A Discrete Büchi Automata Distance for Formal Methods Based Control

Garrett Thomas
Supervisor: Lars Lindemann
Examiner: Professor Dimos V. Dimarogonas

Automatic Control Department
Royal Institute of Technology, KTH

June 28th, 2017

Outline

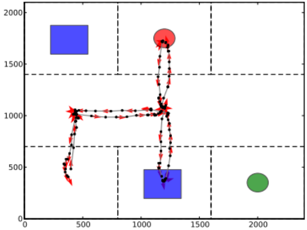
Outline

Linear Temporal Logic

- We will be using **Linear Temporal Logic (LTL)**, defined recursively as $\varphi ::= \top|\alpha|\neg\varphi_1|\varphi_1 \vee \varphi_2|\mathbf{X}\varphi_1|\varphi_1\boldsymbol{\mathcal{U}}\varphi_2$

Linear Temporal Logic

- We will be using **Linear Temporal Logic (LTL)**, defined recursively as $\varphi ::= \top|\alpha|\neg\varphi_1|\varphi_1 \vee \varphi_2|\mathbf{X}\varphi_1|\varphi_1\boldsymbol{\mathcal{U}}\varphi_2$
- Why?LTL formulas are versatile; LTL allows us to encode statements about the robot and workspace, and also how events relate to each other in the time domain.
  Ex. from [Guo15] $\varphi = \diamond(\text{rball} \wedge \diamond\text{basket}) \wedge \diamond\square\text{r1}$
  "Eventually pick up the red ball and put it in one of the baskets. Then go home to r1"



LTL Motion and Task Planning

- LTL Motion and task planning is done in three Steps [BBE+07]

LTL Motion and Task Planning

- LTL Motion and task planning is done in three Steps [BBE+07]
- Specification: Create a graph (the *product automaton*) based on the workspace, robot motion, and LTL formula such that all paths in this graph satisfy the specification.

LTL Motion and Task Planning

- LTL Motion and task planning is done in three Steps [BBE+07]
- Specification: Create a graph (the *product automaton*) based on the workspace, robot motion, and LTL formula such that all paths in this graph satisfy the specification.
- Execution: Find a discrete path in this graph using an optimality criterion.*

LTL Motion and Task Planning

- LTL Motion and task planning is done in three Steps [BBE+07]
- Specification: Create a graph (the *product automaton*) based on the workspace, robot motion, and LTL formula such that all paths in this graph satisfy the specification.
- Execution: Find a discrete path in this graph using an optimality criterion.*
- Implementation: Calculate the continuous controllers such that the continuous path will satisfy the discrete path.

Outline

1. Problem and Motivation
   - Formal Methods Based Control
   - The Product Automaton
   - Another Subsection

Finite-State Transition System

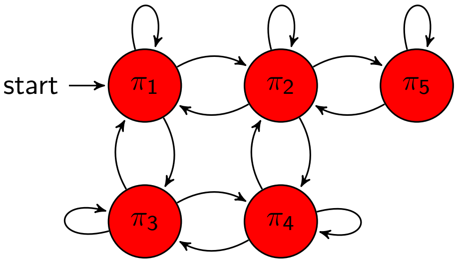
- The product automaton is the of a *finite-state transition system* and a *Büchi automaton*

Finite-State Transition System

- The product automaton is the of a *finite-state transition system* and a *Büchi automaton*

Finite-State Transition System (FTS)

An FTS is a tuple $\mathcal{T} = (\Pi, \rightarrow, \Pi_0, AP, L_D)$ where Π is the set of states, $\rightarrow \subseteq \Pi \times \Pi$ is the transitions, $\Pi_0 \subseteq \Pi$ is the initial state(s), AP is the set of atomic propositions, and $L : \Pi \rightarrow 2^{AP}$ is the labelling function (goes from a state to the set of atomic propositions that are true in that state).



Büchi Automaton

Büchi Automaton

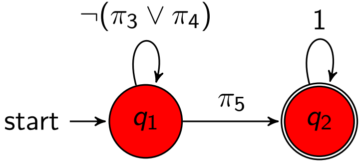
A Büchi automaton is a tuple $\mathcal{A}_\varphi = (\mathcal{Q}, 2^{AP}, \delta, \mathcal{Q}_0, \mathcal{F})$ where $\mathcal{Q}$ is a finite set of states, $\mathcal{Q}_0 \subseteq \mathcal{Q}$ is the set of initial states, 2^{AP} is the alphabet, $\delta : \mathcal{Q} \times 2^{AP} \to 2^{\mathcal{Q}}$ is a transition relation, and $\mathcal{F} \subseteq \mathcal{Q}$ is the set of accepting states.

Büchi Automaton

A Büchi automaton is a tuple $\mathcal{A}_\varphi = (\mathcal{Q}, 2^{AP}, \delta, \mathcal{Q}_0, \mathcal{F})$ where $\mathcal{Q}$ is a finite set of states, $\mathcal{Q}_0 \subseteq \mathcal{Q}$ is the set of initial states, 2^{AP} is the alphabet, $\delta : \mathcal{Q} \times 2^{AP} \to 2^{\mathcal{Q}}$ is a transition relation, and $\mathcal{F} \subseteq \mathcal{Q}$ is the set of accepting states.

- A path on a Büchi automaton is accepting if it passes through an accepting state infinitely many times.
- For any LTL formula φ over AP, there exists a Büchi automaton over 2^{AP} corresponding to φ [BKL08]

Reachability while avoiding regions $\varphi = \neg(\pi_3 \vee \pi_4)\boldsymbol{\mathcal{U}}\pi_5$



Outline

Blocks

Block Title

You can also highlight sections of your presentation in a block, with it's own title

Theorem

There are separate environments for theorems, examples, definitions and proofs.

Example

Here is an example of an example block.

Summary

- The first main message of your talk in one or two lines.
- The second main message of your talk in one or two lines.
- Perhaps a third message, but not more than that.

- Outlook
  - Something you haven't solved.
  - Something else you haven't solved.

For Further Reading I

A. Author.
Handbook of Everything.
Some Press, 1990.

S. Someone.
On this and that.
Journal of This and That, 2(1):50–100, 2000.

References I

Calin Belta, Antonio Bicchi, Magnus Egerstedt, Emilio Frazzoli, Eric Klavins, and George J Pappas, *Symbolic planning and control of robot motion [grand challenges of robotics]*, IEEE Robotics & Automation Magazine **14** (2007), no. 1, 61–70.

Christel Baier, Joost-Pieter Katoen, and Kim Guldstrand Larsen, *Principles of model checking*, MIT press, 2008.

Meng Guo, *Hybrid control of multi-robot systems under complex temporal tasks*, Ph.D. thesis, KTH Royal Institute of Technology, 2015.