Hierarchical Temporal Logic Specifications for Abstract Safety Tasks

Xusheng Luo[1], Changliu Liu[1]
[1]Carnegie Mellon University

Introduction

Questions:
How can we formally and abstractly specify intangible safety constraints, which are more subtle than collision avoidance?

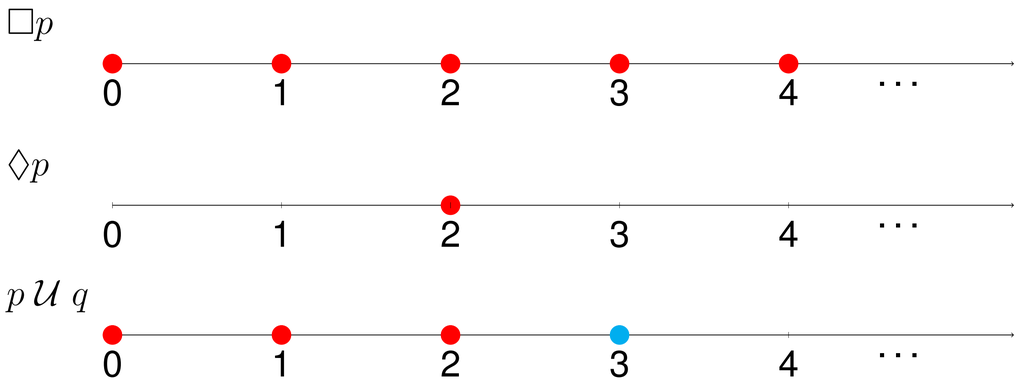
Key contributions:

- We introduced a hierarchical linear temporal logic over finite traces (H-LTL$_f$) that is capable of specifying abstract safety;
- We conducted a user study to show that the H-LTL$_f$ is easier to interpret compared to the standard LTL$_f$;
- We developed a planning algorithm, achieving task allocation and planning for multi-robot systems subject to H-LTL$_f$.

LTL$_f$ and H-LTL$_f$

Semantics of LTL$_f$:

$p =$ $q =$



Scenario 1: Robots are required to distribute documents to desks d_{10}, d_7, and d_5, and avoid public areas while carrying the document. Let carry denote the action of the robot carrying the document.

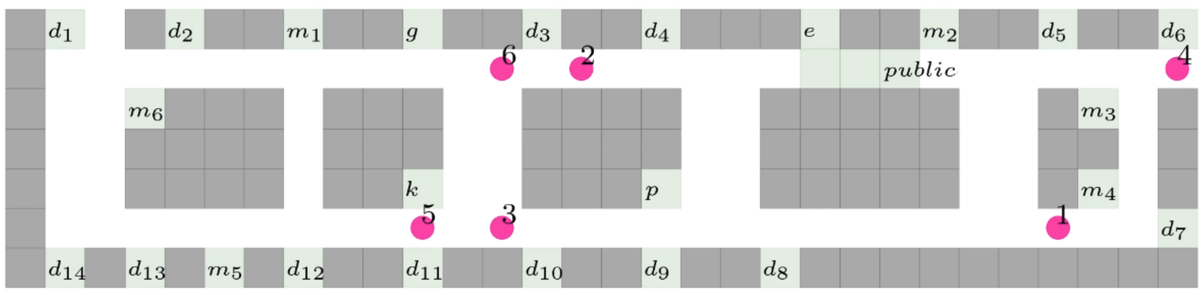


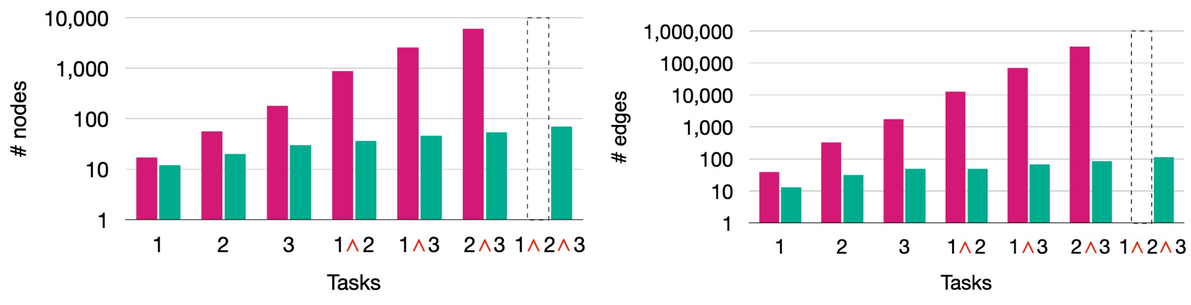
Figure 1: An office building in a grid-based layout.

Standard LTL$_f$:

$$\begin{aligned}\phi = \Diamond(p \wedge \text{carry} \,\mathcal{U}\, (d_{10} \wedge \bigcirc\neg\text{carry})) \\ \wedge \Diamond(p \wedge \text{carry} \,\mathcal{U}\, (d_7 \wedge \bigcirc\neg\text{carry})) \\ \wedge \Diamond(p \wedge \text{carry} \,\mathcal{U}\, (d_5 \wedge \bigcirc\neg\text{carry})) \\ \wedge \Box(\text{carry} \Rightarrow \neg\text{public})\end{aligned} \tag{1}$$

H-LTL$_f$:

$$\begin{aligned}L_1: \quad & \phi_1^1 = \Diamond\phi_2^1 \wedge \Diamond\phi_2^2 \wedge \Diamond\phi_2^3 \\ L_2: \quad & \phi_2^1 = \Diamond(p \wedge \text{carry} \,\mathcal{U}\, (d_{10} \wedge \bigcirc\neg\text{carry})) \wedge \text{notpublic} \\ & \phi_2^2 = \Diamond(p \wedge \text{carry} \,\mathcal{U}\, (d_7 \wedge \bigcirc\neg\text{carry})) \wedge \text{notpublic} \\ & \phi_2^3 = \Diamond(p \wedge \text{carry} \,\mathcal{U}\, (d_5 \wedge \bigcirc\neg\text{carry})) \wedge \text{notpublic} \\ & \text{notpublic} := \Box(\text{carry} \Rightarrow \neg\text{public})\end{aligned} \tag{2}$$



(a) Number of nodes. (b) Number of edges.

Figure 2: Comparison of automaton sizes of LTL$_f$ (red) and H-LTL$_f$ (green).

H-LTL$_f$ is More User-Friendly

- Carried out the survey on Prolific (www.prolific.com), recruiting 25 anonymous participants.
- A robotic arm is tasked to sort blocks into designated areas.

pattern

$\Diamond a \wedge \Diamond b$
Events a and b occur in any order.

$\Diamond(a \wedge \Diamond b)$
First events a occurs and then event b occurs.

$\neg a \,\mathcal{U}\, b$
Events a should not occur until event b occurs

Task:

First place the circle, and then place diamond and star in any order, and lastly place the heart.

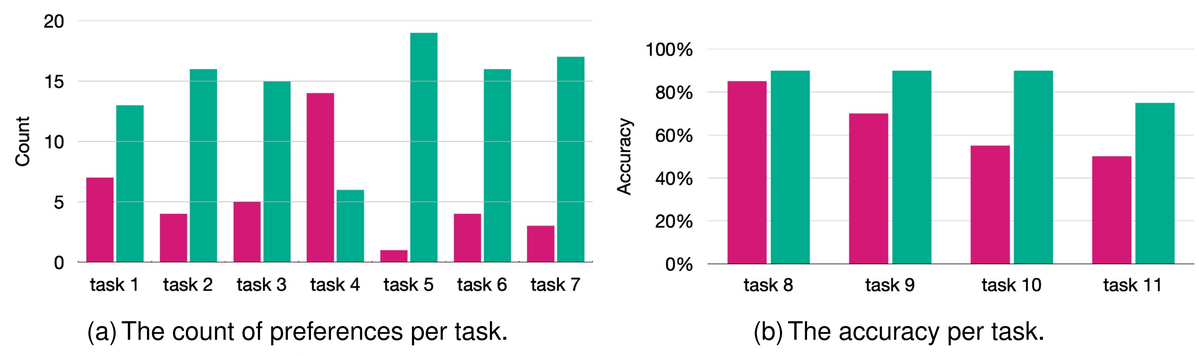
Option 1:

$$\phi = \Diamond(s_{cc} \wedge \Diamond t_{cc}) \wedge \Diamond(s_{dm} \wedge \Diamond t_{dm}) \wedge \Diamond(s_{st} \wedge \Diamond t_{st}) \wedge \Diamond(s_{ht} \wedge \Diamond t_{ht}) \wedge \neg s_{dm} \,\mathcal{U}\, t_{cc} \wedge \neg s_{st} \,\mathcal{U}\, t_{cc} \wedge \neg s_{ht} \,\mathcal{U}\, t_{dm} \wedge \neg s_{ht} \,\mathcal{U}\, t_{st}$$

Option 2:

$$\begin{aligned}L_1: \quad & \phi = \Diamond(a \wedge \Diamond(b \wedge \Diamond c)) \\ L_2: \quad & a = \Diamond(s_{cc} \wedge \Diamond t_{cc}) \\ & b = \Diamond(s_{dm} \wedge \Diamond t_{dm}) \wedge \Diamond(s_{st} \wedge \Diamond t_{st}) \\ & c = \Diamond(s_{ht} \wedge \Diamond t_{ht})\end{aligned}$$

Figure 3: An illustration of the interface used during the user study.

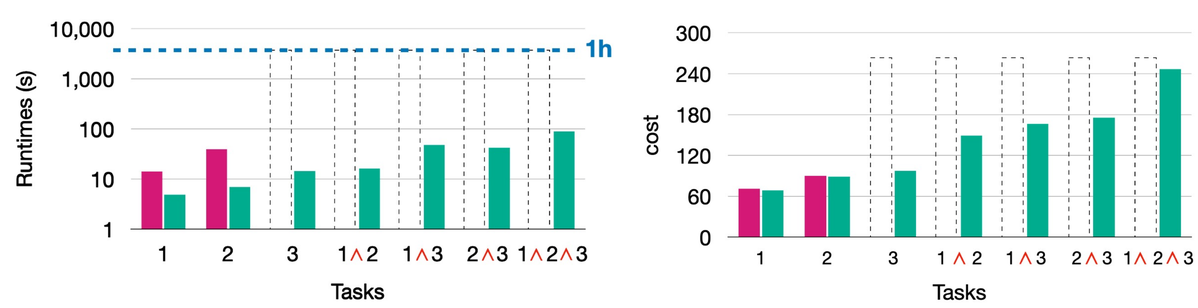


(a) The count of preferences per task. (b) The accuracy per task.

Figure 4: Subjective (a) and objective (b) results of the user study.

- Over 70% of participants favored the hierarchical specification.
- Accuracy for tasks using hierarchical formulas was 86% (20% higher).

Task Allocation and Planning



(a) Runtimes (b) Cost

Figure 5: Comparison between LTL$_f$ (red)and H-LTL$_f$ (green).

- For the standard method, it was unable to return a solution within 1 hour for most scenarios, except for the simpler tasks 1 and 2.

Four robot arms in tight space:

$$\begin{aligned}L_1: \quad & \phi_1^1 = \Diamond(\phi_2^1 \wedge \Diamond(\phi_2^2 \wedge \Diamond\phi_2^3)) \\ L_2: \quad & \phi_2^1 = \Diamond(\text{target1} \wedge \Diamond\text{target3}) \\ & \phi_2^2 = \Diamond(\text{target2} \wedge \Diamond\text{target4}) \\ & \phi_2^3 = \Diamond(\text{target3} \wedge \neg\text{obstacle} \,\mathcal{U}\, \text{target1}).\end{aligned} \tag{3}$$

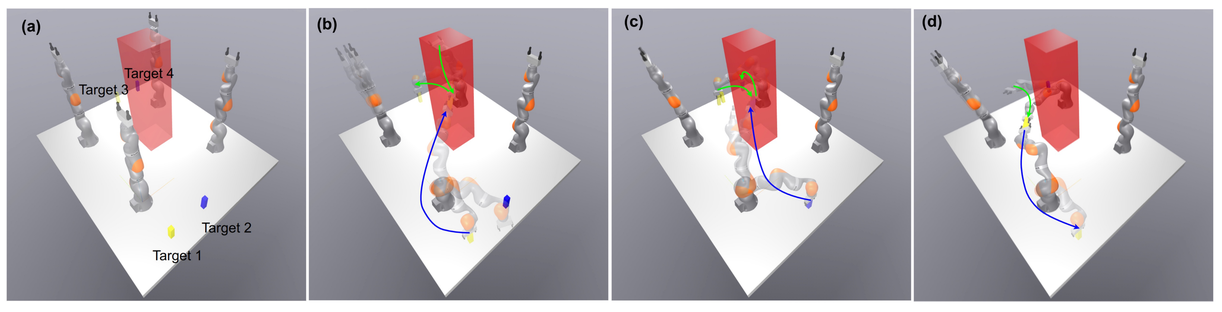


Figure 6: Conditional region avoidance for pick-and-place tasks.