64	8.7	**3.1**	83	73	87	57	1.8	28
LAFTR	84	78	91	62	10	8.2	85	79	91	65	14	**1.5**
PR	84	76	89	61	5.6	4.0	83	71	87	53	**0.1**	33

Table 6.5: Performance of the ML models. Dataset: Adult.

penalizes differences in the prediction rates between different demographic groups [CM21].

- The Hilbert-Schmidt Independence Criterion (HSIC) is a statistical test used to measure the independence of two random variables. Adding an HSIC term measuring the independence between prediction accuracy and sensitive attributes to the loss has been used as a fair learning method [PS+17].
- Learning adversarially fair and transferable representations (LAFTR) is a method proposed by [Mad+18], where the classifier learns an intermediate representation of the data that minimizes classification error while simultaneously minimizing the ability of an adversary to predict sensitive features from the representation.
- Prejudice Remover (PR) [Kam+12] adds a term to the loss that penalizes mutual information between the prediction accuracy and the sensitive attribute.

As a baseline, we trained a fifth classifier simply minimizing empirical risk. We call it the empirical risk minimizer (ERM).

Hyperparameter tuning. Each of the in-processing fairness algorithms depends on the value of certain parameters that indicate the trade-off in the loss function between prediction accuracy and fairness. For each training algorithm, we manually fine-tuned the parameters to obtain a good performance with the same parameter values across all benchmarks. Unfortunately, the parameters of different algorithms have different interpretations and characteristic dimensions, so comparing them is not informative. We detail the ones we used in our experiments, and their meaning.

	gender						race					
	acc	ap	auc	f1	DP	EqOpp	acc	ap	auc	f1	DP	EqOpp
ERM	**65**	63	69	59	16	18	65	63	69	60	14	16
DiffDP	63	63	69	55	12	11	65	62	68	58	9.1	15
HSIC	64	63	69	56	15	9.8	64	62	68	57	**8.2**	**11**
LAFTR	65	**64**	**70**	**60**	17	15	**65**	**64**	**70**	**60**	13	18
PR	63	63	69	55	**12**	**8.0**	64	62	68	57	8.9	13

Table 6.6: Performance of the ML models. Dataset: Compas.

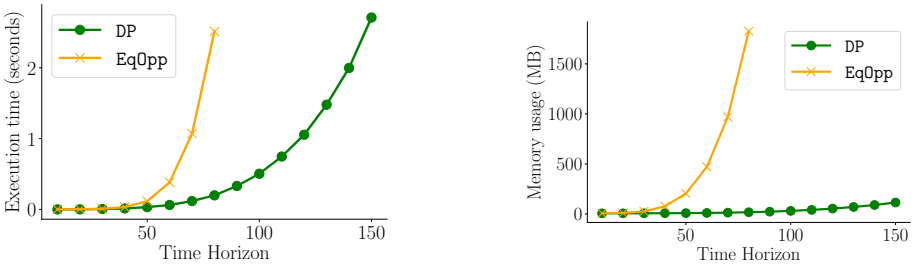
	gender						age					
	acc	ap	auc	f1	DP	EqOpp	acc	ap	auc	f1	DP	EqOpp
ERM	**76**	**87**	**77**	**83**	5.3	5.5	75	**86**	**76**	82	14	15
DiffDP	73	86	74	81	**1.1**	3.5	72	86	75	80	**0.5**	**1.8**
HSIC	73	86	74	81	1.4	**1.1**	74	86	74	82	4.2	6.0
LAFTR	73	87	76	81	8.2	4.1	73	85	74	81	10	6.4
PR	73	86	73	81	5.7	4.1	**75**	86	75	**83**	6.5	4.8

Table 6.7: Performance of the ML models. Dataset: German.

- For DiffDP, a parameter λ controls the contribution of the regularization term in the loss. We tried a range of $\lambda \in [0.5, 10]$. We use $\lambda = 1$.
- In HSIC, a parameter λ controls the importance of the HSIC term in the loss function. We tried a range $\lambda \in [10, 500]$. We use $\lambda = 100$.
- In LAFTR, the loss is composed of three terms: one that penalizes reconstruction error (L_x), one that penalizes prediction error (L_y), and one that penalizes the adversary's error when trying to obtain information about sensitive features from the representation (L_z). Three parameters A_x, A_y, A_z control the weights of each term in the loss. We use $A_x = 8$, $A_y = 4$, $A_z = 2.1$. We tried a range or $[1, 10]$ for each parameter.
- For PR, a parameter λ controls the weight of the loss term that penalizes mutual information between prediction accuracy and the sensitive attribute. We tried a range of $\lambda \in [0.01, 0.5]$. We use $\lambda = 0.06$.

In Tables 6.5, 6.4, 6.6, 6.7 we show the metrics of each trained model on each dataset. For each case, we present accuracy (acc), average precision (ap), area under the curve (auc), and the $F1$ score (f1) as performance metrics, while demographic parity (DP) and equal opportunity (EqOpp) are presented as fairness metrics. The numbers are presented as percentages. In each column, the best performer is marked in boldface.

Approximation of the input distribution. For shield synthesis, we need a distribution of the input space $\theta \in \mathcal{D}(\mathcal{G} \times \mathbb{B} \times \mathbb{C})$. In the ideal case, $\theta \in \mathcal{D}(\mathcal{G} \times \mathbb{B} \times \mathbb{C})$ is the exact joint distribution of group membership, agent recommendation and cost. However, this is unrealistic most of the time, as it assumes knowledge



(a) Time. (b) Memory.

Figure 6.2: Resource usage for shield synthesis with increasing time horizons.

of the underlying distribution and the classifier. Furthermore, the distribution of cost given by the agent may be continuous, but we assume that there is a finite set $\mathbb{C}$ of costs allowed.

For our experiments, we used a simple approach that is agnostic to the ML classifier. We assume there is a cost set of k possible values $\mathbb{C} = \{c_1, \ldots, c_k\}$ uniformly distributed in the interval $[0, 1]$, and that any recommendation is equally likely. Therefore for all $c_i \in \mathbb{C}$, $g \in \mathcal{G}$ and $r \in \mathbb{B}$, we have $\theta(g, b, c_i) = 1/4k$. This approximation is easy to compute and agnostic to the ML classifier.

Cost given by the classifier. While θ is the theoretical distribution that has to be used to synthesize the shield, at the time of deployment the classifier has to choose an output in the form of a recommendation and a cost. In this case, the natural choice is given by the last layer of the neural network. The last layer in a neural network for classification is usually a softmax layer that assigns for each label a value between 0 and 1, that we can interpret it as the "confidence value" that the classifier gives to that label being true. We use this "confidence value" as the cost.

6.5.2 Shield Synthesis Computation Times

As pointed out in Theorem 6.1, our shield synthesis algorithm has a polynomial complexity for both DP and EqOpp, and the degree of the polynomial is the number of counters required to keep track of the fairness property. For `DP` it is sufficient to track 4 counters: the number of instances appeared and accepted of each group. For `EqOpp`, we also need 4 counters for the number of instances appeared and accepted of each group, counting only those for which $z = 1$. Furthermore, we need two extra counters: one to count all instances with $z = 0$, and one to keep track of the last decision for which ground truth has not yet been revealed, for a total of 6 counters.

In Figure 6.2 we show the computation time and memory usage of our shield synthesis algorithm for a fixed problem with increasing time horizon. Figure 6.2 does not show variability, because the synthesis algorithm, as described, is deterministic.

		Q1	Median	Q3	Mean	St. Dev.	Above
DP	No Shield	0.38	0.83	1.59	1.22	1.29	42.46 %
	Static-Fair	0.18	0.42	0.74	0.46	0.31	0.00%
EqOpp	No Shield	0.67	1.76	3.62	2.76	3.01	65.06 %
	Static-Fair	0.00	0.21	0.50	0.27	0.28	0.00 %

Table 6.8: Statistic of normalized fairness for finite horizon shields.

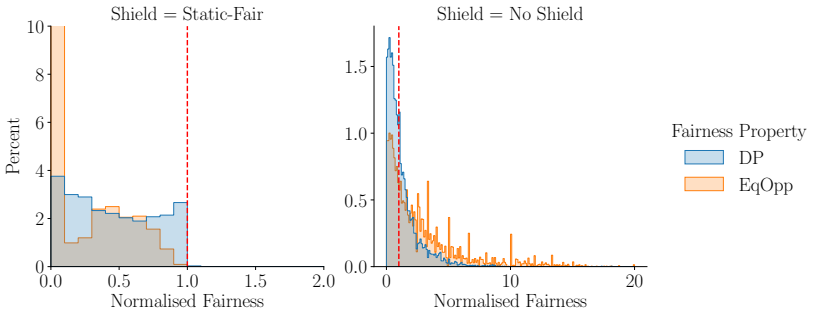


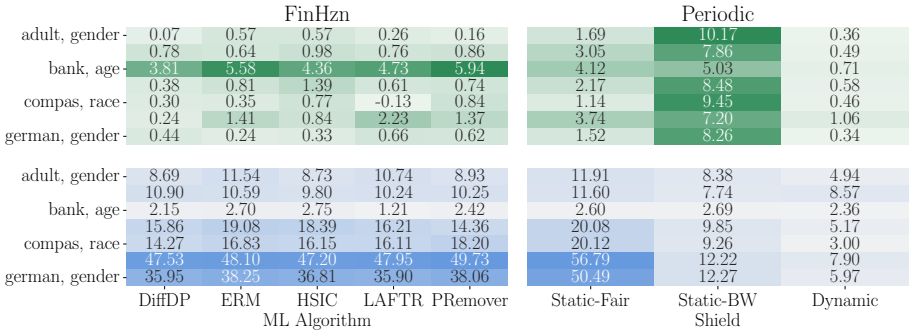
Figure 6.3: Distribution of normalized bias, i.e. Bias / κ, across all runs with (left) and without shield (right) for both demographic parity and equal opportunity.

6.5.3 Performance of Finite Horizon Shields

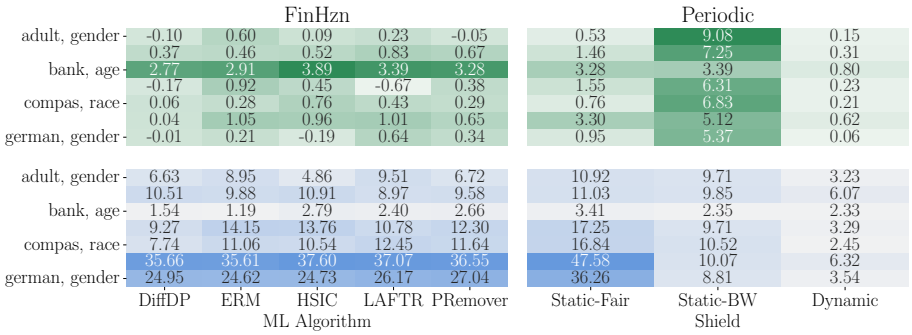
In this group of experiments, we investigate the performance of FinHzn shields on a single period. We use a time horizon of $T = 100$ for DP and $T = 75$ for EqOpp, with fairness thresholds $\kappa \in \{0.05, 0.1, 0.15, 0.20\}$. For each setting we synthesized a FinHzn shield and simulated 30 runs.

Performance in terms of fairness. In Table 6.8 we present the aggregated results of our experiments in terms of normalized fairness, i.e., the fairness value divided by the given fairness threshold. When normalized, a value smaller than 1 indicates that the algorithm is within the constraints, while a value larger than 1 indicates the algorithm is being too biased. In Figure 6.3 we illustrate the same distributions on a more graphical way, by plotting the corresponding value distributions. In the table we summarize the distribution by showing mean, median, standard deviation, Q1 (i.e., the 25% quantile) and Q3 (i.e., the 75% quantile). The last row shows the percentage of the samples that go over the fairness constraint, i.e., with a normalized fairness value larger than 1. As expected, all shielded samples are compliant with the fairness constraint. Note that most runs with shield achieve a fairness value significantly below the threshold. Another common trend is that equal opportunity is in general a harder constraint to satisfy than demographic parity in our experiments.

Performance in terms of utility loss. The utility of classification tasks is measured by classification accuracy. Interventions by the fairness shield, which occasionally convert "correct" classifications into "incorrect" ones for fairness,



(a) DP (top, green) and EqOpp (bottom, blue) with $\kappa = 0.15$.



(b) DP (top, green) and EqOpp (bottom, blue) with $\kappa = 0.2$.

Table 6.9: Comparison of utility loss (in %). Left: finite horizon shields and different ML models. Right: periodic shields only on the ERM model.

typically reduce this utility[4]. We measure *utility loss* on a given run as the difference in utility between the unshielded and shielded runs, relative to the utility of the former.

Table 6.9 (left) shows the average utility loss across all simulations for a threshold of $\kappa = 0.15$ and $\kappa = 0.2$, respectively, for finite horizon shields. We can observe that the mean utility loss is smaller when the classifier is trained to be fair, as fewer interventions are needed. In general, utility loss increases as the bias threshold κ decreases, with more pronounced differences between classifiers for smaller κ. We also observe that that most of the variability comes from the dataset rather than from the ML algorithm. These observations are also supported by Figure 6.4, which provide insight into the distribution of utility loss for each ML algorithm accross all datasets.

Finally, we compared the values of utility loss to the cost incurred by the shield. We do this to validate our approach: we compute shields by minimizing their expected cost, but our real target when deployed is to minimize the utility loss.

[4]The scary quotes in "correct" and "incorrect" are here to emphasize that we are considering correctness with respect to the ground truth of the given — potentially biased — dataset. We do not enter here in the debate of whether a classifier that is more fair and less accurate with respect to the trained data is more or less correct in a general sense.

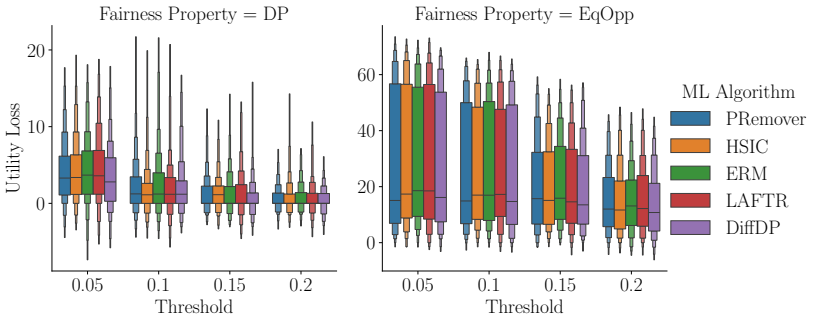


Figure 6.4: Distribution of utility loss (in %) incurred by FinHzn aggregated across all environments for DP (left) and EqOpp (right). The hight of the boxes indicate the spread of the distribution.

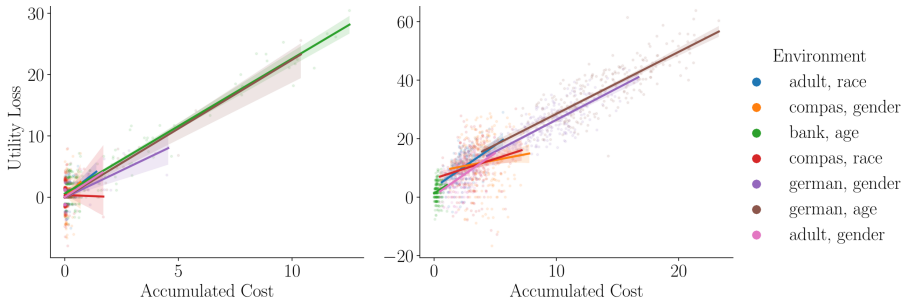


Figure 6.5: Regression plot depicting the relationship between utility loss and cost for $\kappa = 0.2$ for each dataset. The results for other values of κ are analogous. DP (left) and EqOpp (right).

In Figure 6.5 we show that indeed shield cost and utility loss are very much correlated, validating the use of one as a proxy for the other.

6.5.4 Periodic Shielding

In this group of experiments, we investigate the performance of periodic shields. We synthesized Static-Fair, Static-BW, and Dynamic shields with $T = 50$ for DP and EqOpp, with fairness thresholds $\kappa \in \{0.05, 0.1, 0.15, 0.20\}$, and simulated them for 10 periods. We compare the models' performances, with and without shielding, across 20 simulated runs.

Guaranteed fairness vs. actual fairness. Since in the periodic case we have lost the hard fairness guarantees, we first aim to determine how effective are different types of periodic shields in enforcing their corresponding fairness property. We summarize our results in Table 6.10. The "Assumption" column is a reminder of the theoretical assumption under which each of the shields works, as seen in Theorem 6.2 for Static-Fair shields, Assumption 6.1 for Static-BW shields, and Assumption 6.2 for Dynamic shields. For each type of shield and fairness property, we present how often the assumption is satisfied across all experiments, and how often the fairness property is satisfied. Since each assumption

	Assumption	φ	Assumption satisfied	Fairness satisfied
Static-Fair	$\mathtt{den}^{a,b}(\tau_i) = \mathtt{den}^{a,b}(\tau_j)$	DP	0.0 %	95.7 %
		EqOpp	0.0 %	100 %
Static-BW	$\mathtt{den}^{a}(\tau_i), \mathtt{den}^{b}(\tau_i) \geq \lceil \frac{1}{u-l} \rceil$	DP	43.8 %	83.1 %
		EqOpp	4.1 %	56.4 %
Dynamic	$\frac{1}{\mathtt{den}^{a}(\tau\tau')} + \frac{1}{\mathtt{den}^{b}(\tau\tau')} \leq \kappa + \varphi(\tau)$	DP	100 %	100 %
		EqOpp	49.8 %	100 %

Table 6.10: Comparison of different types of fairness shields.

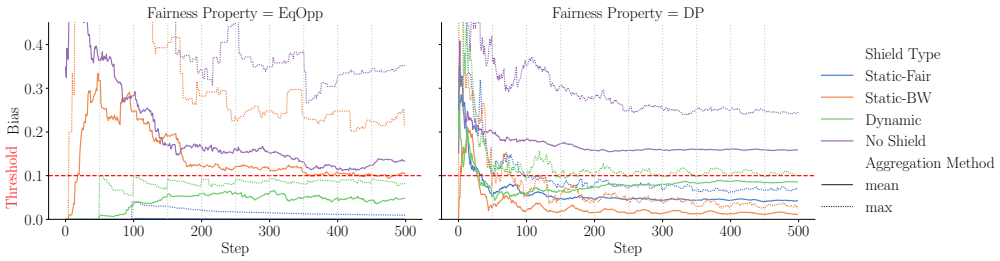
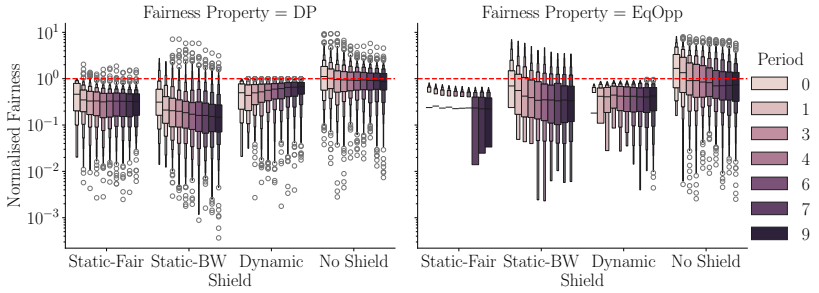


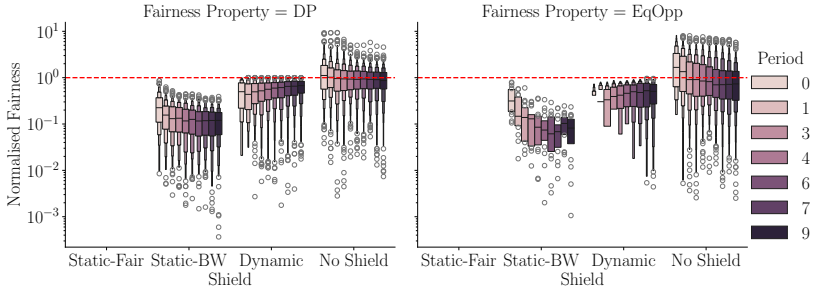
Figure 6.6: Variations of bias over time for the ERM classifier on the Adult dataset with and without periodic shielding.

has its corresponding result guaranteeing fairness (Theorems 6.2, 6.3, and 6.4), we know a priori for each instance that the fairness target is going to be satisfied at least as often as the assumption. We observe that the assumption for Static-Fair is almost never met, the assumption for Static-BW is also often violated, and the assumption for Dynamic is almost always satisfied. Nevertheless, both Static-Fair and Static-BW still perform well as heuristics, with many runs satisfying the fairness constraint. It is notable that Static-Fair offers better empirical performance than Static-BW, even though the balance assumption is almost never met. The more expensive Dynamic shields outperform both static approaches in terms of both assumption satisfaction and fairness satisfaction.

It is in principle unclear what to do in cases where the fairness guarantees are not satisfied. Static-fair shields have a defined behaviour regardless of the input. However, Static-BW and Dynamic shields, we synthesize them by modifying the conditions in the base case of the recursion (Equation (6.10)) for a condition on the new fairness target and the assumption. Concretely, the traces that do not satisfy the fairness guarantee are given cost zero if they also fail to satisfy the assumption. The concrete conditions are detailed in Equations (6.35) and (6.43). This choice ensures that traces outside the assumption do not hinder the optimization process. This translates in deployment as shields that work for ensuring fairness until the trace reaches a point where it can no longer recover from failing the corresponding assumption. If and when this point arrives, the shield "gives up" and becomes transparent until the beginning of the next period.



(a) Distribution of normalized bias for each period for all runs.



(b) Distribution of normalized bias for all runs *where the assumption is satisfied.*

Figure 6.7: Distribution of normalized bias, for each period. Each run below the red line satisfies the fairness condition, in terms of DP (left) and EqOpp (right).

Performance in terms of fairness. As an illustrative example, we show in Figure 6.6 a single run of the ERM trained model on the Adult dataset, with fairness shields synthesized for $\kappa = 0.1$, with `gender` as the sensitive feature. Recall from Table 6.3 that the statistical parity of the dataset is 0.2, so fairness enforcement will be required. The different colors indicate the different types of shields. The main observation is that only Dynamic shields show the truly periodic behaviour, where fairness is guaranteed at the end of each period by a minimal margin.

In Figure 6.7 we provide insight into the distribution of the normalized bias. The former aggregates over all runs, while the latter considers only those runs for Static-BW and Dynamic that satisfy the assumption. We can observe that Static-BW shields has a relatively high rate of violation in general (Figure 6.7a), while at the same time being overly conservative when the assumption is satisfied (Figure 6.7b). This problem does not exist with Dynamic shields, as the normalized fairness of the average run is only slightly below the threshold. In all cases we can report an improvement over the unshielded runs in terms of fairness satisfaction.

Performance in terms of utility loss. Table 6.9 (right) shows the average utility loss for the ERM model across all simulations for a threshold of $\kappa = 0.15$ and $\kappa = 0.2$, respectively, for periodic shields. In general, if the assumptions

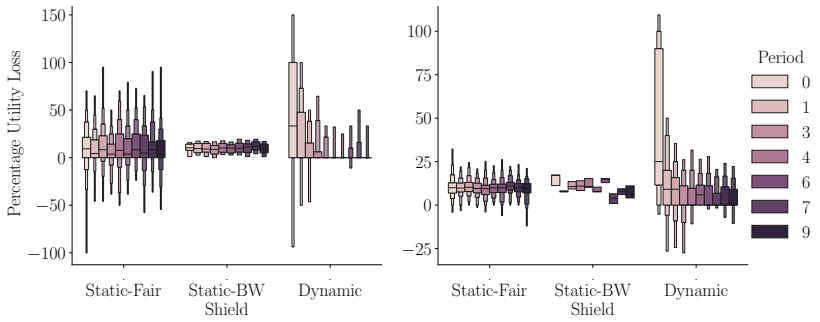


Figure 6.8: Percentage of total utility loss (in %) for each period incurred by Static-Fair, Static-Fair and Dynamic across all runs for $\kappa = 0.15$. DP (left) and EqOpp (right).

are satisfied, Dynamic shields incur the least loss and Static-BW shields incur the most, which is due to their stricter BW objectives. However, an assumption violation forces both Dynamic and Static-BW shields to go inactive incurring no additional utility loss. Therefore, the low utility loss of Static-BW shields in EqOpp can be explained by the frequent assumption violations. For demographic parity, Dynamic shields outperform Static-Fair and Static-BW shields, with Static-BW shields experiencing the highest utility loss due to their stricter BW objective. For EqOpp, the difference is less pronounced, partly because for Static-BW the frequent assumption violation forces the shield to go idle. We also observe, as expected, that the utility loss decreases when increasing κ.

We finish by studying the distribution of utility loss incurred across the different periods, as depicted in Figure 6.8. That is, for each run we normalize the utility loss per period by the total utility loss of the run. We observe that Dynamic shields incurr most of their losses in the earlier periods, a trend not observed for the other shields. Negative values indicate some rare periods in which the shield actually increases the utility of the classifier.

6.6 Discussion

6.6.1 Existence and Composability of Finite Horizon Shields

When approaching the problem of enforcing fairness properties through finite horizon shielding, there are two phenomena that can appear unintuitive in the beginning. The first is the fact that finite horizon shields always exists for DoR properties, independent on the specification threshold, time horizon or trace balance. The second is that finite horizon shields for a certain specification allow arbitrarily biased traces when concatenating them on a static manner — what we called Static-Fair shields. In this section we explore these two phenomena and shed some light on the type of edge cases where we observe them.

6.6.1.1 Existence of Finite Horizon Shields

The set of feasible solutions of the optimization problem in Equation (6.6) is nonempty for DoR properties, because the fairness-shield that always accepts or

always rejects each candidate from each group is a solution that trivially fulfils $\varphi(\tau) \leq \kappa$. In fact, in these cases, the feasible traces always satisfy $\varphi(\tau) = 0$.

Even nontrivial optimal fairness-shields may exhibit such degenerate behaviors at runtime, when the order of appearances of individuals from the two groups is excessively skewed. Consider the following example for $\varphi = \texttt{DP}$ (demographic parity). Let $T \in \mathbb{N}$ be a time horizon and $\kappa < 1/T$. As we know, given a trace τ, demographic parity is defined as

$$\texttt{DP}(\tau) = \left| \frac{n_{a1}(\tau)}{n_a(\tau)} - \frac{n_{b1}(\tau)}{n_b(\tau)} \right|.$$

Suppose at time $T-1$, all the individuals seen so far were from group a (i.e., $n_a = T-1$ and $n_b = 0$). If some of the individuals were accepted and the rest rejected, then $0 < n_{a1} < n_a$, implying $\kappa < \frac{n_{a1}}{n_a} < 1 - \kappa$. Now if the T-th individual x is from group b, n_b becomes 1, and no matter which action the shield picks, DP will be violated: If x is accepted, then $n_{b_1} = \frac{n_{b1}}{n_b} = 1$, and if x is rejected, then $n_{b1} = \frac{n_{b1}}{n_b} = 0$. In both cases, $\texttt{DP}\,(\tau) > \kappa$. Therefore, the shield must have made sure that each individual until time $T-1$, all of whom were from group a, were either accepted or rejected. Luckily, the chances of such skewness of appearance orders are rare in most applications, so that FinHzn as in Definition 6.1 exhibit effective, non-trivial behaviours in most cases, as seen from our experiments.

6.6.1.2 Counterexample Families for Static-Fair Shields Being Not Composable

We have already shown in Example 6.5 that traces can have zero bias in terms of DP, and when composed have a bias arbitrarily close to 1. While the family of counterexamples presented in Example 6.5 is quite degenerate in the sense that acceptance rates are always either 0 or 1, we present here another family of counterexamples that is less degenerate. We write these examples for demographic parity, but the same ideas can be applied to build counterexamples for any DoR property.

Let $T > 0$ and $0 < K < T/2$. The family of counterexamples will be parametrized by (T, K). For a pair (T, K) consider traces τ_1, τ_2 such that $(n_{a1}, n_a, n_{b1}, n_b)(\tau_1) = (1, K, 1, T-K)$, and $(n_{a1}, n_a, n_{b1}, n_b)(\tau_2) = (T-K-1, T-K, K-1, K)$.

In the trace τ_1, exactly one element of each group was accepted, while in the trace τ_2, all but one element of each group were accepted. The values of demographic parity are:

$$\texttt{DP}(\tau_1)_{T,K} = \left| \frac{1}{K} - \frac{1}{T-K} \right| = \frac{T-2K}{(T-K)K}. \tag{6.44}$$

$$\texttt{DP}(\tau_2)_{T,K} = \left| \frac{T-K-1}{T-K} - \frac{K-1}{K} \right| = \frac{T-2K}{(T-K)K}. \tag{6.45}$$

$$\texttt{DP}(\tau_1\tau_2)_{T,K} = \left| \frac{T-K}{T} - \frac{K}{T} \right| = \frac{T-2K}{T}. \tag{6.46}$$

These pairs of traces are not a counterexample for every pair (T, K). However, we can observe that, once fixed K, the limit when $T \to \infty$ of Equation (6.44) and Equation (6.45) is $1/K$, but the limit when $T \to \infty$ of Equation (6.46) is 1. Therefore, for every ε, we can find K large enough such that $1/K < \varepsilon/2$, and then find T large enough such that the corresponding DP values are close enough to the limit.

We now build a different family of counterexamples that show that the condition for correctness of Static-Fair shields given in Theorem 6.2 is as tight as can be.

Theorem 6.5. *For all $\kappa > 0$, there exists κ_1 and κ_2 $\kappa_1 \leq \kappa \leq \kappa_2$, such that for $i \in \{1, 2\}$, there exists t_i and traces τ_i, τ_i' that are $\lfloor \frac{t_i - 1}{2} \rfloor$-balanced such that*

$$\mathtt{DP}(\tau_i) \leq \kappa_i,\ \mathtt{DP}(\tau_i') \leq \kappa_i, \quad \textit{and} \quad \mathtt{DP}(\tau_i \tau_i') > \kappa_i.$$

Before we start the proof, let us unpack the meaning of this theorem: for any value of κ, we can find traces that are just one value off of being $(T/2)$-balanced where composability of Static-Fair shields yields a traces outside of the fairness constraint. The reason why the result is in terms of κ_1, κ_2 surrounding κ is because in the prove we have to use at some point values of κ that have a certain rational form, so we cannot prove our result for all $\kappa \in [0, 1]$, but for all κ in a dense subset of $[0, 1]$.

Proof. We prove this theorem by constructing families of counterexamples. For this proof, we use the (slightly abusive) notation that a trace is composed by its four counters, so $\tau = (n_a(\tau), n_{a1}(\tau), n_b(\tau), n_{b1}(\tau))$.

Let $t = 2T + 1$ with T even. Consider the traces $\tau_1 = (T + 1, T/2 + 1, T, T/2)$ and $\tau_2 = (T, T/2, T + 1, T/2)$. Both traces are T-balanced. Let's compute demographic parity:

$$\mathtt{DP}(\tau_1) = \frac{T/2 + 1}{T + 1} - \frac{T/2}{T} = \frac{1}{2(T + 1)}$$

$$\mathtt{DP}(\tau_2) = \frac{T/2}{T} - \frac{T/2}{T + 1} = \frac{1}{2(T + 1)}$$

$$\mathtt{DP}(\tau_1 \tau_2) = \frac{T + 1}{2T + 1} - \frac{T}{2T + 1} = \frac{1}{2T + 1}$$

It is clear that $\mathtt{DP}(\tau_1) = \mathtt{DP}(\tau_2) < \mathtt{DP}(\tau_1 \tau_2)$. Ideally, we would choose T such that $\frac{1}{2(T+1)} = \kappa$, which can be rewritten to $T = \frac{1 - 2\kappa}{2\kappa}$. However, this may not be an integer. So, given κ, we take

$$T_1 = \left\lfloor \frac{1 - 2\kappa}{2\kappa} \right\rfloor, \quad \text{and} \quad T_2 = \left\lceil \frac{1 - 2\kappa}{2\kappa} \right\rceil,$$

and define $\kappa_i = \frac{1}{2(T_i + 1)}$.

This finishes the construction for an odd t. For an even t, we show a similar construction. Let $t = 2T$. Consider the traces $\tau_1 = (T + 1, 2, T - 1, 1)$, $\tau_2 =$

$(T-1, 1, T+1, 1)$. Both traces are $(T-1)$-balanced. Let's compute demographic parity:

$$\texttt{DP}(\tau_1) = \frac{2}{T+1} - \frac{1}{T-1} = \frac{T-3}{T^2-1}$$

$$\texttt{DP}(\tau_2) = \frac{1}{T-1} - \frac{1}{T+1} = \frac{2}{T^2-1}$$

$$\texttt{DP}(\tau_1\tau_2) = \frac{T+1}{2T+1} - \frac{T}{2T+1} = \frac{1}{2T+1}.$$

This finishes the proof. □

The construction proving the theorem is for time horizons t that are $t \equiv 1 \pmod 4$. Similar constructions can be found for other congruence classes.

6.6.2 Limitations

Static vs. dynamic shielding in the periodic setting. Static shields are computationally cheaper than Dynamic shields and have no runtime overhead, making them ideal for fast decision-making applications like online ad-delivery [Ali+19]. However, they can't adjust decisions based on the actual history, leading to overly restrictive and frequent interventions—particularly in the long run. In contrast, Dynamic shields adapt to historical data, resulting in fewer interventions over time, making them suitable for applications like banking where decision-making can afford longer computation times [Liu+18].

On the assumptions in periodic shielding. All three periodic shielding approaches come with assumptions on the numbers of individuals seen from the two groups in each period. For Static-Fair shields, the assumption provides a tight sufficient condition for the fairness guarantee to be satisfied. For Static-BW and Dynamic shields, the assumptions (Assumption 6.1, 6.2) rule out "edge cases" like in Example 6.5, to give the shield enough advantage to be able to uphold fairness. We argue that in real-world scenarios and particularly for longer time periods, such edge cases are indeed rare.

On the existence of periodic shields. The existence of the optimal T-periodic shield, as defined in Definition 6.2, is left as an open question, but we conjecture it will be true. First, the same argument of Section 6.6.1.1 applies to show that $\Pi_{\texttt{fair-per}}$ is not empty because it contains a trivial "reject-all" shield. We still need to prove that there actually exists a shield that minimizes the expression in Equation (6.8).

A more interesting question is whether there exists an optimal T-periodic shield *that can be computed with finite resources*. We conjecture that such shield does not exist. Our best solution is in the form of dynamic shields, which we can only synthesize on-the-go because a complete description would require infinite memory. And even with dynamic shields, there are still some rare feasible traces that fail the fairness constraint, so dynamic shields are technically not in $\Pi_{\texttt{fair-per}}$.

On the feedback effect in sequential decision-making. In the sequential setting, decisions that seem fair from a standalone perspective may create biases in the population over time [Liu+18; D'A+20; Sun23]. This can be modeled by making the input distribution θ be a function of the trace seen so far. In this chapter, we assumed θ to remain constant, thereby leaving out such feedback effects that are inherent in sequential decision-making. We point out that our basic recursive synthesis algorithm from Section 6.3 could potentially be adapted to trace-dependent θ by modifying Equation (6.11), although a detailed extension is out of the scope of this work.

On considering unrealistic traces. As discussed in Section 6.6.1.1, our shields consider the possibility of very skewed traces. This can be seen as overly conservative, as we could safely assume in most realistic applications that such degenerate traces will not occur, and optimize cost under such assumptions. The price to pay, from a theoretical perspective, is that the probability of an input would be different depending on the trace history. While this is out of the scope of this thesis, we believe this restriction can be modelled using conditional MDPs [Bai+14].

Fairness shields with humans in the loop. In some applications, decisions are made by human experts, and AI-based systems (like classifiers) are deployed to guide the decision-making process [GC19]. In these cases, shields may not have the authority to make final decisions. But they can serve as a runtime "fairness filter," which would modify and de-bias the original outputs of the decision-maker before passing them on to the human expert. This way they can compliment the decision-making process from the fairness standpoint.

6.6.3 Related Work

Existing works on fairness address the question of how to *specify*, *design*, and *verify* AI decision makers that are fair in their decisions. From the specification standpoint, several criteria have been proposed to quantify fairness between groups [Fel+15; HPS16] and between individuals [Dwo+12]. From the design standpoint, many approaches have been developed to ensure that decision-makers are fair with respect to a given fairness objective [HPS16; Gor+19; Zaf+19; Aga+18; WBT21]. From the verification standpoint, several static [Alb+17; BZSL19; Sun+21; GBM21; MAD21; LWW23] and runtime [AV19; Hen+23a; Hen+23b; HKM23] approaches have been invented for verifying how fair or biased a given decision-maker is. Our fairness shielding combines the design and verification aspects, as shields are *verified* to be fair by *design*. Additionally, the design of our fairness shields do not require any knowledge about the underlying decision-maker, and therefore they can be used as trusted third-party intervention mechanisms to guarantee fairness of arbitrary AI-based decision makers.

Traditionally, fairness is defined using the decision-maker's output distribution. However, it has been shown that a decision-maker that is fair according to its output distribution may exhibit biases over short horizons, which could be undesirable in many situations [Ala+24]. To mitigate this issue, we adopt the recently proposed bounded-horizon fairness properties [Ala+24], which require

that decisions remain empirically fair over a given finite horizon. To the best of our knowledge, our work is the first to provide systematic algorithmic support for guaranteeing bounded-horizon fairness properties.

We consider the setting of sequential decision making, where a fairness shield needs to make decisions without knowing the inputs from the future. Similar problems has been extensively studied under the umbrella of optimal stopping problems [Shi07; Ban+18; AKK19; BY24; PV17; Käl22]. The focus of these works has been the analytical design of policies that are as close as possible to the hypothetical policy having the perfect foresight about the future. Unfortunately, statistical properties like fairness remain beyond the reach of existing algorithms from the optimal stopping literature.

Our design algorithms for fairness shields are inspired by a recent work [Can+24a], which proposed sequential decision making algorithms for the general class of finite-horizon statistical properties. They showed that the standard dynamic programming algorithm gets computationally significantly cheaper and produces the same output if the statistically indistinguishable traces are combined together. This idea is mirrored in our design algorithm for finite horizon shields as well, where traces with the same counter values remain indistinguishable.

Chapter 7

Analyzing Intentional Behaviour in Autonomous Agents

Guardeu-vos forces, bona gent, potser ens veurem un altre dia. Sabem que volíeu fer més, però, què hi farem, així és la vida: t'equivoques d'uniforme i dispares a qui més estimes; t'equivoques de remei i va i s'infecta la ferida. [1] — Guillem Gisbert, El Miquel i l'Olga tornen.

7.1 Motivation and Outline

In this chapter, we focus on explaining the decisions of autonomous agents in terms of intentional behaviour. Beyond explainability, understanding intention is also key to accountability. Since formal verification of software for autonomous agents is often infeasible, these agents may cause harm. In such instances, determining whether an agent acted intentionally, negligently, or accidentally helps clarify accountability. The study of intention thus not only strengthens explainability but also serves as an essential tool for assessing responsibility. Because we cannot predict when harm may occur, examining the agent's software after the incident is necessary to address accountability questions. Although a comprehensive liability framework for autonomous agents has yet to be developed, it is reasonable to hold manufacturers of agents that intentionally cause harm to a higher standard than those whose agents cause harm negligently or accidentally. Therefore, defining and understanding intention is crucial for establishing accountability.

Historically, symbolic AI has produced a substantial body of work focused on formally specifying and designing "rational" autonomous agents. Such agents explicitly derive decisions based on their beliefs, desires, and intentions, in the

[1] Save your strength, good people, maybe we'll see each other another day. We know you wanted to do more, but what can we do, that's life: you wear the wrong uniform and shoot the one you love the most; you use the wrong remedy, and the wound gets infected.

so-called BDI approach [Bra87; RG95]. Determining whether an autonomous agent has acted with a given intention is straightforward for BDI agents. Their intentions are explicitly encoded in their inner workings and can, therefore, be readily examined. However, the statistical nature of modern machine-learning-based agents makes interpreting their decision-making in probabilistic settings a much greater challenge, since intentions are not explicitly present in such models.

Traditionally, intention is connected to planning through either cognitive or computational reasoning. Intention is a nuanced term in legal and philosophical contexts; here, we use it in the restricted sense of the "state of the world" the agent plans towards. Whether human or machine, a rational agent with bounded resources must plan towards a goal to successfully achieve it [Bra87; CL90]. Modern machine-learned agents plan implicitly through techniques like reinforcement learning [SB18].

A sizeable portion of the literature on intention in AI relates to the internal beliefs of an agent [RG95; HKW18]. Since we do not model the internal beliefs or reasoning processes of the agent, we can only claim that an agent shows evidence of intending something. While an intentional agent would behave in this way, a random agent might also exhibit such behaviour by chance. We model uncertainty arising from diverse sources as probabilistic behaviour, aligning with modern machine-learning techniques for designing autonomous decision-makers. Therefore, our definitions are inherently quantitative. Rather than stating that an agent shows evidence of intending something, we provide concrete values quantifying the amount of evidence and the confidence level in our assessment.

Quantitative assessment of agent intentions. We consider an autonomous agent operating within a probabilistic environment. Specifically, we model the environment as a Markov Decision Process (MDP), and the agent as a policy within the MDP (recall Section 2.4). We express goals as reaching certain sets of states in an unbounded time horizon. Our aim is to analyze whether the agent's decision-making policy shows evidence of intentional behaviour towards a goal.

At the core of our methodology lie the concepts of *agency* and *intention quotient*. From a given state of the world, an agent employing the optimal policy to reach a goal would achieve it with a certain probability, which we call $P_{\max}$. Conversely, an agent using the optimal policy to avoid the same goal would reach it with a smaller probability, denoted as $P_{\min}$. We define the difference between $P_{\max}$ and $P_{\min}$ as the *agency* – or scope of agency, or extent of agency –, as it indicates how much the agent can affect the outcome in terms of reaching the goal. If the difference is one, it means the agent has complete command over the outcomes: it can ensure either reaching the goal or avoiding it with certainty. If the difference is close to zero, it means the probability of reaching the goal does not change significantly regardless of the agent's actions.

Between $P_{\min}$ and $P_{\max}$ lies the probability that the agent, with its specific policy, will reach the goal; we denote this probability as P_{ag}. The relative position of P_{ag} with respect to $P_{\min}$ and $P_{\max}$ indicates the extent of effort the agent is showing towards reaching the goal, always within the constraints

imposed by the extent of agency allowed to the agent. We call this relative position the intention quotient (IQ). Whenever P_{ag} is close to $P_{\max}$, we say that the agent shows evidence of intentional behaviour, with IQ representing the amount of evidence and the scope of agency representing the confidence level of the assessment. We use probabilistic model checking to compute these probabilities.

Using the concepts of agency and intention quotient, we can assess an agent's intention at a single state of the world. Extending to all relevant states of the world, these concepts can be directly used for a quantitative analysis of intentional behaviour in an agent.

Retrospective methodology for analyzing intentional behaviour. Motivated by the problem of accountability "after the fact", we propose a method to analyse intentional behaviour in a retrospective manner[2]. We assume a given sequence of events has occurred and we aim to assess whether an agent would show intentional behaviour toward reaching a certain goal along said sequence.

Given an agent, a goal, and a sequence of states of the world, which corresponds to a *trace*, we start by computing the intention quotients at the states in the trace, and aggregating them through a weighted average, where the weights are proportional to the agency. If the aggregated intention quotient is not sufficiently high or low, or if the confidence in the assessment (average agency along the trace) is too low, we conclude that the given trace does not provide enough evidence and that we need to analyze counterfactual scenarios.

If the evidence is not sufficient, we generate a diverse set of counterfactual traces close to the original sequence of events under study and repeat our assessment by aggregating results from all states in the counterfactual traces. This loop of counterfactual generation and intention assessment can be repeated until the confidence level of the assessment is sufficiently high or a threshold number of iterations is reached.

This retrospective method is more involved, and it is intended for use in an accountability process after harm has occurred, where the focus is not so much to understand the agent in general but rather to understand the behaviour of the agent in a concrete sequence of events leading to a harmful consequence.

Our framework is strongly inspired by methods for explainability and accountability using counterfactual analysis [WMR17; Gui24]. In computing the intention quotient, we are asking *"what could the agent do differently?"*, and in investigating counterfactual traces we are asking *"what would the agent do in different situations?"*.

Contributions. The contributions of the work presented in this chapter are:

- We present a framework for studying intentional behaviour of agents in MDPs directly from policies. Our method uses model checking to automatically relate the agent's policy to any other possible policy.

[2]For example, an autonomous driver crashes a car against a tree. *After the harm has occurred*, we study the actions of the agent leading to that harm to determine accountability.

- We propose a specific methodology for assessing evidence of intentional behaviour after a concrete sequence of events has happened, designed to be used as part of an accountability process. Furthermore, our method applies counterfactual reasoning to increase the reliability of the assessment.
- To showcase the usefulness of our retrospective method, we provide a case study in which we analyze potential intentional behaviour in the same scenario for different implementations of driving agents.

Outline. In Section 7.2, we describe the main concepts that we use to quantitatively assess intention throughout the paper and how they are grounded in previous notions. In Section 7.3, we present our specific methodology for retrospective analysis of intention, which builds counterfactuals to a reference trace and uses them to make an assessment. We report the results of a case study using our retrospective methodology in a traffic-related scenario in Section 7.4. We conclude the chapter in Section 7.5 discussing potential limitations, extensions, and relation to other work in the literature on intention analysis in AI.

Declaration of sources. This chapter is partially based and reuses material from the following source previously published by the author of this thesis:

[CC+23a] Filip Cano Córdoba, Samuel Judson, Timos Antonopoulos, Katrine Bjørner, Nicholas Shoemaker, Scott J Shapiro, Ruzica Piskac, and Bettina Könighofer. "Analyzing Intentional Behavior in Autonomous Agents under Uncertainty". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org, 2023, pp. 372–381.

7.2 Modelling Intentional Behaviour in Agents on MDPs

In this section, we give the definitions for *evidence of intentional behaviour* of policies in the presence of uncertainty. We use an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$ to model the interaction of the agent and the environment. In the following sections, we will then propose and implement a method to analyze intentional behaviour according to the definitions of this section.

7.2.1 Modelling Environment, Agents, and Intentions

We model the environment as a Markov decision process (MDP) [3] $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$, together with a finite set of atomic propositions AP and a valuation function $\mathrm{Val}\colon \mathrm{AP} \to 2^{\mathcal{S}}$. A state represents "one way the world can exist", so any information available to the agent for deciding what to do is included in the state of the MDP. The set $\mathcal{A}$ contains every possible action that can be taken by the agent. As usual, given $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$, $\mathcal{P}(s, a, s')$ represents the probability to transition to state s' from state s' when executing action a. Also, for each $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $\sum_{s' \in \mathcal{S}} \mathcal{P}(s, a, s') \in \{0, 1\}$.

[3] Recall definitions in Section 2.4.

The literals in AP indicate properties of interest of the MDP, like `goal` or `collision`, and the valuation function Val indicates, for each property, which states satisfy it. The valuation function can be extended to any Boolean formula over AP with the standard conventions as follows. For any pair of formulae $\mathcal{I}, \mathcal{J}$:

- $\text{Val}(\mathcal{I} \wedge \mathcal{J}) = Val(\mathcal{J}) \cap \text{Val}(\mathcal{J})$,
- $\text{Val}(\mathcal{I} \vee \mathcal{J}) = Val(\mathcal{I}) \cup \text{Val}(\mathcal{J})$, and
- $\text{Val}(\neg\mathcal{I}) = \mathcal{S} \setminus \text{Val}(\mathcal{I})$.

Given a Boolean formula $\mathcal{I}$, we denote the set of states where it is satisfied as $S_{\mathcal{I}} \coloneqq \text{Val}(\mathcal{I})$.

The agent is modelled by a memoryless and deterministic *policy* $\pi \colon \mathcal{S} \to \mathcal{A}$ over $\mathcal{M}$ that assigns an action to each state. In Section 7.5.3, we discuss how our method can be extended to consider strategies with non-determinism and memory.

We model potential goals as Boolean expressions over AP and express intentions as reachability properties of goals with an unbounded horizon. Given $\mathcal{I}$, a Boolean expression over AP, and a state $s \in \mathcal{S}$, we are interested in the properties of the type $\varphi = \texttt{Reach}(s, S_{\mathcal{I}})$.

7.2.2 Intention of Agents with Perfect Information

Following classic works in BDI models [RG95], an *intention* of an agent is a set of states the agent committed to reach. In our case, we model the set of states with a Boolean formula $\mathcal{I}$ over AP, whose corresponding set of states is $S_{\mathcal{I}} = \text{Val}(\mathcal{I})$. Therefore, the agent that intends $\mathcal{I}$ should act towards reaching $S_{\mathcal{I}}$ to the best of its knowledge.

Let us assume that the agent has perfect knowledge about the environment and is optimally implemented. For a formula $\mathcal{I}$ to be an intention of an agent, the agent has to implement a policy π that maximizes the probability of reaching $S_{\mathcal{I}}$. Formally, $\mathcal{I}$ is an *intention* of the agent π, if and only if for any $s \in \mathcal{S}$

$$\mathbb{P}_{\pi}(\texttt{Reach}(s, S_{\mathcal{I}})) = \mathbb{P}_{\max}(\texttt{Reach}(s, S_{\mathcal{I}})). \tag{7.1}$$

The policies considered to compute $\mathbb{P}_{\max}$ can be restricted to a set of policies Π, if there are policies that should be excluded for comparison. For example, we may only be interested in policies for comparison that satisfy certain properties like fairness or progress properties. In such cases, the right-hand side of Equation 7.1 transforms into $\mathbb{P}_{\max|\Pi}(\texttt{Reach}(s, S_{\mathcal{I}}))$.

Definition 7.1 (Intention in perfect-information settings)**.** An agent π shows *evidence of intentional behaviour* in a state s towards $\mathcal{I}$ among policies in Π if π maximizes the probability of reaching $S_{\mathcal{I}}$, i.e.,

$$\mathbb{P}_{\pi}(\texttt{Reach}(s, S_{\mathcal{I}})) = \mathbb{P}_{\max|\Pi}(\texttt{Reach}(s, S_{\mathcal{I}})).$$

Note that we do not phrase Definition 7.1 in terms of the optimal policy $\pi_{\max}$, because there may not be a unique policy that maximizes reachability probabilities.

7.2.3 Intention of Agents Under Uncertainty

The definition of intention presented earlier assumes perfect knowledge of the environment and that the agent implements an optimal policy for reaching $S_{\mathcal{I}}$. However, our goal is to analyze intention quantitatively, recognizing that agents acting intentionally do not necessarily follow the optimal policy.

An agent intending to reach $S_{\mathcal{I}}$ may deviate from the optimal policy for various reasons. We distinguish three primary categories of such deviations:

- *Imperfect training.* The agent is trained to reach $S_{\mathcal{I}}$, but training concludes before convergence to an optimal policy.
- *Trade-off among multiple goals.* The agent is trained with several goals simultaneously, with reaching $S_{\mathcal{I}}$ being only one among these objectives. Consequently, the learned policy might be suboptimal due to balancing multiple conflicting goals.
- *Imperfect environment modeling.* The agent is trained to reach $S_{\mathcal{I}}$ in an MDP $\mathcal{M}'$ that slightly differs from the actual environment model $\mathcal{M}$.

In the third scenario, discrepancies between $\mathcal{M}'$ and $\mathcal{M}$ might exist solely in transition probabilities or extend to the action and state spaces. When $\mathcal{M}'$ shares the same action and state sets as $\mathcal{M}$ but differs slightly in transition probabilities, the resulting policy may be suboptimal due to either insufficiently precise modeling of environmental uncertainty or distributional shifts occurring between training and deployment. Such discrepancies naturally emerge from attempts to model real-world uncertainties.

Alternatively, one of the models might represent an abstraction of the other. For instance, an agent could be trained in a continuous environment $\mathcal{M}'$, which must then be abstracted into a discrete model $\mathcal{M}$ for intention analysis. Although abstraction generally preserves overall agent behavior, some fine-grained details might be lost.

In all cases described, we assert that an agent still demonstrates intention through policies that, while potentially suboptimal, remain close to optimal. This observation motivates a relaxation of Definition 7.1, enabling a quantitative measure of intention under conditions of uncertainty, irrespective of the uncertainty's origin.

7.2.3.1 Single State Analysis

In order to analyze an agent π under uncertainty, we first define the *intention quotient* for a state $s \in \mathcal{S}$, which represents how close π is to the policy optimal for satisfying $\mathcal{I}$ from state s.

Definition 7.2 (Intention quotient)**.** Given an agent π at a state $s \in \mathcal{S}$ and a formula $\mathcal{I}$ over AP, the *intention quotient* is defined as follows:

$$\rho_\pi(s, \mathcal{I}) = \frac{\mathbb{P}_\pi(\texttt{Reach}(s, S_{\mathcal{I}})) - \mathbb{P}_{\min|\Pi}(\texttt{Reach}(s, S_{\mathcal{I}}))}{\mathbb{P}_{\max|\Pi}(\texttt{Reach}(s, S_{\mathcal{I}})) - \mathbb{P}_{\min|\Pi}(\texttt{Reach}(s, S_{\mathcal{I}}))}.$$

Whenever $\mathcal{I}$ is clear by context, we may drop it from the notation. In contrast to the case of perfect information, the uncertainty in the agent's knowledge and resources implies uncertainty in the assessment of intentional behaviour.

In general, the higher the value of the intention quotient $\rho_\pi(s, \mathcal{I})$, the more evidence the policy π shows of intentionally trying to satisfy $\mathcal{I}$. The lower the value of $\rho_\pi(s, \mathcal{I})$, the more evidence the policy π shows on acting without the intention to satisfy $\mathcal{I}$, although high values at a single state may be explained by other means.

An additional source of uncertainty is introduced by the agency of a state. In situations where the agent's actions have little effect on satisfying $\mathcal{I}$, there is not enough evidence to support a claim of intentional behaviour. For this reason, we take the agency into account for our assessment of intentional behaviour.

Definition 7.3 (Agency)**.** Given a state $s \in \mathcal{S}$ and a formula $\mathcal{I}$ over AP, the *agency* $\sigma(s, \mathcal{I})$ at a state s is defined as the gap between the best and the worst policy in terms of satisfying $\mathcal{I}$. Formally, it is given by

$$\sigma(s, \mathcal{I}) = \mathbb{P}_{\max|\Pi}(\texttt{Reach}(s, S_{\mathcal{I}})) - \mathbb{P}_{\min|\Pi}(\texttt{Reach}(s, S_{\mathcal{I}})). \tag{7.2}$$

7.2.3.2 Multiple-State Analysis

The concepts of agency and intention quotient apply to a single state in the MDP. However, when studying an agent in particular, we are not only interested in how the agent behaves in one state, but in many states. We extend the definition of agency by averaging the value along a set of states.

Definition 7.4 (Agency for sets of states)**.** For a set of states $S \subseteq \mathcal{S}$ and a formula $\mathcal{I}$, the *agency* of S is

$$\sigma(S, \mathcal{I}) = \frac{1}{|S|} \sum_{s \in S} \sigma(s, \mathcal{I}) \tag{7.3}$$

Since the scope of agency indicates how important is a given state in assessing the outcome of an agent's actions, we aggregate the intention quotients of the individual states using the agency as the weighting factor. This way, the weight of the decision at each state is directly proportional to the impact that an agent can have in that state towards satisfying $\mathcal{I}$.

Definition 7.5 (Intention quotient for sets of states)**.** For an agent π operating around a set of states $S \subseteq \mathcal{S}$, and a formula $\mathcal{I}$ over AP, the intention quotient $\rho_\pi(S, \mathcal{I})$ is given as the weighted average

$$\rho_\pi(S, \mathcal{I}) = \frac{1}{\sum_{s \in S} \sigma(s, \mathcal{I})} \sum_{s \in S} \sigma(s, \mathcal{I}) \rho_\pi(s, \mathcal{I}).$$

We consider two types of sets of states that are of interest from the point of view of studying intentional behaviour: balls and traces.

Balls. A ball around a set $S_{\mathcal{I}}$ represents the states that are *close* to $S_{\mathcal{I}}$, according to some distance in $\mathcal{M}$. Sometimes, instead of analysing the agent in the whole environment, we are interested in how an agent behaves in the vicinity of a set of states. For example, we may be interested in a car's behaviour towards crashing into a wall only when a wall is nearby. This may also be useful for practical reasons: instead of trying to model and understand the agent in a large environment with too many states, we can focus on a ball of a certain radius of influence, where states outside of this radius can be considered unimportant.

Traces. A trace introduces an order and a concept of time passing to a set of states, and we will especially focus on analyzing traces in the retrospective method. Traces are also useful because we know that agents will follow valid traces when deployed. In Figure 7.1 we depict the concepts of agency and intention quotient over a trace. In the hypothetical case represented in Figure 7.1, we see an agent that behaves towards satisfying a certain formula $\mathcal{I}$ most of the time, with most of the states, especially those with high agency, showing a probability close to the maximum. The only exception is the last state, where the probability is closer to the minimum at that state. Even in such a case, this would be too little and too late to exonerate the agent.

This ordering in time allows us to define a notion of commitment. Since very prominent existing theories of intention in autonomous systems take commitment as a central concept [CL90; van+20], we give here our take, defining commitment for traces in a quantitative way, using the concepts of agency and intention quotient.

Definition 7.6 (Commitment along a trace)**.** For an agent π, a trace $\tau = (s_1, \ldots, s_n)$, a threshold $\delta_B \in (0,1)$, and a threshold $\delta_I \in (0,1)$, we say that the agent is *commited* towards satisfying $\mathcal{I}$ if there exists $k \in [1,n]$ such that for all $i \geq k$,

$$\Big((\mathbb{P}_{\max|\Pi}(\texttt{Reach}(s_i, S_{\mathcal{I}})) > \delta_B) \wedge (\mathbb{P}_{\min|\Pi}(\texttt{Reach}(s_i, S_{\mathcal{I}})) < 1 - \delta_B)\Big) \rightarrow \rho_\pi(s_i, \mathcal{I}) \geq \delta_I.$$

The intuition behind this definition is that an agent shows evidence of being committed to satisfying $\mathcal{I}$ if its intention quotient exceeds a certain threshold (δ_I) whenever the agent believes that satisfying $\mathcal{I}$ is still feasible ($\mathbb{P}_{\max|\Pi}(\texttt{Reach}(s_i, S_{\mathcal{I}})) > \delta_B$), and that $\mathcal{I}$ has not yet been achieved ($\mathbb{P}_{\min|\Pi}(\texttt{Reach}(s_i, S_{\mathcal{I}})) < 1 - \delta_B$).

For this definition, δ_I is assumed to be relatively high, while δ_B is close to zero. By setting $\delta_B > 0$, we allow the agent to "give up" if fulfilling $\mathcal{I}$ becomes too unlikely, or to "focus on something else" if reaching $S_{\mathcal{I}}$ is almost certainly guaranteed. The definition can be made stricter by setting $\delta_B = 0$.

7.3 Methodology for the Retrospective Analysis of Intention

7.3.1 Setting and Problem Statement

Setting. We have a model of the environment in the form of an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$ that captures all relevant dynamics and possible interactions for an

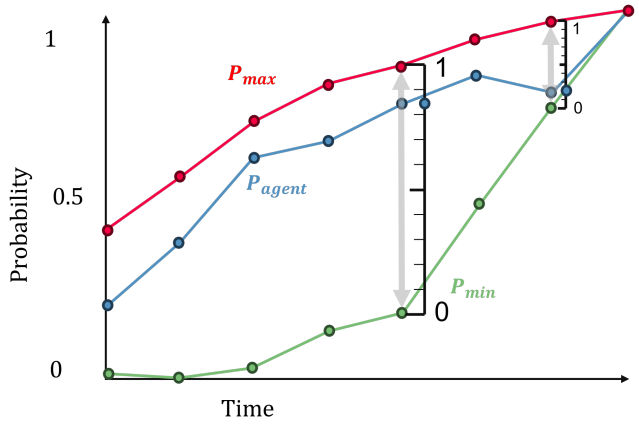


Figure 7.1: Example of the computation of agency and intention quotient. The grey arrows represent agency, while the blue dots inside the 0 to 1 ruler indicates the value of the intention quotient.

agent. We also have a concrete scenario to analyze in the form of a trace $\tau_{ref} = (s_1, \ldots, s_n)$. The trace τ_{ref} is a sequence of visited states in $\mathcal{M}$ that leads to a state in $S_{\mathcal{I}}$, i.e., $s_n \in S_{\mathcal{I}}$. The agents considered comparable are defined by a set of allowed policies Π, and the implementation of the agent under study is given in the form of a policy $\pi \in \Pi$. The underlying intentions of the agent are unknown.

Problem statement. Given this setting, we want to analyze whether there is *evidence of intentional behaviour* of the agent π towards satisfying $\mathcal{I}$ in the scenario represented by τ_{ref}, considering policies in Π.

Example 7.1. *Let us consider a scenario in which an autonomous car collides with a pedestrian crossing the road, as illustrated in Figure 7.2. To analyze to which degree the car is accountable for the accident, we are interested in whether causing harm was the intention of the car. In such an example, $\mathcal{M}$ captures all relevant information necessary to analyze the accident, like positions and velocities of car and pedestrian, car dynamics, road conditions, etc. The scenario $\tau_{ref} = (s_1, \ldots, s_n)$ is defined via the sequence of states prior to the collision. The set of states $S_{\mathcal{I}}$ represents collisions. We want to analyze whether the policy π shows evidence of intentional behaviour towards satisfying $\mathcal{I}$. To avoid unfair comparison with unrealistic policies, we define a set of policies Π that excludes unreasonably slow-moving cars (e.g., cars that stop even though there is no other road user close by).*

Since both agency and intention quotient are quantitative tools, to determine whether there is or there is not evidence of intentional behaviour, we define the following thresholds that indicate how much evidence we need to give a positive or a negative assessment.

Definition 7.7 (Evidence of intentional and non-intentional behaviour in traces)**.** Given lower and upper thresholds $0 \leq \delta_\rho^L < \delta_\rho^U \leq 1$ for intention quotient and an agency threshold $0 < \delta_\sigma < 1$, we say that there is *evidence of intentional*

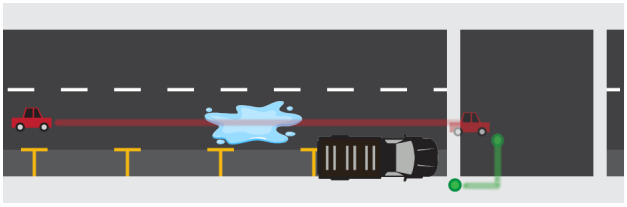


Figure 7.2: Illustration of the scenario in Example 7.1. The red line represents the trajectory of the car, the green dot represents the pedestrian and the green line the trajectory of the pedestrian. There is a water puddle in the road that makes the floor slippery and a parked truck that blocks visibility of the pedestrian.

behaviour towards satisfying $\mathcal{I}$ along a trace τ if

$$\sigma(\tau) \geq \delta_\sigma \quad \text{and} \quad \rho_\pi(\tau) \geq \delta_\rho^U.$$

We say that there is *evidence of non-intentional behaviour* towards satisfying $\mathcal{I}$ along a trace τ if

$$\sigma(\tau) \geq \delta_\sigma \quad \text{and} \quad \rho_\pi(\tau) \leq \delta_\rho^L.$$

Otherwise, i.e., in the cases that

$$\sigma(\tau) < \delta_\sigma \quad \text{or} \quad \delta_\rho^L < \rho_\pi(\tau) < \delta_\rho^U, \tag{7.4}$$

we say that we have *not enough evidence* for intentional behaviour.

The thresholds δ_ρ^L, δ_ρ^U, and δ_σ have to be defined using domain knowledge for each concrete application, and make our method adaptable to different evidence standards required for different accountability processes. For example, to convict a person of a criminal offense, it is typically required to prove the person committed the crime "beyond a reasonable doubt", while in many systems, civil litigations are resolved with the "preponderance of the evidence" standard, which is much less stringent [CS02]. The evidence thresholds can be adapted to suit different standards.

7.3.2 Evidence Augmentation through Counterfactual Generation

In this section, we propose a concrete methodology to analyze retrospectively whether there is evidence an agent acted intentionally towards satisfying $\mathcal{I}$. Our method is illustrated in Figure 7.3.

As depicted in the figure, we start the *analysis of the reference trace* τ_{ref} by computing the intention quotient $\rho_\pi(\tau_{ref})$ and the agency $\sigma(\tau_{ref})$. If $\sigma(\tau_{ref}) \geq \delta_\sigma$, we may be able to draw conclusions about intentional behaviour:

- If $\rho_\pi(\tau_{ref}) \geq \delta_\rho^U$, then we conclude that there is evidence of *intentional behaviour* towards satisfying $\mathcal{I}$.
- If $\rho_\pi(\tau_{ref}) \leq \delta_\rho^L$, then we conclude that there is evidence of *non-intentional behaviour* towards satisfying $\mathcal{I}$.

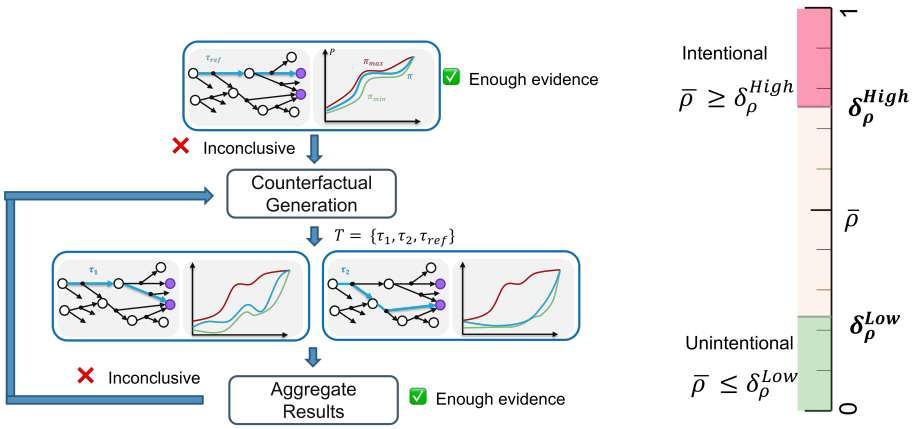


Figure 7.3: Overview of our approach for retrospective analysis of intentional behaviour.

In cases without enough agency, i.e., where $\sigma(\tau_{ref}) < \delta_\sigma$, or where the intention quotient falls between the lower and upper thresholds, i.e., $\delta_\rho^L < \rho_\pi(\tau_{ref}) < \delta_\rho^U$, we say that we do not have enough evidence to reach a conclusion. In such cases, we propose to *generate* more evidence by analyzing *counterfactual scenarios*.

A counterfactual scenario τ is a scenario close to τ_{ref} according to some distance notion. Our method generates a set of counterfactual scenarios T_{cf} and computes whether there is evidence for intentional or non-intentional behaviour for each trace $\tau \in T = T_{cf} \cup \{\tau_{ref}\}$. We fix beforehand the number of counterfactual scenarios to generate to some parameter N.

As before, we draw conclusions about intentional behaviour based on the *aggregated results* of agency $\sigma(T)$ and intention quotient $\rho_\pi(T)$. If $\sigma(T) < \delta_\sigma$ or $\delta_\rho^L < \rho_\pi(T) < \delta_\rho^U$, there is still not enough evidence for intentional or non-intentional behaviour, with $\sigma(T)$ being the agency averaged over all traces in T, and $\rho_\pi(T)$ being the average intention quotient for the set of traces in T.

In such cases, our algorithm iterates back and extends the set T_{cf} by generating N more counterfactual scenarios to be analyzed. The algorithm stops when enough evidence has been generated to draw a conclusion or when the number of generated counterfactual scenarios exceeds some user-defined limit. In the following, we discuss the generation of counterfactual scenarios in detail.

In Figure 7.3 we show this augmentation loop, where at each iteration we may stop if there is enough evidence to draw a conclusion.

7.3.3 Counterfactual Generation

To gather sufficient evidence for our assessment of intentional behaviour, we generate scenarios that serve as counterfactuals for τ_{ref}. There are various approaches to generating counterfactual traces, each requiring different levels of domain knowledge. Here, we present three alternative methods, ordered by decreasing reliance on expert knowledge and involvement. The first approach

is highly dependent on human expertise, while the last operates with minimal human intervention.

7.3.3.1 Counterfactual Generation via a Human Expert

Asking and analyzing counterfactual questions is a standard procedure in accountability processes [MB20]. Usually, such counterfactual questions are proposed by a domain expert. We transfer this concept to analyzing intentional behaviour on MDPs. The counterfactual questions posed by the expert are translated to counterfactual traces T_{cf} in the model $\mathcal{M}$.

Example 7.2. *Recall Example 7.1. Some counterfactual questions posed by an expert in the traffic scenario could be: (Q1) What if the car had driven slower? (Q2) What if the pedestrian had been visible earlier? (Q3) What if the road conditions were different? Each of Q1-Q3 translates to a counterfactual trace, which we can analyze in our framework.*

The method of generating counterfactuals using a human expert imposes a heavy burden of work on the expert. Next, we propose two methods to automatically generate counterfactuals to mitigate the need for human effort.

7.3.3.2 Counterfactual Generation on a Factored MDP

Since $\mathcal{M}$ models the interactions of the agent with its environment, $\mathcal{M}$ is typically given in form of a *factored* MDP. In factored MDPs, the state space of $\mathcal{M}$ is defined in terms of *state variables* $\mathcal{S} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_m$.

In this approach for counterfactual generation, we assume domain knowledge about which variations of state variables generate interesting counterfactual scenarios. In particular, we assume that the state variables can be partitioned into *integral*, which contain the most characteristic information about the sequence of events, and *environmental* or *peripheral* state variables, which define environmental characteristics and are fixed during the sequence of events under study[4].

Example 7.3. *In Example 7.1, integral state variables represent the position and velocity of the car and the position, velocity, and visibility status of the pedestrian. State variables that represent the position of the parked truck, the location of the water puddle and the amount of water in it are properties of the environment that stay fixed during the sequence of events, since they would change at a much slower rate, so they would be tagged as peripheral state variables. Traces with the same sequence of values for the integral state variables and different values for the peripheral state variables can effectively represent the same sequence of events in a slighlty different world.*

Studying agency and intention quotient in traces with modified values of the peripheral state variables is our way of asking counterfactual questions such as *"What would you have done if you could see the pedestrian?"*, or *"What would have happened if the road was not so slippery?"*. To generate informative counterfactuals, we are interested in traces that maintain the values of the integral variables (i.e., maintain the characteristic sequence of events), while changing

[4]Note that the convention on which variables are *integral* and *peripheral* differs from that it in [CC+23a].

some values of the peripheral variables (i.e., changing some of the environmental factors). We do not provide a more formal or concrete definition of integral and peripheral variables, because they have to be defined in each scenario from domain knowledge.

We automatically generate counterfactual traces by exploring variations of the peripheral variables. Let the state space be factored as $\mathcal{S} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_m$, where variables $\mathcal{X}_1, \ldots, \mathcal{X}_k$ are integral and $\mathcal{X}_{k+1}, \ldots, X_m$ are peripheral. For any state $s = (x_1, \ldots, x_m)$, we write its factorization into integral and peripheral variables as $s = (s^{int}||s^{per})$. Let $s^{per}_{ref} = (x_{k+1}, \ldots, x_m)$ be the value of the peripheral variables at any state of τ_{ref}. This is well-defined since the values of peripheral variables do not evolve along the trace. We define the set of counterfactual values as:

$$\mathrm{Cf}_\varepsilon(s^{per}_{ref}) = \{(y_{k+1}, \ldots, y_m) \in \mathcal{X}_{k+1} \times \cdots \times \mathcal{X}_m \ : \ \forall i,\ |x_i - y_i| < \varepsilon_i\},$$

where $\varepsilon = (\varepsilon_{k+1}, \ldots, \varepsilon_m)$ contains, for each peripheral variable, the range of variation that is still considered valid. For a given trace $\tau_{ref} = (s_1, \ldots, s_n)$, the counterfactual traces that we consider are

$$T_C(\tau_{ref}) = \{(s'_1, \ldots, s'_n) \ : \ \exists s^{per}_{cf} \in \mathrm{Cf}_\varepsilon(s^{per}_{ref}),\ \forall i = 1 \ldots n : \\ s'_i = (s^{int}_i||s^{per}_{cf}),\ (s'_1, \ldots, s'_n) \text{ is valid, and } \ s'_n \in S_{\mathcal{I}}\}.$$

To unpack this definition, a trace τ is in $T_C(\tau_{ref})$ if the following conditions are satisfied.

- At each stage of the trace τ, the value of the integral variables corresponds to the value in τ_{ref}.
- The value of the peripheral variables in τ is constant along the trace and close to that of τ_{ref}, as defined by a distance vector ε.
- The trace τ is still a valid trace of the MDP, meaning that for every pair of consecutive states in s'_i, s'_{i+1} in τ, there exists an action a such that the transition $s'_i \xrightarrow{a} s'_{i+1}$ has a non-zero probability.

Note that the search for counterfactual traces is limited to those peripheral variables $\mathcal{X}_i$ for which $\varepsilon_i > 0$, thus by setting some of the ε_i to zero, we can fix their value in the counterfactual generation process.

From T_C, we sample N traces to be used for the counterfactual analysis. For the trace selection, emphasis can be put on traces with higher scopes of agency.

7.3.3.3 Counterfactual Generation Using Distances on MDPs

This method for generating counterfactual scenarios requires to have given a distance $d\colon \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ defined over states in the MDP. Given such a distance metric d over the states, the set of counterfactual traces is given as

$$T_C(\tau_{ref}) = \{(s'_1, \ldots, s'_n) \ : \ \forall i = 1 \ldots n, \\ d(s_i, s'_i) < \eta,\ (s'_1, \ldots, s'_n) \text{ is valid, and } s_n \in S_{\mathcal{I}}\},$$

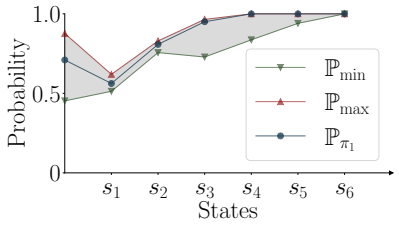


Figure 7.4: Probabilities associated with agency and intention quotient along the reference trace τ_{ref} in the experiments.

where $\eta > 0$ is a distance that represents states being 'close enough' to be compared as counterfactuals.

In case there is no distance defined in the MDP, there are bisimulation distances that are well defined intrinsically in any MDP [Son+16; Fer+06; Fer03; WDS19]. They depend on the intrinsic structure of the MDP, defined mainly by similarities in terms of the transition function. The main caveat of this approach is that distances are expensive to compute, and the explanation of why two states are assigned a given distance becomes more obscure to the user.

7.4 Experimental Validation

In this section, we showcase our retrospective method on a traffic-related scenario related to Examples 1-2, and illustrated in Figure 7.2. In this scenario, a car was driving on a road with a crosswalk. A pedestrian at the crosswalk decided to cross. Close to the crosswalk, there was a parked truck that blocked the visibility of the car. Furthermore, the previous rainy conditions generated a water puddle that made the road slippery in the region covered by the puddle. In this region, both braking and accelerating are less effective than normal, as the friction between the tires and the road is weak. While crossing, the pedestrian was hit by the car. We want to study the behaviour of the car for signs of the hit being intentional.

All experiments were executed on an Intel Core i5 CPU with 16GB of RAM running Ubuntu 20.04. We use a modified version of Tempest [Pra+21a] as our model-checking engine.

7.4.1 Model of Environment

The environment is modeled as an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$. The set of states is a triple $\mathcal{S} = \mathcal{S}^{car} \times \mathcal{S}^{ped} \times \mathcal{S}^{env}$, where $\mathcal{S}^{car}$ models the position and velocity of the car, $\mathcal{S}^{ped}$ models the position of the pedestrian, and $\mathcal{S}^{env}$ models other properties that do not change during a scenario. These properties include the slipperiness factor of the road and the existence of the truck blocking the car's view of the pedestrian.

The car's position is defined via the integers x_c and y_c with $0 \leq x_c \leq 60$ m and $3 \leq y_c \leq 13$ m. The velocity of the car is in $\{0, 1, \ldots, 5\}$ m/s. The position of the car is updated at each step, assuming a uniform motion at the current

velocity. The car has the following set of actions $\mathcal{A}$: hitting the brakes, pressing down on the accelerator, and coasting. If the car is on a non-slippery part of the road, the action of accelerating increases the velocity stochastically (by 1 or 2 m/s), braking decreases the velocity stochastically (by 1 or 2 m/s) and coasting maintains or decreases the velocity (by 1 m/s). If the car is on a slippery part of the road, the probabilistic consequences of the selected action on the velocity are different and include the possibility of no modification to the current velocity for both the actions of braking and accelerating.

The pedestrian's position is given via the integers x_p and y_p with $0 \leq x_p \leq 60$ m and $0 \leq y_p \leq 15$ m. The pedestrian can move 1 m in any direction, or not move at all. The probabilities of moving in each direction are given by a stochastic model of the pedestrian, designed in such a way that the pedestrian favours crossing the street through the crosswalk while avoiding being hit by the car. The probabilities in the pedestrian's position update can be influenced by a hesitance factor, which captures how likely it is that the pedestrian puts themselves at a hitting distance from the car. The resulting MDP consists of about 120k states and 400k transitions, so the model-checking calls have a trivial computational cost, generally under one second.

7.4.2 Analysis of a Trace

In the described environment, we are given a scenario τ_{ref} as illustrated in Figure 7.2, and an agent $\pi \colon \mathcal{S} \to \mathcal{A}$[5]. As thresholds to evaluate evidence of intention, we use $\delta_\rho^L = 0.25$, $\delta_\rho^U = 0.75$ and $\delta_\sigma = 0.5$ as reasonable arbitrary choices. In real-world scenarios, these thresholds should be adapted to concrete problem and evidence standard, and ideally agreed upon beforehand by all stakeholders.

We restrict the set of policies Π to policies that do not stop the car if no pedestrian is within a range of 15m of the car. The collision states are described by the formula

$$\mathcal{I} = (|x_p - x_c| \leq 5) \vee (|y_p - y_c| \leq 5)\,.$$

Given this setting, we analyze τ_{ref} for evidence of intentional behaviour towards reaching the set of states $S_\mathcal{I}$. Therefore, we first compute the agency and intention quotient along τ_{ref}.

Results of analysing τ_{ref}. In Figure 7.4, we show the results of the model checking calls for reaching $S_\mathcal{I}$ for states in τ_{ref}. The lower line (▽) represents $\mathbb{P}_{\min}$, the upper (△) represents $\mathbb{P}_{\max}$ and the line in the middle (●) represents $\mathbb{P}_\pi$ for every state in τ_{ref}. The shaded area, between $\mathbb{P}_{\min}$ and $\mathbb{P}_{\max}$, represents the agency at each state. The figure shows the agent is close to the line of $\mathbb{P}_{\max}$, but the agency is very small, with $\rho_\pi(\tau_{ref}) = 0.73$ and $\sigma(\tau_{ref}) = 0.18$. Since $\sigma(\tau_{ref}) < \delta_\sigma$, our method concludes that there is not enough evidence for intentional behaviour yet and moves on to the step of generating counterfactual scenarios.

[5]Both the trace and the agent are handcrafted for this experiment to illustrate our method. The agent is programmed to get to the end of the street and opportunistically hit the pedestrian if possible. The trace has the car collisiding with the pedestrian. The same analysis method would apply to different agents and any trace ending in a collision.

	sl_{init}	sl_{end}	sl_{fact}	h_{fact}	vis
Value τ_{ref}	20	45	2.5	0.5	1
Range	$[10, 30]$	$[35, 55]$	$[1, 4]$	$[0.1, 0.9]$	$\{0, 1\}$

Table 7.1: Ranges to use in counterfactual generation.

Counterfactual analysis. We generate counterfactual scenarios by exploiting domain knowledge about integral and peripheral variables of the MDP. We change the values of the following peripheral variables:

- *Slipperiness range.* The street is considered to be slippery between the positions sl_{init} and sl_{end}.
- *Slipperiness factor.* The strength of the slippery effect is measured by the slippery factor sl_{fact}, which is analogous to the inverse of the friction coefficient in classical dynamics. The effect of slipperiness is to make the acceleration and brake less effective, increasing the probability that both acceleration and brake have no effect on the speed of the car. The larger the value of sl_{fact}, the more effect, with $sl_{fact} = 1$ being the minimum value, where the road is considered to be 'not slippery at all'.
- *Hesitancy factor.* The pedestrian, in general, tends to cross the street through the crosswalk. The hesitancy factor modifies the probabilistic model of the pedestrian, to make them more or less prone to put themselves at a hitting distance from the car. In the limit, a pedestrian with hesitancy factor $h_{fact} = 0$ is a completely cautious pedestrian, that under no circumstance would put themself at a position where they could be hit by the car. On the contrary, a pedestrian with hesitancy factor $h_{fact} = 1$ completely disregards the position and velocity of the car, and would not hesitate to cross even with a fast car approaching.
- *Visibility.* In the given scenario, there is a truck blocking the visibility of the car, corresponding to $vis = 1$. In case $vis = 0$, the visibility block is eliminated.

The variables and the ranges considered for generating counterfactuals are summarized in Table 7.1.

Results of analyzing counterfactual scenarios. We build the counterfactuals in batches of $N = 5$, by sampling uniformly on the ranges described in Table 7.1. We show the results in terms of intention quotient and agency in Table 7.2. We report the averaged values and standard deviations over 5 runs. As we can see from the table, with 21 traces in T we have $\rho_\pi(T) > \delta_\rho^U = 0.75$ and $\sigma_\pi(T) > \delta_\sigma = 0.5$. Thus, our method concludes that the agent under study does present evidence of intentional behaviour to hit the pedestrian.

7.4.3 Comparative Analysis of Several Agents

In this section, we illustrate how our method can be used to compare different agents in terms of intentional behaviour. We compare three different agents π_1, π_2, π_3 in the same scenario τ_{ref}. The agent π_1 corresponds to the policy π

$\lvert T\rvert$	6	11	16	21
$\rho_\pi(T)$	0.78 ± 0.03	0.81 ± 0.02	0.83 ± 0.02	0.84 ± 0.01
$\sigma_\pi(T)$	0.33 ± 0.02	0.44 ± 0.03	0.48 ± 0.01	0.50 ± 0.01
time (s)	53 ± 16	147 ± 42	227 ± 32	318 ± 64

Table 7.2: Results of the counterfactual evaluation.

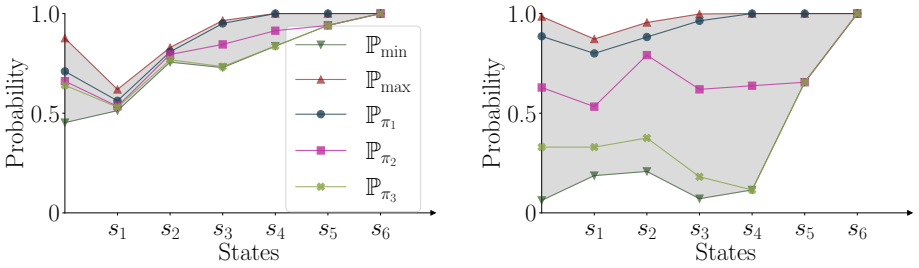


Figure 7.5: Comparison of τ_{ref} (left) with a high-agency counterfactual scenario (right).

in Section 7.4.2. The agent π_2 is designed as a reckless driver that completely disregards the position of the pedestrian, while π_3 is designed as a cautious driver.

In Figure 7.5 we show the probabilities for reaching $S_{\mathcal{I}}$ for the policies π_1, π_2, π_3 for two different traces: left for τ_{ref}, right for a counterfactual trace $\tau \in T$ with a high agency. The figure illustrates how even a single counterfactual trace can be a powerful tool for distinguishing between policies that seem impossible to differentiate with any confidence in the original trace τ_{ref}. In general, high agency values are achieved by minimizing the slippery range and factor, increasing the hesitancy of the pedestrian and eliminating the visibility block.

A second insight is illustrated in Table 7.3. In this table, for each agent π_1, π_2, π_3, we show the number of counterfactuals needed to generate enough evidence of intentional behaviour, together with the final values of the intention quotient and agency. Both π_1 and π_3 are clear-cut, but for π_2 our algorithm reaches the limit of $|T| = 100$ without finding enough evidence. In this case, the intention quotient of the agent seems to converge to a value of about 0.53, sitting in the middle of the lower and upper threshold.

Finally, in Figure 7.6, we show the values of intention quotient against the scope

	π_1	π_2	π_3
$\lvert T\rvert$	21	100	26
$\rho_\pi(T)$	0.86	0.53	0.14
$\sigma_\pi(T)$	0.52	0.64	0.50

Table 7.3: Final values of $\rho_\pi(T)$ and $\sigma_\pi(T)$ for different strategies.

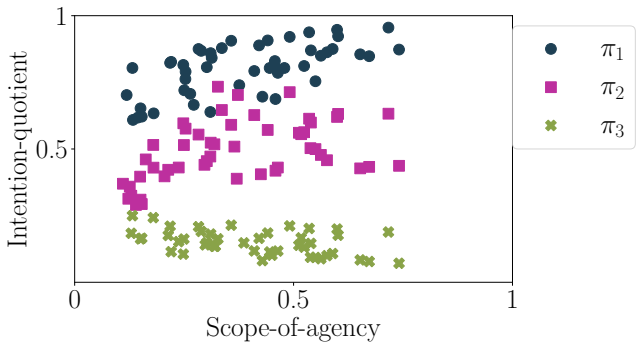


Figure 7.6: Scatter plot of intention quotient vs agency for different agents.

of agency for 100 counterfactual traces sampled from the ranges in Table 7.1. This serves as a visual representation of the same facts presented in Table 7.3, concluding that π_1 (–●–) is clearly showing evidence of intentionally hitting the pedestrian, π_2 (–■–) is showing evidence of intentionally hitting the pedestrian in a lower magnitude, which would be considered enough or not depending on the thresholds, and π_3 (–✖–) is showing clear evidence of acting without the intention of hitting the pedestrian.

The results of these experiments are in agreement with the way we designed the agents. This just serves to showcase our method. A more thorough validation would require a human-subject experiment, where real users give their subjective perception of the intention of different agents, and we measure how close their perception is to our notion. This is, however, out of the scope of this thesis.

7.5 Discussion

7.5.1 Limitations

We believe that our approach has great potential. However, there are aspects that need to be addressed to make the method applicable in challenging scenarios.

Modelling the agent and the environment. Our method requires having a *correct model of the environment* that captures everything relevant to analyze a scenario. In many cases, such models are not available. Recent work on digital twin technologies [Jon+20] and the existence of realistic simulators [Dos+17] provides optimism for more and more accurate models of agents and their environment. Our method also requires the *agent be given as a policy in an MDP*. In case we are given a different implementation, e.g., as a neural network, we would need a sample-efficient method to translate the implementation into a policy in the MDP, at least for the relevant parts of the state space.

Computational complexity. While current probabilistic model-checking engines achieve impressive performance [Bud+21], *computing exact probabilities is costly* (polynomial complexity). An alternative would be to use statistical

model checking [AP18], which is less demanding, albeit also less precise. Statistical model checking has been successfully used to validate autonomous driving modules [Bar+19].

Knowledge of the agent's beliefs. An intrinsic limitation of studying policies in MDPs is the lack of knowledge of the agent's beliefs about the world. Belief plays a fundamental role in the study of intentions: an agent that intends $S_{\mathcal{I}}$ must act believing that their acts are a good strategy to reach $S_{\mathcal{I}}$ [Bra87]. Belief is also central to the definitions of responsibility and blameworthiness in structural causal models [CH04; HKW18]. Partially for this reason, together with the uncertainties derived from a probabilistic setting, we can make claims about evidence of intentional behaviour instead of supporting stronger claims on the actual intention of the agent.

For example, an autonomous driving agent may have a faulty perception element that confuses the numbers 30 and 80 in speed limit signals. Thus, when the agent is in a low-speed area with a speed limit of 30 km/h, it accelerates to 80 km/h. This agent may show evidence of intentionally overspeeding only in low-speed areas with our definition, while an inspection of the internal belief system may show that it is actually just trying to go as fast as the speed limit. While this is an inherent limitation of our method, we still think our method is valuable for an accountability process. From the perspective of other road users, overspeeding does not happen by accident or in some failure cases, but it is rather a systematic flaw that is functionally equivalent to intentional overspeeding. By functionally equivalent we mean that the behaviour of our faulty agent and the behaviour of an intentionally harmful agent are the same.

Distinction between negligence and intentional harm. As we have described in the previous overspeeding example, our method may characterize negligent or faulty systems as intentional when they are functionally equivalent to intentionally harmful systems.

In our framework, negligence and recklessness can be expressed as mid-range intention quotients towards a harmful set of states, especially in counterfactual traces. In our running example studied in Section 7.4, an intentionally harmful driver would behave very differently when the pedestrian is far from crossing the street, waiting for them to be vulnerable, while a reckless or negligent driver would not care about the state of the pedestrian. This difference is then reflected in the values of the intention quotients.

7.5.2 Avoidance Properties

In our analysis, we have focused on reachability properties, answering questions of the type "Does the agent show evidence of intending to reach a state that satisfies $\mathcal{I}$?". A symmetric approach would be to consider avoidance properties, as defined in Equation 2.8.

With this spirit, we could rephrase the definitions of agency and intention quo-

tients from Definitions 7.3 and 7.2 as:

$$\sigma'(s,\mathcal{I}) = \mathbb{P}_{\max|\Pi}(\texttt{Avoid}(s,S_{\mathcal{I}})) - \mathbb{P}_{\min|\Pi}(\texttt{Avoid}(s,S_{\mathcal{I}})) \quad \text{and}$$
$$\rho'_\pi(s,\mathcal{I}) = \frac{\mathbb{P}_\pi(\texttt{Avoid}(s,S_{\mathcal{I}})) - \mathbb{P}_{\min|\Pi}(\texttt{Avoid}(s,S_{\mathcal{I}}))}{\mathbb{P}_{\max|\Pi}(\texttt{Avoid}(s,S_{\mathcal{I}})) - \mathbb{P}_{\min|\Pi}(\texttt{Avoid}(s,S_{\mathcal{I}}))}.$$

With the next result, we will show that the reachability and avoidance versions are very much related to one another.

Proposition 7.1. *Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P})$ be an MDP, $\pi\colon \mathcal{S} \to \mathcal{A}$ be a policy, $s \in \mathcal{S}$, and $\mathcal{I}$ a formula over* AP*. The following holds:*

- $\sigma'(s,\mathcal{I}) = \sigma(s,\mathcal{I})$*, and*
- $\rho'_\pi(s,\mathcal{I}) = 1 - \rho_\pi(s,\mathcal{I})$*.*

Proof. By the definition of avoidance properties, we know that

$$\mathbb{P}_{\max|\Pi}(\texttt{Avoid}(s,S_{\mathcal{I}})) = 1 - \mathbb{P}_{\min|\Pi}(\texttt{Reach}(s,S_{\mathcal{I}})) \quad \text{and}$$
$$\mathbb{P}_{\min|\Pi}(\texttt{Avoid}(s,S_{\mathcal{I}})) = 1 - \mathbb{P}_{\max|\Pi}(\texttt{Reach}(s,S_{\mathcal{I}})).$$

The agency result follows directly.

Similarly, for the intention quotient, we have

$$\begin{aligned}\rho'_\pi(s,\mathcal{I}) &= \frac{\mathbb{P}_\pi(\texttt{Avoid}(s,S_{\mathcal{I}})) - \mathbb{P}_{\min|\Pi}(\texttt{Avoid}(s,S_{\mathcal{I}}))}{\sigma(s,\mathcal{I})} \\ &= \frac{1 - \mathbb{P}_\pi(\texttt{Reach}(s,S_{\mathcal{I}})) - (1 - \mathbb{P}_{\max|\Pi}(\texttt{Reach}(s,S_{\mathcal{I}})))}{\sigma(s,\mathcal{I})} \\ &= \frac{\mathbb{P}_{\max|\Pi}(\texttt{Reach}(s,S_{\mathcal{I}})) - \mathbb{P}_\pi(\texttt{Reach}(s,S_{\mathcal{I}}))}{\sigma(s,\mathcal{I})}. \end{aligned} \tag{7.5}$$

On the other hand

$$\begin{aligned}1 - \rho_\pi(s,\mathcal{I}) =& \Big[\mathbb{P}_{\max}(\texttt{Reach}(s,S_{\mathcal{I}})) - \mathbb{P}_{\min|\Pi}(\texttt{Reach}(s,S_{\mathcal{I}})) - \\ & \big(\mathbb{P}_\pi(\texttt{Reach}(s,S_{\mathcal{I}})) - \mathbb{P}_{\min|\Pi}(\texttt{Reach}(s,S_{\mathcal{I}}))\big)\Big] / \sigma(s,\mathcal{I}) \\ =& \frac{\mathbb{P}_{\max|\Pi}(\texttt{Reach}(s,S_{\mathcal{I}})) - \mathbb{P}_\pi(\texttt{Reach}(s,S_{\mathcal{I}}))}{\sigma(s,\mathcal{I})}. \end{aligned} \tag{7.6}$$

The proof is concluded by observing that the expressions in Equations 7.5 and 7.6 are the same. □

7.5.3 Generalized Policies

We briefly discuss how to treat policies with memory and non-determinism. Our definitions naturally extend to non-deterministic policies with memory, although it is not evident whether the probabilities required to measure intention quotients (Definition 7.2) are easy to compute.

Computing extreme probabilities, i.e., $\mathbb{P}_{\max}$ and $\mathbb{P}_{\min}$, is equally hard for general policies, since the maximum and the minimum can be achieved with memoryless deterministic policies. If the policy has a finite amount μ of memory, $\mathbb{P}_\pi(\texttt{Reach}(s, S_{\mathcal{I}}))$ can be computed using probabilistic model checking, with a cost of μ times that of the memoryless case [BK08]. In case the non-determinism is unknown to us, to compute $\mathbb{P}_\pi(\texttt{Reach}(s, S_{\mathcal{I}}))$ we need to sample the decisions of the agent often enough to get an accurate approximation of its decision-making probabilities, making it more costly, although recent heuristics for determinization may help [Ash+20].

7.5.4 Single-Agent Setting

In our framework, all relevant parts of the environment are modeled by an MDP, and all the agency in the model is attributed to the agent, i.e., the only actor choosing actions in the MDP is the agent. We argue that this decision is reasonable to study the behaviour of an individual agent: from the perspective of an agent, it makes no difference whether the decisions of other actors are governed by a sophisticated policy or by random events in the environment, as long as the MDP model contains accurate transition probabilities. The emergence of intrinsically multi-agent phenomena, like shared intentions in cooperative settings, would require a multi-agent extension of our framework and is out of the scope of this thesis. In particular, we do not explore how to assign moral responsibility to large groups of agents (the so-called "problem of many hands" [Tho80; Poe15]). Another problem we do not explore is the existence of responsibility voids [BH11], i.e., situations in which a group of agents should be held accountable for an outcome, while at the same time, no individual agent intended that outcome.

7.5.5 Related Work

Intention in artificial intelligence. The concept of intention is a contested term in artificial intelligence. Since the early work from Bratman [Bra87], it has been used in the design of rational agents [Woo03]. A consensus on a formal definition remains, however, an open problem. In their seminal book on multiagent systems [SLB08], Shoham and Leyton-Brown call the attempt on a formal definition of intention *the road to hell*. We comment on some relevant concepts of intention in artificial intelligence and how they relate to our quantitative measure.

The work in [RG95; RG91] is the main conceptualization of agents with the belief-desire-intention models, and BDI models have also been used to model agency [Geo+98]. A good survey of the BDI literature can be found in [Woo03], and a more recent one in [DSML20]. On a more specific note, [SP11] builds an analogy between optimal BDI-based and MDP-based agents, that serves us as our basis for the definition of intention in MPD agents under perfect information.

Cohen and Levesque's work [CL90] is foundational to the concept of commitment as part of intention. In simple terms, they define intention through the notion of persistent goals. A persistent goal is a goal to which an agent remains committed until it believes either the goal is unattainable or it has been achieved. While we do not adopt their formalism, our concept of commitment

aligns with this high-level idea: an agent demonstrates commitment to a goal if its intention quotient remains persistently high from a certain point onward, and only decreases significantly when the agent believes the goal has been accomplished or is no longer feasible. Since our approach is quantitative, we translate the agent's beliefs about goal achievement or feasibility into quantitative measures, using maximum and minimum probabilities to capture these judgments. The formalism of [CL90] has been criticized for being too convoluted, and more modern approaches include [Sin92; Wob95; HW03; DHJW07; HL04; Her+17; van+20]. However, these modern approaches focus on providing a more usable formalism and do not challenge the core idea of persistent goals.

More recent contributions include [Mot+23] on a formalization of a logic for intention in probabilistic models, [Zha+23] on recognizing intentions when studying multiple agents, and [War+24] on modelling intentions as instrumental goals.

Intention in philosophy. Characterizing and understanding the concept of intention in rational agents, both humans and non-humans, is one of the fundamental problems in the philosophy of action [Pau20]. Most influential is the work of G.E.M. Anscombe [Ans57], who poses the problem of intention presenting itself in three forms: (i) intention for the future, as I intend to finish this thesis by the end of the year; (ii) intention with which someone acts, as in I am typing these words in order to have my thesis finished; and (iii) intentional action, as I am working on my thesis intentionally. Since these three forms are distinct, but we use the same concept for them, a theory of intention has to be such that it reconciles them. Much effort has been dedicated to the building of theories that explain the unity of these facets, see [Set22] for a summary of the main theories. While we get most of our inspiration in the planning theory of intention [Bra87; Bra99], it is important to note that much of the debate circles around beliefs and states of mind, which we do not model. Therefore, from a functional perspective, our concept of intention quotient is consistent with other theories [Vel07; Mel92; Dav63], since typical objections such as intending something believed to be impossible, intending something while not doing it, or intending A while believing that B is better; do not affect the functional effect of intending something. The concept of agency has also been extensively studied in this context [Lis21; Sha14; BH11; Geo+98], although most of the problematization concentrates on the relation between individual and collective agency.

There is an ongoing debate in the philosophy of mind, between those that consider that an agent's reasoning is sufficient to explain their actions [Qui69], and those who maintain that extrinsic information must be imported through a "Principle of Charity" [Dav63]. By building a model of the agent's knowledge (the MDP) to inquire about their behaviour, we are assuming the latter position. Recent work attempts to answer similar questions from the former [Jud+24b; Jud+24a].

Responsibility and accountability. The concepts of intention and agency are also fundamental as they relate to concepts in moral responsibility [BH12; Sca10]. The concept of agency is a necessary element in assigning responsibility, leading to issues when the agency is diluted among many individu-

als [Sha14; BH11]. Intention and agency are also very important in the context of accountability processes; both in criminal [Moo03; Moo10; Pau14] and civil cases [GCC06; GJ91].

Causality and blame attribution. A basic element for a complete accountability process is the study of *causality* [HP05a; HP05b], which is also a necessary condition for legal responsibility [Moo19; Moo09]. The foundational work of [CH04] introduced a quantitative notion of causality, by studying degrees of responsibility and blame. Responsibility and blame allocation have been extensively developed in the context of non-probabilistic structures (see, e.g., [Ale+17] for the characterization of complexity or [YD16] for a multi-agent framework). More recent and more closely related to our approach is the work of [BFM21b; BFM21a], studying responsibility allocation and blame attribution in Markovian models. The study of harm from a causality perspective is also gaining attention recently, with [BCH22; BCH23] studying harm from an actual causality perspective, and [RBT22] studying harm from a probabilistic perspective, heavily relying on counterfactuals. Counterfactual analysis [Lew13] is a key concept in causality [Pea09; LGZ13], used in an analogous way to our generation of counterfactual scenarios. We go one step further by relating the implementation of the agent to the best and worst implementation for reaching an intended event.

Another recent approach to blame attribution is [TSR21], which studies multi-agent Markov decision processes from a game-theoretic perspective, and [Dat+15], which builds on actual causes as a theory for accountability.

Policy-discovery methods. Since the popularization of reinforcement learning, there exist several methods for obtaining representations of a black-box agent, by studying traces of such agents. In inverse reinforcement learning (IRL) [NR00; AD21], the agent is assumed to be maximizing an unknown reward function, and the objective is to find the reward function that best explains the agent's performance over a set of traces [BJD23; Big+21]. A similar approach is imitation learning, where an agent has to learn to perform a task from successful demonstrations. The demonstrations can either be provided by a human, by an expert autonomous agent, or be the result of filtering the best traces from random execution [Hus+17]. These methods could potentially be used as a pre-processing step to apply our framework to black box agents. In any case, the obtained representations must be accurate enough before using them for any accountability process.

There is also literature on RL methods that hide their true goals or intentions, generally known as deceptive RL methods [MS17; LM23], so IRL methods could be vulnerable to deceptive RL agents.

Explainability. Explainability in machine learning has gained much traction in recent years [DVK17; LL17; DLH19; MCB20; Bai+21c] as a useful tool for both development and accountability. One of the most influential works in *explainability* of AI is [Mil19], which studies how explainability should rely on concepts from social sciences. More recently [Win+21] uses the built-in notions of desire, beliefs, and intentions to study explainability of BDI models,

relying on concepts from the sociology literature. While the main paradigm in explainable reinforcement learning is applying techniques from explainable machine learning [PV20], our analysis of intentional behaviour can be used as a method to aid the interpretability of agents operating in MDPs, using concepts from the philosophy of action [Bra87].

Chapter 8

Conclusion

S'ha acabat el bròquil. [1] — Catalan popular saying.

8.1 Future Work

There are many avenues for future endeavours that are ripe for exploring.

8.1.1 Shields for Safety.

Shielding in the deterministic setting has been recently extended for specifications in the safety fragment of LTL modulo theories [Rod+25] and with abstracted MDPs [CBG25], which offer the potential to study the delayed setting in new shielding use-cases.

Another natural extension is to develop shields for models with continuous time and states, using tools like control barrier functions [Ame+19].

Shielding is mainly thought of as a method that is agnostic to the controller. However, learning performance and safety cooperatively is an approach that has had some recent success [Cha+23], and it would be enlightening to explore how the shield can improve the training process of the agent, or how the agent can inform the shield on more efficient interventions.

From the user perspective, the shield is a sort of black box that decides on the safety of a given action to follow a certain specification. However, there is no more information to the user on why a given action may be unsafe. Self-explainable shields could include a language model layer that would explain a concrete decision in terms of potential transitions by the environment, or could abstract shields into more succinct representations like decision trees, maybe trading minimality of intervention for a more understandable shield.

[1] The broccoli is over.

8.1.2 Fairness in Bounded Horizons.

Similarly to shields for safety, shields for fairness are considered agnostic to the agent, and we want to explore how such shields can be used to improve the learning process, effectively turning our post-processing fairness intervention into an in-processing one.

In this thesis we leave open the question of whether optimal T-periodic shields exist and can be described with finite resources. We believe they do exist, but cannot be described with finite resources. However, it may be possible to still obtain T-periodic shields sacrificing some of the cost-optimality with hard fairness guarantees. Our closest solution is that of dynamic shields, but they are still limited in the sense that they cannot guarantee fairness for some traces.

As we have described them now, fairness shields operate in windows of T decisions. While this is natural in some use cases, it is unnatural in others, and we want to explore ways to eliminate this window-like constraint in future work.

Finally, our T-periodic shields guarantee fairness in the sense that the bias is smaller than a certain threshold. Typically, fairness properties are defined as the bias tending to zero as the sequence gets longer. Therefore, there are traces that satisfy fairness in the periodic sense but not in the more classical long-run average sense. Understanding these traces and modifying our shielding methods to prevent them would go a long way toward unifying the concepts of fairness for bounded, periodic, and unbounded horizons.

8.1.3 Intention Analysis

In future work, we want to extend our current analysis by considering a multitude of possibly conflicting intentions of the agent, as has been done with other intention approaches [Zha+23].

Another interesting line of work is to extend the study of intentional behaviour to multi-agent systems, in which cooperative or competitive intentions may arise, and study the emergence of responsibility voids [BH11].

We also want to study long executions, where the agent has time for reconsideration, and where it would be very helpful to use the notion of commitment that is so central in many theories of intention [CL90; van+20].

Furthermore, we want to transcend the simple toy example shown in this thesis and implement our framework to study reinforcement learning agents in challenging application areas.

8.2 Concluding Remarks

The rapid advancement of AI technologies has brought both immense opportunities and significant challenges. This thesis has explored key issues in ensuring AI systems operate safely, fairly, and transparently. By focusing on formal methods, verification techniques, and reinforcement learning safety mechanisms, we have contributed to the development of more robust AI systems that align with ethical and legal standards.

One of the central themes of this thesis has been shielding mechanisms, which provide runtime guarantees to AI systems by enforcing constraints on their behaviour. Our work on deterministic shielding in the presence of delayed observations demonstrates how real-world uncertainties can be systematically addressed to ensure safety. Similarly, our contributions to probabilistic shielding illustrate the potential of balancing safety guarantees with the need for flexible and efficient AI decision-making, particularly in applications such as autonomous valet parking.

Beyond safety, this thesis has examined fairness in AI decision-making, particularly in sequential settings. We introduced fairness shields as a mechanism for enforcing group fairness constraints over finite and periodic horizons. By formulating fairness as an optimization problem with hard fairness constraints and soft intervention costs, we developed shields that can correct biased decision-making processes while minimizing unnecessary alterations.

Transparency and accountability remain crucial for AI systems, particularly those operating in high-stakes environments. Our proposed framework for measuring intentional behaviour in reinforcement learning agents provides a novel approach to evaluating AI decision-making processes. By quantifying agency and intention quotient, we offer a methodology that aids in both explainability and accountability, enabling better assessments of AI responsibility in cases of failure or harm.

Looking ahead, the intersection of neurosymbolic AI, reinforcement learning safety, and algorithmic fairness presents exciting opportunities to further advance the field. Additionally, as regulatory landscapes evolve, the need for robust and interpretable AI systems will only grow, reinforcing the importance of the work presented in this thesis.

In conclusion, responsible deployment of AI systems is a multifaceted challenge that requires a combination of theoretical insights and practical implementations. By leveraging formal methods, we take a step towards AI systems that not only perform effectively but also uphold critical societal values. This thesis contributes to this broader goal, laying the groundwork for future advancements in trustworthy AI.

List of Publications

Publications the thesis is based on

[CC+23a] Filip Cano Córdoba, Samuel Judson, Timos Antonopoulos, Katrine Bjørner, Nicholas Shoemaker, Scott J Shapiro, Ruzica Piskac, and Bettina Könighofer. "Analyzing Intentional Behavior in Autonomous Agents under Uncertainty". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org, 2023, pp. 372–381.

[CC+23b] Filip Cano Córdoba, Alexander Palmisano, Martin Fränzle, Roderick Bloem, and Bettina Könighofer. "Safety Shielding under Delayed Observation". In: *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)* 33.1 (2023), pp. 80–85.

[Can+25a] Filip Cano, Thomas A. Henzinger, Bettina Könighofer, Konstantin Kueffner, and Kaushik Mallik. "Fairness Shields: Safeguarding against Biased Decision Makers". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2025.

Other peer-reviewed publications

[Tap+22] Martin Tappler, Filip Cano Córdoba, Bernhard K. Aichernig, and Bettina Könighofer. "Search-Based Testing of Reinforcement Learning". In: *Proceedings of the International Joint Conference of Artificial Intelligence (IJCAI)*. ijcai.org, 2022, pp. 503–510.

[Bjø+23] Katrine Bjørner, Samuel Judson, Filip Cano, Drew Goldman, Nick Shoemaker, Ruzica Piskac, and Bettina Könighofer. "Formal XAI via Syntax-Guided Synthesis". In: *Bridging the Gap Between AI and Reality (AISoLA)*. 2023.

[Ben+23] Saddek Bensalem, Panagiotis Katsaros, Dejan Nickovic, Brian Hsuan-Cheng Liao, Ricardo Ruiz Nolasco, Mohamed Abd El Salam Ahmed, Tewodros A. Beyene, Filip Cano, Antoine Delacourt, Hasan Esen, Alexandru Forrai, Weicheng He, Xiaowei Huang, Nikolaos Kekatos, Bettina Könighofer, Michael Paulitsch, Doron Peled, Matthieu Ponchant, Lev Sorokin, Son Tong, and Changshun Wu. "Continuous Engineering for Trustworthy Learning-Enabled Autonomous Systems". In: *Bridging the Gap Between AI and Reality (AISoLA)*. 2023.

[Jud+24a] Samuel Judson, Matthew Elacqua, Filip Cano, Timos Antonopoulos, Bettina Könighofer, Scott J. Shapiro, and Ruzica Piskac. "'Put the Car on the Stand': SMT-based Oracles for Investigating Decisions". In: *Proceedings of the Symposium on Computer Science and Law (CSLAW)*. ACM, 2024, pp. 73–85.

[Jud+24b] Samuel Judson, Matthew Elacqua, Filip Cano, Timos Antonopoulos, Bettina Könighofer, Scott J. Shapiro, and Ruzica Piskac. "soid: A Tool for Legal Accountability for Automated Decision Making". In: *Proceedings of the International Conference on Computer Aided Verification (CAV)*. Lecture Notes on Computer Science. Springer, 2024, pp. 233–246.

[Can+24a] Filip Cano, Thomas A. Henzinger, Bettina Könighofer, Konstantin Kueffner, and Kaushik Mallik. "Abstraction-Based Decision Making for Statistical Properties". In: *International Conference on Formal Structures for Computation and Deduction (FSCD)*. vol. 299. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, 2:1–2:17.

Preprints and technical reports

[Can+25b] Filip Cano, Fabio Rutter, Bernhard Ramsauer, Oliver Hofmann, and Bettina Könighofer. "Building Ensembles of Molecules via Deep Reinforcement Learning". In: *Preprint pending review* (2025).

[K̈+24] Bettina Könighofer, Joshua A. Kroll, Ruzica Piskac, Michael Veale, and Filip Cano Córdoba. "Accountable Software Systems (Dagstuhl Seminar 23411)". In: *Dagstuhl Reports* 13.10 (2024), pp. 24–49.

[Des+24] Jyotirmoy Deshmukh, Bettina Könighofer, Dejan Ničković, and Filip Cano. "Safety Assurance for Autonomous Mobility (Dagstuhl Seminar 24071)". In: *Dagstuhl Reports* 14.2 (2024), pp. 95–119.

Bibliography

If I have seen further it is by standing on the shoulders of Giants.
— Isaac Newton

[Ach+17] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. "Constrained policy optimization". In: *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR. 2017, pp. 22–31.

[AD21] Saurabh Arora and Prashant Doshi. "A survey of inverse reinforcement learning: Challenges, methods and progress". In: *Artificial Intelligence* 297 (2021), p. 103500.

[Aga+18] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. "A reductions approach to fair classification". In: *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR. 2018, pp. 60–69.

[AKK19] Stefan Ankirchner, Maike Klein, and Thomas Kruse. "A verification theorem for optimal stopping problems with expectation constraints". In: *Applied Mathematics & Optimization* 79 (2019), pp. 145–177.

[Ala+24] Parand A. Alamdari, Toryn Q. Klassen, Elliot Creager, and Sheila A. Mcilraith. "Remembering to Be Fair: Non-Markovian Fairness in Sequential Decision Making". In: *Proceedings of the International Conference on Machine Learning (ICML)*. Vol. 235. PMLR, 2024, pp. 906–920.

[Alb+17] Aws Albarghouthi, Loris D'Antoni, Samuel Drews, and Aditya V Nori. "Fairsquare: probabilistic verification of program fairness". In: *Proceedings of the ACM on Programming Languages (OOPSLA)* 1 (2017), pp. 1–30.

[Alb21] Aws Albarghouthi. *Introduction to Neural Network Verification*. 2021. arXiv: 2109.10317.

[Ale+17] Gadi Aleksandrowicz, Hana Chockler, Joseph Y. Halpern, and Alexander Ivrii. "The computational complexity of structure-based causality". In: *Journal of Artificial Intelligence Research* 58 (2017), pp. 431–451.

[Ali+19] Muhammad Ali, Piotr Sapiezynski, Miranda Bogen, Aleksandra Korolova, Alan Mislove, and Aaron Rieke. "Discrimination through optimization: How Facebook's Ad delivery can lead to biased outcomes". In: *Proceedings of the ACM on Human-Computer Interaction (HCI)* 3.CSCW (2019), pp. 1–30.

[Als+18] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. "Safe Reinforcement Learning via Shielding". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2018, pp. 2669–2678.

[Alt21] Eitan Altman. *Constrained Markov Decision Processes*. Routledge, 2021.

[Ame+19] Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. "Control Barrier Functions: Theory and Applications". In: *Proceedings of the European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[Ans57] Gertrude Elizabeth Margaret Anscombe. *Intention*. Harvard University Press, 1957.

[AP18] Gul Agha and Karl Palmskog. "A Survey of Statistical Model Checking". In: *ACM Transactions on Modeling and Computer Simulation* 28.1 (2018), pp. 1–39.

[Ash+20] Pranav Ashok, Mathias Jackermeier, Pushpak Jagtap, Jan Křetínský, Maximilian Weininger, and Majid Zamani. "dt-Control: decision tree learning algorithms for controller representation". In: *Proceedings of the International Conference on Hybrid Systems: Computation and Control (HSCC)*. ACM, 2020, 17:1–17:7.

[AV19] Aws Albarghouthi and Samuel Vinitsky. "Fairness-aware programming". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT)*. 2019, pp. 211–219.

[Bai+14] Christel Baier, Joachim Klein, Sascha Klüppelholz, and Steffen Märcker. "Computing Conditional Probabilities in Markovian Models Efficiently". In: *Proceedings of the Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. Springer Berlin Heidelberg, 2014, pp. 515–530.

[Bai+21a] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. "Recent Advances in Adversarial Training for Adversarial Robustness". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. Survey Track. ijcai.org, 2021, pp. 4312–4321.

[Bai+21b] Yunjun Bai, Ting Gan, Li Jiao, Bican Xia, Bai Xue, and Naijun Zhan. "Switching controller synthesis for delay hybrid systems under perturbations". In: *Proceedings of the International Conference on Hybrid Systems: Computation and Control (HSCC)*. ACM, 2021, 3:1–3:11.

[Bai+21c] Christel Baier, Clemens Dubslaff, Florian Funke, Simon Jantsch, Rupak Majumdar, Jakob Piribauer, and Robin Ziemek. "From Verification to Causality-Based Explications". In: *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*. Vol. 198. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 1:1–1:20.

[Bal92] Silvano Balemi. "Communication delays in connections of input/output discrete event processes". In: *Proceedings of the Conference on Decision and Control (CDC)*. 1992, pp. 3374–3379.

[Ban+18] Elena Bandini, Andrea Cosso, Marco Fuhrman, and Huyên Pham. "Backward SDEs for optimal control of partially observed path-dependent stochastic systems: a control randomization approach". In: *The Annals of Applied Probability* 28.3 (2018), pp. 1634–1678.

[Bar+19] Mathieu Barbier, Alessandro Renzaglia, Jean Quilbeuf, Lukas Rummelhard, Anshul Paigwar, Christian Laugier, Axel Legay, Javier Ibañez-Guzmán, and Olivier Simonin. "Validation of Perception and Decision-Making Systems for Autonomous Driving via Statistical Model Checking". In: *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 252–259.

[BCH22] Sander Beckers, Hana Chockler, and Joseph Y. Halpern. "A Causal Analysis of Harm". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 35. Curran Associates, Inc., 2022, pp. 2365–2376.

[BCH23] Sander Beckers, Hana Chockler, and Joseph Y. Halpern. "Quantifying Harm". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org, 2023, pp. 363–371.

[Bee+24] Maurice H ter Beek, Rod Chapman, Rance Cleaveland, Hubert Garavel, Rong Gu, Ivo ter Horst, Jeroen JA Keiren, Thierry Lecomte, Michael Leuschel, Kristin Yvonne Rozier, et al. "Formal methods in industry". In: *Formal Aspects of Computing* 37.1 (2024), pp. 1–38.

[Beh+07] Gerd Behrmann, Agnès Cougnard, Alexandre David, Emmanuel Fleury, Kim Guldstrand Larsen, and Didier Lime. "UPPAAL-Tiga: time for playing games!" In: *Proceedings of the International Conference on Computer Aided Verification (CAV)*. Vol. 4590. Lecture Notes in Computer Science. Springer, 2007, pp. 121–125.

[Ben+23] Saddek Bensalem, Panagiotis Katsaros, Dejan Nickovic, Brian Hsuan-Cheng Liao, Ricardo Ruiz Nolasco, Mohamed Abd El Salam Ahmed, Tewodros A. Beyene, Filip Cano, Antoine Delacourt, Hasan Esen, Alexandru Forrai, Weicheng He, Xiaowei Huang, Nikolaos Kekatos, Bettina Könighofer, Michael Paulitsch, Doron Peled, Matthieu

Ponchant, Lev Sorokin, Son Tong, and Changshun Wu. "Continuous Engineering for Trustworthy Learning-Enabled Autonomous Systems". In: *Bridging the Gap Between AI and Reality (AISoLA)*. 2023.

[Ber+08] Dietmar Berwanger, Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Sangram Raje. "Strategy Construction for Parity Games with Imperfect Information". In: *Proceedings of the International Conference on Concurrency Theory (CONCUR)*. Vol. 5201. Lecture Notes in Computer Science. 2008, pp. 325–339.

[Ber+17] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. *A convex framework for fair regression*. 2017. arXiv: 1706.02409.

[Ber+21] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. "Fairness in criminal justice risk assessments: The state of the art". In: *Sociological Methods & Research* 50.1 (2021), pp. 3–44.

[BFM21a] Christel Baier, Florian Funke, and Rupak Majumdar. "A Game-Theoretic Account of Responsibility Allocation". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org, 2021, pp. 1773–1779.

[BFM21b] Christel Baier, Florian Funke, and Rupak Majumdar. "Responsibility Attribution in Parameterized Markovian Models". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2021, pp. 11734–11743.

[BH11] Matthew Braham and Martin van Hees. "Responsibility voids". In: *The Philosophical Quarterly* 61.242 (2011), pp. 6–15.

[BH12] Matthew Braham and Martin van Hees. "An Anatomy of Moral Responsibility". In: *Mind* 121.483 (2012), pp. 601–634.

[BHN23] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning Limitations and Opportunities*. MIT Press, 2023.

[Big+21] Ariyan Bighashdel, Panagiotis Meletis, Pavol Jancura, and Gijs Dubbelman. "Deep adaptive multi-intention inverse reinforcement learning". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*. Springer, 2021, pp. 206–221.

[BIP88] Michael E. Bratman, David J. Israel, and Martha E. Pollack. "Plans and resource-bounded practical reasoning". In: *Computational Intelligence* 4 (1988), pp. 349–355.

[BJD23] Ariyan Bighashdel, Pavol Jancura, and Gijs Dubbelman. "Model-free inverse reinforcement learning with multi-intention, unlabeled, and overlapping demonstrations". In: *Machine Learning* 112.7 (2023), pp. 2263–2296.

[Bjø+23] Katrine Bjørner, Samuel Judson, Filip Cano, Drew Goldman, Nick Shoemaker, Ruzica Piskac, and Bettina Könighofer. "Formal XAI via Syntax-Guided Synthesis". In: *Bridging the Gap Between AI and Reality (AISoLA)*. 2023.

[BK08] Christel Baier and Joost-Pieter Katoen. "Principles of Model Checking". In: MIT Press, 2008.

[BK96] Barry Becker and Ronny Kohavi. *Adult*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20. 1996.

[Blo+15] Roderick Bloem, Bettina Könighofer, Robert Könighofer, and Chao Wang. "Shield Synthesis: - Runtime Enforcement for Reactive Systems". In: *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. Springer, 2015, pp. 533–548.

[BLS24] Asger Horn Brorholt, Kim Guldstrand Larsen, and Christian Schilling. *Compositional Shielding and Reinforcement Learning for Multi-Agent Systems*. 2024. arXiv: 2410.10460.

[Blu+18] Avrim Blum, Suriya Gunasekar, Thodoris Lykouris, and Nati Srebro. "On preserving non-discrimination when combining expert advice". In: *Advances in Neural Information Processing Systems (NeurIPS)* 31 (2018).

[Bou+19] Maxime Bouton, Jesper Karlsson, Alireza Nakhaei, Kikuo Fujimura, Mykel J. Kochenderfer, and Jana Tumova. *Reinforcement Learning with Probabilistic Guarantees for Autonomous Driving*. 2019. arXiv: 1904.07189.

[Bra87] Michael E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, 1987.

[Bra99] Michael E. Bratman. *Faces of Intention: Selected Essays on Intention and Agency*. Cambridge Studies in Philosophy. Cambridge University Press, 1999.

[Bud+21] Carlos E. Budde, Arnd Hartmanns, Michaela Klauck, Jan Křetínskỳ, David Parker, Tim Quatmann, Andrea Turrini, and Zhen Zhang. "On Correctness, Precision, and Performance in Quantitative Verification". In: *Proceedings of the International Symposium on Leveraging Applications of Formal Methods (ISoLA)*. Springer, 2021, pp. 216–241.

[But01] Giorgio Buttazzo. "Artificial consciousness: Utopia or real possibility?" In: *Computer* 34.7 (2001), pp. 24–30.

[BY24] Erhan Bayraktar and Song Yao. "Optimal stopping with expectation constraints". In: *The Annals of Applied Probability* 34.1B (2024), pp. 917–959.

[BZSL19] Osbert Bastani, Xin Zhang, and Armando Solar-Lezama. "Probabilistic verification of fairness properties via concentration". In: *Proceedings of the ACM on Programming Languages (OOPSLA)* 3 (2019), pp. 1–27.

[Can+24a] Filip Cano, Thomas A. Henzinger, Bettina Könighofer, Konstantin Kueffner, and Kaushik Mallik. "Abstraction-Based Decision Making for Statistical Properties". In: *International Conference on Formal Structures for Computation and Deduction (FSCD)*. Vol. 299. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, 2:1–2:17.

[Can+24b] Filip Cano, Thomas A. Henzinger, Bettina Könighofer, Konstantin Kueffner, and Kaushik Mallik. *Fairness Shields: Safeguarding against Biased Decision Makers (extended version)*. 2024. arXiv: 2412.11994.

[Can+25a] Filip Cano, Thomas A. Henzinger, Bettina Könighofer, Konstantin Kueffner, and Kaushik Mallik. "Fairness Shields: Safeguarding against Biased Decision Makers". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2025.

[Can+25b] Filip Cano, Fabio Rutter, Bernhard Ramsauer, Oliver Hofmann, and Bettina Könighofer. "Building Ensembles of Molecules via Deep Reinforcement Learning". In: *Preprint pending review* (2025).

[Car+23] Steven Carr, Nils Jansen, Sebastian Junges, and Ufuk Topcu. "Safe reinforcement learning via shielding under partial observability". In: AAAI Press, 2023. isbn: 978-1-57735-880-0.

[CBG25] Edwin Hamel-De le Court, Francesco Belardinelli, and Alexander W. Goodall. "Probabilistic Shielding for Safe Reinforcement Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2025.

[CC+23a] Filip Cano Córdoba, Samuel Judson, Timos Antonopoulos, Katrine Bjørner, Nicholas Shoemaker, Scott J Shapiro, Ruzica Piskac, and Bettina Könighofer. "Analyzing Intentional Behavior in Autonomous Agents under Uncertainty". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org, 2023, pp. 372–381.

[CC+23b] Filip Cano Córdoba, Alexander Palmisano, Martin Fränzle, Roderick Bloem, and Bettina Könighofer. "Safety Shielding under Delayed Observation". In: *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)* 33.1 (2023), pp. 80–85.

[CD+17] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. "Algorithmic decision making and the cost of fairness". In: *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2017, pp. 797–806.

[CH04] Hana Chockler and Joseph Y. Halpern. "Responsibility and Blame: A Structural-Model Approach". In: *Journal of Artificial Intelligence Research* 22 (2004), pp. 93–115.

[CH20] Simon Caton and Christian Haas. "Fairness in machine learning: A survey". In: *ACM Computing Surveys* (2020).

[Cha+23] Krishnendu Chatterjee, Thomas A Henzinger, Mathias Lechner, and Đorđe Žikelić. "A learner-verifier framework for neural network controllers and certificates of stochastic systems". In: *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. Springer. 2023, pp. 3–25.

[Che+16] M. Chen, M. Fränzle, Y. Li, P. Mosaad, and N. Zhan. "Validated Simulation-Based Verification of Delayed Differential Dynamics". In: *Proceedings of the International Symposium on Formal Methods (FM)*. Vol. 9995. Lecture Notes in Computer Science. 2016, pp. 137–154.

[Che+18] Mingshuai Chen, Martin Fränzle, Yangjia Li, Peter Nazier Mosaad, and Naijun Zhan. "What's to Come is Still Unsure - Synthesizing Controllers Resilient to Delayed Interaction". In: *Proceedings of the International Symposium on Automated Technology for Verification and Analysis (ATVA)*. Springer, 2018, pp. 56–74.

[Che+19] Richard Cheng, Gábor Orosz, Richard M. Murray, and Joel W. Burdick. "End-to-End Safe Reinforcement Learning through Barrier Functions for Safety-Critical Continuous Control Tasks". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* 33.01 (2019), pp. 3387–3395.

[Che+20] Yifang Chen, Alex Cuellar, Haipeng Luo, Jignesh Modi, Heramb Nemlekar, and Stefanos Nikolaidis. "Fair contextual multi-armed bandits: Theory and experiments". In: *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. PMLR. 2020, pp. 181–190.

[Che+21] Mingshuai Chen, Martin Fränzle, Yangjia Li, Peter N. Mosaad, and Naijun Zhan. "Indecision and delays are the parents of failure—taming them algorithmically by synthesizing delay-resilient control". In: *Acta Informatica* 58.5 (2021), 497–528.

[CL90] Philip R. Cohen and Hector J. Levesque. "Intention is choice with commitment". In: *Artificial Intelligence* 42.2 (1990), pp. 213–261.

[CM13] Antonio Chella and Riccardo Manzotti. *Artificial consciousness*. Andrews UK Limited, 2013.

[CM21] Ching-Yao Chuang and Youssef Mroueh. "Fair Mixup: Fairness via Interpolation". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net, 2021.

[Com19] European Commission. *Ethics guidelines for trustworthy AI*. Publications Office, 2019. DOI: doi/10.2759/346720.

[Cre+07] Tanya L. Crenshaw, Elsa L. Gunter, Craig L. Robinson, Lui Sha, and P. R. Kumar. "The Simplex Reference Model: Limiting Fault-Propagation Due to Unreliable Components in Cyber-Physical System Architectures". In: *Proceedings of the International Real-Time Systems Symposium (RTSS)*. IEEE, 2007, pp. 400–412.

[CS02] Kevin M Clermont and Emily Sherwin. "A comparative view of standards of proof". In: *The American Journal of Comparative Law* 50 (2002), p. 243.

[CŽ13] Toon Calders and Indrė Žliobaitė. "Why unbiased computational processes can lead to discriminative decision procedures". In: *Discrimination and Privacy in the Information Society: Data mining and profiling in large databases* (2013), pp. 43–57.

[D'A+20] Alexander D'Amour, Hansa Srinivasan, James Atwood, Pallavi Baljekar, David Sculley, and Yoni Halpern. "Fairness is not static: deeper understanding of long term fairness via simulation studies". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT)*. 2020, pp. 525–534.

[Dat+15] Anupam Datta, Deepak Garg, Dilsun Kaynar, Divya Sharma, and Arunesh Sinha. "Program Actions as Actual Causes: A Building Block for Accountability". In: *Proceedings of the Computer Security Foundations Symposium (CSF)*. IEEE. 2015, pp. 261–275.

[Dav+13] Alexandre David, Kim G. Larsen, Marius Mikucionis, Omer Nguena-Timo, and Antoine Rollet. "Remote Testing of Timed Specifications". In: *Proceedings of the International Conference on Testing Software and Systems (ICTSS)*. Vol. 8254. Lecture Notes in Computer Science. Springer, 2013, pp. 65–81.

[Dav63] Donald Davidson. "Actions, Reasons, and Causes". In: *The Journal of Philosophy* 60.23 (1963), pp. 685–700.

[Deb19] Jan Debille. *Good digital twins don't lie*. Visited on 31/01/2025. 2019. url: `https://blogs.sw.siemens.com/simcenter/good-digital-twins-dont-lie/`.

[Des+24] Jyotirmoy Deshmukh, Bettina Könighofer, Dejan Ničković, and Filip Cano. "Safety Assurance for Autonomous Mobility (Dagstuhl Seminar 24071)". In: *Dagstuhl Reports* 14.2 (2024), pp. 95–119.

[DF18] Julia Dressel and Hany Farid. "The accuracy, fairness, and limits of predicting recidivism". In: *Science advances* 4.1 (2018), eaao5580.

[DG+19] Giuseppe De Giacomo, Luca Iocchi, Marco Favorito, and Fabio Patrizi. "Foundations for restraining bolts: Reinforcement learning with LTL_f/LDL_f restraining specifications". In: *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*. Vol. 29. AAAI Press, 2019, pp. 128–136.

[DG+20] Giuseppe De Giacomo, Luca Iocchi, Marco Favorito, and Fabio Patrizi. "Restraining bolts for reinforcement learning agents". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. Vol. 34. 09. 2020, pp. 13659–13662.

[DHJW07] Wiebe van Der Hoek, Wojciech Jamroga, and Michael Wooldridge. "Towards a theory of intention revision". In: *Synthese* 155 (2007), pp. 265–290.

[DI19] Cynthia Dwork and Christina Ilvento. "Fairness Under Composition". In: 2019.

[DLH19] Mengnan Du, Ninghao Liu, and Xia Hu. "Techniques for Interpretable Machine Learning". In: *Communications of the ACM* 63.1 (2019), pp. 68–77.

[Dos+17] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the Conference on Robot Learning (CoRL)*. PMLR, 2017.

[DSML20] Lavindra De Silva, Felipe Rech Meneguzzi, and Brian Logan. "BDI agent architectures: A survey". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org, 2020, pp. 4914–4921.

[Dur19] Rick Durrett. *Probability: theory and examples*. Vol. 49. Cambridge university press, 2019.

[DVK17] Finale Doshi-Velez and Been Kim. *Towards a Rigorous Science of Interpretable Machine Learning*. 2017. arXiv: 1702.08608.

[Dwo+12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. "Fairness through awareness". In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS)*. ACM, 2012, 214–226.

[Els+21] Ingy Elsayed-Aly, Suda Bharadwaj, Christopher Amato, Rüdiger Ehlers, Ufuk Topcu, and Lu Feng. "Safe Multi-Agent Reinforcement Learning via Shielding". In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. ACM, 2021, pp. 483–491.

[Elz+19] Hadi Elzayn, Shahin Jabbari, Christopher Jung, Michael Kearns, Seth Neel, Aaron Roth, and Zachary Schutzman. "Fair algorithms for learning in allocation problems". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT)*. 2019, pp. 170–179.

[Eur21] European Commission. *Proposal for a Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts*. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206. Accessed: 2024-08-01. 2021.

[Fel+15] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. "Certifying and removing disparate impact". In: *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*. 2015, pp. 259–268.

[Fer03] Norman Francis Ferns. "Metrics for Markov Decision Processes". In: (2003).

[Fer+06] Norm Ferns, Pablo Samuel Castro, Doina Precup, and Prakash Panangaden. "Methods for Computing State Similarity in Markov Decision Processes". In: *Proceedings Conference in Uncertainty in Artificial Intelligence (UAI)*. 2006.

[Fis+21] Michael Fisher, Viviana Mascardi, Kristin Yvonne Rozier, Bernd-Holger Schlingloff, Michael Winikoff, and Neil Yorke-Smith. "Towards a framework for certification of reliable autonomous systems". In: *Autonomous Agents and Multi-Agent Systems* 35 (2021), pp. 1–65.

[FJW11] Joan Feigenbaum, Aaron D. Jaggard, and Rebecca N. Wright. "Towards a Formal Model of Accountability". In: *Proceedings of the New Security Paradigms Workshop (NSPW)*. 2011, pp. 45–56.

[FJW20] Joan Feigenbaum, Aaron D. Jaggard, and Rebecca N. Wright. "Accountability in Computing: Concepts and Mechanisms". In: *Foundations and Trends in Privacy and Security* 2.4 (2020), pp. 247–399.

[FP19] Yliès Falcone and Srinivas Pinisetty. "On the Runtime Enforcement of Timed Properties". In: *Proceedings of the International Conference on Runtime Verification (RV)*. Vol. 11757. Lecture Notes on Computer Science. Springer, 2019, pp. 48–69.

[GBA21] Ather Gattami, Qinbo Bai, and Vaneet Aggarwal. "Reinforcement Learning for Constrained Markov Decision Processes". In: *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR. 2021, pp. 2656–2664.

[GBM21] Bishwamittra Ghosh, Debabrota Basu, and Kuldeep S Meel. "Justicia: A stochastic SAT approach to formally verify fairness". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. Vol. 35. 2021, pp. 7554–7563.

[GC19] Ben Green and Yiling Chen. "The principles and limits of algorithm-in-the-loop decision making". In: *Proceedings of the ACM on Human-Computer Interaction (HCI)* 3.CSCW (2019), pp. 1–24.

[GCC06] Peter Z. Grossman, Reed W. Cearley, and Daniel H. Cole. "Uncertainty, Insurance and the Learned Hand Formula". In: *Law, Probability and Risk* 5.1 (2006), pp. 1–18.

[Ge+21] Yingqiang Ge, Shuchang Liu, Ruoyuan Gao, Yikun Xian, Yunqi Li, Xiangyu Zhao, Changhua Pei, Fei Sun, Junfeng Ge, Wenwu Ou, et al. "Towards long-term fairness in recommendation". In: *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*. ACM, 2021, pp. 445–453.

[Geo+98] Michael Georgeff, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. "The belief-desire-intention model of agency". In: *Proceedings of the International Workshop on Agent Theories, Architectures, and Languages (ATAL)*. Springer. 1998, pp. 1–10.

[Gia+21] Mirco Giacobbe, Mohammadhosein Hasanbeig, Daniel Kroening, and Hjalmar Wijk. "Shielding Atari Games with Bounded Prescience". In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. ACM, 2021, pp. 1507–1509.

[GJ91] Thomas C. Galligan Jr. "Strict Liability in Action: The Truncated Learned Hand Formula". In: *Lousiana Law Review* 52 (1991), p. 323.

[GK21] Swati Gupta and Vijay Kamble. "Individual fairness in hindsight". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 6386–6420.

[Gor+19] Paula Gordaliza, Eustasio Del Barrio, Gamboa Fabrice, and Jean-Michel Loubes. "Obtaining fairness using optimal transport theory". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2357–2365.

[Gra+22] Riccardo Grazzi, Arya Akhavan, John I.F. Falk, Leonardo Cella, and Massimiliano Pontil. "Group meritocratic fairness in linear contextual bandits". In: *Advances in Neural Information Processing Systems (NeurIPS)* 35 (2022), pp. 24392–24404.

[Gro+22a] Timo P. Gros, Holger Hermanns, Jörg Hoffmann, Michaela Klauck, Maximilian A. Köhl, and Verena Wolf. "MoGym: Using Formal Models for Training and Verifying Decision-making Agents". In: *Proceedings of the International Conference on Computer Aided Verification (CAV)*. Ed. by Sharon Shoham and Yakir Vizel. Springer International Publishing, 2022, pp. 430–443.

[Gro+22b] Dennis Gross, Nils Jansen, Sebastian Junges, and Guillermo A. Pérez. "COOL-MC: A Comprehensive Tool for Reinforcement Learning and Model Checking". In: *Proceedings of the International Symposium on Dependable Software Engineering. Theories, Tools, and Applications (SETTA)*. Vol. 13649. Lecture Notes in Computer Science. Springer, 2022, pp. 41–49.

[GSS15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2015.

[Gui24] Riccardo Guidotti. "Counterfactual explanations and how to find them: literature review and benchmarking". In: *Data Mining and Knowledge Discovery* 38.5 (2024), pp. 2770–2824.

[Han+24] Xiaotian Han, Jianfeng Chi, Yu Chen, Qifan Wang, Han Zhao, Na Zou, and Xia Hu. "FFB: A Fair Fairness Benchmark for In-Processing Group Fairness Methods". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2024.

[Har09] John Harrison. *Handbook of practical logic and automated reasoning*. Cambridge University Press, 2009.

[He+23] Xiangkun He, Jingda Wu, Zhiyu Huang, Zhongxu Hu, Jun Wang, Alberto Sangiovanni-Vincentelli, and Chen Lv. "Fear-Neuro-Inspired Reinforcement Learning for Safe Autonomous Driving". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023), pp. 1–13.

[Hen+22] Christian Hensel, Sebastian Junges, Joost-Pieter Katoen, Tim Quatmann, and Matthias Volk. "The probabilistic model checker Storm". In: *International Journal on Software Tools for Technology Transfer* 24.4 (2022), pp. 589–610.

[Hen+23a] Thomas A. Henzinger, Mahyar Karimi, Konstantin Kueffner, and Kaushik Mallik. "Monitoring Algorithmic Fairness". In: *Proceedings of the International Conference on Computer Aided Verification (CAV)*. Springer-Verlag, 2023, 358–382.

[Hen+23b] Thomas A. Henzinger, Mahyar Karimi, Konstantin Kueffner, and Kaushik Mallik. "Runtime Monitoring of Dynamic Fairness Properties". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT)*. ACM, 2023, pp. 604–614.

[Her+17] Andreas Herzig, Emiliano Lorini, Laurent Perrussel, and Zhanhao Xiao. "BDI logics for BDI architectures: old problems, new perspectives". In: *KI-Künstliche Intelligenz* 31 (2017), pp. 73–83.

[HFM17] Zhenqi Huang, Chuchu Fan, and Sayan Mitra. "Bounded invariant verification for time-delayed nonlinear networked dynamical systems". In: *Nonlinear Analysis: Hybrid Systems* 23 (2017), pp. 211–229.

[HH14] Arnd Hartmanns and Holger Hermanns. "The Modest Toolset: An Integrated Environment for Quantitative Modelling and Verification". In: *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. Vol. 8413. Lecture Notes in Computer Science. Springer, 2014, pp. 593–598.

[HKM23] Thomas A. Henzinger, Konstantin Kueffner, and Kaushik Mallik. "Monitoring algorithmic fairness under partial observations". In: *Proceedings of the International Conference on Runtime Verification (RV)*. Springer, 2023, pp. 291–311.

[HKW18] Joseph Y. Halpern and Max Kleiman-Weiner. "Towards Formal Definitions of Blameworthiness, Intention, and Moral Responsibility". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2018, pp. 1853–1860.

[HL04] Andreas Herzig and Dominique Longin. "C&L Intention Revisited". In: vol. 4. 2004, pp. 527–535.

[HL72] Frederick A. Hosch and Lawrence H. Landweber. "Finite Delay Solutions for Sequential Conditions". In: *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*. North-Holland, Amsterdam, 1972, pp. 45–60.

[Hof94] Hans Hofmann. *Statlog (German Credit Data)*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NC77. 1994.

[HP05a] Joseph Y. Halpern and Judea Pearl. "Causes and Explanations: A Structural-Model Approach. Part I: Causes". In: *The British Journal for the Philosophy of Science* 56.4 (2005), pp. 843–887.

[HP05b] Joseph Y. Halpern and Judea Pearl. "Causes and Explanations: A Structural-Model Approach. Part II: Explanations". In: *The British Journal for the Philosophy of Science* 56.4 (2005), pp. 889–911.

[HPS16] Moritz Hardt, Eric Price, and Nati Srebro. "Equality of Opportunity in Supervised Learning". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2016, pp. 3315–3323.

[Hus+17] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. "Imitation learning: A survey of learning methods". In: *ACM Computing Surveys* 50.2 (2017), pp. 1–35.

[HW03] Wiebe Van der Hoek and Michael Wooldridge. "Towards a logic of rational agency". In: *Logic Journal of IGPL* 11.2 (2003), pp. 135–159.

[HWL24] Pierre Haritz, David Wanke, and Thomas Liebig. "Enhancing Safety for Autonomous Agents in Partly Concealed Urban Traffic Environments Through Representation-Based Shielding". In: *Proceedings of the Intelligent Vehicles Symposium (IV)*. IEEE. 2024, pp. 1758–1763.

[HZ22] Yaowei Hu and Lu Zhang. "Achieving long-term fairness in sequential decision making". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. Vol. 36. 2022, pp. 9549–9557.

[Jag+09] Radha Jagadeesan, Alan Jeffrey, Corin Pitcher, and James Riely. "Towards a Theory of Accountability and Audit". In: *European Symposium on Research in Computer Security (ESORICS)*. Springer. 2009, pp. 152–167.

[Jan+20] Nils Jansen, Bettina Könighofer, Sebastian Junges, Alex Serban, and Roderick Bloem. "Safe Reinforcement Learning Using Probabilistic Shields (Invited Paper)". In: *Proceedings of the International Conference on Concurrency Theory (CONCUR)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 3:1–3:16.

[Jon+20] David Jones, Chris Snider, Aydin Nassehi, Jason Yon, and Ben Hicks. "Characterising the Digital Twin: A systematic literature review". In: *CIRP Journal of Manufacturing Science and Technology* 29 (2020), pp. 36–52.

[Jud+24a] Samuel Judson, Matthew Elacqua, Filip Cano, Timos Antonopoulos, Bettina Könighofer, Scott J. Shapiro, and Ruzica Piskac. "'Put the Car on the Stand': SMT-based Oracles for Investigating Decisions". In: *Proceedings of the Symposium on Computer Science and Law (CSLAW)*. ACM, 2024, pp. 73–85.

[Jud+24b] Samuel Judson, Matthew Elacqua, Filip Cano, Timos Antonopoulos, Bettina Könighofer, Scott J. Shapiro, and Ruzica Piskac. "soid: A Tool for Legal Accountability for Automated Decision Making". In: *Proceedings of the International Conference on Computer Aided Verification (CAV)*. Lecture Notes on Computer Science. Springer, 2024, pp. 233–246.

[Jum+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[K̈+24] Bettina Könighofer, Joshua A. Kroll, Ruzica Piskac, Michael Veale, and Filip Cano Córdoba. "Accountable Software Systems (Dagstuhl Seminar 23411)". In: *Dagstuhl Reports* 13.10 (2024), pp. 24–49.

[Käl22] Sigrid Källblad. "A dynamic programming approach to distribution-constrained optimal stopping". In: *The Annals of Applied Probability* 32.3 (2022), pp. 1902–1928.

[Kam+12] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. "Fairness-aware classifier with prejudice remover regularizer". In: *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*. Springer. 2012, pp. 35–50.

[Kat16] Joost-Pieter Katoen. "The Probabilistic Model Checking Landscape". In: *Proceedings of the Symposium on Logic in Computer Science (LICS)*. ACM, 2016, pp. 31–45.

[KB15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2015.

[KC12] Faisal Kamiran and Toon Calders. "Data preprocessing techniques for classification without discrimination". In: *Knowledge and Information Systems* 33.1 (2012), pp. 1–33.

[Kir+16] Lauren Kirchner, Surya Mattu, Jeff Larson, and Julia Angwin. "Machine Bias". In: *ProPublica* (2016). URL: `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing` (visited on 06/04/2024).

[Kir+21] B. Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez. "Deep reinforcement learning for autonomous driving: A survey". In: *IEEE Transactions on Intelligent Transportation Systems* 23.6 (2021), pp. 4909–4926.

[Kno16] Geert-Jan Alexander Knoops. *Mens Rea at the International Criminal Court*. Brill Academic Publishers, 2016.

[KNP11] Marta Kwiatkowska, Gethin Norman, and David Parker. "PRISM 4.0: Verification of Probabilistic Real-time Systems". In: *Proceedings of the International Conference on Computer Aided Verification (CAV)*. Springer, 2011, pp. 585–591.

[Koc+23] Niklas Kochdumper, Hanna Krasowski, Xiao Wang, Stanley Bak, and Matthias Althoff. "Provably Safe Reinforcement Learning via Action Projection using Reachability Analysis and Polynomial Zonotopes". In: *IEEE Open Journal of Control Systems* 2 (2023), pp. 79–92.

[Kön+17] Bettina Könighofer, Mohammed Alshiekh, Roderick Bloem, Laura Humphrey, Robert Könighofer, Ufuk Topcu, and Chao Wang. "Shield synthesis". In: *Formal Methods in System Design* 51.2 (2017), pp. 332–361.

[Kön19] Bettina Könighofer. "Shield synthesis: runtime enforcement for reactive systems". PhD thesis. Graz University of Technology, 2019.

[Kro+17] Joshua A. Kroll, Joanna Huey, Solon Barocas, Edward W. Felten, Joel R. Reidenberg, David G. Robinson, and Harlan Yu. "Accountable Algorithms". In: *University of Pennsylvania Law Review* 165.3 (2017), p. 633.

[KTV10] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. "Accountability: Definition and Relationship to Verifiability". In: *Proceedings of Conference on Computer and Communications Security (CCS)*. ACM, 2010, pp. 526–535.

[KWA20] Hanna Krasowski, Xiao Wang, and Matthias Althoff. "Safe Reinforcement Learning for Autonomous Lane Changing Using Set-Based Prediction". In: *Proceedings of the International Conference on Intelligent Transportation Systems (ITSC)*. 2020, pp. 1–7.

[KZ15a] Felix Klein and Martin Zimmermann. "How Much Lookahead is Needed to Win Infinite Games?" In: *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*. Vol. 9135. Lecture Notes in Computer Science. Springer, 2015, pp. 452–463.

[KZ15b] Felix Klein and Martin Zimmermann. "What are strategies in delay games? Borel determinacy for games with lookahead". In: *Proceedings of the Conference on Computer Science Logic (CSL)*. Vol. 41. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2015, pp. 519–533.

[Lew13] David Lewis. *Counterfactuals*. Originally published in 1973. John Wiley & Sons, 2013.

[LGZ13] David A. Lagnado, Tobias Gerstenberg, and Ro'i Zultan. "Causal responsibility and counterfactuals". In: *Cognitive Science* 37.6 (2013), pp. 1036–1073.

[Li+23] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. "Trustworthy AI: From principles to practices". In: *ACM Computing Surveys* 55.9 (2023), pp. 1–46.

[Lin+24] Haohong Lin, Wenhao Ding, Zuxin Liu, Yaru Niu, Jiacheng Zhu, Yuming Niu, and Ding Zhao. "Safety-aware causal representation for trustworthy offline reinforcement learning in autonomous driving". In: *IEEE Robotics and Automation Letters* (2024).

[Lis21] Christian List. "Group agency and artificial intelligence". In: *Philosophy & Technology* 34.4 (2021), pp. 1213–1242.

[Liu+18] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. "Delayed impact of fair machine learning". In: *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR. 2018, pp. 3150–3158.

[LL17] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in Neural Information Processing Systems (NeurIPS)* 30 (2017).

[LM23] Alan Lewis and Tim Miller. "Deceptive Reinforcement Learning in Model-Free Domains". In: *Proceedings of the International Conference on Automatic Planning and Scheduling (ICAPS)*. AAAI Press, 2023, pp. 587–595.

[LWW23] Yannan Li, Jingbo Wang, and Chao Wang. "Certifying the Fairness of KNN in the Presence of Dataset Bias". In: *Proceedings of the International Conference on Computer Aided Verification (CAV)*. Springer, 2023.

[Mad+18] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. "Learning adversarially fair and transferable representations". In: *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2018, pp. 3384–3393.

[MAD21] Anna Meyer, Aws Albarghouthi, and Loris D'Antoni. "Certifying Robustness to Programmable Data Bias in Decision Trees". In: *Advances in Neural Information Processing Systems (NeurIPS)* 34 (2021), pp. 26276–26288.

[MB20] Peter Menzies and Helen Beebee. "Counterfactual Theories of Causation". In: *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, 2020.

[MCB20] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. "Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges". In: *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*. Springer. 2020, pp. 417–431.

[MCR12] Sérgio Moro, Paulo Cortez, and Paulo Rita. *Bank Marketing*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5K306. 2012.

[Mel92] Alfred R. Mele. *Springs of Action: Understanding Intentional Behavior*. Oxford University Press, 1992.

[Mil19] Tim Miller. "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial Intelligence* 267 (2019), pp. 1–38.

[Mni+15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533.

[Moo03] Michael S. Moore. "For What Must We Pay-Causation and Counterfactual Baselines". In: *San Diego Law Review* 40 (2003), pp. 1181–1272.

[Moo09] Michael S. Moore. *Causation and Responsibility: An Essay in Law, Morals, and Metaphysics*. Oxford University Press, 2009.

[Moo10] Michael S. Moore. *Act and Crime: The Philosophy of Action and its Implications for Criminal Law*. Oxford University Press, 2010.

[Moo19] Michael Moore. "Causation in the Law". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Winter 2019. Metaphysics Research Lab, Stanford University, 2019.

[Mot+23] Nima Motamed, Natasha Alechina, Mehdi Dastani, Dragan Doder, and Brian Logan. "Probabilistic Temporal Logic for Reasoning about Bounded Policies". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org, 2023, pp. 3296–3303.

[MS17] Peta Masters and Sebastian Sardina. "Deceptive Path-Planning." In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2017, pp. 4368–4375.

[MW24] Tobias Meggendorfer and Maximilian Weininger. "Playing Games with Your PET: Extending the Partial Exploration Tool to Stochastic Games". In: *Proceedings of the International Conference on Computer Aided Verification (CAV)*. Ed. by Arie Gurfinkel and Vijay Ganesh. Springer Nature Switzerland, 2024, pp. 359–372.

[NHR99] Andrew Y. Ng, Daishi Harada, and Stuart Russell. "Policy invariance under reward transformations: Theory and application to reward shaping". In: *Proceedings of the International Conference on Machine Learning (ICML)*. Vol. 99. Citeseer. 1999, pp. 278–287.

[Nil98] Johan Nilsson. "Real-time control systems with delays". In: *Department of Automatic Control* 1049 (1998).

[NR00] Andrew Y. Ng and Stuart Russell. "Algorithms for Inverse Reinforcement Learning". In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2000, pp. 663–670.

[Obe+19] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. "Dissecting racial bias in an algorithm used to manage the health of populations". In: *Science* 366.6464 (2019), pp. 447–453.

[O'k+20] Matthew O'kelly, Hongrui Zheng, Dhruv Karthik, Rahul Mangharam, Hugo Jair Escalante, and Raia Hadsell. "F1TENTH: An Open-source Evaluation Environment for Continuous Control and Reinforcement Learning". In: *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*. Vol. 123. PMLR, 2020, pp. 77–89.

[Pan+17] Xinlei Pan, Yurong You, Ziyan Wang, and Cewu Lu. "Virtual to Real Reinforcement Learning for Autonomous Driving". In: *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2017.

[Pau14] Sarah Paul. "Embarking on a Crime". In: *Law and the Philosophy of Action*. Ed. by Enrique Villanueva V. Rodopi, 2014, pp. 101–24.

[Pau20] Sarah Paul. *Philosophy of Action: A Contemporary Introduction*. Routledge, 2020.

[Pea09] Judea Pearl. *Causality*. Cambridge University Press, 2009.

[PJ05] Stephen Prajna and Ali Jadbabaie. "Methods for Safety Verification of Time-Delay Systems". In: *Proceedings of the Conference on Decision and Control (CDC)*. IEEE, 2005, pp. 4348–4353.

[Poe15] Ibo Van de Poel. "The problem of many hands". In: *Moral responsibility and the problem of many hands*. Routledge, 2015, pp. 50–92.

[Pra+21a] Stefan Pranger, Bettina Könighofer, Lukas Posch, and Roderick Bloem. "TEMPEST - Synthesis Tool for Reactive Systems and Shields in Probabilistic Environments". In: *Proceedings of the International Symposium on Automated Technology for Verification and Analysis (ATVA)*. Vol. 12971. Lecture Notes in Computer Science. Springer, 2021, pp. 222–228.

[Pra+21b] Stefan Pranger, Bettina Könighofer, Martin Tappler, Martin Deixelberger, Nils Jansen, and Roderick Bloem. "Adaptive Shielding under Uncertainty". In: *Proceedings of the American Control Conference (ACC)*. IEEE, 2021, pp. 3467–3474.

[PS+17] Adrián Pérez-Suay, Valero Laparra, Gonzalo Mateo-García, Jordi Muñoz-Marí, Luis Gómez-Chova, and Gustau Camps-Valls. "Fair Kernel Learning". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*. Springer, 2017, pp. 339–355.

[PT20] Ana Pereira and Carsten Thomas. "Challenges of machine learning applied to safety-critical cyber-physical systems". In: *Machine Learning and Knowledge Extraction* 2.4 (2020), pp. 579–602.

[PV17] Aaron Zeff Palmer and Alexander Vladimirsky. "Optimal stopping with a probabilistic constraint". In: *Journal of Optimization Theory and Applications* 175 (2017), pp. 795–817.

[PV20] Erika Puiutta and Eric Veith. "Explainable Reinforcement Learning: A Survey". In: *Proceedings of the International Cross-Domain Conference for Machine Learning and Knowledge Extraction (CD-MAKE)*. Springer. 2020, pp. 77–95.

[Qui69] Willard Van Orman Quine. *Ontological Relativity and Other Essays*. Columbia University Press, 1969.

[RBT22] Jonathan Richens, Rory Beard, and Daniel H. Thompson. "Counterfactual harm". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 35. 2022, pp. 36350–36365.

[Ren+19] Matthieu Renard, Yliès Falcone, Antoine Rollet, Srinivas Pinisetty, Thierry Jéron, and Hervé Marchand. "Enforcement of (Timed) Properties with Uncontrollable Events". In: *International Colloquium on Theoretical Aspects of Computing (ICTAC)*. Lecture Notes on Computer Science. 2019, pp. 542–560.

[RG91] Anand S. Rao and Michael P. Georgeff. "Modeling Rational Agents within a BDI-Architecture". In: *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*. Morgan Kaufmann, 1991, pp. 473–484.

[RG95] Anand S. Rao and Michael P. Georgeff. "BDI Agents: From Theory to Practice". In: *Proceedings of the International Conference on Multiagent Systems (ICMAS)*. MIT Press, 1995, pp. 312–319.

[Rod+25] Andoni Rodriguez, Guy Amir, Davide Corsi, Cesar Sanchez, and Guy Katz. "Shield Synthesis for LTL Modulo Theories". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2025.

[Sax+19] Dhruv Mauria Saxena, Sangjae Bae, Alireza Nakhaei, Kikuo Fujimura, and Maxim Likhachev. "Driving in Dense Traffic with Model-Free Reinforcement Learning". In: *Proceedings International Conference on Robotics and Automation (ICRA)* (2019), pp. 5385–5392.

[SB18] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Sca10] Thomas Michael Scanlon. *Moral Dimensions: Permissibility, Meaning, Blame*. Harvard University Press, 2010.

[Sch+15] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. "Trust Region Policy Optimization". In: *Proceedings of the International Conference on Machine Learning (ICML)*. Vol. 37. PMLR, 2015, pp. 1889–1897.

[Set22] Kieran Setiya. "Intention". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta and Uri Nodelman. Winter 2022. Metaphysics Research Lab, Stanford University, 2022.

[Set+98] Danbing Seto, Bruce Krogh, Lui Sha, and Alongkrit Chutinan. "The Simplex architecture for safe online control system upgrades". In: *Proceedings of the American Control Conference (ACC)*. IEEE, 1998, pp. 3504–3508.

[SG21] Harini Suresh and John Guttag. "A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle". In: *Proceedings of the ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO)*. ACM, 2021.

[SGD23] Meirav Segal, Anne-Marie George, and Christos Dimitrakakis. "Policy Fairness and Unknown Bias Dynamics in Sequential Allocations". In: *Proceedings of the Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO)*. ACM, 2023, pp. 1–10.

[Sha14] Scott J. Shapiro. "Massively Shared Agency". In: *Rational and Social Agency: The Philosophy of Michael Bratman* (2014), pp. 257–293.

[Shi07] Albert N. Shiryaev. *Optimal stopping rules*. Vol. 8. Springer Science & Business Media, 2007.

[Sil+16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587 (2016), pp. 484–489.

[Sin+21] Maulshree Singh, Evert Fuenmayor, Eoin P. Hinchy, Yuansong Qiao, Niall Murray, and Declan Devine. "Digital twin: Origin to future". In: *Applied System Innovation* 4.2 (2021), p. 36.

[Sin92] Munindar P. Singh. "A critical examination of the Cohen-Levesque theory of intention". In: *Proceedings of the European Conference on Artificial Intelligence (ECAI)*. 1992, pp. 364–368.

[SJS21] Thiago D. Simão, Nils Jansen, and Matthijs T. J. Spaan. "AlwaysSafe: Reinforcement Learning without Safety Constraint Violations during Training". In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. ACM, 2021, pp. 1226–1235.

[SLB08] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[Son+16] Jinhua Song, Yang Gao, Hao Wang, and Bo An. "Measuring the Distance between Finite Markov Decision Processes". In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. ACM, 2016, pp. 468–476.

[SP11] Gerardo I. Simari and Simon D. Parsons. *Markov Decision Processes and the Belief-Desire-Intention Model: Bridging the Gap for Autonomous Agents*. New York: Springer, 2011.

[SPB19] Morgan Klaus Scheuerman, Jacob M Paul, and Jed R Brubaker. "How computers see gender: An evaluation of gender classification in commercial facial analysis services". In: *Proceedings of the ACM on Human-Computer Interaction (HCI)* 3.CSCW (2019), pp. 1–33.

[SPC23] Ashish Kumar Shakya, Gopinatha Pillai, and Sohom Chakrabarty. "Reinforcement learning algorithms: A brief survey". In: *Expert Systems with Applications* 231 (2023), p. 120495.

[Sun+21] Bing Sun, Jun Sun, Ting Dai, and Lijun Zhang. "Probabilistic verification of neural networks against group fairness". In: *Proceedings of the International Symposium on Formal Methods (FM)*. Springer. 2021, pp. 83–102.

[Sun23] Yi Sun. "Algorithmic Fairness in Sequential Decision Making". PhD thesis. Massachusetts Institute of Technology, 2023.

[Tap+22] Martin Tappler, Filip Cano Córdoba, Bernhard K. Aichernig, and Bettina Könighofer. "Search-Based Testing of Reinforcement Learning". In: *Proceedings of the International Joint Conference of Artificial Intelligence (IJCAI)*. ijcai.org, 2022, pp. 503–510.

[Tho80] Dennis F. Thompson. "The Moral Responsibility of Public Officials: The Problem of Many Hands". In: *American Political Science Review* 74.4 (1980), pp. 905–916.

[Tho95] Wolfgang Thomas. "On the synthesis of strategies in infinite games". In: *Proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS)*. Springer, 1995, pp. 1–13.

[Tow+24] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. *Gymnasium: A Standard Interface for Reinforcement Learning Environments*. 2024. arXiv: 2407.17032.

[Tri04] Stavros Tripakis. "Decentralized control of discrete-event Systems With bounded or Unbounded Delay communication". In: *IEEE Transactions on Automatic Control* 49.9 (2004), pp. 1489–1501.

[TSR21] Stelios Triantafyllou, Adish Singla, and Goran Radanovic. "On Blame Attribution for Accountable Multi-Agent Sequential Decision Making". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 34. Curran Associates, Inc., 2021, pp. 15774–15786.

[van+20] Marc van Zee, Dragan Doder, Leendert van der Torre, Mehdi Dastani, Thomas Icard, and Eric Pacuit. "Intention as commitment toward time". In: *Artificial Intelligence* 283 (2020), p. 103270.

[VDL24] Fabian Vu, Jannik Dunkelau, and Michael Leuschel. "Validation of Reinforcement Learning Agents and Safety Shields with ProB". In: *Proceedings of the NASA Formal Methods Symposium (NFM)*. Springer. 2024, pp. 279–297.

[Vel07] J. David Velleman. "What good is a will?" In: *Action in context* (2007), pp. 193–215.

[Wan+23] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. "A survey on the fairness of recommender systems". In: *ACM Transactions on Information Systems* 41.3 (2023), pp. 1–43.

[War+24] Francis Rhys Ward, Matt MacDermott, Francesco Belardinelli, Francesca Toni, and Tom Everitt. "The Reasons that Agents Act: Intention and Instrumental Goals". In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2024, pp. 1901–1909.

[WBT21] Min Wen, Osbert Bastani, and Ufuk Topcu. "Algorithms for fairness in sequential decision making". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR. 2021, pp. 1144–1152.

[WD92] Christopher J.C.H. Watkins and Peter Dayan. "Q-learning". In: *Machine Learning* 8 (1992), pp. 279–292.

[WDS19] Hao Wang, Shaokang Dong, and Ling Shao. "Measuring Structural Similarities in Finite MDPs". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org, 2019, pp. 3684–3690.

[Whi22] White House OSTP. *Blueprint for an AI Bill of Rights: Making Automated Systems Work for the American People*. `https://www.whitehouse.gov/ostp/ai-bill-of-rights/`. Accessed: 2024-08-01. 2022.

[Wie+23] Patrick Wienhöft, Marnix Suilen, Thiago D. Simão, Clemens Dubslaff, Christel Baier, and Nils Jansen. "More for Less: Safe Policy Improvement with Stronger Performance Guarantees". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org, 2023, pp. 4406–4415.

[Wil+22] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre-Alvise Rebuffi, Ira Ktena, Krishnamurthy Dj Dvijotham, and Ali Taylan Cemgil. "A Fine-Grained Analysis on Distribution Shift". In: *Proceedings of the International Conference on Learning Representations*. 2022.

[Win+21] Michael Winikoff, Galina Sidorenko, Virginia Dignum, and Frank Dignum. "Why bad coffee? Explaining BDI agent behaviour with valuings". In: *Artificial Intelligence* 300 (2021), p. 103554.

[WMR17] Sandra Wachter, Brent Mittelstadt, and Chris Russell. "Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR". In: *Harvard Journal of Law & Technology* 31.2 (2017), pp. 841–887.

[Wob95] Wayne Wobcke. "Plans and the revision of intentions". In: *Australian Workshop on Distributed Artificial Intelligence (DAI)*. Springer. 1995, pp. 100–114.

[Woo03] Michael Wooldridge. *Reasoning about rational agents*. MIT press, 2003.

[Woo+09] Jim Woodcock, Peter Gorm Larsen, Juan Bicarregui, and John Fitzgerald. "Formal methods: Practice and experience". In: *ACM Computing Surveys* 41.4 (2009), pp. 1–36.

[WS20] Akifumi Wachi and Yanan Sui. "Safe reinforcement learning in constrained Markov decision processes". In: *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR. 2020, pp. 9797–9806.

[WT18] Min Wen and Ufuk Topcu. "Constrained cross-entropy method for safe reinforcement learning". In: *Advances in Neural Information Processing Systems (NeurIPS)* 31 (2018).

[WZ20] Sarah Winter and Martin Zimmermann. "Finite-state strategies in delay games". In: *Information and Computation* 272 (2020), p. 104500.

[XZL22] Chengbin Xuan, Feng Zhang, and Hak-Keung Lam. "SEM: Safe exploration mask for q-learning". In: *Engineering Applications of Artificial Intelligence* 111 (2022), p. 104765.

[Yan+23a] Qisong Yang, Thiago D. Simão, Nils Jansen, Simon H. Tindemans, and Matthijs T. J. Spaan. "Reinforcement Learning by Guided Safe Exploration". In: *Proceedings of the European Conference on Artificial Intelligence (ECAI)*. Vol. 372. IOS Press, 2023, pp. 2858–2865.

[Yan+23b] Wen-Chi Yang, Giuseppe Marra, Gavin Rens, and Luc De Raedt. "Safe Reinforcement Learning via Probabilistic Logic Shields". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org, 2023, pp. 5739–5749.

[YD16] Vahid Yazdanpanah and Mehdi Dastani. "Distant group responsibility in multi-agent systems". In: *Proceedings of the International Conference on Principles of Practice in Multi-Agent Systems (PRIMA)*. Springer, 2016, pp. 261–278.

[Zaf+19] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. "Fairness constraints: A flexible approach for fair classification". In: *The Journal of Machine Learning Research* 20.1 (2019), pp. 2737–2778.

[Zem+13] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. "Learning fair representations". In: *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR. 2013, pp. 325–333.

[Zha+23] Zhang Zhang, Yifeng Zeng, Wenhui Jiang, Yinghui Pan, and Jing Tang. "Intention recognition for multiple agents". In: *Information Sciences* 628 (2023), pp. 360–376.

[Zim17] Martin Zimmermann. "Finite-State Strategies in Delay Games". In: *Proceedings of the International Symposium on Games, Automata, Logics, and Formal Verification (GandALF)*. Vol. 256. 2017, pp. 151–165.

[ZL21] Xueru Zhang and Mingyan Liu. "Fairness in learning-based sequential decision algorithms: A survey". In: *Handbook of Reinforcement Learning and Control*. Springer, 2021, pp. 525–555.

[ZLM18] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. "Mitigating unwanted biases with adversarial learning". In: *Proceedings of the Conference on AI, Ethics, and Society (AIES)*. ACM, 2018, pp. 335–340.

[ZMS23] Quan Zhou, Jakub Mareček, and Robert Shorten. "Fairness in Forecasting of Observations of Linear Dynamical Systems". In: *Journal of Artificial Intelligence Research* 76 (2023), pp. 1247–1280.

Nomenclature

But Taborlin knew the name of all things, and so all things were his to command. — Patrick Rothfuss, The Name of the Wind.

Symbol	Usage
a	generic action, $a \in \mathcal{A}$
	generic number $a \in \mathbb{R}$
b	generic action when a is already in use, $b \in \mathcal{A}$, for example in Fig. 2.1
	generic number $b \in \mathbb{R}$
	generic element of $\mathbb{B}$, $b \in \mathbb{B}$
c	generic cost, $c \in \mathbb{C}$, Chap. 6
$cost(\tau; s)$	cost incurred by a fairness shield on a trace τ up to a certain time $s \leq \lvert\tau\rvert$, Eq. 6.2, Chap. 6
d	used to denote a generic probability distribution, Sec 2.2
	when computing maximally permissive strategies under delay, the value of the intermediate delay, Sec. 2.3.2, Alg. 1
	used to denote a generic distance function in MDPs, Sec. 2.4, Chap. 7
f	used to denote a generic function $f\colon X \to Y$
	in classification problems, used to denote an ML-based classifier, Chap. 6
g	used to denote a generic group in group fairness, typically $g \in \{a, b\}$, Chap. 6
i, j	used as generic counters
k	used to indicate the length of a trace in reachability and avoidance properties in MDPs, Eq. 2.6
	used as a generic counter
l	lower bound on welfare for bounded welfare shields, Sec. 6.4.1.2, Chap. 6
m	when computing maximally permissive strategies under delay, the value of the intermediate memory, $m = \min(d, \mu)$, Alg. 1

Notation index, lowercase latin alphabet, part 1.

Symbol	Usage
n	used in general to denote lengths of traces or sequences
n_A	in group fairness measures, number of candidates in a tracce of group A, Chap. 6
n_B	in group fairness measures, number of candidates in a tracce of group B, Chap. 6
n_A^1	in group fairness measures, number of accepted candidates in a tracce of group A, Chap. 6
n_B^1	in group fairness measures, number of accepted candidates in a tracce of group B, Chap. 6
o	used to denote a generic observation in the reactive decision making framework, $o \in \mathcal{O}$, Chap. 3
p	in fairness shield, the number of counters required by the statistic to compute the fairness property, Sec. 6.2.2, Thm. 6.1, Chap. 6
r	a generic radius of a ball, Sec. 2.4
	recommendation of the ML-based classifier, Chap. 6
s	generic state of a state set, $s \in \mathcal{S}$
	used to denote a point or time in the trace in Eq. 6.2
s_0	initial state of a safety game, Sec. 2.3, 3.2.1, Chap. 4
	when it is unique, initial state of an MDP, Sec. 2.4
t	used as a generic time or length of a trace
u	upper bound on welfare for bounded welfare shields, Sec. 6.4.1.2, Chap. 6
$u_{i,j}^k$	k-th velocity datapoint for reference action α_i and reference velocity v_j, Eq. (5.1), Chap. 5
v	used to indicate a generic velocity, Chaps. 4, 5, 7
$v(\tau)$	value function associated with trace τ, Sec. 6.3, Eq. 6.9
w	sometimes used to refer to a generic word of an alphabet, Sec. 2.1
x	generic element of a set $x \in X$
	used to denote a generic input in fairness shields, Chap. 6
y	used to denote a generic action in the action register for safety games under delay, Sec. 2.3.2, Chap. 4
	in fairness shields indicates the final `accept/reject` decision of the shield, $y \in \mathcal{Y}$, Chap. 6

Notation index, lowercase latin alphabet, part 2.

Symbol	Usage
A	in group fairness, abstract groups are typically names groups A and B, used mostly in Chap. 6
Acc	winning condition of a two-player game, given by a set of accepting traces, Sec. 2.3
Ag	agent in the reactive decision making framework, Chap. 3
AP	set of atomic propositions to label an MDP, Chap. 7
B	in group fairness, abstract groups are typically names groups A and B, used mostly in Chap. 6
$B_r(x)$	ball of radius r centered at point x, Sec. 2.4
`Beh`	Set of behaviours of a strategy in a safety game, Eq. (2.2)
`Cyl`	cylinder set construction, Sec. 2.4.1
`D`	"down" action in gridworlds, Chap. 4
F	generic cummulative distribution function of a random variable, Sec. 2.2
$F_k(s, \overline{\sigma})$	k-forward multiset of states, Def. 4.1, Chap. 4
$\mathtt{FT}^t_{\theta,\pi}$	set of feasible traces of length t sampling inputs from distribution θ and using shield π, Sec. 6.2.1, Chap. 6
G	discounted return in RL, Sec. 2.4.3 random variable representing group membership of a candidate in fairness classification problems, Sec. 2.5, Chap. 6
`L`	"left" action in gridworlds, Chap. 4
N	number of samples used to build the transition probability function for the car model in Chap. 5
`N`	"neutral" or "no operation" action in gridworlds, Chap. 4
P_{ag}	set of positions of the ego car, Sec. 4.5
P_{env}	set of positions of the environment, either other car or pedestrian, Sec. 4.5
`R`	"right" action in gridworlds, Chap. 4
$R_{\mu,T}$	when μ is a fairness statistic and T is a time horizon, $R_{\mu,T}$ is the range of values that μ can take on traces of length up to T, Theorem 6.1, Chap. 6
S	set of states of a deterministic two-player game, Sec. 2.3, 3.2.1, Chap. 4
S_{ag}	set of states controlled by the agent of a deterministic two-player game, Sec. 2.3, 3.2.1, Chap. 4
S_{ag*}	set of states controlled by the agent of a deterministic two-player game, plus an extra void state ε, used for defining strategies in delayed games, Sec. 2.3.2, Chap. 4
S_{env}	set of states controlled by the environment of a deterministic two-player game, Sec. 2.3, 3.2.1, Chap. 4
$\text{Supp}(f)$	support of a function f, Sec. 2.1

Notation index, uppercase latin alphabet, part 1.

Symbol	Usage
T	set of target states in reachability properties, Sec. 2.4.2
	time horizon when computing fairness shields, Chap. 6
`U`	"up" action in gridworlds, Chap. 4
V	
V_{ag}	set of velocities of the ego car, Sec. 4.5
V_{env}	set of velocities of the environment's car, Sec. 4.5
Val	valuation function in a labelled MDP, Val: AP $\to 2^{\mathcal{S}}$, Chap. 7
W	winning region of a safety game, Eq. (2.3), Chaps. 2-4
$\texttt{WF}^g$	welfare function of group $g \in \mathcal{G}$, Chap. 6
X	used to denote a generic set
	used to denote a generic random variable

Notation index, uppercase latin alphabet, part 2.

Symbol	Usage
$\mathcal{A}$	Set of actions available to the agent in all formalisms
$\mathcal{A}_{env}$	Set of actions of the environment in safety games, Sec. 2.3, Eq. (2.1)
$\mathcal{B}$	Borel σ-algebra, Sec. 2.2
$\mathcal{D}$	set of probability distributions, given a set X, $\mathcal{D}(X)$ is the set of distributions over X, defined in Sec. 2.2, and used through all the thesis
$\mathcal{F}$	generic σ-algebra, Sec. 2.2
	set of safe states of a safety game, Sec. 2.3, Sec. 3.4.1, Sec. 3.4.2, Chap. 4
	in classification problems, the input is factored as $\mathcal{G} \times \mathcal{F}$, where $\mathcal{G}$ is the space of the protected features, and $\mathcal{F}$ is that of the other features, Sec. 2.5
$\mathcal{G}$	game graph of a two-player determinisitc game, Sec. 2.3, Chap. 4
	protected feature (a.k.a. group membership) in fairness for classification problems, Sec. 2.5, Chap. 6
$\mathcal{G}_{\delta,\mu}$	game graph, emphasizing that the plays are with delay δ and the strategies are allowed a memory μ, Sec. 2.3, Chap. 4

Notation index, `mathcal` latin alphabet, part 1.

Symbol	Usage
$\mathcal{I}$	generic Boolean formula over the set of atomic propositions associated with the MDP, that defines a potential *intention* in Chap. 7
$\mathcal{I}_s$	auxiliary set of states used in Alg. 1
$\mathcal{I}_{s,y}$	auxiliary set of states used in Alg. 1
$\mathcal{J}$	set of reachable states from s_0, Algorithm 2
	generic Boolean formula over the set of atomic propositions associated with the MDP, that defines a potential *intention* in Chap. 7
$\mathcal{L}$	Loss function of a classification problem, Sec. 2.5
	set of correct traces, Chap. 3, Def. 3.9
$\mathcal{L}_{T,\lambda,k}$	set of correct traces for probabilistic shields in MDPs, where T is a subset of states to reach, $\lambda \in (0,1)$ is the safety threshold and k is the step horizon, Sec. 3.4.3
$\mathcal{M}$	used throughout the thesis to indicate a Markovian model, either a Markov chain or a Markov decision process, depending on the context
$\mathcal{M}_{\text{car}}$	MDP model of the car, Chap. 5
$\mathcal{M}_{\text{ped}}$	Markov chain model of the pedestrian, Chap. 5
$\mathcal{O}$	"Big O" notation for stating complexity results
	observation space in reactive decision making framework, Chap. 3
$\mathcal{P}$	transition probability function of a Markovian model, either an MDP or a Markov chain, depending on the context, defined in Sec. 2.4, used throughout
$\mathcal{R}$	reward function in an RL problem, Sec. 2.4.3
$\mathcal{S}$	set of states of a Markovian model, either an MDP or a Markov chain, depending on the context, defined in Sec. 2.4, used throughout
$\mathcal{T}$	transition relation of a safety game, Sec. 2.3, Sec. 3.2.1
$\mathscr{T}$	environment transition function in the reactive decision making framework, Chap. 3
$\mathcal{X}$	input space in fairness shields, Chap. 6
$\mathcal{Y}$	output (or decision) space in fairness shields, $\lvert\mathcal{Y}\rvert = 2$, Chap. 6
$\mathcal{Z}$	in classification problems with a protected feature, $\mathcal{Z}$ represents the set of non-protected features, Chap. 6

Notation index, `mathcal` latin alphabet, part 2.

Symbol	Usage
$\mathbb{B}$	Boolean domain, $\mathbb{B} = \{\bot, \top\}$, sometimes equivalently $\mathbb{B} = \{0, 1\}$, Sec. 2.1, Chap. 6
$\mathbb{C}$	Set of costs in Chapter 6. It is understood that $\mathbb{C}$ is finite.
$\mathbb{E}$	Expected value, defined in Sec. 2.2, used throughout Chap. 6
$\mathbb{E}[cost; \theta, \pi, t]$	expected cost of a trace of length t produced sampling inputs from $\theta \in \mathcal{D}(\mathcal{X})$ and using the shield π, Chap. 6, Eq. (6.3)
$\mathbb{E}[cost \mid \tau; \theta, \pi, t]$	expected cost of a trace of length t produced sampling inputs from $\theta \in \mathcal{D}(\mathcal{X})$, and using the shield π with τ as a prefix, Chap. 6, Eq. (6.4)
$\mathbb{J}$	interference set of a shield, Def. 3.10, Chap. 3
$\mathbb{N}$	Set of natural numbers, $\mathbb{N} = \{0, 1, 2, \dots\}$
$\mathbb{P}$	probability measure in a probability space, Sec. 2.2
	probability measure over sets of finite traces, Chap. 6
$\mathbb{P}^{\mathcal{M}}$	probability measure associated with the Markov chain $\mathcal{M}$, Sec. 2.2
$\mathbb{P}^{\mathcal{M}}_{\pi}/\mathbb{P}_{\pi}$	probability measure associated with the MDP $\mathcal{M}$ and the policy π. Whenever $\mathcal{M}$ is clear from context, we may drop it from the notation, Sec. 2.2, Chap. 5, Chap. 7
$\mathbb{P}^{\mathcal{M}}_{\max}$	probability measure associated with the MDP $\mathcal{M}$ and the policy that maximizes a certain property, Chap. 5, Chap. 7
$\mathbb{P}^{\mathcal{M}}_{\min}$	probability measure associated with the MDP $\mathcal{M}$ and the policy that minimizes a certain property, Chap. 5, Chap. 7
$\mathbb{P}^{\mathcal{M}}_{\max\|\Pi}/\mathbb{P}_{\max\|\Pi}$	probability measure associated with the MDP $\mathcal{M}$ and the policy that maximizes a certain property among policies in the set Π, Chap. 5, Chap. 7
$\mathbb{P}^{\mathcal{M}}_{\min\|\Pi}/\mathbb{P}_{\min\|\Pi}$	probability measure associated with the MDP $\mathcal{M}$ and the policy that minimizes a certain property among policies in the set Π, Chap. 5, Chap. 7
$\mathbb{R}$	Set of real numbers.
$\mathbb{R}_{\geq 0}$	Set of non-negative real numbers.
$\mathbb{Z}$	Set of integer numbers.

Table 1: Notation index, `mathbb` latin alphabet.

Symbol	Usage
α	each of the individual actions in the available set of actions, $\mathcal{A} = \{\alpha_1, \ldots, \alpha_n\}$, Chap. 5
γ	discount factor in RL, Sec. 2.4.3
	experimental proportionality factor between Δx and u, Equation 5.1
δ	delay in safety games, Sec. 2.3.2, Chap. 4
	delay in reactive decision making, Sec. 3.2.4, Sec. 3.4.2
$\delta_{\max}$	cutoff value of the delay when computing controllability values, Chap. 4
δ_ρ^L, δ_ρ^U	thresholds for intention quotient, used in the retrospective method for assessing intentional behaviour Sec. 7.3, Def. 7.7
δ_σ	threshold for agency, Sec. 7.3, Def. 7.7
δ_B, δ_I	thresholds on "belief" and intention quotient to define a notion of commitment, Def. 7.6
ε	representation of a general unobserved state in safety games under delay, Sec. 2.3.2
	used to indicate a generic small number, Ex. 4.2
	$\varepsilon = (\varepsilon_{k+1}, \ldots, \varepsilon_m)$ indicates, for each integral variable, the range of variation to consider counterfactuals valid when generating counterfactuals on a factored MDP, Sec. 7.3.3.2
η	distance between two traces to consider a counterfactual valid, Sec. 7.3.3.3
θ	probability distribution of the input in classification problems, $\theta \in \mathcal{D}(\mathcal{X})$, Sec. 2.5, Chap. 6
ι	distribution of initial states of an MDP, Sec. 2.4
κ	threshold on the fairness metric, as part of the specification of fairness shields, Chap. 6, Eq. (6.5)
λ	safety threshold in probabilistic shields, Sec. 3.4.3, Chap. 5
	parameter that regulates fairness interventions in different in-processing fairness algorithms, Sec. 6.5.1
μ	generic probability measure, Sec. 2.2
	memory in a strategy for a safety game with delay, Sec. 2.3.2, Chap. 4
	statistic that maps a trace to the relevant counters used to compute a fairness property, defined in Sec. 6.2.2, used throughout Chap. 6
μ_{pos}	multiplier to convert positions between local-continuous and local-discrete coordinates, Chap. 5
μ_{vel}	multiplier to convert velocities between local-continuous and local-discrete coordinates, Chap. 5
ν	used as a counter in the proof of Thm. 3.3
ξ	a generic strategy in a safety game, Sec. 2.3, Chap. 3
	sometimes, when clear from context, especially in Chap. 4, ξ denotes the maximally permissive winning strategy
$\xi_{\texttt{max.perm.}}$	maximally permissive winning strategy of a safety game, Eq. (2.4)
$\xi_{\delta,\mu}$	in Chap. 4, to specify that the strategy works with delay δ and memory μ
π	policy in an MDP, Sec. 2.4, Sec. 2.4.3, Chap. 7
	agent policy function, Chap. 3. Sometimes π is used to refer to the agent $Ag = (\mathcal{O}, \mathcal{A}, \pi)$ following π

Notation index, lowercase greek alphabet, part 1.

Symbol	Usage
σ	elements of a word, Sec. 2.1
	generic action in a safety game, usually to denote actions of the action memory, Sec. 2.3, Sec. 3.2.1, Sec. 3.4.2, Chap. 4
	agency of a state or a set of states, Chap 7, Def. 7.3
$\overline{\sigma}$	action memory or register, $\overline{\sigma} = (\sigma_1, \ldots, \sigma_\mu)$, Chap. 4
σ_{ped}	parameter of the model of the pedestrian in Chap. 5, indicating how volatile is their behaviour
τ	trace (a.k.a. path) in a safety game, Sec. 2.3, Sec. 3.2.1, Chap. 4
	trace in reactive decision making, $\tau \in (\mathcal{O} \times \mathcal{A})^*$, Chap. 3
	trace of a fairness shield, $\tau \in (\mathcal{X} \times \mathcal{Y})^*$, Chap. 6
	trace of states of the MDP, Chap. 7
τ_A	action trace, $\tau_A \in \mathcal{A}^*$, Chap. 3
τ_O	observation trace, $\tau_O \in \mathcal{O}^*$, Chap. 3
τ_{ref}	reference trace in the retrospective method to analyze intention, Sec. 7.3, Sec. 7.4
φ	fitness function, Chap. 4
	fairness metric, Chap. 6
	a generic reachability property, Chap. 7
φ_c	controllability fitness function, Sec. 4.3, Chap. 4
φ_r	robustness fitness function, Sec. 4.3, Chap. 4
χ	a *deterministic* winning strategy, usually to build a post-shield, Sec. 3.4.1, Sec. 3.4.2, Chap 4
ω	sample of a probability space $\omega \in \Omega$, Sec. 2.2, Sec. 3.2
	finite trace prefix in the cylinder set construction, Sec. 2.4.1
	used to denote infinite repetitions, e.g., X^ω is the set of infinite sequences of elements in X

Notation index, lowercase greek alphabet, part 2.

Symbol	Usage
Δ	increment of a variable, e.g., Δt, Sec. 4.5, Chap. 5
Θ	distribution of input $\Theta_{\mathcal{X}} \in \mathcal{D}(\mathcal{O})$, Sec. 3.2.3
Π	Subset of available policies when computing maximum and minimum reachability properties in MDPs, Sec. 2.4.2, Chap. 7
	Set of agents to which a shield is restricted to work with, Chap. 3
	set of all fairness shields, Chap. 6
$\Pi^t_{\texttt{fair}}$	set of all fairness shields with bounded horizon t
$\Pi_{\texttt{fair}}$	set of fair shields for a given specification, Chap. 6
$\Pi_{\texttt{fair-per}}$	set of all periodic fair shields for a given specification, Chap 6, Eq. (6.7)
$\Pi_{\texttt{BW}}$	set of all fair shields with respect to a bounded welfare specification, Chap. 6
$\Pi_{\texttt{fair-dyn}}$	set of all dynamic fair shields for a given specification, Chap. 6
$\Pi(\mathcal{G})$	set of plays in a deterministic two player game $\mathcal{G}$, Sec. 2.3
$\Pi_U(s)$	set of paths or traces in a safety game starting from s that end outside of the winning region in exactly $\delta + k$ transitions, in the proof of Thm. 4.2
Σ	generic alphabet, Sec. 2.1
	subset of actions in a safety game, $\Sigma \in 2^{\mathcal{A}}$, Sec. 2.3
Σ_Π	set of shields associated with a set of agents Π, Chap. 3, Eq. (3.3)
Ω	sample set of a measurable space or a probability space, Sec. 2.2
$\Omega^{\mathcal{M}}_\pi$	sample space of the probability measure associated with an MDP $\mathcal{M}$ and a policy π, Sec. 2.4
$\Omega^{\mathscr{E},Ag}$	set of all observation-action traces associated with an environment $\mathscr{E}$ and an agent Ag, Sec. 3.2
$\Omega^{\mathscr{E},Ag}_k$	set of all observation-action traces of length k associated with an environment $\mathscr{E}$ and an agent Ag, Sec. 3.2

Notation index, uppercase greek alphabet.

Symbol	Usage
$\shield$	Shield, Chap. 3
$\shield^{pre}_{Ag}$	pre-shield induced by an agent Ag, defined to follow the actions of the agent, Def. 3.5, Chap. 3
$\shield^{pos}_{Ag,Ag_{det}}$	post-shield induced by an agent Ag and a determinization of the agent, Ag_{det}, Def. 3.6, Chap. 3
$\mathbb{1}$	Indicator function, for subset set $X \subseteq \mathcal{X}$, $\mathbb{1}_X \colon \mathcal{X} \to \{0,1\}$ is defined as $\mathbb{1}_X(x) = 1$ if $x \in X$, and $\mathbb{1}_X(x) = 0$ if $x \notin X$, Sec. 2.1
2^X	when X is a set, 2^X denotes the power set of X, that is, the set of subsets of X, Sec. 2.1
$f(X), f^{-1}(Y)$	when $f \colon \mathcal{X} \to \mathcal{Y}$ is a function and $X \subseteq \mathcal{X}$, $f(X)$ is the image set of X; for $Y \subseteq \mathcal{Y}$, $f^-1(Y)$ is the antiimage set, Sec. 2.1
$\gg$, $\ll$	much greater / much smaller than, Sec. 2.1, Sec. 6.2.1
$\lfloor a \rfloor$	floor of a, i.e., the greatest integer that is smaller or equal to a, Sec. 2.1
$\lceil a \rceil$	ceiling of a, i.e., the smallest integer that is greater or equal to a, Sec. 2.1
$\lfloor a \rceil$	rounded of a, i.e., the closest integer to a, and the ceiling of a if a is equidistant to $\lfloor a \rfloor$ and $\lceil a \rceil$, Sec. 2.1
$\lvert a \rvert$	when a is a number, $\lvert a \rvert$ denotes the absolute value of a
$\lvert v \rvert$	when v is a vector, $\lvert v \rvert$ denotes the magnitude of the vector
$\lvert \tau \rvert$	when τ is a word, a trace, or a sequence of some kind, $\lvert \tau \rvert$ denotes its length
$\lvert X \rvert$	when X is a set, $\lvert X \rvert$ denotes the cardinality of the set, i.e., the number of elements
$\xrightarrow{u}$	in safety games, when $s \in S_{env}$ and $s' \in S_{ag}$, we use $s \xrightarrow{u} s'$ to denote that there is an environment transition from s to s', without specifying an action of the environment; where u stands for "undefined", Sec. 2.3, 3.2.1, Chap. 4
$\emptyset$	the empty set

Notation index, special symbols.