Formal Analysis of Lane-Changing Algorithms for Autonomous Vehicles using Probabilistic Model Checking

By

Muhammad Bilal Sarwar
(Registration No: 00000402631)

Department of Engineering

School of Interdisciplinary Engineering and Sciences

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

Formal Analysis of Lane-Changing Algorithms for Autonomous Vehicles using Probabilistic Model Checking

By

Muhammad Bilal Sarwar

(Registration No: 00000402631)

A thesis submitted to the National University of Sciences and Technology, Islamabad,

in partial fulfillment of the requirements for the degree of

Master of Science in

Computational Sciences and Engineering

Supervisor: Dr. Osman Hasan

School of Interdisciplinary Engineering and Sciences

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

Annex A to NUST Letter No.
0972/102/Exams/Thesis-Cert
dated 23 Dec 16.

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by **Mr. Muhammad Bilal Sarwar** Registration No. **00000402631** of **SINES** has been vetted by undersigned, found complete in all aspects as per NUST Statutes/Regulations, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature with stamp: ____________________

Name of Supervisor: **Dr. Osman Hasan**

Date: 31/01/2025

(DR. OSMAN HASAN)
Pro-Rector (Academics)
National University of Sciences
and Technology, Isla abad

Signature of HoD with stamp: ____________________

Date: 04/02/2025

Dr. Mian Ilyas Ahmad
HoD Engineering
Professor
SINES - NUST, Sector H-12
Islamabad

Countersign by

Signature (Dean/Principal): ____________________

Date: 4/2/25

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in this thesis, entitled "**Formal Analysis of Lane-Changing Algorithms for Autonomous Vehicles using Probabilistic Model Checking**" was conducted by Mr. **Muhammad Bilal Sarwar** under the supervision of **Dr. Osman Hasan**. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **School of Interdisciplinary Engineering & Sciences** in partial fulfillment of the requirements for the degree of Master of Science in Field of Computational Sciences & Engineering, Department of Engineering, National University of Sciences and Technology, Islamabad.

Student Name: Muhammad Bilal Sarwar Signature: Bilal Sarwar

Examination Committee:

a) External Examiner 1: Name Dr. Mian Ilyas Ahmad Signature: ....

Professor, SINES

b) External Examiner 2: Name Dr. Muhammad Tariq Saeed Signature: ....

Associate Professor, SINES

Name of Co-Supervisor: Dr. Ammar Mushtaq Signature: ....

Name of Supervisor: Dr. Osman Hasan Signature:

Name of Dean/HOD: Dr. Syed Irtiza Ali Shah Signature: ....

AUTHOR'S DECLARATION

I **Muhammad Bilal Sarwar** hereby state that my MS thesis titled "**Formal Analysis of Lane-Changing Algorithms for Autonomous Vehicles using Probabilistic Model Checking**" is my own work and has not been submitted previously by me for taking any degree from National University of Sciences and Technology, Islamabad or anywhere else in the country/ world.

At any time if my statement is found to be incorrect even after I graduate, the university has the right to withdraw my MS degree.

Student Signature: Bilal Sarwar

Name: Muhammad Bilal Sarwar

Date: 31/01/2025

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled "**Formal Analysis of Lane-Changing Algorithms for Autonomous Vehicles using Probabilistic Model Checking**" is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and National University of Sciences and Technology (NUST), Islamabad towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the University reserves the rights to withdraw/revoke my MS degree and that HEC and NUST, Islamabad has the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized thesis.

Student Signature: Bilal Sarwar

Name: Muhammad Bilal Sarwar

Date: 31/01/2025

To my beloved Parents

&

Brothers

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Almighty ALLAH, the Creator and Sustainer of all beings.

I would like to express my heartfelt thanks and sincere gratitude to my supervisor, Dr. Osman Hasan, for his unwavering support, constant encouragement, and invaluable guidance throughout my research. His accessibility, insights into research, and vast knowledge in the field of formal methods have greatly strengthened this work. His emphasis on excellence kept me well-directed and focused, and his cheerful and enthusiastic encouragement was a source of strength in completing this thesis.

I would like to thank my co-supervisor, Dr. Ammar Mushtaq, for his continuous support and guidance on my thesis and for providing space in his CAL lab. I am profoundly grateful to Dr. Mian Ilyas Ahmad and Dr. Muhammad Tariq Saeed for their invaluable contributions as members of my Guidance & Examination Committee (GEC). Their insightful feedback and unwavering commitment to academic excellence have been instrumental in shaping the direction and quality of this research. Their guidance has not only enriched my thesis but also significantly enhanced my learning experience.

I would like to express my gratitude to Dr. Hammad Cheema, who was the Principal of SINES at that time, for granting me access to the SuperComputing Lab to conduct my research. Very special thanks go out to my colleagues in SINES; without their motivation and encouragement, I would not have reached this point. I would like to thank my friends and colleagues, including Yasir Bhatti, Arqam Habib, Rishaeel Haider, Haris Sohail, Farhan Siddiqui, Gulzar Ali, Suleman Munawar, Faiq Shahbaz, Abdul Rehman, Sabih, Hamza Saleem, and Shaheer for their encouragement during my thesis.

Finally, I would like to thank my parents for their prayers and unwavering support, both financially and morally. Most importantly, I wish to thank all members of my family, especially my elder brother Ali Sarwar, for his motivation, financial assistance, and moral support throughout my higher studies. His unwavering guidance at every stage of my academic journey paved the way for me to reach this goal.

Contents

List of Tables

List of Figures

LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS

AV Autonomous Vehicle

NHTSA National Highway Traffic Safety Administration

DARPA Defense Advanced Research Projects Agency

PRM Probabilistic Roadmaps

RRT Rapidly-exploring Random Trees

MDP Markov Decision Process

CTL Computational Tree Logic

PCTL Probabilistic Computational Tree Logic

MOBIL Minimizing Overall Braking Induced by Lane Changes

Abstract

Lane-changing algorithms play a critical role in ensuring passenger safety and traffic efficiency in the dynamic and stochastic environment of Autonomous Vehicles (AVs). Despite their safety-critical nature, these algorithms are predominantly analyzed using computer simulation. While simulations provide valuable insights, their sampling-based nature inherently limits their ability to capture all potential corner cases, which can lead to overlooked safety-critical scenarios. To address these limitations, we advocate for the use of probabilistic model checking as a more rigorous approach for the formal analysis of lane-changing algorithms. Probabilistic model checking is a formal verification technique that systematically explores all possible behaviors of a system within its modeled environment. Unlike simulations, it provides mathematical guarantees about the system's behavior by exhaustively analyzing the model. This makes it particularly effective for identifying and addressing rare but critical scenarios that simulations might miss.

Our proposed approach leverages Markov Decision Processes (MDPs) to model the stochastic dynamics of AV lane-changing maneuvers. MDPs represent the probabilistic nature of real traffic and the AV behavior and provide detailed dynamics of the system. Then properties of interest are formally specified using Probabilistic Computation Tree Logic (PCTL), a powerful logic for expressing complex temporal and probabilistic properties. Using the probabilistic model checker PRISM, we formally verified critical properties of the MOBIL (Minimizing Overall Braking Induced by Lane Changes) model, a widely adopted framework for AV lane change. We specifically investigated the safety and lane change efficiency, temporal performance, and system robustness under dynamic traffic conditions. We demonstrate with this work that probabilistic model checking not only surpasses the simulation limitations but also gives a complete and rigorous safety and reliability assurance framework for AV lane-changing algorithms. By formalizing and verifying these critical properties, this work establishes a foundation for developing more dependable and efficient AV systems that can robustly navigate the complexities and uncertainties of real-world traffic conditions.

Keywords: Formal Analysis, Probabilistic Analysis, Lane-Changing Algorithms, Formal Verification, Model Checking, PRISM, MOBIL, Autonomous Vehicles

Chapter 1

Introduction

1.1 Motivation

> *"Autonomous driving is both very, very sophisticated but on the other hand very, very accurate. The tolerance for failure is almost zero"*
>
> *Amnon Shashua* *(Co-founder of Mobileye)*

Autonomous Vehicles (AVs) represent a transformative advancement in modern transportation technology, with profound potential to enhance safety, efficiency, and accessibility on a global scale. These systems use sensors, machine learning algorithms, and sophisticated control mechanisms to move around in the complex, real-world surroundings, with minimal human involvement [1]. Central to the operational efficacy of AVs is the process of path planning [2]—a computational process by which an AV determines the optimal route and corresponding maneuvers required to progress from its current location to a designated destination.

Lane changing [3] is a crucial and one of the most complex tasks of path planning and its effect is substantial on safety and efficiency of roadway systems. Because of the dynamic, sometimes stochastic nature, of road environments, it is very important for AVs to make lane-changing decisions efficiently, almost optimally, in near real-time to avoid collisions. Lane changing involves transitioning between lanes to optimize traffic flow, maintain safety, or adhere to traffic regulations while interacting with other vehicles in shared roadways.

The challenges posed by lane changing underscore the need for robust verification frameworks. The imperative for safe and effective lane changing is further underscored by empirical evidence from real-world incidents. According to the NHTSA documentation, all 11 crashes of Tesla models on autopilot system reveal potential challenges in autonomous systems in terms of path planning and lane changing [4]. Also, NHTSA predicts that there will be thousands of crashes with Level 2 [5] AVs (with require driver supervision) and about 200 attended incidents involving higher level AVs (with some driver interventions to full automation) per year [6]. These projections make one case for rigorous verification frameworks that ensure AVs execute as intended, perform the desired behaviors, and fulfill their safety constraints in their decision-making. Robust lane-changing algorithms not only reduce the risk of vehicular accidents but also offer broader societal benefits, including alleviating traffic congestion, reducing emissions, and enhancing the overall reliability of transportation infrastructure.

The reliability and correctness of lane-changing algorithms in AVs cannot be fully ensured using traditional techniques like simulation or road testing alone. Although these methods are useful, they are missing the rigor needed to account for those rare but critical edge cases that could compromise safety. This need for robustness became evident when incidents such as the 2018 accident in Tempe, Arizona, where a pedestrian was run over by an autonomous Uber vehicle, came to light [7]. This incident exposed to light fundamental inadequacies of algorithmic and systems design; which is why their stringent, mathematically rigorous methods are needed to ensure operational safety and reliability. To demonstrate, with 95% confidence, that the rate of failure of AVs is 20% less than for human drivers (1.09 fatalities per 100 million miles, US, 2013), even despite an AV fleet of 100 vehicles operating 24/7, 365 days/year at an average speed of 25 mph, would take around 400 years of simulation time [8]. In fact the inherent limitations of empirical testing have already been seen in the real world. For example, CalTech developed an autonomous vehicle known as Alice for DARPA Urban Challenge [9]. Alice's goal was to navigate through an urban environment where its tasks included parking and obeying traffic regulations. Nonetheless, during competition, Alice showed unsafe behavior almost *'owning'* a collision. Subsequently, the root cause was found in the adverse interaction between the reactive obstacle avoidance subsystem and the reacting path planner. The failure was in a very specific set of circumstances and, even with a lot of testing, it would have been very hard to notice on something that would only happen in one out of a few million [10]. Only formal verification methods, such as probabilistic model checking, provide the capacity to rigorously analyze and ensure the reliable performance of such systems across all possible scenarios.

1.2 Autonomous Vehicles

Autonomous vehicles represent a paradigm shift in transportation. With a suite of sensors like LiDAR, radar, cameras, and ultrasonic sensors they allow the perception of the environment [11]. Real-time decisions are made using the processed data which has been done through advanced algorithms and machine learning techniques. Their transformative potential extends across multiple dimensions: This will reduce accidents, optimize traffic flow, and expand mobility access. These systems are a major step forward in tackling world transportation problems.

The Society of Automotive Engineers (SAE) classifies AV automation capabilities into six levels, ranging from Level 0 (no automation) to Level 5 (full automation) [5]:

- **Level 0 (No Automation):** The driver handles all driving tasks without support.
- **Level 1 (Driver Assistance):** Systems provide limited assistance, such as cruise control or lane centering.
- **Level 2 (Partial Automation):** The system assumes control of steering and acceleration but requires driver oversight.
- **Level 3 (Conditional Automation):** The system handles full driving tasks under specific conditions but may require human intervention.
- **Level 4 (High Automation):** Autonomous operation in most environments, excluding extreme conditions.
- **Level 5 (Full Automation):** Complete autonomy in all scenarios without driver input.

Despite a lot of progress, there are still some hurdles to AVs development. In order to navigate real-world traffic, systems for decision-making need to be designed capable of making decisions in the presence of uncertainty, like that from the presence of an aggressive driver or change in the environment that occurs suddenly. Additionally, social acceptance and regulation challenges remain an issue, specifically the ethical dilemma and liability in the crash scenarios [12]. As AVs approach Level 5 automation, it will be imperative that decision-making systems are reliable.

However, safety remains a central concern. Implicit assumptions of human behavior and unpredictable behavior of human drivers are a big challenge for automated systems to predict and adapt appropriately. Therefore, the evidence demonstrates the need for rigorous verification frameworks to certify the safety and reliability of AV systems for high-risk tasks such as lane changing where the complexity of the problem is expanded.

1.3 Lane-Changing Algorithms

Lane-changing algorithms [3] are an essential part of path planning process of AVs and they have to successfully contend with the intricacies of the shared roadways that pose autonomous and human driven vehicles with varying degrees of predictability. Despite being fundamental to AV navigation, the operation of lane changing remains a challenging task. A successful lane-changing algorithm must take into account vehicle kinematics, dynamic traffic conditions and well as regulatory adherence and be tractable to real time execution [13]. These algorithms are critical to the safety and traffic efficiency because of their complexity.

Three principal approaches to lane-changing algorithms have been developed:

1. **Rule-Based Algorithms:** These depend on predefined heuristics, including the maintenance of minimum safe distances or the prioritization of right-of way. However, these are interpretable and computationally efficient, but their rigid structure makes them less adaptable to unpredicted traffic scenarios [14].

2. **Machine Learning-Based Algorithms:** Large datasets are used to predict the optimal maneuver using such machine learning techniques as neural networks,

reinforcement learning approaches, etc. Although these methods are robust to diverse conditions, they tend to be opaque and hard to hold accountable in safety critical domains [15].

3. **Hybrid Methods:** Combining the benefits of rule-based and machine learning approaches, hybrid methods aim to balance interpretability and adaptability. For complex scenarios, they apply machine learning and stick to rule–based safety constraints [16].

Despite advancements, lane-changing algorithms face significant challenges:

- **Handling Edge Cases:** A major problem is that algorithms may fail to generalize sufficiently, resulting in too many rare and unforeseen traffic scenarios being risky.
- **Human Interaction:** Human drivers are erratic and context dependent; autonomous systems must operate side by side with them.
- **Safety vs. Efficiency Trade-Offs:** The persistent challenge is to balance "cautious maneuvering" to stay out of a collision's way with traffic flow optimization.

To address these challenges, various analysis techniques have been employed. Traditional analysis methods for lane-changing algorithms include paper-and-pencil proofs, simulations, and experimentation. Each of these techniques has its own strengths and limitations. Paper-and-pencil proofs offer theoretical insights and mathematically rigorous arguments about system behavior. However, they are prone to human error and impractical for exhaustive analysis of large, complex systems with numerous interacting components. Using simulations and experimentation, system performance can be predicted cost-effectively without deploying the system. Their process of evaluating real-world scenarios is not rigorous enough and often misses the rare bugs in the simulations. However, these approaches won't be able to find a bug in the system if that corner case occurs. When doing real experimentation, a huge waste (loss) of resources could happen if an unforeseen anomaly arises when the system goes live. To build a system, that is precise and accurate from all aspects, formal verification is used. The approach uses theorem proving and model checking to verify safety-critical systems formally. Deep theoretical guarantees make theorem proving a match for proving system properties using mathematical logic. Model checking explores all possible states to ensure correctness and reliability. They both catch corner case bugs that get missed in a simulation or a test. The major drawback of model checking is its state space explo-

Table 1.1. Comparison of Analysis Techniques

Techniques	**Pros**	**Cons**
Paper-and-pencil Proofs	• Completeness	• Human-error prone • Practically impossible to analyze large and complex systems
Simulations	• User friendly • Quick insights about the working of the force algorithm	• Incomplete • Impossible to predict all corner cases
Experiments	• Real-time: hardware/software interaction • Quick insights about the working of the force algorithm	• Incomplete • Impossible to predict all corner cases • High cost
Formal Verification (Model Checking)	• Completeness • Rigorous testing for all scenarios	• State-space explosion

sion: requirements that grow exponentially with system complexity cause intractable computational demands. Abstraction and decomposition will help temper this issue. Yet, model checking is indispensable for highly safety-critical systems since it provides unmatched reliability. Table 1.1 presents a detailed comparison of these methods.

Formal methods [17] have the potential to cater for the above-mentioned challenges. However, the environmental uncertainties, the unpredictable traffic patterns, and obstacle presence, in the case of lane changing pose substantial challenges for traditional formal verification methods. To capture these uncertainties and unpredictable aspects, we propose a probabilistic model checking based approach to verify the safety and performance of lane-changing algorithms. Probabilistic model checking [18] entails constructing mathematical representations, such as Markov Decision Processes (MDPs) [19], to capture the probabilistic dynamics of the system. MDPs allow decision making within an uncertain environment that is captured through probabilistic transitions and multiple decision pathways. Probabilistic model checking also facilitates a comprehensive verification process involving probabilistic properties expressed in Probabilistic Computation Tree Logic (PCTL) [20].

Overall, while traditional methods provide useful insights, formal verification techniques such as model checking offer a systematic and robust way to analyze lane-changing algorithms, ensuring safety, reliability, and performance in AVs.

1.4 Problem Statement

The increasing adoption of Autonomous Vehicles (AVs) introduces significant challenges in ensuring safety, reliability, and efficiency in complex driving scenarios. Among these, lane-changing maneuvers are particularly critical due to their impact on traffic flow, collision avoidance, and compliance with driving regulations. AVs must perform lane changes dynamically while considering the presence of other vehicles, uncertain driver behaviors, and unpredictable environmental conditions.

Existing methods for verifying lane-changing algorithms, such as simulations and real-world testing, fail to provide exhaustive safety guarantees, as they cannot capture rare but critical edge cases. Moreover, traditional verification techniques, including paper-and-pencil proofs and heuristic-based evaluations, lack the scalability and rigor necessary to analyze the vast number of possible interactions in dynamic road environments. These limitations highlight the need for a robust, formal verification framework capable of ensuring the correctness of lane-changing algorithms under probabilistic uncertainties.

This thesis addresses the problem of formally analyzing lane-changing algorithms for AVs using probabilistic model checking. By leveraging MDPs and PCTL, this study aims to verify the safety and performance of AV lane-changing strategies rigorously. The proposed approach provides a mathematically sound framework to assess AV decision-making in uncertain environments, ensuring that lane changes are performed optimally while adhering to safety constraints. The research bridges the gap between theoretical formal methods and real-world AV challenges, contributing to the development of safer and more reliable autonomous driving systems.

1.5 Related Work

In this section, we review related work, highlighting key advancements and methodologies in the field, while identifying gaps that our study aims to address.

1.5.1 Traditional and Sampling-Based Approaches

Lane changing has garnered significant scholarly attention due to its critical role in ensuring safety, efficiency, and reliability for AVs. Traditional lane-changing methodologies encompass a range of graph-based algorithms, such as Dijkstra's algorithm [21] and A* [22]. They proved to be very effective in the static or moderately changing environment where the positions of the obstacles and road participants are easy to predict. The algorithms that derive from this idea have been fundamental tools for computing shortest paths, especially in static environments, because they allow efficient shortest path computation. However, they can often be insufficient in highly dynamic environments that introduce rapid changes of traffic elements which go beyond the scope of static planning paradigms. To overcome these shortcomings, sampling-based methods including the Probabilistic Roadmaps (PRM) [23] and Rapidly-exploring Random Trees (RRT) [24] have received more attention lately. These methods incorporated stochasticity to cope with the dynamic nature of complex environments. The benefit of generating feasible trajectories for both holonomic and non-holonomic systems is provided. In addition, the RRT* [25] can even guarantee the asymptotic optimality of the generated path, formally ensuring the almost-sure convergence to global optimal solutions, with increasing number of samples. The sampling-based methods and their variants have been widely applied [21, 24] to AVs. The computational complexity of the sampling procedure, however, is high, making their practical use limited.

1.5.2 Machine Learning Approaches

Machine learning, particularly using reinforcement learning, is being increasingly used to improve the decision making in uncertain and dynamic environments. For example, Yang et al. [26] presented a model consisting of combining Long Short-Term Memory (LSTM) networks with Deep Deterministic Policy Gradients (DDPG) to enhance trajectory prediction in lane changes. By incorporating temporal learning into AV decision-making, they improved average single step rewards by 7.4% over traditional methods. Furthermore, hybrid frameworks that integrate reinforcement learning with traditional rule-based methods were introduced to benefit from the adaptability of learning-based techniques while keeping the interpretability of deterministic models. A framework combining Deep Q-Learning with rule-based constraint was introduced by Ghimire et al. [27], resulting in a safety rate of 0.8 while ensuring improved decision-making efficiency. Tian et al. [28] used personalized lane change assistance systems based on mode predictive control in conjunction with the end-to-end imitation learning. This approach is adapted to the particular driver's behavior, giving it a driver tailored driving experience, which is useful for human like AV behaviour modeling.. Moreover, Shi et al. [29] proposed a method to use imitation learning as reinforcement learning to initialize learned lane-changing behaviors, which improve collision rates and speed performances considerably.

1.5.3 Formal Verification in Lane Changing

Formal methods have been used to analyze lane-changing algorithms. For example, Zita et al. [30] formally analyzed the lane changing module of an AV, finding defects both in the model and its implementation. These results demonstrate the ability of formal methods to dramatically improve software reliability for autonomous systems. Similarly, Yang et al. [31] applied probabilistic model checking to multimodal transportation systems, optimizing path planning by dynamically adjusting congestion probabilities based on IoT sensor data. Dhonthi et al. [32] has also applied Signal

Temporal Logic (STL) to verify the safety and the efficiency of AV path planning. They verified AV trajectories in scenarios that may arise in real time (e.g., automated valet parking) given temporal logic specifications. This approach assures that safety standards are fulfilled by varying constraints that demonstrate that the AV behaves appropriately to static and dynamic obstacles. Chen and Li [33] also suggest a hybrid approach that combines data driven and model driven approaches to advance formal modeling and analysis of the driving scenarios. They abstract complex models and map them to formal structures to provide a mechanism that improves the safety analysis of AV path planning.

Lane changing is vital for AV systems, influencing their safety, efficiency, and reliability. Traditional graph-based methods like Dijkstra's and A* excel in static environments but struggle with the dynamic and uncertain nature of real-world traffic. Sampling-based techniques, such as PRM and RRT, add stochasticity but are computationally expensive for real-time applications. Machine learning approaches, while adaptive, often lack formal safety guarantees. Formal methods offer rigorous verification but fail to capture the stochastic aspects of lane changes. Bridging these gaps requires a unified approach using MDPs and probabilistic model checking to ensure robust and reliable AV lane-changing algorithms.

To bridge these gaps, there is a critical need for a unified approach that leverages the strengths of probabilistic modeling and formal verification. Such an approach should address the inherent uncertainty and dynamic nature of lane-changing scenarios while providing rigorous guarantees regarding safety and efficiency. By capturing the unpredictable aspects of lane changing using a rich formalism like MDPs and employing probabilistic model checking for their analysis, we can advance the development of robust and reliable lane-changing algorithms for AVs operating in safety-critical environments.

1.6 Proposed Framework

The objective of this thesis is primarily focused on the formal modeling, analysis, and verification of lane-changing algorithms for AVs using probabilistic model checking. The proposed methodology establishes a structured framework that ensures the safety, reliability, and efficiency of AV lane-changing strategies in dynamic traffic environments. Specifically, this thesis develops a framework encompassing the following capabilities:

1. The ability to formally model the parameters of the ego vehicle and surrounding vehicles, including position, speed, acceleration, and lane information. This foundational representation ensures accurate depiction of real-world traffic scenarios.

2. The ability to formalize decision-making algorithms, such as rule-based methods, reinforcement learning, and fuzzy logic, and evaluate their performance against temporal logic-based safety and performance requirements.

3. The ability to represent and analyze the stochastic interactions between the ego vehicle and its environment using an MDP. This enables precise modeling of uncertainties and dynamic decision outcomes.

4. The ability to verify lane-changing strategies using a probabilistic model checker (e.g., PRISM). This includes specifying safety constraints (e.g., collision avoidance) and performance objectives (e.g., optimizing travel time) in temporal logic and systematically evaluating these properties.

The proposed framework, illustrated in Figure 1, outlines the methodology for the formal analysis of lane-changing algorithms. The grey-shaded boxes in the figure represent the key contributions of this thesis, which form the essential components for formalizing and verifying lane-changing strategies in AVs. The input to this framework, represented by the rectangles with curved bottoms, includes the modeling parameters of the ego vehicle and surrounding vehicles, as well as the decision-making algorithms to be evaluated. The first step in the framework involves building a formal model of the ego vehicle and its environment, using the parameters outlined above, in the form of an MDP. This formalism captures the probabilistic and dynamic nature of traffic interactions.

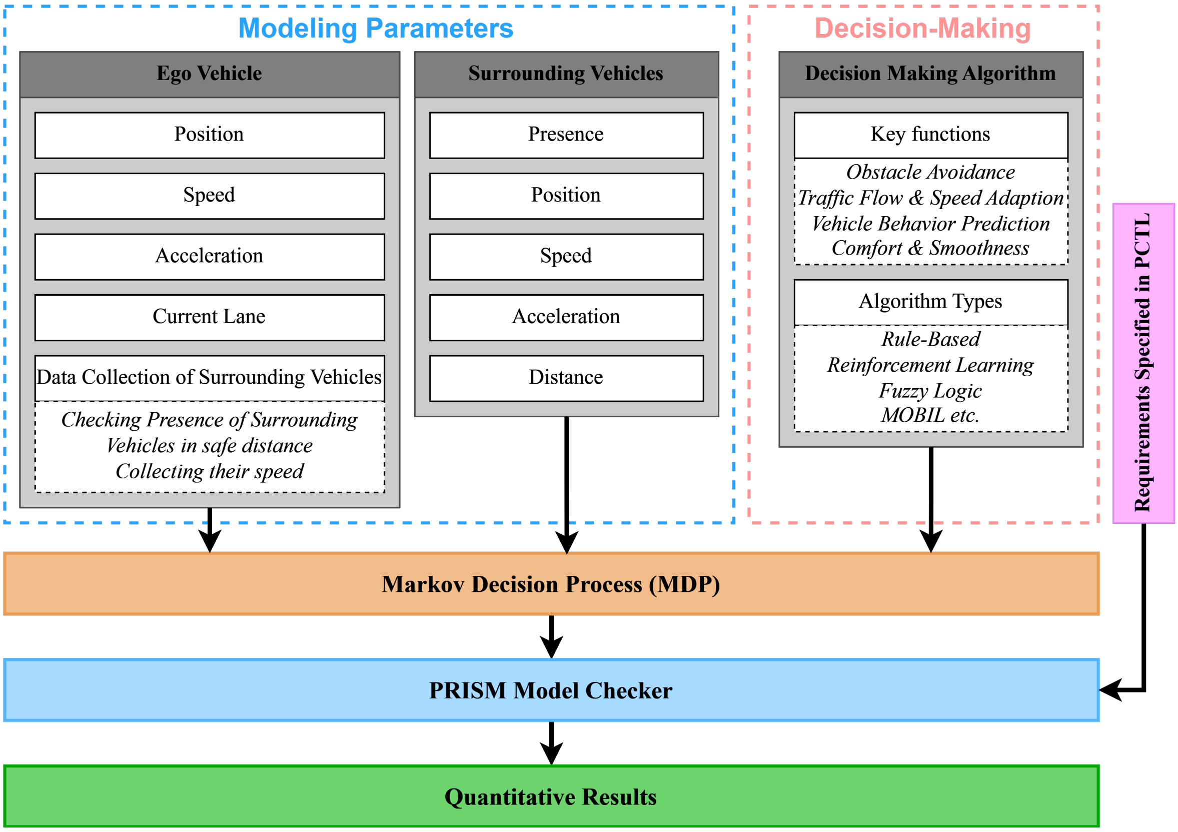

Figure 1.1: Proposed Framework

To enable this step, the thesis formalizes the decision-making algorithms, including rule-based, reinforcement learning, and fuzzy logic, and integrates them into the MDP framework. The algorithms are modeled to consider key functions such as obstacle avoidance, traffic flow adaptation, and vehicle behavior prediction. The second step involves specifying system properties in PCTL, such as safety and performance criteria. These properties serve as the basis for evaluating the effectiveness of lane-changing strategies.

The third step in the proposed approach involves verifying the specified properties using a probabilistic model checker. The PRISM model checker is utilized to evaluate metrics such as the probability of successful lane changes, collision risks under various traffic conditions, and expected improvements in time and fuel efficiency. To streamline the verification process, this thesis builds a library of pre-verified properties, including classical safety and performance criteria, to minimize the effort required for interactive verification.

Finally, the output of the framework comprises the quantitative results of system properties. These results provide a rigorous certification of the safety and performance of the lane-changing algorithms for the given traffic scenarios. By leveraging this framework, the thesis establishes a robust methodology for ensuring the correctness and continuous improvement of lane-changing strategies in autonomous driving systems.

1.7 Thesis Contributions

In summary, the main focus of this thesis is on the modeling, analysis, and verification of lane-changing algorithms for AVs using a probabilistic formal framework. This approach leverages Markov Decision Processes and temporal logic verification to ensure both safety and performance in dynamic traffic environments. In this endeavor, this thesis makes the following contributions.

1. It presents a formal framework for modeling and verifying lane-changing algorithms in AVs, leveraging MDPs to capture the probabilistic nature of traffic dynamics and interactions.

2. It introduces a method to formally specify and verify key safety and performance requirements using Probabilistic Computation Tree Logic (PCTL) and the PRISM model checker, ensuring rigorous evaluation of lane-changing strategies.

3. It provides an algorithm-agnostic structure to analyze a wide variety of decision-making approaches, including rule-based methods, reinforcement learning, fuzzy logic, etc., making the framework adaptable to diverse scenarios.

4. It presents quantitative performance and safety metrics for lane-changing strategies, offering a systematic approach to guide iterative improvement and ensure robust, real-world applicability of autonomous vehicle systems.

1.8 Organization of the Thesis

The rest of this thesis is structured as follows. In Chapter 2, we provide an introduction to the theoretical foundations essential for understanding the complex topics discussed throughout the thesis. This chapter outlines key concepts and background information that will facilitate the reader's comprehension of the advanced material presented in later chapters. It acts as a primer, ensuring that readers are adequately prepared for the more technical discussions that follow.

Chapter 3 focuses on the general formalizations that underpin lane-changing algorithms. In this chapter, we present a rigorous treatment of the formalized frameworks that form the core of the research. The emphasis is placed on the structure and relevance of these models, which are integral to the objectives of the thesis. This chapter establishes a solid theoretical foundation, serving as a reference point for subsequent analyses and discussions.

In Chapter 4, we formalize and verify the MOBIL model within a specific framework. This chapter examines the model's principles, mechanisms, and applicability while detailing the rigorous verification methods and results. By integrating analysis and verification, we demonstrate the model's robustness, reliability, and theoretical soundness, laying the foundation for its broader applicability and practical use.

Finally, Chapter 5 concludes the thesis with a synthesis of the main findings and contributions. This chapter recaps the key results and highlights areas for future research, proposing potential directions for extending the work and addressing unresolved challenges in the field.

Chapter 2

Preliminaries

In this chapter, we introduce some preliminary material on which our work builds. First, we describe the model that represents autonomous vehicle behaviors and their interactions with dynamic environments, specifically MDPs. We also introduce relevant concepts such as policies and reward structures, which are central to decision-making in MDPs. Then, we discuss how to formally specify system properties, which we then reason about, using PCTL. Next, we describe how to incorporate such formal models along with probabilistic requirements to assess and guarantee that systems behave reliably. Finally, we discuss the probabilistic model checking approach, with an emphasis on the PRISM tool, and highlight the unique challenges associated with the formal verification of autonomous vehicle systems.

2.1 Formal Verification

Formal methods consist of applying formal reasoning on the core functionality and behavior of any real-time system by using mathematics to model the system first and to eliminate defects in the system design. These are widely used to verify safety-critical systems in which failure leads to financial loss or could even result in humans being lost. Such systems are exemplified by car lane-changing algorithms in autonomous vehicles, which are critical for functional and logical verification. Minor errors in the design of software or hardware can have such catastrophic outcomes as collisions or system failures that would compromise both passenger safety and public confidence in autonomous vehicle technologies.

First, we abstract a system into a mathematical model that captures its key behaviors and constraints in order to verify the system. The formal logic (e.g. temporal logic) expresses properties or specifications that the system is supposed to satisfy. Then tools or techniques analyze the model against these properties to assure compliance. This process entails entering all possible states that the system can attain and recognizing any irregularities or mistakes. Formal verification has a rigorous mathematical foundation and offers the confidence that in all circumstances the system will operate as expected, even in obscure edge cases that are hard to catch with normal testing methods.

Formal verification employs two principal techniques: Model checking and Theorem proving. In Theorem proving, we create a formal proof to make sure that a system is correct. This technique usually concludes a certain property from a set of axioms by means of logical inference following a human expert. On the other hand, model checking is an automated technique in which the state space of a system model is systematically explored to guarantee that it satisfies given properties.

To be specific, in the context of this thesis about analyzing lane-changing algorithms for autonomous vehicles, we prefer model checking as our technique. This is because model checking, in particular, is very well suited to dynamic system verification of finite state spaces like lane-changing algorithms. It allows us to perform automated analysis of these systems without having to handle probabilistic behaviors found in real-world applications. Although theorem proving and theorem proving systems are laudable, they tend to be complex and require manual intervention, which have themselves limited their practicality for systems as complex and dynamic as those of autonomous vehicles.

For analyzing lane-changing algorithms, we chose model checking over theorem proving because it is faster and able to scale better than theorem proving. It is more effective as volatile traffic can be (and is) mathematically modeled. Probabilistic model checking is especially suitable given the nature of lane-changing algorithms for AVs, which are probabilistic in behaviors such as traffic conditions and sensor inaccuracies. These uncertainties are modeled efficiently using Markov chains and explore large state spaces. Automated exhaustive analysis and error path identification are made possible on tools like PRISM [34], Spin [35], Storm [36] and UPAAL [37]. They can discover error paths, and verify safety and performance properties crucial to the system. Probabilistic model checking is an ideal tool for developing safety-critical applications such as autonomous vehicle lane-changing algorithms because we are able to model and analyze probabilistic behaviors. Such tools allow developers to program the system for certainty, even when the system is otherwise likely to operate under the ambiguities of uncertain and dynamic conditions.

While powerful, theorem proving is not used in this thesis because the theorem proving methods are not sufficient for verifying the complex, dynamic behavior of systems like lane-changing algorithms for autonomous vehicles. Constructing proofs for large and intricate systems is time-consuming, error-prone, and requires substantial manual effort and expert guidance in theorem proving. Also, it does not provide the automation and scalability of model checking, and so is less useful for real applications based on finite state models with probabilistic behavior. However, while verification of lane-changing algorithms has to deal with dynamic and probabilistic behavior, probabilistic model checking offers an efficient and automated procedure to verify uncertain systems, making it a more attractive option.

2.2 Formal Probabilistic Modeling and Verification

A formal framework for analyzing and ensuring the reliability of autonomous vehicle's lane-changing algorithms under uncertain and dynamic conditions is given by probabilistic modeling and verification. This chapter covers the basic concepts and tools, such as the Markov Decision Processes, Probabilistic Computation Tree Logic, and the PRISM model checker, required to formally analyze lane-changing algorithms.

2.2.1 Markov Decision Processes (MDPs)

Markov Decision Processes [19] allow modeling systems that exhibit both nondeterministic and probabilistic behavior. That's why we use *Markov Decision Processes (MDPs)* to model the ego vehicle, surrounding vehicles, and lane-changing algorithms.

Definition 2.1 (MDP). *An MDP is defined as tuple* $\mathcal{M} = (\mathcal{S}, s', \mathcal{A}, \mathcal{P}, AP, L)$*, where:*

- $\mathcal{S}$ *is finite set of states;*
- $s' \in \mathcal{S}$ *is the initial state;*
- $\mathcal{A}$ *is a finite set of actions;*
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ *is a probabilistic transition function, where* $\forall s \in \mathcal{S}, a \in \mathcal{A} : \sum_{s' \in \mathcal{S}} \mathcal{P}(s, a, s') \in \{0, 1\}$*;*

- *AP is a set of atomic propositions;*

- *$L : \mathcal{S} \rightarrow 2^{AP}$ is a labelling function, such that $q \in L(s)$ if and only if q is true in $s \in \mathcal{S}$.*

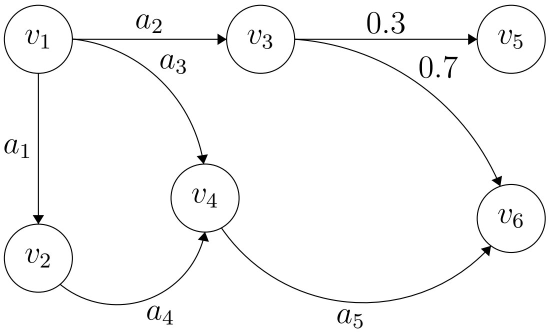

Figure 2.1: A Markov decision process (MDP) with 6 states. The labels of each state are $v_1, v_2, v_3, v_4, v_5, v_6$. The actions are a_1, a_2, a_3, a_4, a_5.

In each state s of an MDP $\mathcal{M}$, a choice is made between the actions that are enabled in s. These actions form the set $\mathcal{A}_s$ =$\{a \in \mathcal{A} \mid \mathcal{P}(s, a, s') > 0 \ for \ some \ s \in \mathcal{S}\}$. When an action $a \in \mathcal{A}_s$ is selected in state s, the probability of transitioning to the next state s' is give by $\mathcal{P}(s, a, s')$. A sequence of such transitions, denoted $\sigma = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \ldots$, where $\mathcal{P}(s_i, a_i, s_{i+1}) > 0$ for $i \in \mathbb{N}$, represents an (infinite) path through the MDP. A finite path $\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \ldots \xrightarrow{a_{n-1}} s_n$ is defined as prefix of an infinite paths. The set of all finite and infinite paths of $\mathcal{M}$ starting from state s are denoted by $\mathsf{FPath}_{\mathcal{M},s}$ and $\mathsf{IPath}_{\mathcal{M},s}$, respectively. The choice of action to take at each step of the execution of an MDP $\mathcal{M}$ is made by a policy, which can base its decision on the history of $\mathcal{M}$ up to the current state.

Definition 2.2 (Policy). *A policy for MDP $\mathcal{M}$ is a function $\pi : FPath_{\mathcal{M},s'} \rightarrow \mathcal{A}$ such that, for any path ρ ending in state s_n, we have $\pi(\rho) \in \mathcal{A}_{s_n}$.*

In this work, we will use *memoryless policies* $\pi : \mathcal{S} \to \mathcal{A}$, which only base their choice of action on the current state, and *finite-memory policies*, which track a finite set of "modes" needed, in conjunction with the current state, to choose an action. For a particular policy π, we can define a probability space $\Pr^{\pi}_{\mathcal{M},s}$ over the set of infinite paths $\mathsf{IPath}_{\mathcal{M},s}$.

Furthermore, for a measurable function $X : \mathsf{IPath}_{\mathcal{M},s} \to \mathbb{R}$, we write $E^{\pi}_{\mathcal{M},s}(X)$ for the expected value of X with respect to $\Pr^{\pi}_{\mathcal{M},s}$.

Finally, we define MDP *reward structures.* We use a variant that assigns non-negative values to state-action-state triples.

Definition 2.3 (Reward Structure)**.** *A reward structure in the context of an MDP* $\mathcal{M} = (\mathcal{S}, s', \mathcal{A}, \mathcal{P}, AP, L)$*, is formally defined as a function* $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geq 0}$*, where* $\mathcal{R}(s, a)$ *assigns a real-valued positive reward to each transition between states* s *and* s' *due to action* a*.*

Of particular interest in this thesis is the expected minimum cumulative reward under constraints related to critical state penalties and lane changes until a target is reached.

Definition 2.4 (Expected Cumulative Reward)**.** *For a reward structure* $\mathcal{R}$ *on an MDP* $\mathcal{M}$ *and a target label* $b \in AP$*, we define the function* $cumul^b_{\mathcal{R}}$ *as:*

$$cumul^b_{\mathcal{R}}(s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \ldots) = \sum_{i=0}^{n_b - 1} \mathcal{R}(s_i, a_i, s_{i+1}),$$

where n_b is the first index for which $b \in L(s_{n_b})$. For the cases where $b \notin L(s_i)$ $\forall i$, we define $n_b = \infty$.

The expected cumulative reward under a policy π on $\mathcal{M}$ is then defined as:

$$E^{\pi}_{\mathcal{M},s}(cumul^{b}_{\mathcal{R}}),$$

where the expectation is computed over the possible paths generated by following policy π, starting from state s.

2.2.2 Probabilistic Computation Tree Logic (PCTL)

Probabilistic Computation Tree Logic (PCTL) [20] is the extension of Computation Tree Logic (CTL) [38] which allows reasoning about infinite sequences of states..The PCTL grammar syntax defines state formulas and path formulas for reasoning about probabilistic systems.

Definition 2.5 (PCTL Syntax). *A state formula ϕ is defined as:*

$$\varphi ::= true \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid P_{\sim q}[\phi],$$

where p is an atomic proposition ($p \in AP$), $\sim$ is a comparison operator ($\sim \in \{<, \leq, >, \geq\}$), q is a probability threshold ($q \in [0, 1]$), and ϕ is a path formula.

Path formulas, denoted by ϕ, are defined as:

$$\phi ::= X\varphi \mid \varphi U \varphi \mid \varphi R \varphi.$$

In this grammar, $X\varphi$ means φ holds in the next state, $\varphi U \psi$ means ψ eventually holds with φ continuously true until then, and $\varphi R \psi$ means φ holds until ψ or forever if ψ never holds.

The **operator semantics** include logical operators $\neg\varphi$ (negation) and $\varphi \wedge \psi$ (conjunction). Probability quantification is expressed using $P_{\sim q}[\phi]$, specifying that the probability of ϕ satisfies $\sim q$. Derived operators include *eventually* $(F\varphi)$ defined as true $U\varphi$, and *always* $(G\varphi)$ defined as $\neg F \neg \varphi$. These allow reasoning about probabilistic temporal properties in a system.

2.2.3 PCTL Specifications for MDPs

For a Markov Decision Process (MDP) $\mathcal{M}$ and a PCTL formula φ defined over atomic propositions AP, the probability of a path satisfying φ from state s under policy π is denoted by $Pr^{\pi}_{\mathcal{M},s}(\varphi)$. This is given as:

$$Pr^{\pi}_{\mathcal{M},s}(\varphi) = Pr^{\pi}_{\mathcal{M},s}(\{\sigma \in \mathsf{IPath}_{\mathcal{M},s} \mid \sigma \models \varphi\}),$$

where $\text{IPath}_{\mathcal{M},s}$ represents all paths starting from s. The maximum probability of satisfying φ across all policies is $Pr^{\max}_{\mathcal{M},s}(\varphi)$.

Additionally, the **expected accumulated reward** until a co-safe PCTL formula φ is satisfied is defined for reward structure r as:

$$\text{cumul}^{\pi}_{\mathcal{R}}(s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \dots) = \sum_{i=0}^{n_{\varphi}-1} \mathcal{R}(s_i, a_i, s_{i+1}),$$

where n_{φ} is the first index where φ holds. The expected reward under π is $\mathbb{E}^{\pi}_{\mathcal{M},s}(\text{cumul}^{\pi}_{\mathcal{R}})$, and its maximum over all policies is $\mathbb{E}^{\max}_{\mathcal{M},s}(\text{cumul}^{\pi}_{\mathcal{R}})$.

2.2.4 PRISM Model Checker

The probabilistic model checker PRISM [34] is a widely used probabilistic model checker designed for analyzing various probabilistic models, including MDPs. PRISM provides a high-level modeling language, i.e., the PRISM language, based on the Reactive Modules formalism for constructing and analyzing complex systems [39]. In PRISM, models are described as a system of interacting modules where a module consists of a finite set of variables, which, taken as a whole, describes the state of the corresponding module. The transitions of a module are specified through a series of guarded commands in the following format:

```
[action] <guard> → <prob>:<update> + ··· + <prob>:<update>
```

A command does contain a guard, and optionally an action label, and a probabilistic choice between updates. A guard is a logical predicate over variables, which allows command execution only when it is true. The system moves to a new state according to the specified updates with probabilities given by `⟨prob⟩`. Actions facilitate the interaction between modules by allowing synchronization and communication between several components.

Support for reward structures in PRISM is provided through reward commands, defined as follows:

```
<action><guard>:<reward>
```

representing the rewards accumulated when taking an action in a state that is satisfying the guard.

Chapter 3

Formal Framework for Lane Change Analysis

This chapter lays down the formal groundwork needed to analyze and verify lane-changing algorithms for autonomous vehicles. Vehicle dynamics, surrounding traffic and environmental constraints are modeled precisely for Lane change which is a complex and critical safety maneuver. This section provides a basis for probabilistic model checking and the verifications of algorithmic properties, after we define the key concepts and definitions used in the formalization.

3.1 Formal Definitions and Theoretical Constructs

This section introduces the formal definitions, propositions, and lemmas that establish the foundational framework for analyzing lane-changing algorithms. Each concept is defined rigorously to facilitate the development of verifiable and testable models.

To formalize the modeling of a roadway environment for AVs, we first define the scenario configuration, which outlines the basic structure and components of the roadway.

Definition 3.1 (Scenario Configuration)**.** *Consider a roadway with a fixed number of lanes $L >= 1$. An autonomous ego vehicle and multiple other vehicles are situated on this roadway. Each vehicle occupies a position in a discrete set of reference points along and cross the lanes. The state of each vehicle is described at discrete time steps.*

The state of the autonomous ego vehicle is characterized by key parameters that describe its operational status, position, and speed within the roadway environment.

Definition 3.2 (Ego Vehicle State)**.** *An ego vehicle is characterized by a state tuple (s, l, v), where: $s \in \mathcal{S}$ is the operational state of the vehicle (e.g., cruising, changing lanes, keeping lane), $l \in \{1, 2, \cdots, N\}$ donates the current lane number, with N being the total number of lanes and $v \in \mathcal{V}$ represents the speed levels of the vehicle, where $\mathcal{V}$ is a finite set of possible speeds.*

Similarly, to account for interactions with other vehicles, we define the surrounding vehicle states, which detail the presence, speed, and distance of other vehicles relative to the ego vehicle.

Definition 3.3 (Surrounding Vehicles)**.** *For each position relative to the ego vehicle (e.g., ahead in the current lane, behind in right lane), we define: $presence_i \in \{true, false\}$ is a Boolean variable indicating the presence of a vehicle at position i, $speed_i \in \mathcal{V}$ is the speed of the vehicle at position i, if present, and $distance_i \in \mathcal{D}$ is the the distance of vehicle at position i from ego vehicle, where $\mathcal{D}$ is a finite set of possible distances.*

Definition 3.4 (State Space)**.** *Let $\mathcal{S}$ be the finite state space capturing all the relevant system aspects. Each state $s \in \mathcal{S}$ is defined by tuple:*

$$s = (EgoVehicleState, \{presence_i, speed_i, distance_i\}_{i \in \mathcal{J}})$$

For each potentially occupied position $i \in \mathcal{J}$ (e.g., front, behind, neighbouring lanes), $presence_i \in \{true, false\}$ and $speed_i \in \mathcal{V} \cup \{0\}$ represents the speed of the vehicle if present, or 0 if not present.

Definition 3.5 (Transition Model)**.** *The system evolves based on a probabilistic transition model:*

$$\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$$

where $\mathcal{P}(\mathcal{S})$ is the probability distribution over the global state space. For a given state s and action $a, \mathcal{T}(s, a)(s')$ represents the probability of transitioning to state s'.

Definition 3.6 (Utility Function)**.** *Ego vehicle is associated with a utility function*

$$\mathcal{U}_{\text{ego}} : \mathcal{S} \times \mathcal{A}_{\text{ego}} \to \mathbb{R}$$

capturing the preferences from various outcomes. The utility function may consider safety, efficiency, and comfort.

The goal of a lane-changing algorithm is to maximize $\mathcal{U}_{\text{ego}}$ under the constraints of the system dynamics.

Definition 3.7 (Feasibility of Lane Change)**.** *A lane change for the ego vehicle from lane l to l' (where $l' \in \{l-1, l+1\}$) is feasible if:*

1. *l' is within the set of lanes $\mathcal{L}$, and*

2. *The lateral gap in lane l' is sufficient to accommodate the ego vehicle i.e., $\forall i \in \mathcal{J}_{l'}$, if $presence_i = true$, then $distance_i \geq d_{min}$*

Where $\mathcal{J}_{l'}$ is the set of positions (e.g., ahead, behind) relative to ego vehicle in lane l', $distance_i$ is the distance between ego vehicle and vehicle at position i and d_min is safety threshold ensuring sufficient clearance for lane change.

Lemma 3.8 (Collision-Free Dynamics)**.** *If the safety threshold d_{min} is strictly enforced, then any sequence of lane-changing actions will result in collision free lane change. Formally:*

$$\forall s \in \mathcal{S},\ \textit{if}\ \forall i \in \mathcal{J}_{l'},\ \textit{if presence}_i = \textit{true, then distance}_i \geq d_{min}.$$

Proof. The safety condition guarantees that the distance between the ego vehicle and vehicles in the target lane is large enough at any point prior to a lane change. The enforcement of $distance_i \geq d_{min}$ prevents any overlap in positions after the lane change, thereby avoiding collision.

Lemma 3.9 (Data Collection Validity)**.** *The ego vehicle must ensure that data collection from surrounding vehicles is complete for the decision-making algorithm to be executable:*

$$DataCollection = \bigwedge_{i=1}^{n} datacheck_i > 0$$

where $datacheck_i$indicates successful data acquisition (e.g., presence and speed) of the surrounding vehicles i.

Proof. The decision-making algorithm of the ego vehicle relies on complete and accurate data from surrounding vehicles to predict their behaviors and ensure safe navigation. Without this data, the algorithm cannot make reliable decisions, rendering it inexecutable.

Proposition 3.10 (Optimization Objective). *The objective of lane-changing algorithm is to maximize the expected cumulative reward over a planning horizon while satisfying all the safety constraints. Formally,*

$$\min_{\sigma} \mathbf{E}^{\sigma} \left[\sum_{t=0}^{T} (s_t, a_t) \right]$$

subject to safety criterion, and operational constraints.

Proof. By formulating the problem as an MDP with associated rewards and penalties, standard optimization techniques can be applied to find the policy σ that yields the highest expected reward, reflecting efficient and safe driving behavior.

3.2 Verification Properties for Lane-Changing Algorithms

Next, we propose some of the key properties that are needed to verify lane-changing algorithms. We formalize these properties to provide a rigorous basis for proving lane-changing behaviors correct and robust in dynamic environments.

3.2.1 Safety Properties

Based on the general formalizations described above, the ego vehicle must always operate without colliding with other vehicles, to ensure that ego vehicle always satisfies the safety criteria in current lane or in target lane while changing the lane.

$$\mathbf{P}_{\geq 1}[\mathbf{G}\ (\phi_{safe})]$$

where ϕ_{safe} is a safety invariant ensuring sufficient clearance and no collision.

3.2.2 Liveness Properties

These are some liveness properties which can be verified for the formal analysis of lane-changing algorithms.

3.2.2.1 Eventual Progress

The ego vehicle must eventually take an action to ensure the system progress:

$$\mathbf{P}_{>0}[\mathbf{F}\ (action \in stay, change\_lane)]$$

3.2.2.2 Avoiding stagnation

The ego vehicle should avoid remaining in an unsafe or critical state indefinitely:

$$\mathbf{P}_{>0}[\mathbf{F}\ (\phi_{safe})]$$

3.2.2.3 Lane Change Feasibility

The ego vehicle must eventually have the ability to perform a safe lane change:

$$\mathbf{P}_{>0}[\mathbf{F}\ (action = change \wedge \phi_{safetarget})]$$

3.2.3 Performance Properties

Now we present some performance-based properties which are:

3.2.3.1 Time Efficiency

The algorithm should maximize the time efficiency:

$$\mathbf{E}_{max}[Time\ Efficiency]$$

3.2.3.2 Minimizing Critical State

The ego vehicle should avoid spending excessive time in critical states:

$$\mathbf{E}_{max}[Time\ in\ Critical\ States]$$

3.2.3.3 Avoiding Excessive Deceleration

The ego vehicle should minimize deceleration event unless necessary for safety:

$$\mathbf{E}_{min}[Total\ Deceleration]$$

3.3 Concluding Remarks

In this chapter, we began by defining the formal constructs necessary for analyzing lane-changing algorithms, including state space, transition models, and utility functions. We then discussed the key verification properties categorized into safety, liveness, and performance. The motivations of this chapter were twofold: to establish a rigorous foundation for evaluating lane-changing algorithms and to provide a systematic approach for verifying their safety, feasibility, and efficiency. The next chapter will delve into the probabilistic model checking of these algorithms, building on the formal framework presented here.

Chapter 4

Case Study: Formalization and Verification of MOBIL

In this chapter, we present the formal foundations required for analyzing and verifying the MOBIL algorithm, a widely recognized framework for lane-changing decisions in autonomous vehicles. Lane-changing is a critical aspect of autonomous driving, requiring meticulous modeling of vehicle behavior, interactions with surrounding traffic, and adherence to safety and feasibility constraints. This chapter introduces the core definitions, lemmas, and propositions necessary for formalizing the MOBIL algorithm, serving as a foundation for probabilistic model checking and verification of its properties. The formalization encompasses essential components, such as incentive and safety criteria, lane change feasibility, and factors like politeness and traffic density, ensuring a comprehensive understanding of the algorithm's functionality. Additionally, we categorize and verify the algorithm's safety, liveness, and performance properties, providing a systematic evaluation of its effectiveness in dynamic traffic scenarios.

4.1 Formalization of MOBIL

The MOBIL model [40] is being considered as a de facto benchmark lane-changing model in traffic simulations, and as such is a useful proxy for understanding and evaluating automated driving behaviors. Therefore, analysis results from MOBIL can be viewed as an informative starting point for the design and verification of more advanced lane-changing algorithms towards safer and more reliable autonomous driving systems.

The MOBIL lane-changing model is formalized where the speed of the ego vehicle is discretized with three distinct states as defined in *definition **3.2***. Secondly, this abstraction makes the continuous nature of speed manageable for our lane-changing decisions by simplifying it into manageable categories. It discretizes the dynamics and simplifies its analysis while preserving the important dynamics to the model to capture all the corner cases.

Definition 4.1 (Ego Vehicle State in MOBIL)**.** *An ego vehicle in the MOBIL model is characterized by a state tuple* (s, l, v, a)*, where:* $s \in \{0, 1, 2, 3, 4, 5, 6, 7\}$ *representst the operational state,* $l \in \{1, 2, 3\}$ *denotes the current lane number,* $v \in \{1, 2, 3\}$ *represent the discretized speed level:* $v = 1 : 0 - 40km/h$,$v = 2 : 41 - 80km/h$*, and* $v = 3 : 81 - 120km/h$ *and* $a \in \{0, 1\}$ *indicates whether the ego vehicle has collected environmental data* $(a = 1)$*.*

Building upon *definition **3.3*** of surrounding vehicles, we define specific positions relative to the ego vehicle that are critical for lane-changing decisions in MOBIL. We have made some assumptions that an ego vehicle only considers surrounding vehicles within its unsafe distance. Relative positions from vehicles across three lanes are identified to locate surrounding vehicles. For example, the surrounding vehicles of an ego vehicle in the middle lane (Lane 2) is at positions 1,2,3,4,6,7,8, and 9 depending on their presence. The ego vehicle position in the center lane is depicted in Figure 4.1 and the corresponding position in the surrounding vehicles is also shown in Figure 4.2. Consider the case where the vehicle that is directly in front of the ego vehicle (Position 2) is within the unsafe distance. In this scenario, the ego vehicle must make a lane change.

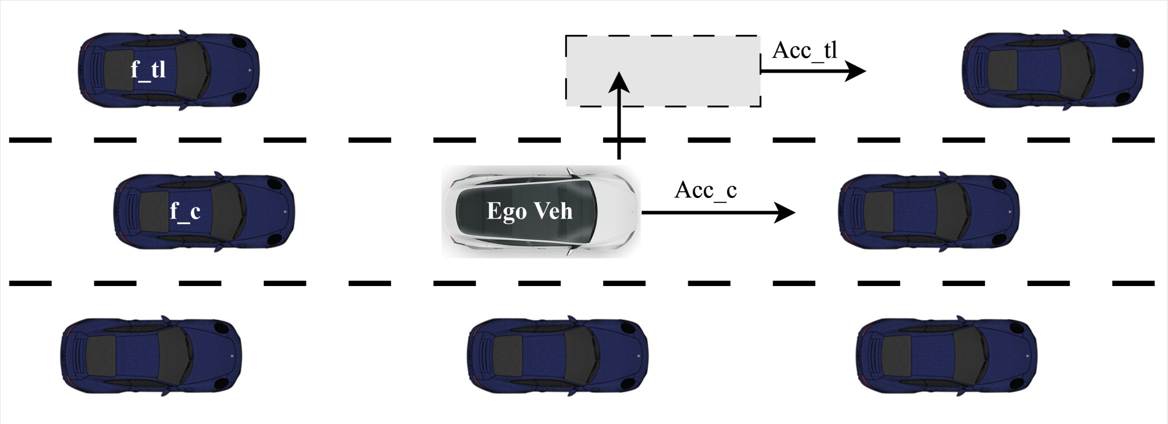

Figure 4.1: Ego Vehicle considering lane change to left as vehicle is in front (old and new followers are denoted by f_c and f_{tl} respectively).

This detailed enumeration of positions ensures MOBIL comprehensively accounts for all relevant surrounding vehicles during decision-making.

Definition 4.2 (Surrounding Vehicles in MOBIL)**.** *For each relevant position relative to the ego vehicle, MOBIL defines: $presence_i \in \{true, false\}$ indicates the presence of a vehicle at position i and $speed_i \in \{0, 1, 2, 3\}$ is discretized speed of the vehicle at position i, where $speed_i = 0$ is when no vehicle is present, or data not collected. Other speeds are the same as for ego vehicle defined in definition 4.1.*

Definition 4.3 (Incentive Criterion in MOBIL)**.** *The incentive criterion in MOBIL evaluates the net benefit of lane change by considering both the ego vehicle's acceleration*

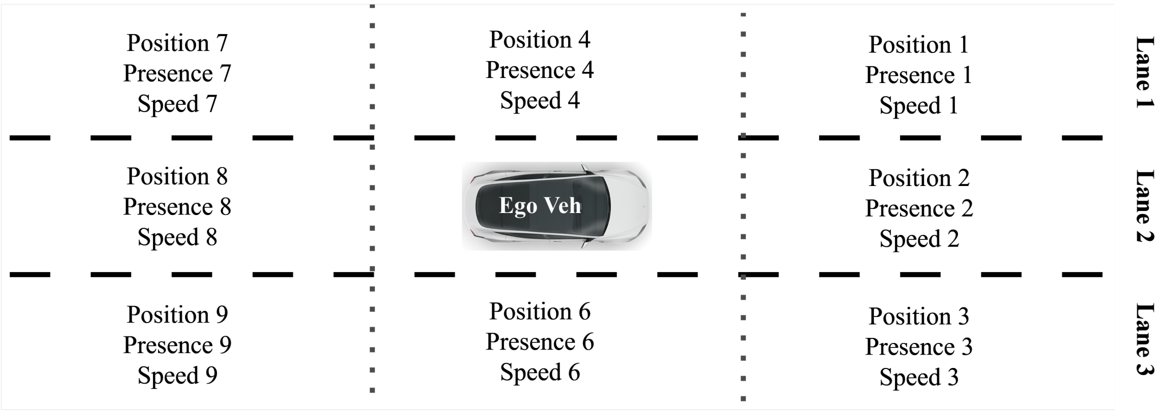

Figure 4.2: Ego Vehicle Lane Change and Relative Positions of Surrounding Vehicles.

and the impact on surrounding vehicles:

$$Incetive\ Criterion = \Delta acc_{ego} + p(\Delta acc_{new\_follower} + \Delta acc_{old\_follower}) > a_{thr}$$

Definition 4.4 (Safety Criterion in MOBIL). *The safety criterion in MOBIL ensures that any lane change does not adversely affect the safety of surrounding vehicles:*

$$Safety\ Criterion = acc_{ft} \geq b_{safe}$$

More specifically, for left and right lane,

$$Safety\_left = acc_{ftl} \geq b_{safe} \text{ and } Safety\_right = acc_{ftr} \geq b_{safe}$$

Where $acc_{f_{tl}}$is the acceleration of new follower in target lane (left or right) after the lane change and b_{safe} is the safety threshold (e.g,$b_{safe} = -1$) represents the maximum acceptable deceleration to prevent unsafe conditions.

Definition 4.5 (Lane Change Feasibility Conditions in MOBIL). *A lane change to the left or the right in MOBIL is feasible if all the following conditions are met:*

1. *Incentive Criterion* $> a_{thr}$

2. *Safety Criterion holds true*

3. *Lane Availability*

Proposition 4.6 (Lane-Changing Decision Rule in MOBIL)**.** *An ego vehicle employing the MOBIL model will initiate a lane change to l' iff all feasibility conditions in Definition **4.5** are satisfied.*

Proof. If any of the feasibility conditions (Incentive, Safety, or Lane Availability) are not satisfied, initiating the lane change would either fail to provide sufficient benefit, compromise safety, be impossible due to lane constraints, or violate regulations. Therefore, in the absence of any of these conditions, the lane change cannot be initiated. On the other hand, if all feasibility conditions are met, the lane change offers a net benefit, does not compromise safety, and adheres to all necessary constraints and regulations. In this case, initiating the lane change is both appropriate and justified.

Lemma 4.7 (Safety Precedence)**.** *If the Safety Criterion is not satisfied, the ego vehicle must not change lanes, regardless of Incentive Criterion.*

Proof. In autonomous vehicle operations, safety is paramount. Even if a lane change offers significant benefits to the ego vehicle (e.g., increased speed), violating the Safety Criterion can lead to hazardous situations such as collisions or forced braking of other vehicles. Therefore, safety considerations override any potential incentives for lane changing.

Lemma 4.8 (Collision-Free Lane Changes)**.** *If the safety criterion is strictly enforced, then lane changes will always be collision-free:*

$$\forall s \in \mathcal{S}, Safety_{left} \lor Safety_{right} \Rightarrow no\ Collisions$$

Proof. The safety criterion requires that the acceleration of the following vehicle in the target lane does not fall below b_{safe}, ensuring sufficient safety margin to prevent collisions.

Defining safety-critical states is essential for ensuring the system's ability to respond effectively to hazardous scenarios. A formal definition of such states is outlined below:

Definition 4.9 (Critical State)**.** *A Critical State in MOBIL occurs when the ego vehicle cannot safely continue in the current lane or execute a safe lane change. Formally:*

$$Critical\ State \Rightarrow \neg Safety_{Current} \land \neg(Safety_{left} \lor Safety_{right})$$

In such a state, the vehicle may need to perform emergency maneuvers or take corrective actions to maintain safety.

Definition 4.10 (Politeness Factor)**.** *The Politeness Factor p in MOBIL reflects the ego vehicle's consideration for the acceleration changes of other drivers.* $p = 0$*: Egoistic behavior, where only ego vehicle's benefits are considered in lane-changing decisions.* $p = 1$*: Altruistic behavior, where the ego vehicle also accounts for the impact of its action on surrounding vehicles' acceleration.*

Lemma 4.11 (Effect of Traffic Density on Politeness factor)**.** *Higher politeness factors p generally lead to fewer lane changes by the ego vehicle, as it becomes more considerate of the impact on surrounding vehicles. However, the relationship between p and lane changes is influenced by traffic density, with counterintuitive behaviors possible at extreme politeness levels or under specific conditions (e.g., discretized models, traffic density effects).*

Proof. The politeness factor p in MOBIL scales the ego vehicle's consideration for surrounding vehicles during lane changes. At $p = 0$, the ego vehicle prioritizes only its own acceleration incentive, often resulting in fewer cooperative lane changes, especially in dense traffic. At $p = 1$, the ego vehicle considers others' incentives, leading to more cooperative lane changes that optimize overall traffic flow. This relationship is influenced by traffic density and discretization effects, which may amplify counterintuitive behaviors.

4.2 Verification Results

In this section, we assess the correctness and performance of MOBIL based on the proposed probabilistic model checking approach. We mainly verify the safety, liveness, and performance properties under different MOBIL parameter configurations by building on the generic properties, outlined in Chapter 3.

4.2.1 Safety Properties

We have verified some safety properties to ensure the safety of ego vehicle, which are:

4.2.1.1 Deadlock

A deadlock property ensures that the ego vehicle will not indefinitely wait to change lanes due to conflicting conditions or resource availability. Hence, we verify the following property to ensure that.

$$\mathbf{E}[\mathbf{G} \neg \text{"deadlock"}]$$

4.2.1.2 Recovery from Unsafe Conditions

$$\mathbf{P} \geq 0.90 \left[(\text{"Critical\_State"})\, \mathbf{U} \leq 10\, (s = 0)\right]$$

Once the ego vehicle enters a critical state, this property confirms it can return to a safe driving condition within a bounded number of steps. By verifying “Recovery from Unsafe Conditions” we ensure the system cannot remain indefinitely in an unsafe configuration, underscoring the vehicle’s ability to promptly regain safety. For example, we define a critical state for the ego vehicle as follows:

$$\textsf{label}\ \text{"Critical\_State"}\ =\ s = 7;$$

This property holds *“true”* in PRISM and ensures that the ego vehicle will safely transition from a critical state to a safe state with at least 90% probability within 10 steps.

4.2.2 Liveness Properties

Here we present some liveness properties that ensure that the ego vehicle is making progress.

4.2.2.1 Eventual Progress

$$\mathbf{P} \geq 1 \left[\mathbf{F}\,(\text{"Lane\_Changed"} \vee \text{"Keeping\_Lane"})\right]$$

This property guarantees with probability 1 that the system will ultimately change lanes or remain in its lane.”*Keeping_Lane*” and ”*Lane_Changed*” are the representation of the states in which the vehicle has been keeping the lane and changing the lane, respectively, in the model and given as:

$$\textsf{label}\ \text{"Lane\_Changed"} = s = 2 \vee s = 3;$$

$$\textsf{label}\ \text{"Keeping\_Lane"} = s = 1;$$

In the PRISM model checker, this property holds *"true"*, which ensures that the system ensures progress to one of these two outcomes.

4.2.2.2 Eventually Make a Justified Lane Change

This property in PRISM is expressed as:

$$\mathbf{P_{max}} = ?[\mathbf{F}((incentive\_left \lor incentive\_right) \land "Lane\_Changed")]$$

This property computes the maximum probability that, at some later time in the future, the system performs a lane change ($Lane\_Changed$) with incentive to change the lane ($incentive\_left$ or $incentive\_right$). This makes sure that lane change takes place only with the existence of a valid incentive. The incentive formulas are defined as:

$$\text{incentive\_left} = \Delta\text{acc}_{\text{ego\_left}} + p \cdot (\Delta\text{acc}_{\text{new\_follower\_left}} + \Delta\text{acc}_{\text{old\_follower}}) > a_{thr}$$
$$\text{incentive\_right} = \Delta\text{acc}_{\text{ego\_right}} + p \cdot (\Delta\text{acc}_{\text{new\_follower\_right}} + \Delta\text{acc}_{\text{old\_follower}}) > a_{thr}$$

PRISM returns $P_{max} = 1$, indicating that ego vehicle is guaranteed to eventually perform a justified lane.

4.2.3 Performance Properties

Using reachability rewards [39] in PRISM, we evaluate our lane-changing model (MOBIL) under different parameter configurations with respect to various reward-based properties. Two of these properties, in particular, will guide our selection of the optimal parameter values:

$$\mathbf{R}\{Lane\_Changes\}_{min} = ?\ [\mathbf{C} \leq \mathbf{T}]$$

We leverage upon cumulative reward properties [39] to find the minimum expected number of lane changes by ego vehicle within time bound T in our model. Similarly,

$$\mathbf{R}\{Critical\_State\_Penalty\}_{min} = ?\ [\mathbf{C} \leq \mathbf{T}]$$

which measures the minimum expected critical state penalty accumulated before termination within time-bound T.

We instantiate our model with varying values of two MOBIL parameters: the politeness factor, denoted as p), which takes values from the set $\{0, 0.25, 0.50, 0.75, 1\}$, and the acceleration threshold or incentive criterion, denoted by a_{thr}, which can be 0.1, 0.5, or 1. Table 4.1 summarizes the resulting variables of these two parameters after 200 discrete steps (T=200). Each cell corresponds to (p, a_{thr})) setting and shows Minimum Expected Lane Changes and Minimum Expected Critical State (penalty).

Overall, a higher politeness factor p generally increases lane changes since the ego vehicle accommodates its neighbors more frequently. Meanwhile, lowering the incentive threshold (a_{thr}) tends to boost the number of lane changes but can reduce the likelihood of entering high-risk states. The fewest lane changes (2.452) occur with ($p = 0, a_{thr} = 1$, though this setting has a relatively high critical-state penalty (4.481). Conversely, the lowest penalties (around 3.746–3.754) appear when p is higher or a_{thr} is lower, at the cost of increased maneuvers. Balancing these factors, a moderate setting—such as ($p = 0.5, a_{thr} = 1$)—offers a middle ground between limiting lane changes and

Table 4.1. Minimum Expected Lane Changes and Critical States for Different Parameters

p	$\mathbf{a_{thr}}$	**Minimum Expected Lane Changes**	**Minimum Expected Critical State**
0	0.1	3.767	4.324
0.25	0.1	4.744	3.885
0.5	0.1	5.025	3.77
0.75	0.1	5.499	3.746
1	0.1	5.492	3.746
0	0.5	3.767	4.324
0.25	0.5	3.847	4.002
0.5	0.5	4.693	3.828
0.75	0.5	5.067	3.754
1	0.5	5.492	3.746
0	1	2.452	4.481
0.25	1	3.364	4.085
0.5	1	3.847	3.885
0.75	1	4.725	3.77
1	1	4.767	3.754

mitigating risky maneuvers. Accordingly, the subsequent properties will be verified using this moderate parameter configuration ($p = 0.5, a_{thr} = 1$).

4.2.3.1 Minimum Expected Number of Lane Changes

$$\mathbf{R}\{Lane\_Changes\}_{min} = ?\ [\mathbf{C} \leq 200]$$

We performed the experiment again, measuring Minimum Expected Lane Changes over

200 discrete steps, this last time with a moderate threshold parameter of $a_{thr} = 1$. The graphs resulting from this are shown in Figure 4.3 and prove Lemma 4. In general, we find that lane changes decrease as politeness factors p increase because the ego vehicle is more considerate. However, under predetermined discretization conditions or when the levels of politeness are extreme, traffic density can lead to counterintuitive behaviors.

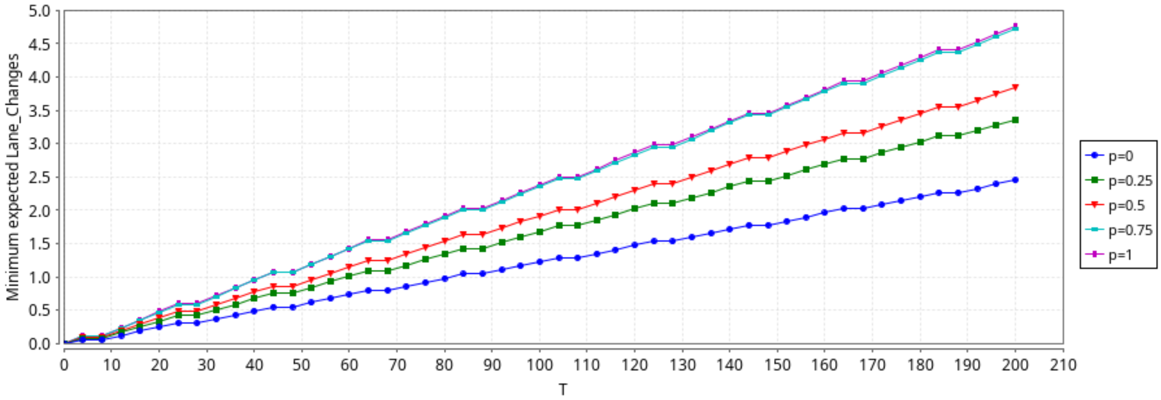

Figure 4.3: Expected Lane Changes over Time Steps

4.2.3.2 Minimum Expected Time Efficiency

This property is defined as:

$$\mathbf{R}\{Time\_Efficiency\}_{min} = ?\ [\mathbf{C} \leq 200]$$

We performed an experiment to evaluate the Minimum Expected Time Efficiency for $T \leq 200$. As shown in Figure 4.4, the ego vehicle is more time-efficient at $p = 1$ than at $p = 0.5$ and $p = 0$, demonstrating that it can maintain safety while increasing its lane-changing rate to achieve better travel times.

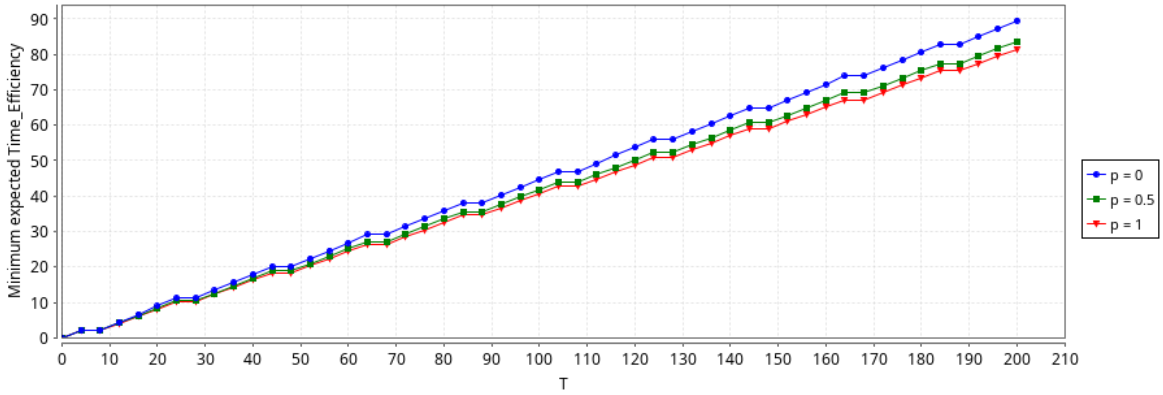

Figure 4.4: Time Efficiency at different p values

4.2.3.3 Minimum Deceleration

$$\mathbf{R}\{Acceleration\_Management\}_{min} = ?\ [\mathbf{F}\ Lane\_Changed \vee Critical\_State]$$

We conducted this experiment at $a_{thr} = 1$ with varying values of politeness factor p, the results as shown in Figure 4.5, reveal that $p = 0.5$ outperforms $p = 0$ yet remains slightly below $p = 1$.

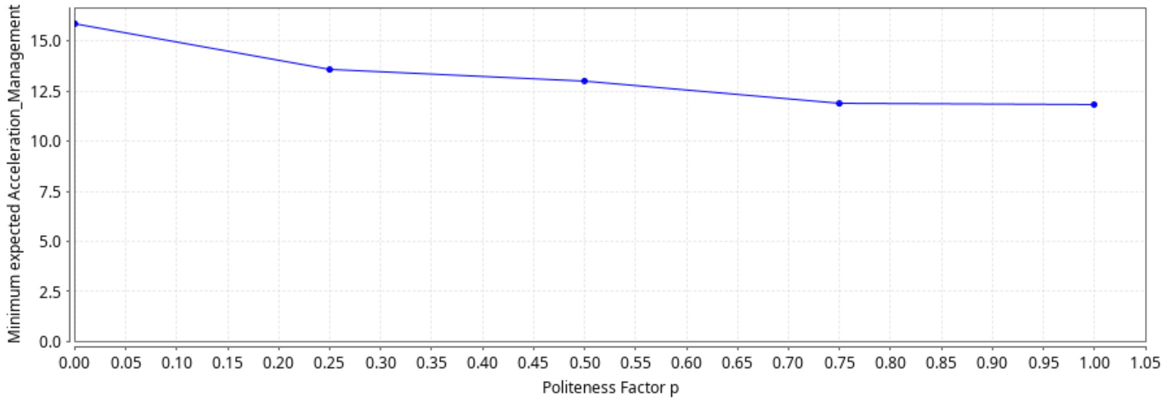

Figure 4.5: Minimum Deceleration over different politeness factor

However, since $p = 1$ leads to an increased number of critical states, $p = 0.5$ and $a_{thr} = 1$ represent a balanced choice to achieve both safety and efficiency.

Comparing the proposed PRISM based verification results with the simulation-based results presented in the original MOBIL paper [40], highlights that both the approaches are complementary in nature, and the proposed approach has some distinguishing advantages. As an example, the simulation results in MOBIL paper show a peak of 1,100 lane changes per hour per km under symmetric rules ($p = 0$) and 600 changes under $p = 1$ (more cooperating behavior). These results are useful for understanding the behavior of real vehicles in real time, but the proposed based analysis provides a systematic quantification of these dynamics as a function of varying parameters and demonstrates the existence of balanced configurations like $p = 0.5$, $a_{thr} = 1$ which result in a moderate amount of lane changes (4.767) while keeping the penalty under critical states to a minimum (3.754). Moreover, unlike the MOBIL simulations which observe safety through emergent behaviors as well as velocity adjustments, our approach guarantees critical properties, such as $a \geq 90\%$ recovery probability from unsafe states and absence of deadlocks for all modeled conditions. Additionally, the MOBIL simulations focus on emergent dynamics in particular scenarios leading to challenges on generalizing results or examining rare edge cases. However, in comparison our approach, completely checks all possible states and so is correct and robust by definition beyond the scope of simulation studies. On the other hand, the models used in simulation-based analysis are closer to reality compared to our discretized models. These comparisons show that the two approaches are complementary in nature and have to play together to provide more comprehensive insights about the underlying lane-changing algorithm.

4.3 Concluding Remarks

In this chapter, we formalized and verified the MOBIL algorithm for autonomous vehicle lane-changing by constructing a rigorous mathematical framework. The properties analyzed were safety, liveness and performance. The liveness properties ensured the support to smooth traffic flow for the algorithm while the safety properties ensured the prevention of deadlocks and unsafe conditions. Based on this analysis, parameter configurations were identified that exhibit a tradeoff between lane change frequency and critical penalty minimization.

The formal methods were shown to be robust for analyzing lane-changing algorithms, with high probability recovery from unsafe states and guarantees for avoiding deadlock. This approach revealed balanced configurations that achieved efficiency while maintaining cooperative behavior. In doing so, these findings reconfirm the necessity of building in rigorous analysis in the development of autonomous driving systems from the very beginning for safety and reliability reasons.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we presented a groundbreaking approach to the formal analysis of lane-changing algorithms for AVs using probabilistic model checking. This comprehensive methodology represents a leap forward in systematically formalizing and verifying lane-changing strategies. By employing a generic framework, we not only analyzed critical safety, liveness, and performance properties but also deconstructed the complex lane-changing process into manageable subtasks, enabling separate model utilization for verification.

Our application of this approach to the widely recognized MOBIL algorithm underscored its effectiveness. Through meticulous analysis, we identified key parameters that significantly impact lane-changing decisions and elucidated the influence of varying thresholds and parameter values. Furthermore, we delved into the effects of traffic density on the ego vehicle's decision-making, uncovering nuanced insights into this pivotal aspect. This research establishes a robust foundation for the ongoing formal analysis of lane-changing algorithms, reaffirming the indispensable role of formal methods in enhancing the safety and reliability of AVs.

5.2 Future Work

To expand the horizons of this research, several directions can be explored:

- **Symmetric vs. Asymmetric Traffic Rules**: While our research has been thoroughly conducted under symmetric traffic rules, future studies can boldly delve into the complexities of asymmetric traffic rules, mirroring the intricate and diverse traffic networks found across Europe and other regions. This exploration can potentially reveal novel insights into the adaptability of autonomous systems.

- **Multi-Agent Scenarios**: A transformative leap can be made by extending the proposed approach to encompass dynamic interactions between multiple autonomous agents. This expansion will not only address cooperative strategies but also pioneer understanding in competitive scenarios, paving the way for harmonious coexistence in increasingly crowded autonomous ecosystems.

- **Dynamic and Learning-Based Models**: Pushing the envelope further, future investigations can integrate state-of-the-art dynamic models alongside advanced reinforcement learning techniques. This paradigm shift could enable autonomous systems to evolve in real-time, crafting lane-changing strategies that are not only adaptive but also capable of responding to unforeseen challenges with unparalleled agility.

- **Real-Time Decision-Making**: The frontier of real-time decision-making beckons with its promise of precision and reliability. Developing cutting-edge methodologies to verify and execute split-second decisions can revolutionize the safety and efficiency of lane-changing maneuvers, ensuring impeccable performance under the pressure of real-world demands.

- **Scalability and Efficiency**: To truly elevate this framework, optimizing for scalability is imperative. By enabling the analysis of exceedingly complex traffic scenarios and massive datasets, the research can lay the groundwork for addressing the sprawling networks of modern metropolises and beyond.

- **Broader Application**: The versatility of this framework holds untapped potential for a wider spectrum of lane-changing algorithms. Systematically applying it to diverse methodologies could uncover universal principles, fostering a profound understanding of autonomous behaviors across a multitude of contexts.

This future work will build upon the foundation laid in this study, propelling the field of autonomous vehicle research toward more sophisticated and practical solutions for real-world deployment.

Bibliography

[1] J. Anderson, N. Kalra, K. Stanley, P. Sorensen, C. Samaras, and T. Oluwatola, Autonomous Vehicle Technology: A Guide for Policymakers. RAND Corporation, 2016.

[2] M. Reda, A. Onsy, A. Y. Haikal, and A. Ghanbari, "Path planning algorithms in the autonomous driving system: A comprehensive review," Robotics and Autonomous Systems, vol. 174, p. 104630, Apr. 2024.

[3] S. Moridpour, M. Sarvi, and G. Rose, "Lane changing models: a critical review," Transportation Letters, vol. 2, p. 157–173, July 2010.

[4] S. Posada, "Autopilot & first responder scenes," Preliminary Evaluation Report PE 21-020, National Highway Traffic Safety Administration, August 2021.

[5] SAE International, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," Tech. Rep. J3016, SAE International, 2021. Edition: April 2021.

[6] National Highway Traffic Safety Administration, "Standing general order 2021-01: Incident reporting for automated driving systems (ads) and level 2 advanced driver

assistance systems (adas)," tech. rep., National Highway Traffic Safety Administration, June 2021.

[7] National Transportation Safety Board, "Collision between vehicle controlled by developmental automated driving system and pedestrian," Highway Accident Report NTSB/HAR-19/03, National Transportation Safety Board, Washington, DC, 2019.

[8] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?," tech. rep., RAND Corporation, 2016.

[9] Defense Advanced Research Projects Agency, "DARPA Urban Challenge." `https://www.darpa.mil/about-us/timeline/darpa-urban-challenge`.

[10] T. Wongpiromsarn, Formal Methods for Design and Verification of Embedded Control Systems: Application to an Autonomous Vehicle. PhD thesis, California Institute of Technology, 2010.

[11] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus, and M. Ang, "Perception, planning, control, and coordination for autonomous vehicles," Machines, vol. 5, p. 6, Feb. 2017.

[12] N. J. Goodall, "Ethical decision making during automated vehicle crashes," Transportation Research Record: Journal of the Transportation Research Board, vol. 2424, p. 58–65, Jan. 2014.

[13] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," ROBOMECH Journal, vol. 1, July 2014.

[14] M. Treiber and A. Kesting, Traffic Flow Dynamics: Data, Models and Simulation. Springer Berlin Heidelberg, 2013.

[15] H. Krasowski, X. Wang, and M. Althoff, "Safe reinforcement learning for autonomous lane changing using set-based prediction," in 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), IEEE, Sept. 2020.

[16] R. Zhou, H. Cao, J. Huang, X. Song, J. Huang, and Z. Huang, "Hybrid lane change strategy of autonomous vehicles based on soar cognitive architecture and deep reinforcement learning," Neurocomputing, vol. 611, p. 128669, Jan. 2025.

[17] O. Hasan and S. Tahar, "Formal verification methods," in Encyclopedia of Information Science and Technology (M. Khosrow-Pour, ed.), pp. 7162–7170, IGI Global, 2015.

[18] C. Baier and J.-P. Katoen, Principles of Model Checking. Cambridge, MA: The MIT Press, 2008.

[19] M. L. Puterman, Markov decision processes. Wiley Series in Probability & Mathematical Statistics: Applied Probability & Statistics, Nashville, TN: John Wiley & Sons, May 1994.

[20] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," in Formal Aspects of Computing, vol. 6, pp. 512–535, Springer, 1994.

[21] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, p. 269–271, Dec. 1959.

[22] P. Hart, N. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, p. 100–107, 1968.

[23] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” IEEE Transactions on Robotics and Automation, vol. 12, no. 4, p. 566–580, 1996.

[24] S. M. LaValle, Planning Algorithms. Cambridge, UK: Cambridge University Press, 2006.

[25] S. Karaman and E. Frazzoli, “Incremental sampling-based algorithms for optimal motion planning,” 2010.

[26] Z. Yang, Z. Wu, Y. Wang, and H. Wu, “Deep reinforcement learning lane-changing decision algorithm for intelligent vehicles combining lstm trajectory prediction,” World Electric Vehicle Journal, vol. 15, p. 173, Apr. 2024.

[27] M. Ghimire, M. R. Choudhury, and G. S. S. H. Lagudu, “Lane change decision-making through deep reinforcement learning,” 2021.

[28] H. Tian, C. Wei, C. Jiang, Z. Li, and J. Hu, “Personalized lane change planning and control by imitation learning from drivers,” IEEE Transactions on Industrial Electronics, vol. 70, p. 3995–4006, Apr. 2023.

[29] J. Shi, T. Zhang, J. Zhan, S. Chen, J. Xin, and N. Zheng, “Efficient lane-changing behavior planning via reinforcement learning with imitation learning initialization,” in 2023 IEEE Intelligent Vehicles Symposium (IV), p. 1–8, IEEE, June 2023.

[30] A. Zita, S. Mohajerani, and M. Fabian, "Application of formal verification to the lane change module of an autonomous vehicle," in 2017 13th IEEE Conference on Automation Science and Engineering (CASE), p. 932–937, IEEE, Aug. 2017.

[31] X. Yang, Y. Wei, L. Shi, and L. Chen, "Applying probabilistic model checking to path planning for a smart multimodal transportation system using iot sensor data," Mobile Networks and Applications, vol. 28, p. 382–393, Feb. 2023.

[32] A. Dhonthi, N. Schischka, E. M. Hahn, and V. Hashemi, "Autonomous vehicles path planning under temporal logic specifications," 2024.

[33] B. Chen and T. Li, "Formal modeling and verification of autonomous driving scenario," in 2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE), p. 313–321, IEEE, Mar. 2021.

[34] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in Computer Aided Verification, Lecture notes in computer science, pp. 585–591, Springer Berlin Heidelberg, 2011.

[35] G. Holzmann, "The model checker spin," IEEE Transactions on Software Engineering, vol. 23, p. 279–295, May 1997.

[36] C. Hensel, S. Junges, J.-P. Katoen, T. Quatmann, and M. Volk, "The probabilistic model checker storm," Int. J. Softw. Tools Technol. Transf., vol. 24, pp. 589–610, Aug. 2022.

[37] K. G. Larsen, P. Pettersson, and W. Yi, "Uppaal in a nutshell," International Journal on Software Tools for Technology Transfer, vol. 1, p. 134–152, Dec. 1997.

[38] E. M. Clarke and E. A. Emerson, "Design and synthesis of synchronization skeletons using branching time temporal logic," in Logics of Programs, pp. 52–71, Berlin/Heidelberg: Springer-Verlag, 2005.

[39] "Prism model checker manual." `http://www.prismmodelchecker.org/manual`, 2023. Accessed: 2025-01-03.

[40] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," Transportation Research Record: Journal of the Transportation Research Board, vol. 1999, p. 86–94, Jan. 2007.

LIST OF PUBLICATIONS

1. **MB. Sarwar**, and O. Hasan, “Formal Analysis of Lane Changing Algorithms using Probabilistic Model Checking,” *Submitted to NASA Formal Methods (NFM-2025) (Under Review).*

2. **MB. Sarwar**, GM. Raza, MA. Sarwar, and B.-S. Kim, “Revolutionizing ICT with AI and ML: A Comprehensive Study of Current Applications and Future Potential,” *IEIE Transactions on Smart Processing & Computing*, vol. 13, no. 5, pp. 514–522, Oct. 2024. DOI: 10.5573/IEIESPC.2024.13.5.514