STL Notes

David Kooi

May 28, 2019

1 Autonomous Vehicles

1.1 Intersection Management

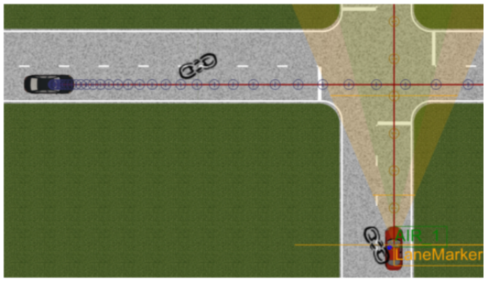

Figure 1: Intersection Scenario

In [?] the ego-vehicle's behavior at an intersection is defined using STL. The system has a state:

$$\mathbf{x_t} = [x_t^{ego}\ y_t^{ego}\ x_t^{adv}\ y_t^{adv}\ v_t^{ego}\ v_t^{adv}\ ]^T \tag{1}$$

Figure 1 shows the ego-vehicle in blue and the adversary-vehicle in red. The authors of [?] use the following specification to describe the behavior: ϕ_s: "When the ego-vehicle is within 2 meters of the adversary-vehicle the ego-vehicle will stop for two seconds."

$$\phi_s = \square(|x_t^{ego} - x_t^{adv}| < 2) \implies \square_{[0,2]}(|v_t^{ego}| < 0.1) \tag{2}$$

1.1.1 Alternate Formulation

The above STL specification relies on the adversary-vehicle to stop at the intersection, but a car should always stop at an intersection. Inspired from [?] the following specification describes: "When the ego-vehicle reaches an intersection it will stop for one second and only cross whens safe (ϕ_{cross}) otherwise it will stay stopped. ($\phi_{stopped}\mathcal{U}_{[1,3]}\phi_{stopped}$)."

$$\phi_{stopped} = (|v_t^{ego}| == 0) \tag{3}$$

$$\phi_{cross} = \square_{[1,3]}(|y_t^{adv} - \mathbf{Y^{INT}}| > 0.1) \wedge (x_t^{ego} > 5) \tag{4}$$

$$\phi_{int} = \square(|x_t^{ego} - \mathbf{X^{INT}}| < 0.1) \implies \square[0.5,1](\phi_{stopped}) \wedge (\phi_{stopped}\mathcal{U}_{[1,3]}\phi_{cross} \vee \phi_{stopped}\mathcal{U}_{[1,3]}\phi_{stopped}) \tag{5}$$

Where $\mathbf{X^{INT}} = -5$ and $\mathbf{Y^{INT}} = -5$ are the positional coordinates of the intersection.

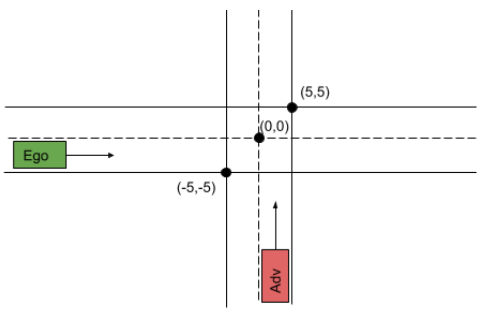

Figure 2: Alternate Scenario

2 STL for Hybrid Systems

2.1 Hybrid Time STL

2.1.1 Hybrid System Solutions

Solutions to hybrid systems are defined on ordinary and discrete time:

1. Ordinary Time: $t \in \mathbb{R}_{\geq 0} := [0, \infty)$
2. Discrete Time: $j \in \mathbb{N} := \{0, 1, 2, ...\}$

Definition 2.1 (Hybrid Time Instance)**.** A *hybrid time instance* is given by

$$(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$$

Definition 2.2 (Compact Hybrid Domain)**.** A set $E \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ is a *compact hybrid domain* if it can be written as

$$E = \bigcup_{j=0}^{J-1} ([t_j, t_{j+1}] \times \{j\}) \tag{6}$$

for a finite sequence of times $0 = t_0 \leq t_1 \leq t_2 \leq ... \leq t_J$.

Definition 2.3 (Hybrid Domain)**.** A set $E \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ is a *hybrid domain* if for each $(T, J) \in\ E$

$$E \cap ([0, T] \times \{0, 1, ..., J\})$$

is a compact hybrid time domain.

Definition 2.4 (Compact Hybrid Shift)**.** Given a compact hybrid domain E and (t^*, j^*), the forward compact hybrid shift of E by (t^*, j^*) is

$$(t^*, j^*) + E = \bigcup_{j=0}^{J-1} ([t_j + t^*, t_{j+1} + t^*] \times \{j + j^*\} \tag{7}$$

and the backward hybrid shift of E by (t^*, j^*) is

$$E - (t^*, j^*) = \bigcup_{j=0}^{J-1} ([t_j - t^*, t_{j+1} - t^*] \times \{j - j^*\} \tag{8}$$

for a finite sequence of times $0 = t_0 \leq t_1 \leq t_2 \leq ... \leq t_J$ and E satisfying (6).

Definition 2.5 (Compact Hybrid Interval)**.** A *compact hybrid interval* is defined by two hybrid time instances (t^*_A, j^*_A) and (t^*_B, j^*_B), where $t^*_A + j^*_A \leq t^*_B + j^*_B$, and an hybrid domain E, over the range $[t^*_A, t^*_B] \times \{j^*_A, ..., j^*_B\}$, by

$$\mathcal{I} := (t^*_A, j^*_A) + E = \bigcup_{j=0}^{J-1} ([t_j + t^*_A, t_{j+1} + t^*_A] \times \{j + j^*_A\} \tag{9}$$

- where $J = j^*_B - j^*_A$ and $t_0 = 0 \leq t_1 \leq t_2 \leq ... \leq t_J = t^*_B - t^*_A$.

Definition 2.6 (Open Hybrid Interval)**.** An open hybrid interval is defined by two hybrid time instances, (t^*_A, j^*_A) and (t^*_B, j^*_B), on hybrid domain E, over the range $[t^*_A, t^*_B] \times \{j^*_A, ..., j^*_B\}$, by the following cases:

1. Left open interval

$$\mathcal{I} := \Big((t^*_A, t_1] \times \{j^*_A\} \Big) \cup \Big\{ (t^*_A + t_1, j^*_A + 1) + E \Big\} = \bigcup_{j=0}^{J-1} ([t_j + t^*_A + t_1, t_{j+1} + t^*_A + t_1] \times \{j + j^*_A + 1\} \tag{10}$$

   - where $J = j^*_B - j^*_A$ and $t_0 = 0 \leq t_1 \leq t_2 \leq ... \leq t_J = t^*_B - t^*_A - t_1$;

2. Right open interval

$$\mathcal{I} := \Big\{ (t^*_A, j^*_A) + E \Big\} \cup \Big([t_{j^*_B}, t^*_B) \times \{j^*_B\} \Big) = \bigcup_{j=0}^{J-1} ([t_j + t^*_A, t_{j+1} + t^*_A] \times \{j + j^*_A\} \cup \Big([t_{j^*_B}, t^*_B) \times \{j^*_B\} \Big) \tag{11}$$

   - where $J = j^*_B - j^*_A - 1$ and $t_0 = 0 \leq t_1 \leq t_2 \leq ... \leq t_J = t_{j^*_B} - t^*_A$, and $t_J \leq t_{j^*_B}$; ;

3. Open interval

$$\mathcal{I} := \left((t_A^*, t_1] \times \{j_A^*\}\right) \cup \left\{(t_A^* + t_1, j_A^* + 1) + E\right\} \cup \left([t_{j_B^*}, t_B^*) \times \{j_B^*\}\right) \tag{12}$$

$$= \left((t_A^*, t_1] \times \{j_A^*\}\right) \cup \left\{ \bigcup_{j=0}^{J-1} ([t_j + t_A^* + t_1, t_{j+1} + t_A^*] \times \{j + j_A^* + 1\} \right\} \cup \left([t_{j_B^*}, t_B^*) \times \{j_B^*\}\right) \tag{13}$$

- where $J = j_B^* - j_A^* - 1$ and $t_0 = 0 \leq t_1 \leq t_2 \leq ... \leq t_J = t_{j_B^*} - t_A^*$, and $t_J \leq t_{j_B^*}$;

Remark: Authors of [**?**] denote a ordinary time interval $\mathcal{I}$, relative to ordinary time t as:

$$t + \mathcal{I} := \{t + t' : t' \in \mathcal{I}\}$$

The same notation can be used denoting a hybrid temporal interval, $\mathcal{I}$ relative to hybrid time instance (t, j):

$$(t, j) + I := \{(t + t', j + j') : (t', j') \in I\}$$

2.2 Continuous Time STL

A STL formula ϕ is defined recursively as:

$$\psi ::= p^{\mu} \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \Diamond_{\mathcal{I}}\psi \mid \psi_1 U_{\mathcal{I}} \psi_2 \tag{14}$$

Where ψ_1, ψ_2 are STL formula and $\mathcal{I}$ is an ordinary time interval of form $\mathcal{I} = (t_a, t_b), (t_a, t_b], [t_a, t_b)$, or $[t_a, t_b]$ with $t_a < t_b$ and $\mathcal{I} \subset \mathbb{R}_{\geq 0}$.

Inductive STL Definition

$$\begin{aligned}
&\phi \models \psi && \Leftrightarrow (\phi, t) \models p \\
&(\phi, t) \models p^{\mu} && \Leftrightarrow \mu(\phi(t)) > 0 \\
&(\phi, t) \models \Box_{[a,b]}\psi && \Leftrightarrow \forall t' \in [t+a, t+b]\ s.t\ (\phi, t') \models \psi
\end{aligned}$$

2.2.1 Possible Notion of Hybrid STL

Recursive STL Definition

$$\psi ::= p^{\mu} \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \Diamond_I \psi \mid \psi_1 U_I \psi_2 \tag{15}$$

Where ψ_1, ψ_2 are STL formula and I is a hybrid time interval.

Inductive STL Definition

$$\begin{aligned}
&\phi \models \psi && \Leftrightarrow (\phi,(t,j)) \models p \\
&(\phi,(t,j)) \models p^{\mu} && \Leftrightarrow \mu(\phi(t)) > 0 \\
&((\phi,(t,j)) \models \Box_I \psi && \Leftrightarrow \forall (t',j') \in (t,j) + I \; s.t. \; (\phi,(t',j')) \models \psi
\end{aligned}$$

2.2.2 Previous Notions of Hybrid STL

From Atreya - Modeling Hybrid Systems using Signal Temporal Logic

Inductive STL Definition

$$((\phi,(t,j)) \models \Box_{(t,j)} \psi \Leftrightarrow \forall (t',j') \in dom\ \phi, t' + j' > t + j \; s.t \; (\phi,(t',j')) \models \psi \tag{16}$$

.1 Lane Changing

The authors of [?] use STL and Hamilton Jacobi reachability to combine temporal specification with quantification of a reachable set: STL temporal specifications are related to reachability operators.

.1.1 Example

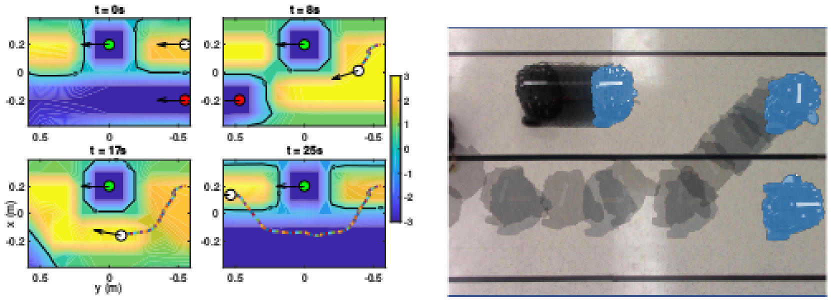

Figure 3: Lane Change Scenario

In Figure 3 white represents the autonomous car, green a slow car, and red an adjacent car. The white car's lane change specification is given as: "Within 25 seconds, satisfy general safety ϕ until you can pass. Otherwise stay within the lane. Also ensure you are always within 5 seconds from re-entering the lane.(ψ^{lane}.)"

$$\phi = \phi^{off-road} \wedge \phi^{on-road} \wedge \phi^{avoid} \tag{17}$$

$$\Box_{[0,25]} \Diamond_{[0,5]} \psi^{lane} \wedge (\phi \mathcal{U}_{[0,25]} \psi^{pass} \vee \phi \mathcal{U}_{[0,25]} \psi^{stay}) \tag{18}$$

.2 Solutions for Discrete Systems

Consider a discrete system of the form:

$$x^+ \in G(x, \gamma) \tag{19}$$

Let $x \in \mathbb{R}^n$ be states of the discrete system.
Let $\gamma \in \mathbb{R}^m$ be input to the discrete system.
Discrete time is denoted: $k \in \mathbb{N}$
Then, given an initial state, x_0 and an input $k \mapsto \gamma(k)$, a state trajectory(solution) can be constructed. The domain of γ is given: $dom\ \gamma = \{0, 1, 2, ..., K\} \cap \mathbb{N}$ where $K \in \mathbb{N} \cup \{\infty\}$. Solutions are defined as: $k \mapsto \phi(k)$ such that the following are satisfied:

1. $\phi(0) = x_0$
2. $dom\ \phi = dom\ \gamma$ if $K = \infty$
3. $dom\ \phi \setminus \{max\ dom\ \phi\} = dom\ \gamma$ if K is finite
4. $\forall k \in dom\ \gamma$: $\phi(k+1) = G(\phi(k), \gamma(k))$

.2.1 Discrete Time STL

$$\psi ::= p^\mu \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \Diamond_I \psi \mid \psi_1 U_I \psi_2 \tag{20}$$

Where ϕ_1, ϕ_2 are STL formula and I is an interval on $dom\ \phi$.

Inductive STL Definition

$$\begin{aligned}
\sigma \models \psi \quad &\Leftrightarrow (\sigma, k) \models p \\
(\sigma, k) \models p^\mu \quad &\Leftrightarrow \mu(x_k, \gamma_k) > 0 \\
(\sigma, k) \models \Box_{[n,m]}\psi &\Leftrightarrow \forall k' \in [k' + n, k' + m]\ s.t\ (\sigma, k') \models \psi
\end{aligned}$$