Automatically Modifying Conflicting Specifications

Georgios E. Fainekos
Arizona State University

fainekos at asu edu
http://www.public.asu.edu/~gfaineko
Humans & Robots in the near future

Mr Roboto, go to the storage room, take three blankets and deliver them to examination room C2 and then to rooms B2 and A2. But don't go through corridor 1, there is a meeting going on!
Humans & Robots in the near future

Mr Roboto, go to the storage room, take three blankets and deliver them to examination room C2 and then to rooms B2 and A2. But don’t go through corridor 1, there is a meeting going on!

From Natural Language to Formal Specifications:
Humans & Robots in the near future

\[ G(\neg \text{corr}_1) \land F(\text{stor} \land F(\text{exam}_C \land F(\text{exam}_A \land F(\text{exam}_B) ) ) ) \]

Why you didn’t say so before?!?
Humans & Robots in the near future

Linear Temporal Logic Planning for Robotics as Model Checking:
Linear Temporal Logic Planning for Robotics as Model Checking

Lahijanian, Kloetzer, Itani, Belta, Andersson, ICRA 2009

Bobadilla, Sanchez, Czarnowski, Gossman, LaValle, RSS 2011

LaViers, Chen, Belta, Egerstedt, IEEE RAM 2011
Humans & Robots in the near future

I am sorry, I cannot do it!!!

F(stor ∧ F(exam_C2 ∧ F exam_A2 ∧ F exam_B2 ))

G(¬corr_1) ∧ F(stor ∧ F(exam_C2 ∧ F exam_A2 ∧ F exam_B2 ))

Why not?

Well, what can you do?

Sounds good! Do that.
Overview & Problem Statement

Robot(s) model
\[ \dot{x} = f(x, p, u) \quad X_0 \subseteq X \]
\[ y = g(x, p, u) \]

Environment

Specification in Natural Language

Linear Temporal Logic

TL Planning Algorithm Based on Model Checking

Planning succeeds

Built Feedback controllers

This presentation

Planning fails

Return to the user:
1. Why the specification failed
2. A specification that can be realized on the system
   • and it is as close as possible to the initial intention of the user
Linear Temporal Logic: Semantic Intuition

\( a \) - a now

\( G a \) - always a

\( F a \) - eventually a

\( X a \) - next state a

\( a U b \) - a until b

\( a B b \) - a before b
Supervisory Controller

$\varphi = (\neg q_1)U(q_2 \land (\neg q_1)Uq_3)$

*Simplified figure since, for instance, "q2 $\land$ q3" cannot happen in this example*
Supervisory Controller

\[ \varphi = (\neg q_1)U(q_2 \land (\neg q_1)Uq_3) \]

Product Automaton

For any initial state

Find a path to an accepting state

and

Find a path from an accepting state back to itself

Supervisor
Supervisory Controller

\[ \varphi = (\neg q_1) U (q_2 \land (\neg q_1 \land \neg q_4) U q_3) \]

For any initial state

\[ \varphi = (\neg q_1) U (q_2 \land (\neg q_1 \land \neg q_4) U q_3) \]

Supervisor

Product Automaton

Cannot find a path to an accepting state

or

Cannot find a path from an accepting state back to itself
Problem 1: Why did the specification failed?

• Solution straightforward:
  • Find the set of reachable atomic propositions \( R_{\Pi} \)
  • If an atomic proposition in the specification \( \phi \) is not in \( R_{\Pi} \), then the planning failed due to an unreachable state in the system

\[
\varphi = F \text{ ROOM2}
\]

• If all the atomic propositions in the specification \( \varphi \) are in \( R_{\Pi} \), then the planning failed due to “logical inconsistencies”

\[
\varphi = (\neg \text{COR}) \cup \text{ROOM1}
\]
Revising the specification

- Ok, $\phi = (\neg q_1) U (q_2 \land (\neg q_1 \land \neg q_4) U q_3)$ cannot be satisfied.

What specification can be satisfied?

- We must restrict our search space
  - “Irrelevant” solutions are not an option, e.g., $\phi = G q_5$
    - E.g., you ask the robot to bring oranges and the robot responds: “Actually, I can only bring apples”
  - Non-minimal solutions are not desirable, e.g., $\phi = true$
    - E.g., you ask the robot to bring oranges and the robot responds: “Actually, I can only stay here and do nothing or I can visit all the rooms in the house. Please choose.”
Partial Order on the LTL formulas

• Let $\phi_1$ and $\phi_2$ be 2 LTL formulas, then

$\phi_1 < \phi_2$ if $\phi_1 \Rightarrow \phi_2$

• Remark: In order to have a lattice, we need to consider the congruence under the equivalence relation of LTL formulas
  - e.g. $F(p_1 \lor p_2) \equiv Fp_1 \lor Fp_2$

• Hence, the conjunction and disjunction become the meet and join operations over the lattice
Examples

Notation

\( \varphi_1 < \varphi_2 \)

\[ \begin{align*}
\varphi_2 \\
Fp \\
p \\
false
\end{align*} \]

\[ \begin{align*}
\varphi_1 \\
true \\
p_1 \lor p_2 \\
G(p_1 \land p_2)
\end{align*} \]

false \( < p < Fp < true \)

\( F(p_1 \lor p_2) \)

\( \varphi_2 \)

\[ \begin{align*}
false \\
true \\
p_1 \lor p_2 \\
p_1 \land p_2 \\
G(p_1 \land p_2)
\end{align*} \]
Possible space of solutions ...

\[ F(\varphi, T) = \{ \varphi' \in \text{LTL} \mid T \models \varphi' \text{ and } \varphi < \varphi' \} \]
Possible space of solutions ...

\[ F(\varphi, T) = \{ \varphi' \in LTL \mid T \models \varphi' \text{ and } \varphi < \varphi' \} \]

- However, searching for a minimal solution in \( F(\varphi, T) \) is not enough ...

\[ \varphi = \pi_0 \land F(\pi_2 \land F \pi_1) \]

\[ \varphi' = (\pi_0 \land F(\pi_2 \land F \pi_1)) \lor G(\pi_0 \land \neg \pi_1 \land \neg \pi_2) \]
Modified possible space of solutions...

$$RF(\varphi, T) = \{\varphi' \in LTL \mid T \vDash \varphi' \text{ and } \varphi \prec \varphi'\} \cap \text{rel}(\varphi)$$

where $\text{rel}(\varphi)$ removes some “irrelevant” solutions

• Example: $\text{rel}(\varphi)$ is the set of all formulas where each atomic proposition $\pi$ in $\varphi$ is replaced by a formula in its upper bound
  • e.g.

    \[
    \begin{align*}
    \text{true} \\
    \text{F}\pi \\
    \pi
    \end{align*}
    \]
Automatic Revision and Minimality due to unreachable atomic propositions

Theorem: Let LTL formula $\phi$ be unsatisfiable on system $T$ and $U$ be the set of unreachable atomic propositions in $T$. Set $\phi' = \text{rem}_U(\phi)$. Then,

1. we have $\phi \prec \phi'$.
2. $\phi'$ is minimal in $RF(\phi, T)$

where:

1. $\text{rem}_U(\phi)$ replaces each atomic proposition in $\phi$ that is also in $U$ with $true$
2. $\phi'$ is minimal in $RF(\phi, T)$ if for any other $\psi$ in $RF(\phi, T)$ we have $\psi \prec \phi'$, then $\psi \equiv \phi'$.
Automatic Revision and Minimality due to "logical inconsistencies"

- Can we find the minimal revision in RF(φ,T)?
  - Obviously, the problem is decidable
  - Assume that we are going to maximally relax (i.e., set to true) each atomic proposition in φ
  - We need to consider $2^{|\text{AP}(\phi)|}$ combinations of maximal relaxations
  - For each combination, we need to run the LTL planning algorithm which is of complexity $|T|2^{O(|\phi|)}$
  - Worst case complexity $|T|2^{O(|\phi|)}2^{|\text{AP}(\phi)|} = |T|2^{O(|\phi|)+|\text{AP}(\phi)|}$
Label each edge in the product automaton with the set of atomic propositions from the spec that must be removed for edge to become enabled.
Problem: Find the path on the graph with the smallest number of AP to be removed.

Product Automaton

Path weight \( \{\pi_1, \pi_2, \pi_3\} \)

Path weight \( \{\pi_3\} \)
Main result: NP-Completeness

**Minimal Accepting Path (MAP) Problem:**
Given an instance of the minimal accepting path problem \((G, Y, L, v_0, F)\) and a bound \(W\), the decision of problem of whether there exists a truth assignment \(Z \subseteq Y\) such that \(|Z| \leq W\) is NP-Complete.

Proof* by reduction from 3-CNF-SAT.

* The proof does not appear in the ICRA 2012 paper, so please ask me for the technical report or wait for the journal version.
Intuition on why the problem is hard
Using SAT solvers* and ASP to get a solution

• We use variables of the form \( \text{REACH}(v_0, v) \) and \( \text{REACH}(v_f, v) \) for every vertex \( v \in V \) and \( v_f \in F \).

• Reachability from \( v_0 \) and from \( v_f \in F \):
  • If \( (v_0, v) \in E \), then \( \text{REACH}(v_0, v) \Leftrightarrow \bigwedge_{y \in L(v_0,v)} y \)
  • If \( (v_0, v) \notin E \), then \( \text{REACH}(v_0, v) \Leftrightarrow \bigvee_{(u,v) \in E} (\text{REACH}(v_0, u) \land \bigwedge_{y \in L(u,v)} y) \)
  • If \( (v_f, v) \in E \), then \( \text{REACH}(v_f, v) \Leftrightarrow \bigwedge_{y \in L(v_f,v)} y \)
  • If \( (v_f, v) \notin E \), then \( \text{REACH}(v_f, v) \Leftrightarrow \bigvee_{(u,v) \in E} (\text{REACH}(v_f, u) \land \bigwedge_{y \in L(u,v)} y) \)

• “Lasso” condition:
  • \( \bigvee_{v_f \in F} (\text{REACH}(v_0, v_f) \land \text{REACH}(v_f, v_f)) \)

*Yices and/or Z3
Experimental Results: SAT problem

The experiments were run on the ASU supercomputing center* which consists of clusters of Dual 4-core processors, 16GB Intel(R) Xeon(R) CPU, X5355 @2.66 GHz.

<table>
<thead>
<tr>
<th>Nodes n</th>
<th>Edges → Sparse: 2n − 2</th>
<th>Medium: 3n</th>
<th>Dense: n^2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>avg</td>
<td>max</td>
</tr>
<tr>
<td>10</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>100</td>
<td>0.3</td>
<td>0.6</td>
<td>1.5</td>
</tr>
<tr>
<td>200</td>
<td>1.8</td>
<td>4.7</td>
<td>24.1</td>
</tr>
<tr>
<td>300</td>
<td>5.9</td>
<td>15.4</td>
<td>76.3</td>
</tr>
<tr>
<td>400</td>
<td>14.7</td>
<td>58.2</td>
<td>244.9</td>
</tr>
<tr>
<td>500</td>
<td>33.2</td>
<td>125.7</td>
<td>473.0</td>
</tr>
</tbody>
</table>

*Our implementations do not utilize the parallel architecture.
Experimental Results: 2 Approximation Algorithm*

<table>
<thead>
<tr>
<th>Nodes</th>
<th>ASP</th>
<th>AAMRP</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>avg</td>
<td>max</td>
</tr>
<tr>
<td>9</td>
<td>0.003</td>
<td>0.0071</td>
<td>0.012</td>
</tr>
<tr>
<td>100</td>
<td>0.099</td>
<td>0.1954</td>
<td>1.405</td>
</tr>
<tr>
<td>196</td>
<td>0.335</td>
<td>1.25058</td>
<td>6.003</td>
</tr>
<tr>
<td>324</td>
<td>0.899</td>
<td>5.3316</td>
<td>14.731</td>
</tr>
<tr>
<td>400</td>
<td>1.267</td>
<td>12.87</td>
<td>35.58</td>
</tr>
<tr>
<td>529</td>
<td>3.086</td>
<td>34.1642</td>
<td>103.638</td>
</tr>
</tbody>
</table>

**TABLE I**

**Numerical Experiments:** Number of nodes versus the results of ASP solver and AAMRP. Under the ASP and AAMRP columns, the numbers indicate computation times in sec. RATIO indicates the experimentally observed approximation ratio to the optimal solution.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>ASP</th>
<th>AAMRP</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>avg</td>
<td>max</td>
</tr>
<tr>
<td>9</td>
<td>0.003</td>
<td>0.0097</td>
<td>0.039</td>
</tr>
<tr>
<td>100</td>
<td>0.378</td>
<td>18.4679</td>
<td>3502.343</td>
</tr>
<tr>
<td>196</td>
<td>3.336</td>
<td>31.995</td>
<td>685.819</td>
</tr>
<tr>
<td>306</td>
<td>9.801</td>
<td>75.524</td>
<td>2795.337</td>
</tr>
<tr>
<td>400</td>
<td>21.744</td>
<td>124.7486</td>
<td>164.5459</td>
</tr>
<tr>
<td>506</td>
<td>58.67</td>
<td>241.167</td>
<td>1054.98</td>
</tr>
</tbody>
</table>

**TABLE II**

**Numerical Experiments:** Number of nodes versus the results of ASP solver and AAMRP. Under the ASP and AAMRP columns, the numbers indicate computation times in sec. RATIO indicates the experimentally observed approximation ratio to the optimal solution.

*Algorithm under review*
Experimental Results:
2 Approximation Algorithm*

Preliminary results on scalability using a prototype Python implementation:

<table>
<thead>
<tr>
<th>Nodes</th>
<th>ASP min</th>
<th>ASP avg</th>
<th>ASP max</th>
<th>AAMRP min</th>
<th>AAMRP avg</th>
<th>AAMRP max</th>
<th>succ</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td>24.438</td>
<td>168.2133</td>
<td>237.758</td>
<td>0.125</td>
<td>0.23</td>
<td>0.323</td>
<td>10/10</td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>0/10</td>
<td>15.723</td>
<td>76.164</td>
<td>128.471</td>
<td>9/10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20164</td>
<td>0/10</td>
<td>50.325</td>
<td>570.737</td>
<td>1009.675</td>
<td>8/10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50176</td>
<td>0/10</td>
<td>425.362</td>
<td>1993.449</td>
<td>4013.717</td>
<td>3/10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60025</td>
<td>0/10</td>
<td>6734.133</td>
<td>6917.094</td>
<td>7100.055</td>
<td>2/10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE III

Numerical Experiments: Number of nodes versus the results of ASP solver and AAMRP. Under the ASP and AAMRP columns the numbers indicate computation times in sec. RATIO indicates the experimentally observed approximation ratio to the optimal solution.

*Algorithm under review
As automatic synthesis methods for software and, in particular, for embedded control software move from theory and prototypes into technology, we will need methods for specification debugging and automatic revision.
Conclusions

• Contributions
  1. We have defined the **Minimal Specification Revision Problem**
     • **Important for user-friendly temporal logic planning frameworks**
  2. We showed NP-completeness of MSRP even in its simplest version
  3. We have provided a SAT encoding of the MSRP
  4. We have developed an approximation algorithm

• Future work
  1. Define appropriate search space restrictions for different classes of planning problems
  2. Extend the theory to LTL games
  3. Provide direct feedback in natural language
Thank you! Questions?

- **Acknowledgements:**
  - Students: Kangjin Kim (PhD)
  - Collaborators: Sriram Sankaranarayanan (University of Colorado, Boulder)
  - Support: NSF CNS 1116136

- **References:**