Unified Automata-based Framework for Planning with Relaxed Specifications

Disha Kamale, Eleni Karyofylli, Cristian Ioan Vasile,

Electrical and Computer Engineering

Abstract:

it is important to acknowledge the need for relaxed specification semantics in the cases where the robot is unable to execute its mission with the current specifications. To address this situation, an automata-based framework for path planning with relaxed specifications is introduced with a focus on symbolic path planning . A three-way product automaton construction method is established. This product-automaton helps to calculate minimal relaxation policies for the robots using shortest path algorithms and it can identify the motion, satisfaction, specification, and available relaxations of the robot.

Background:

- Automata-based approaches are being utilized increasingly to find solutions to problems related to robotic path planning with automata being generated from the specifications of the task using temporal logic formulae.
- A Deterministic Finite State Automaton is: a tuple that consists of a finite set of states, an initial state, the input alphabet, the transition function, and a set of accepting states.
- A Transition System is: a tuple composed of a finite set of states, the initial state, a set of transitions, a set of properties, a labeling function and a weight function.
- A Weighted Finite State Edit System is: a weighted DFA whose alphabet captures word edit operations.
- A three way product automaton is composed of a DFA, TS, WFSE and it is a way of computing shortest paths and optimal trajectories Different problems have been addressed in the case studies; in this case study: minimum violation (substitution), minimum revision (deletion)

$$\mathcal{A} = (S_{\mathcal{A}}, s_0^{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}}, F_{\mathcal{A}})$$

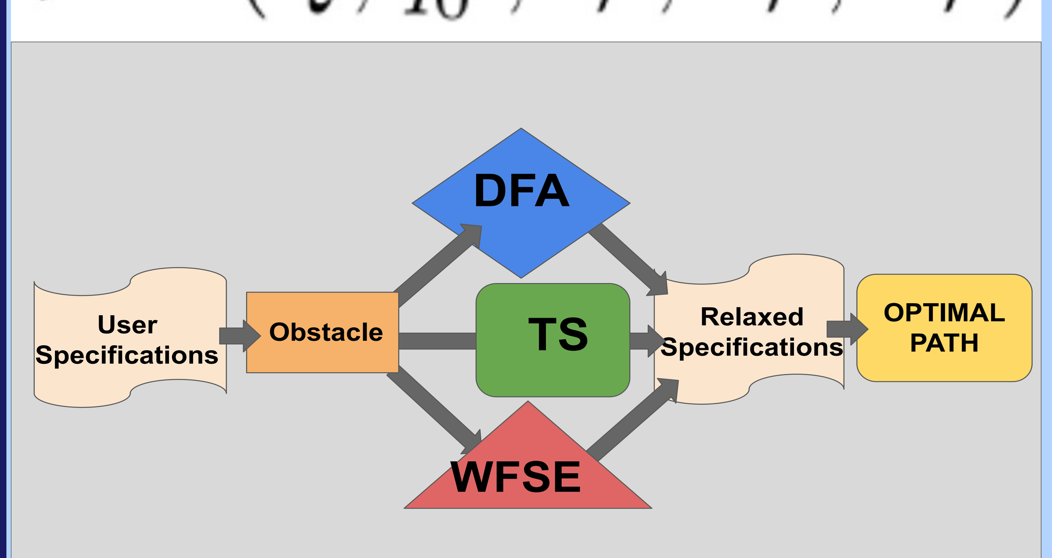
$$\mathcal{T} = (X, x_0^{\mathcal{T}}, \delta_{\mathcal{T}}, AP, h, w_{\mathcal{T}})$$

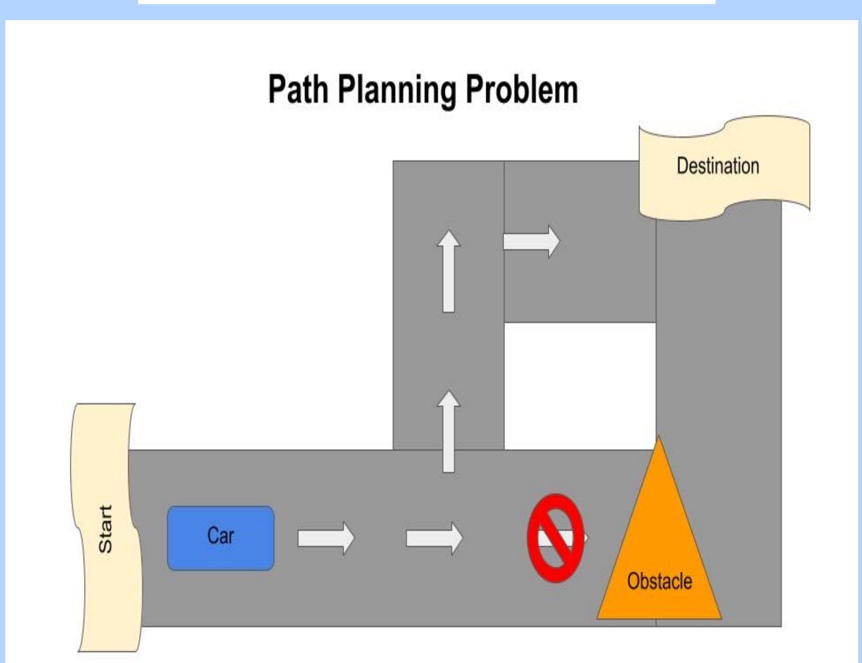
$$\mathcal{E} = (Z_{\mathcal{E}}, z_0^{\mathcal{E}}, \Sigma_{\mathcal{E}}, \delta_{\mathcal{E}}, F_{\mathcal{E}}, w_{\mathcal{E}})$$

$$\mathcal{P} = (Q, q_0^{\mathcal{P}}, \delta_{\mathcal{P}}, F_{\mathcal{P}}, w_{\mathcal{P}})$$

$$\mathcal{P}_{\mathcal{E}} = (Q^{\mathcal{E}}, q_0^{\mathcal{E}}, \delta_{\mathcal{P}}^{\mathcal{E}}, F_{\mathcal{P}}^{\mathcal{E}}, w_{\mathcal{P}}^{\mathcal{E}})$$

$$\mathcal{P}_{\mathcal{E}} = \mathcal{T} \times_{\mathcal{E}} A$$





Algorithm 1: Optimal Planning Algorithm – Plan()

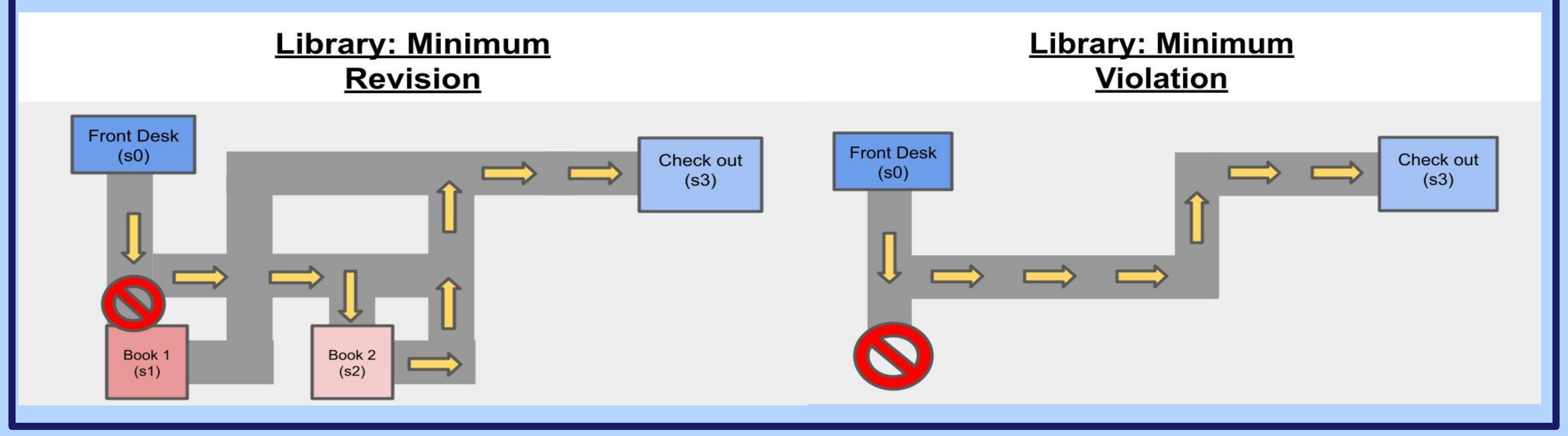
Input: $\mathcal{T}$ – transition system
Input: $\mathcal{A}$ – specification DFA
Input: $\mathcal{E}$ – user preference specification WFSE
Input: $w_{\mathcal{P}}^{\mathcal{E}}$ – product weight function
Output: optimal policy for the TS $\mathcal{T}$

Objective:

• To present a case study that emphasizes the framework's runtime performance.

Case Study 2: Runtime Performance

• Consider a robot in a library is assigned the mission of retrieving user-specified books.
• The specification DFA is composed of 8 states: initial state (FD), final state (CO), and six states representing 6 books.
• In the case that a book is not attainable due to an obstacle, the robot still partially concludes the task through a relaxed task specification.
• In this case study the two relaxation rules are: 1) Minimum Revision: if there is a book similar to the one requested by the user, then the similar book is selected to substitute the book not found; 2) Minimum Violation: if no other similar book exists, the robot return to the final state. In order to test the runtime performance, we varied the size of the TS from 8 to 100 nodes.

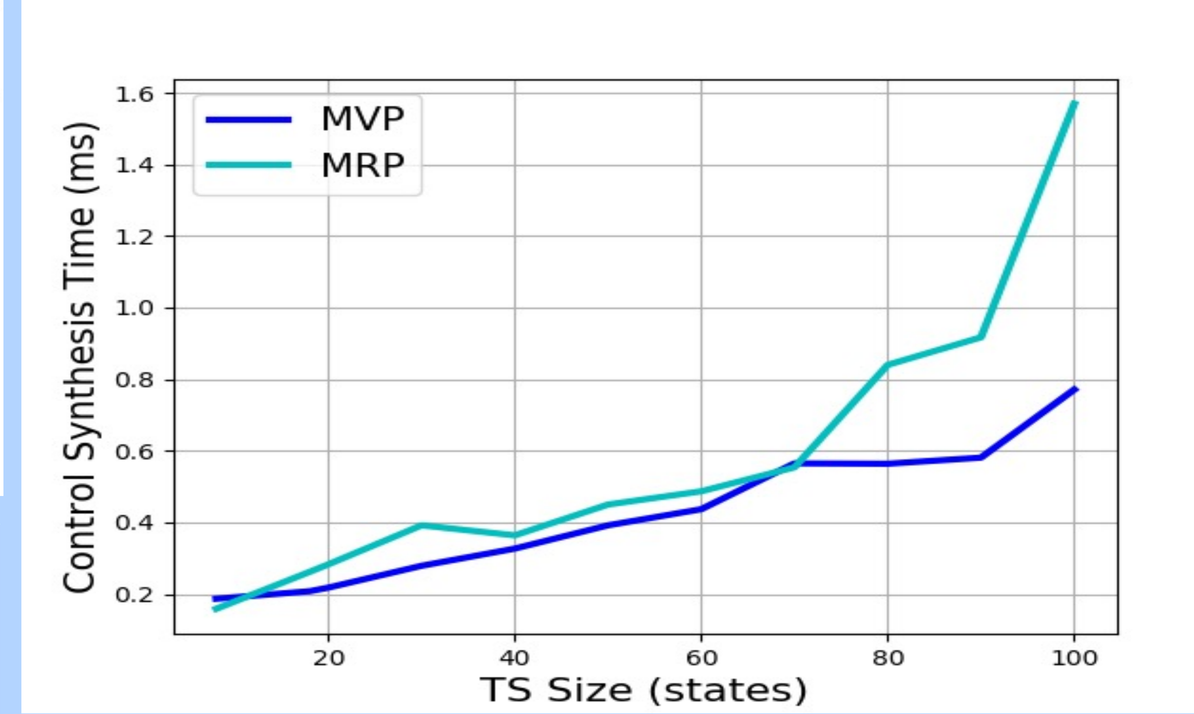


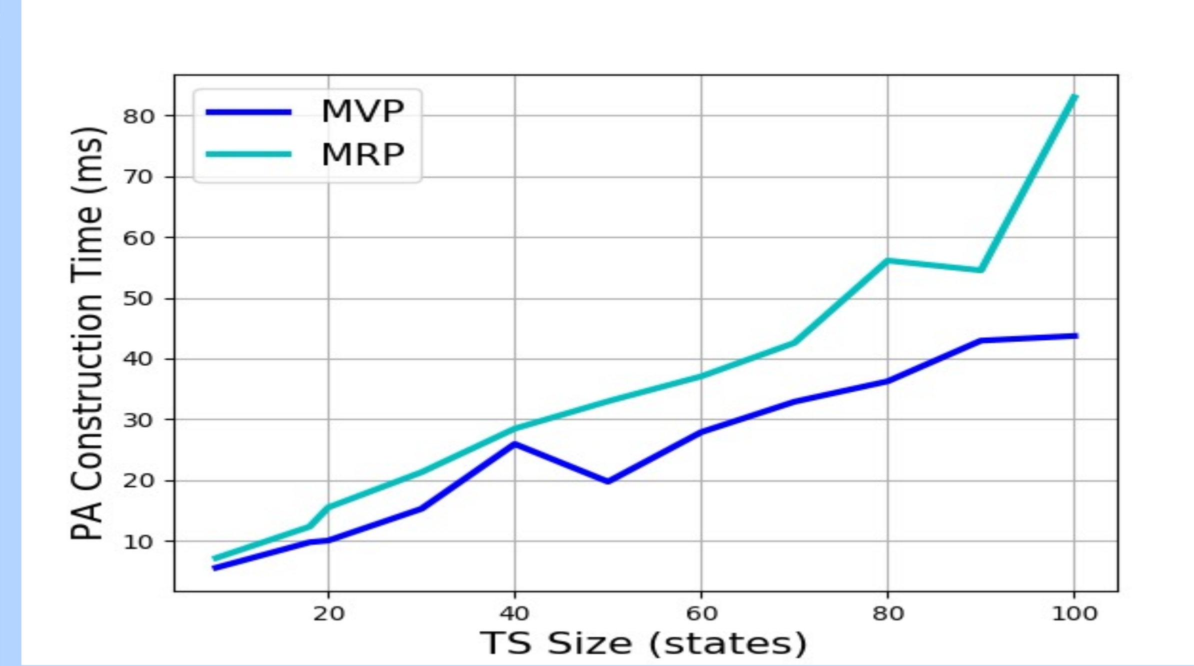
Results:

- From the data collected and shown in the graphs above, the computation time for the three-way product and optimal planning present a linear relationship with the size of the TS.
- This result matches our initial projection.

References :

D. Kamale, E. Karyofylli, and C. Vasile, Unified automata-based framework for planning with relaxed specifications (in review)





User Preference	Specification	Obstacle	Optimal Trajectory
MRP	2	Yes	('fd', 'fd', 'fd', 'q3', 'co', 'co')
MVP	6	Yes	('fd', 'fd', 'fd', 'fd', 'fd', 'co')

Conclusions:

- In this paper, "we propose the construction of a three-way product automaton that allows word to word translations and temporal relaxations at the same time".
- We show how the different instances of specification relaxations can be implemented through case studies and demonstrate that the product automaton increases linearly in time as the TS increases.

Next Steps:

- For the future work, we are planning on further developing the case studies.
- More specifically we will be implementing word to word translations.

LEHIGH UNIVERSITY | P.C. ROSSIN COLLEGE OF ENGINEERING AND APPLIED SCIENCE