Efficient Reactive Controller Synthesis for a Fragment of Temporal Logic

Eric M. Wolff¹
Ufuk Topcu² and Richard M. Murray¹
¹Caltech and ²UPenn

ICRA 2013
Planning in a Dynamic Environment

- Dynamic obstacles
- Safe navigation and repetitive tasks
Our Contributions

• Introduce expressive and efficient fragment of linear temporal logic
  – 2-EXP improvement over LTL
  – Non-deterministic and stochastic systems
  – Simple and extensible framework (e.g. optimality)
Outline

• Motivation
• Preliminaries
• Feasible control policies
• Examples
Non-deterministic Transition Systems

- A non-deterministic transition system (NTS) is a tuple $T = (S, A, R, s_0, AP, L)$ where
  - $S$ is a finite set of states,
  - $A$ is a finite set of actions (e.g., motion primitives),
  - $R: S \times A \rightarrow 2^S$ is the transition function,
  - $s_0$ is the initial state,
A non-deterministic transition system (NTS) is a tuple $T = (S, A, R, s_0, AP, L)$ where

- $S$ is a finite set of states,
- $A$ is a finite set of actions (e.g., motion primitives),
- $R: S \times A \rightarrow 2^S$ is the transition function,
- $s_0$ is the initial state,
- $AP$ is a finite set of atomic propositions,
- $L: S \rightarrow 2^{AP}$ is a labeling function.
Control Policies

- Control policy:
  - $\mu: S \to A$ (memoryless)
  - $\mu: S \times M \to A \times M$ (finite-memory, $M = \text{memory set}$)

![Diagram of control policies](image)
Control Policies

• Control policy:
  – $\mu: S \to A$ (memoryless)
  – $\mu: S \times M \to A \times M$ (finite-memory, $M =$ memory set)

• Two-player game:
  – 1) **System** picks action using control policy
  – 2) **Environment** picks next state
Control Policies

- Control policy:
  - $\mu: S \rightarrow A$ (memoryless)
  - $\mu: S \times M \rightarrow A \times M$ (finite-memory, $M =$ memory set)

- Two-player game:
  - 1) **System** picks action using control policy
  - 2) **Environment** picks next state

- $T^\mu(s)$: set of runs from state $s$ under policy $\mu$
Hierarchical Control Architecture

We focus on the discrete planning layer
Discrete plan is executed at continuous level

1. AlurHLP00, BeltaH06, BhatiaLV10, HabetsCS06, KaramanF09, KloetzerB08, and more
Specification Language

• We consider formulas of the form:

\[ \varphi = \varphi_{\text{safe}} \land \varphi_{\text{resp}} \land \varphi_{\text{per}} \land \varphi_{\text{task}} \land \varphi_{\text{resp}}^{\text{ss}}, \]

where

\[ \varphi_{\text{safe}} := \square \psi_1, \]

Safety
Specification Language

• We consider formulas of the form:

$$\varphi = \varphi_{\text{safe}} \land \varphi_{\text{resp}} \land \varphi_{\text{per}} \land \varphi_{\text{task}} \land \varphi_{\text{resp}}^{\text{ss}},$$

where

$$\varphi_{\text{safe}} := \square \psi_1,$$

$$\varphi_{\text{resp}} := \bigwedge_{j \in I_2} \square (\psi_{2,j} \implies \Diamond \phi_{2,j}),$$

Safety

Response
Specication Language

• We consider formulas of the form:

\[ \varphi = \varphi_{\text{safe}} \land \varphi_{\text{resp}} \land \varphi_{\text{per}} \land \varphi_{\text{task}} \land \varphi_{\text{ss}}^{\text{resp}}, \]

where

\[ \varphi_{\text{safe}} := \Box \psi_1, \]
\[ \varphi_{\text{resp}} := \bigwedge_{j \in I_2} \Box (\psi_{2,j} \implies \Diamond \phi_{2,j}), \]
\[ \varphi_{\text{per}} := \Diamond \Box \psi_3, \]

Safety
Response
Persistence (stability)
Specication Language

• We consider formulas of the form:

\[ \varphi = \varphi_{\text{safe}} \land \varphi_{\text{resp}} \land \varphi_{\text{per}} \land \varphi_{\text{task}} \land \varphi_{\text{resp}}^{ss}, \]

where

\[ \varphi_{\text{safe}} := \Box \psi_1, \]

\[ \varphi_{\text{resp}} := \bigwedge_{j \in I_2} \Box (\psi_{2,j} \implies \Diamond \phi_{2,j}), \]

\[ \varphi_{\text{per}} := \Diamond \Box \psi_3, \]

\[ \varphi_{\text{task}} := \bigwedge_{j \in I_4} \Box \Diamond \psi_{4,j}, \]

Safety
Response
Persistence (stability)
Repeated tasks
Specification Language

• We consider formulas of the form:

\[ \varphi = \varphi_{\text{safe}} \land \varphi_{\text{resp}} \land \varphi_{\text{per}} \land \varphi_{\text{task}} \land \varphi_{\text{resp}}^{\text{ss}}, \]

where

\[ \varphi_{\text{safe}} := \Box \psi_1, \]
\[ \varphi_{\text{resp}} := \bigwedge_{j \in I_2} \Box (\psi_{2,j} \implies \Diamond \phi_{2,j}), \]
\[ \varphi_{\text{per}} := \Diamond \Box \psi_3, \]
\[ \varphi_{\text{task}} := \bigwedge_{j \in I_4} \Box \Diamond \psi_{4,j}, \]
\[ \varphi_{\text{resp}}^{\text{ss}} := \bigwedge_{j \in I_5} \Diamond \Box (\psi_{5,j} \implies \Diamond \phi_{5,j}). \]

Safety
Response
Persistence (stability)
Repeated tasks
Steady-state response
Related Work

• GR(1) [Piterman06]
  \[(\square \Diamond p_1 \land \cdots \land \square \Diamond p_m) \rightarrow (\square \Diamond q_1 \land \cdots \land \square \Diamond q_n)\]

• Related logics: AlurT04, Ehlers11, MalerPS95

• How this work differs:
  – More system guarantees than GR(1)
  – No environment liveness assumptions
Problem Statement

• Given:
  – Non-deterministic system
  – Temporal logic specification $\varphi$ (in fragment)

• Problem: Create control policy $\mu$ such that that $T^\mu(s_0)$ satisfies $\varphi$
Value Function and Reachability

- $V^c_B(s)$: minimum cost to reach set B from state s under all resolutions of the non-determinism

$$V^c_{B,T}(s) = \min_{a \in A(s)} \max_{t \in R(s,a)} V^c_{B,T}(t) + c(s, a, t)$$
Value Function and Reachability

- $V^c_B(s)$: minimum cost to reach set $B$ from state $s$ under all resolutions of the non-determinism

$$V^c_{B, T}(s) = \min_{a \in A(s)} \max_{t \in R(s,a)} V^c_{B, T}(t) + c(s, a, t)$$

- Example
  - $V^c_4(1) = \infty$
  - $V^c_4(2) = \infty$
  - $V^c_4(3) = 1$
  - $V^c_4(4) = 0$
  - $\text{CPre}(4) = \{ 3, 4 \}$
Outline

• Motivation
• Preliminaries
• Feasible control policies
  – Create “safe” subgraph
  – Compute task cycle (liveness)
• Examples
Removing Unsafe Behaviors

1. Remove unsafe states from $T (\varphi_{\text{safe}}, \varphi_{\text{per}})$
2. Remove unsafe transitions from $T (\varphi_{\text{resp}}, \varphi_{\text{ss \, resp}})$
Removing Unsafe Behaviors

1. Remove unsafe states from $T$ ($\varphi_{safe}$, $\varphi_{per}$)
2. Remove unsafe transitions from $T$ ($\varphi_{resp}$, $\varphi_{ss \, resp}^{ss}$)

$\varphi_{safe} = [] B$
Repeated Tasks

1. Remove unsafe states from $T (\varphi_{safe}, \varphi_{per})$
2. Remove unsafe transitions from $T (\varphi_{resp}, \varphi^{ss}_{resp})$
3. Compute task cycle (generalized Büchi) on $T_{inf}$
Generalized Büchi Game

\[ \varphi_{\text{task}} = []<> F1 \land []<> F2 \]

Iterate until sets stop changing.

ChatterjeeHP06, McNaughton93
Generalized Büchi Game

$\varphi_{\text{task}} = []<> A \land []<> B$

Cannot reach F1

F1

F2

CPre(F1)
Generalized Büchi Game

\( \varphi_{task} = []<> A & []<> B \)
Generalized Büchi Game

\[ \varphi_{\text{task}} = []<> A \land []<> B \]

Cannot reach F2

ChatterjeeHP06, McNaughton93
Generalized Büchi Game

\[ \varphi_{\text{task}} = []<> A \land []<> B \]
Generalized Büchi Game

\( \varphi_{\text{task}} = []<> A & []<> B \)
Generalized Büchi Game

\[ \varphi_{\text{task}} = []<> A \land []<> B \]

ChatterjeeHP06, McNaughton93
Generalized Büchi Game

\[ \varphi_{\text{task}} = []<> A \& []<> B \]

DONE!
Reactive Synthesis Summary

• What have we computed?
  – Largest possible task sets $F_j$
  – Largest possible winning set $W$
  – Finite-memory control policies

• Time complexity:
  – $S = \# \text{ states}, R = \# \text{ transitions}, \varphi = \# \text{ specs}$

<table>
<thead>
<tr>
<th>Language</th>
<th>DTS</th>
<th>NTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>$O(</td>
<td>\varphi</td>
</tr>
<tr>
<td>GR(1)</td>
<td>$O(</td>
<td>\varphi</td>
</tr>
<tr>
<td>LTL</td>
<td>$O(2^{(</td>
<td>\varphi</td>
</tr>
</tbody>
</table>
Example: Pickup and delivery

• System:
  – Robot and obstacle move to adjacent regions

• Specs:
  – Always avoid collisions
  – Repeatedly visit Pickup and Dropoff locations
Results

Prior work

Our work
Conclusions

• Main results
  – 2-EXP improvement over LTL
  – Non-deterministic and stochastic systems
  – Simple and extensible framework (e.g. optimality)
  – Promising alternative to GR1

• These are the easy tasks!
Thanks!

• Questions?

• Funding
  – NDSEG fellowship
  – Boeing
  – AFOSR

Eric Wolff (www.cds.caltech.edu/~ewolff/)
Backup
Do Standard Methods Work?

\[ \varphi_{\text{resp}} := \bigwedge_{j \in I_2} \mathbf{□} (\psi_{2,j} \implies \mathbf{O} \phi_{2,j}) \]
Acceptance Conditions

- Let $\sigma = s_0s_1s_2 \ldots$ be a run of the system
- Let $\psi$ be a propositional formula. It holds at states in $[[\psi]]$
- $\text{Vis}(\sigma) =$ set of states visited
- $\text{Inf}(\sigma) =$ set of states visited repeatedly

- $\sigma \models \Box \psi$ iff $\text{Vis}(\sigma) \subseteq [[\psi]]$,
- $\sigma \models \Diamond \Box \psi$ iff $\text{Inf}(\sigma) \subseteq [[\psi]]$,
- $\sigma \models \Box \Diamond \psi$ iff $\text{Inf}(\sigma) \cap [[\psi]] \neq \emptyset$,
- $\sigma \models \Box(\psi \implies \Diamond \phi)$ iff $\sigma_i \notin [[\psi]]$ or $\sigma_{i+1} \in [[\phi]]$ for all $i$,
- $\sigma \models \Diamond \Box(\psi \implies \Diamond \phi)$ iff there exists an index $j$ such that $\sigma_i \notin [[\psi]]$ or $\sigma_{i+1} \in [[\phi]]$ for all $i \geq j$. 


Modeling Non-determinism

state = (UAV, car)

\[(u_1,c_2) \xrightarrow{} (u_1,c_1) \]

\[(u_1,c_2) \xrightarrow{} (u_1,c_2) \]

\[(u_9,c_1) \quad (u_9,c_2) \]