Planning in FOND domains with TEGs expressed in LTL

Eleni Triantafillou
Under the supervision of
Sheila McIlraith
Outline

• Automated Planning
• FOND Domains
• Linear Temporal Logic (LTL)
• Controller Synthesis: The problem and the approach
Planning Problem

• Planning Domain
  – S: set of states
  – A: set of actions
  – s0: Initial State
  – f(a, s): transition function
  – Prop: set of propositions

• Goal
  – G: set of goal states
Stealing Cookies
Planning Domain

**Propositions:**
- EatingCookie
- Caught
- Hungry
- JarReachable
- PinkSocks
- Hiding

**Actions:**
- OpenJar
- CloseJar
- Run!
- StealCookie
- Hide
- EatCookie

**Initial State:**
(JarReachable, Hungry)
Specifying Actions

Action: Run!

- **Preconditions**: StoleCookie, Hiding, Caught
- **Effects**
  - Adds: Running
  - Deletes: Hiding
Planning Goal

Goal: (EatingCookie)
FOND

• Type of domains that are:
  – Fully-Observable: The planner has complete information about the initial state
  – Non-deterministic: There may be more than one possible values of f(a, s), for some action a that is applicable in some state s.
Non-Determinism

Jar tips over?

Or not?
Types of Plans for FOND

• Policy: A function that maps states to actions

• Types of policies:
  – **Weak**: Achieves the goal under one set of outcomes
  – **Strong**: Always achieves the goal in finite steps
  – **Strong Cyclic**: Every state reachable from the initial state can *eventually* reach the goal (PRP)
LTL

□ \( \varphi \)  “Always \( \varphi \)”

◇ \( \varphi \)  “Eventually \( \varphi \)”

○ \( \varphi \)  “Next \( \varphi \)”

\( \varphi \) U \( \psi \)  “\( \varphi \) Until \( \psi \)”
Temporally Extended Goals (TEGs)

\[ \square \left( \text{CookiesRemaining} \implies \Diamond \left( \text{StealingCookie} \land \left( \bigcirc \text{EatingCookie} \right) \right) \right) \]

\( (\text{StealingCookie} \cup (\text{Caught} \lor \text{JarEmpty})) \)
Compiling the LTL away

**Theorem:**
For every LTL formula $\phi$, there exist a Büchi Automaton that accepts all and only the models of $\phi$

A Büchi automaton: $<\Sigma, Q, Q0, \rho, F>$ where
- $\Sigma$: input Alphabet
- $Q$: set of states
- $q0$: initial state
- $\rho$: transition function
- $F$: set of accepting states
Controller Synthesis

The problem:

**Input:** A Non-Deterministic domain D representing some system and an LTL formula $\phi$.

**Output:** A Finite State Controller (FSC) that fairly realizes $\phi$ on D.
Controller Synthesis

Given a domain $D = \langle A, \text{Prop}, S, s_0, f \rangle$, a FSC for $D$ is a tuple:

$\text{FSC} = \langle C, c_0, \Gamma, \Lambda, \delta, \Omega \rangle$ where

- $C$: set of controller states
- $c_0$: initial controller state
- $\Gamma = S$: input alphabet
- $\Lambda = A$: output alphabet
- $\delta$: transition function
- $\Omega$: controller output function
The Approach

Planning Domain $D = \langle A, \text{Prop}, S, s_0, f \rangle$

Automaton $A = \langle S, Q, q_0, \rho, F \rangle$

Domain $D' = \langle A, \text{Prop}', (S, Q), (s_0, q_0), f' \rangle$
Reference

Paper on Controller Synthesis representing the current state-of-the-art:

Authors: Patrizi, Lipovetzky, Geffner
Published in IJCAI’13
Thank you!

Questions?