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

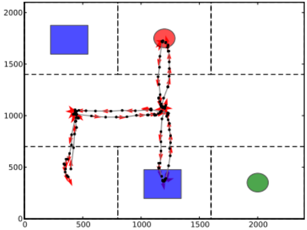
Outline

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 $\varphi ::= \top|\alpha|\neg\varphi_1|\varphi_1 \vee \varphi_2|\mathbf{X}\varphi_1|\varphi_1\mathcal{U}\varphi_2$
- 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] $\varphi = \diamond(\text{rball} \wedge \diamond\text{basket}) \wedge \diamond\Box\text{r1}$
  "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
   - Execution: Path Finding
2. Our Contribution
   - Büchi Distance and Algorithm
3. Performance on Common Formulas
   - Workspace
   - Reachability While Avoiding Regions
   - Sequencing

Finite-State Transition System

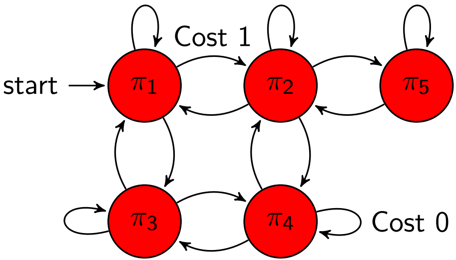
- The product automaton is the of a *finite-state transition system* and a *Büchi automaton*

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).



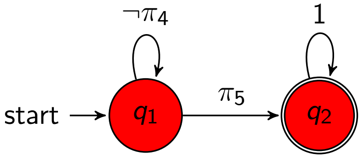
Büchi Automaton

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.

- 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 $\varphi = \neg\pi_4 \boldsymbol{\mathcal{U}} \pi_5$

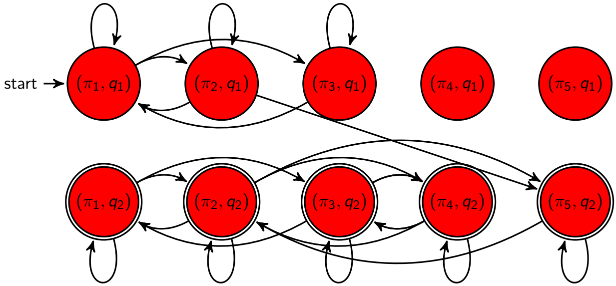


Product Automaton

Product Automaton

$\mathcal{A}_p = \mathcal{T}_w \otimes \mathcal{A}_\varphi = (Q', \delta', Q'_0, \mathcal{F}', W_p)$, where
$Q' = \Pi \times Q = \{\langle \pi, q\rangle \in Q' | \forall \pi \in \Pi, \ \ \forall q \in Q\}$; $\delta' : Q' \rightarrow 2^{Q'}$.
$\langle \pi_j, q_n\rangle \in \delta'(\langle \pi_i, q_m\rangle)$ iff $(\pi_i, \pi_j) \in \rightarrow_c$ and $q_n \in \delta(q_m, L_d(\pi_j))$;
$Q'_0 = \{\langle \pi, q\rangle | \pi \in \Pi_0, \ \ q_0 \in Q_0\}$, $\mathcal{F}' = \{\langle \pi, q\rangle | \pi \in \Pi, q \in \mathcal{F}\}$

- Also a Büchi automaton



Outline

Accepted Algorithm

- We prefer a prefix, suffix structure
  $R = \langle R_{pre}, R_{suf} \rangle = q'_0 q'_1 \ldots q'_f [q'_f q'_{f+1} \ldots q'_n]^\omega$

Here is the algorithm currently used in the literature [Guo15],[FGKGP09],[KB08],[STBR10]

Procedure 1 OptRun() [Guo15]

Input: Input $\mathcal{A}_p$

Output: R_{opt}

1: For initial state $q'_0 \in \mathcal{Q}'_0$, find the optimal path to each $q'_f \in \mathcal{F}$.
2: For each accepting state $q'_f \in \mathcal{F}'$, calculate the optimal path back to q'_f.
3: Find the pair of $(q'_{0,opt}, q'_{f,opt})$ that minimizes the total cost
4: Optimal accepting run R_{opt}, prefix: shortest path from q'_{0*} to q_{f*}; suffix: the shortest cycle from q'_{f*} and back to itself.

Outline

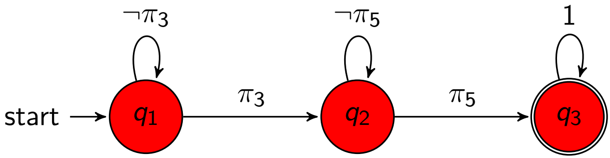
Büchi Distance Measure

- We introduce a discrete Büchi distance measure, which is the least number of transitions possible to get to an accepting state

Büchi Distance Measure

- We introduce a discrete Büchi distance measure, which is the least number of transitions possible to get to an accepting state

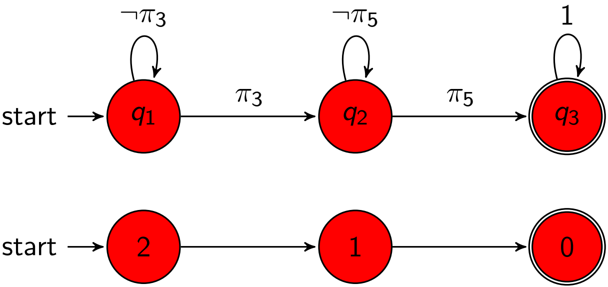
Ex. Sequencing $\diamond(\pi_3 \wedge \diamond\pi_5)$



Büchi Distance Measure

- We introduce a discrete Büchi distance measure, which is the least number of transitions possible to get to an accepting state

Ex. Sequencing $\diamond(\pi_3 \wedge \diamond\pi_5)$



- Denoted $d_p : \mathcal{Q} \to \mathbb{Z}$, e.g. $d_p(q_2) = 1$

Note: Transitions with && are removed from LTL2BA because regions do not overlap

Our Algorithm

- Motivation for our algorithm: $\mathcal{F}' = \{\langle \pi, q \rangle | \pi \in \Pi, q \in \mathcal{F}\}$
- We greedily find the optimal path which reduces the Büchi distance at each step

Procedure 2 GreedyRun()

Input: Input $\mathcal{A}_{p,d}$
Output: R_g

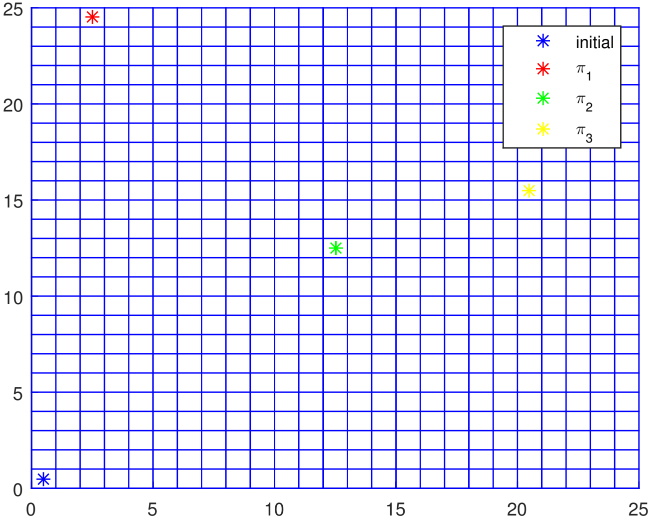
```
LEVEL = d_p(q'_0 ∈ Q'_0)
while LEVEL > 0 do
    find optimal path down to q'_n s.t. d_p(q'_n) == LEVEL − 1
    Level = Level - 1
Find optimal path from q'_n back to itself
Accepting run R_g, prefix: the optimal paths calculated in the while loop concatenated together; suffix: optimal path from q'_n back to itself.
```

Why?

- **We approximate the globally optimal path with a series of locally optimal paths**
- **We sacrifice a degree of optimality for easier computation!**

Outline

Workspace



Outline

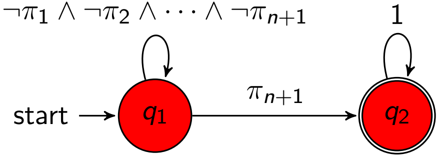


Figure: Büchi automaton corresponding to $\neg(\pi_1 \vee \pi_2 \vee \ldots \pi_n)\mathcal{U}\pi_{n+1}$

$$d_p(q_1) = 1 \text{ and } d_p(q_2) = 0$$

Outline

- We look at the sequencing formula $\diamond(\pi_1 \wedge \diamond(\pi_2 \wedge \diamond\pi_3))$

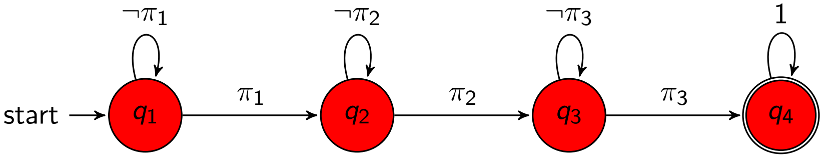


Figure: Büchi Automaton Corresponding to $\diamond(\pi_3 \wedge \diamond\pi_5)$

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.

Georgios E Fainekos, Antoine Girard, Hadas Kress-Gazit, and George J Pappas, *Temporal logic motion planning for dynamic robots*, Automatica **45** (2009), no. 2, 343–352.

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

References II

Marius Kloetzer and Calin Belta, *A fully automated framework for control of linear systems from temporal logic specifications*, IEEE Transactions on Automatic Control **53** (2008), no. 1, 287–297.

Stephen L Smith, Jana Tumova, Calin Belta, and Daniela Rus, *Optimal path planning under temporal logic constraints*, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, 2010, pp. 3288–3293.