Details of the First-Order Logic Representation in "Model Predictive Control of Priced Timed Automata Encoded with First-Order Logic"

Efe C. Balta*†, Ilya Kovalenko†, Isaac A. Spiegel†, Dawn M. Tilbury, Kira Barton

Rev 1: October 12, 2020

1 Introduction

This document provides technical details to assist in the implementation of techniques presented in:

> E. C. Balta, I. Kovalenko, I. A. Spiegel, D. M. Tilbury, and K. Barton, "Model Predictive Control of Priced Timed Automata Encoded with First-Order Logic," *IEEE Transactions on Control Systems Technology,* Conditionally accepted, 2020.

This text was included in the peer-reviewed and accepted manuscript submitted to *IEEE Transactions on Control Systems Technology*, but was cut in the final version to meet the page limit.

This document overviews the methods used to express a priced timed automaton (PTA) in first-order logic (FOL). Note that this document uses the same assumptions and notation as [1]. Using the FOL representation described in the rest of this document, [1] describes how to formulate a model predictive control (MPC) problem using the FOL PTA representation (i.e., the PTA-MPC problem).

2 First-order Logic Representation of the Graph

Suppose each location in Q is represented as the j^{th} standard basis vector in $\mathbb{B}^{n_q}$, $\mathbf{q}^j$, where $n_q = |Q|$ and $\mathbb{B} = \{0, 1\}$. Similarly, each edge in E can be represented as the jth standard basis vector in $\mathbb{B}^{n_e}$, $\mathbf{e}^j$ where $n_e = |E|$. Then, let $A \in \{-1, 0, 1\}^{n_q \times n_e}$ represent the incidence matrix of $\mathcal{A}$ where an element $A(i, j) = 1$ (positive incidence) indicates that $\mathbf{e}^j$ transitions to $\mathbf{q}^i$ and $A(i, j) = -1$ (negative incidence) indicates that $\mathbf{e}^j$ transitions from $\mathbf{q}^i$.

Define the input transition matrix $B_{in} \triangleq \max(A, 0)$, where max is computed element-wise for each element of the first argument and $\triangleq$ denotes a definition for the left-hand side, that maps an edge to the location it transitions to. Then, $\mathbf{q}_{i+1} = B_{in}\mathbf{e}_{i+1}$ where $\mathbf{e}_{i+1}$ is the edge from a location $\mathbf{q}_i$ to $\mathbf{q}_{i+1}$, and $\mathbf{q}_{i+1}$ is the location following the discrete transition. Similarly, define the output transition matrix $B_{out} \triangleq \max(-A, 0)$ that maps an edge to the location from where it transitioned. Therefore, $\mathbf{q}_i = B_{out}\mathbf{e}_{i+1}$.

Next, define $\tilde{A} \triangleq \max(-A^T B_{in}, 0)$ as the matrix that maps an edge $\mathbf{e}^j$ into the set of edges that are enabled after the execution of $\mathbf{e}^j$. Here, enabling is defined in terms of connectivity and not the guards and invariants, so that if an edge is negatively incident to a location, then that edge is enabled at that location. A vector of edges enabled following the traversal of any edge $\mathbf{e}_i$ can thus be defined as $\tilde{\mathbf{e}}_{i+1} = \tilde{A}\mathbf{e}_i$ with initial condition $\tilde{\mathbf{e}}_1 = \max(-A^T, 0)\mathbf{q}_0$.

---

†The authors have equal contributions to this work. *Corresponding Author.

This work is supported by the National Science Foundation (CNS Grant #1544678, CAREER Award #1351469, and a Graduate Fellowship), a gift from Rockwell Automation, and NIST Award No.70NANB19H090.

Efe C. Balta, Ilya Kovalenko, Isaac A. Spiegel, Dawn M. Tilbury, and Kira Barton are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 {baltaefe, ikoval, ispiegel, tilbury, bartonkl} @umich.edu

Algorithm 1 Global clock valuation constraint assignment

1: **function** GLOCLOCK($E, B_{in}, B_{out}, Q, val(\mathbf{c}), U$)
2:   *Initialize*: $\Phi \leftarrow \{\}, \Lambda \leftarrow \{\}$
3:   **for all** $\mathbf{q}^i \in Q \setminus \mathbf{q_0}$ **do**
4:    $\delta_{in} \leftarrow B_{in}^T \mathbf{q}^i$
5:    **if** $\delta_{in} \in E$ **then** $\mathbf{q}_m = B_{out}\delta_{in}$
6:     $\Lambda \leftarrow \Lambda \cup \{c_g(\mathbf{q}^i) = c_l(\mathbf{q}^i) + c_g(\mathbf{q}^m)\}$
7:    **else**
8:     Decompose $\delta_{in} = \sum_j \mathbf{e}^j$
9:     **for all** $\mathbf{e}^j \in \delta_{in}$ **do** $\mathbf{q}^m = B_{out}\mathbf{e}^j$
10:      $\phi_m \leftarrow \{c_g(\mathbf{q}^i) = c_l(\mathbf{q}^i) + c_g(\mathbf{q}^m)\}$
11:      $\Phi \leftarrow \Phi \cup \{\bigvee_{k\in[1,N]} u_k(\mathbf{e}^j) \Leftrightarrow \phi_m\}$
12:     **end for**
13:    **end if**
14:   **end for**
15:   **Output:** $\mathcal{C} \leftarrow \Phi \cup \Lambda$
16: **end function**

To constrain the discrete transition to one of the possible choices in the vector $\tilde{\mathbf{e}}$, the "multiple exclusive or" Boolean function is introduced:

$$\texttt{mXor}(\mathbf{a}, \mathbf{b}) \triangleq \mathbf{a}^T\mathbf{b} == 1 \tag{1}$$

where $\mathbf{a}$ and $\mathbf{b}$ are binary column vectors of equal length. This function enables the encoding of the automaton connectivity as a set of first-order logical constraints on the edges taken during discrete transitions: each edge $\mathbf{e}_i$, denoting the i^{th} discrete transition in a path, must satisfy $\texttt{mXor}(\mathbf{e}_i, \tilde{\mathbf{e}}_i)$.

Finally, a first-order logic MPC horizon, N_{mpc}, is defined as the length of the horizon for the MPC formulation. The decision variables of the first-order logic MPC are the discrete transitions of a path on a PTA. Then, the sequence $U = \{u_1, \ldots, u_{N_{mpc}}\}$ denotes the decision variables for the first-order logic MPC problem, so that $u_i \in \mathbb{B}^{n_e}$.

3 First-order Logic Representation of Time

The guards and invariants of the set $\mathcal{B}(C)$ of the PTA must also be represented in first-order logic. With Assumption 3 in [1], the PTA has two clocks: a global clock and a local clock. Local clock valuations are denoted with $c_l(\mathbf{q}^j)$ and represent the time spent at a given location. Valuation of a global clock at state $\mathbf{q}^j$ is denoted with $c_g(\mathbf{q}^j)$. A global clock valuation represents the time spent since the beginning of a path. The vector of all clock valuations is defined as $val(\mathbf{c}) = [c_l(\mathbf{q}^0), c_g(\mathbf{q}^0), \ldots, c_l(\mathbf{q}^{n_q}), c_g(\mathbf{q}^{n_q})]^T$. This schema allows the encoding of the algebraic constraints in $\mathcal{B}(C)$ in an algorithmic way for each state in the PTA.

By Assumption 3 in [1], the global clock of a PTA is never reset. This poses a consistency constraint on the valuations of the global clock at different $\mathbf{q}^i$, such that the global clock valuation at $\mathbf{q}^i$ must be $c_g(\mathbf{q}^i) = c_l(\mathbf{q}^i) + c_g(\mathbf{q}^m)$ if and only if a discrete transition from $\mathbf{q}^m$ to $\mathbf{q}^i$ is taken in a path (i.e. $\big(c_g(\mathbf{q}^i) = c_l(\mathbf{q}^i) + c_g(\mathbf{q}^m)\big) \Leftrightarrow e^j$, where e^j is the edge from $\mathbf{q}^m$ to $\mathbf{q}^i$). Algorithm 1 presents the global clock valuation constraint assignment. The assignment is performed between all the locations in the PTA except for the initial location q_0. Let $\delta_{in} \in \mathbb{B}^{n_e}$ denote a vector that corresponds to the edges with positive incidence to $\mathbf{q}^i$. Then, each 1 in the vector δ_{in} corresponds to an $\mathbf{e}^j \in E$. So, if $\delta_{in} \in E$, there is a single incoming edge (positive incidence) to the location $\mathbf{q}^i$. The constraint given in line 6 is assigned for the single incoming edge case. If δ_{in} contains multiple ones, the vector is decomposed into the unit basis vectors $\mathbf{e}^j$ and the proposition ϕ_m is created for each $\mathbf{e}^j$. After evaluating the proposition ϕ_m, the constraint given in line 11 is evaluated. The constraint reads as: the proposition ϕ_m is true if and only if the discrete transition belonging to the edge e^j is executed within the MPC-horizon N_{mpc}. By using Algorithm 1, the consistency constraint on the global clock valuations is maintained.

Algorithm 2 presents the guard condition assignment for the clock valuations in $val(\mathbf{c})$. The assignment is performed for all locations in the PTA except for the initial location $\mathbf{q}_0$. Let the vector $\delta_{out} \in \mathbb{B}^{n_e}$ denote

Algorithm 2 Guard condition assignment

1: **function** ASSGNGUARD($E, B_{in}, B_{out}, Q, val(\mathbf{c}), U$)
2: *Initialize*: $\Xi \leftarrow \{\}, \Omega = \{\}$
3: **for all** $\mathbf{q}^i \in Q \setminus \mathbf{q_0}$ **do**
4: $\delta_{out} \leftarrow B_{out}^T \mathbf{q}^i$
5: **if** $\delta_{out} \notin E$ **then** $\delta_{out} = \sum_j e^j,\ \Theta \leftarrow \{\}$
6: **for all** $e^j \in \delta_{out}$ **do** $\theta \leftarrow e^j$
7: **if** $g^{e^j+1} = g^\theta$ **then** ▷ equivalent guards
8: $\Theta \leftarrow \Theta \cup \theta$
9: **else**
10: $\varepsilon \leftarrow (c_l(\mathbf{q}^i) \models g^{e^j})$
11: $\Xi \leftarrow \Xi \cup \{\bigvee_{k\in[1,N]} u_k(e^j) \Leftrightarrow \varepsilon\}$
12: **end if**
13: **end for**
14: $\varepsilon \leftarrow (c_l(\mathbf{q}^i) \models g^{e^j})$
15: $\Xi \leftarrow \Xi \cup \{\bigvee_{e^j\in\Theta} \bigvee_{k\in[1,N]} u_k(e^j) \Leftrightarrow \varepsilon\}$
16: **else** $\delta_{out} \in E$
17: $\Omega \leftarrow \Omega \cup \{c_l(\mathbf{q}^i) \models g^{e^j}\}$
18: **end if**
19: **end for**
20: **Output:** $\mathcal{G} \leftarrow \Xi \cup \Omega$
21: **end function**

a vector that corresponds to the edges with negative incidence to $\mathbf{q}^i$. If δ_{out} contains more than one edge, it is decomposed into the edges it corresponds to ($\delta_{out} = \sum_j \mathbf{e}^j$). The algorithm first checks if the multiple negatively incident edges have the same guard condition g^{e^j}.

The proposition in line 11 of Algorithm 2 is assigned to ensure that $c_l(\mathbf{q}^i)$ will satisfy the guard condition if a discrete transition belonging to edge e^j is executed. To have a feasible satisfaction problem in first-order logical constraints, all the Boolean equation constraints of a system must evaluate true. The constraints evaluated at line 11 of Algorithm 2 make the equivalent guard conditions of two outgoing edges (negative incidence) infeasible when one of the edges has an associated discrete transition executed. To circumvent this situation for equivalent guard conditions on outgoing edges, the proposition in line 15 is evaluated. The proposition states that a local clock value satisfies the equivalent guard condition if and only if either of the edges with the equivalent guard condition is executed.

Finally, the invariants of the PTA can be encoded with first-order logic via linear constraints. Each location of the PTA has a nonnegative number of invariants, η^i, associated with it. The invariants are put in the standard form as:

$$\Psi^i \left(\begin{smallmatrix} y \\ x \end{smallmatrix}\right) \leq \mathbf{t}^i \tag{2}$$

where x and y are the global and local clock valuations at a location, $\Psi^i \in \mathbb{R}^{\eta^i \times 2}$, and $\mathbf{t}^i \in \mathbb{R}^{\eta^i}$. The vectors representing all of the invariants can be built by letting $\Psi = diag(\Psi^1, \ldots, \Psi^{n_q})$ and $\mathbf{t} = [\mathbf{t}^1 \ldots \mathbf{t}^{n_q}]^T$ satisfying:

$$\mathbf{\Psi}\, val(\mathbf{c}) \leq \mathbf{t} \tag{3}$$

where $\Psi \in \mathbb{R}^{\eta \times 2n_q}$, $\mathbf{t}^i \in \mathbb{R}^\eta$, and η is the number of invariant constraints. This forms the first-order logic representation of the invariants of the PTA.

The proposed encoding scheme for the first-order logic representation of time allows for an efficient formulation of the MPC problem in which the clock valuations only at the discrete transitions are considered. Presented time encoding scheme differs from the existing first-order logic time encoding schemes in literature such as [2]. The existing encoding schemes encode the time of delay and discrete transitions of a PTA in fixed discretized intervals, resulting in an increased number of constraints to represent time when compared to the proposed scheme here. The first-order logic formulation of the PTA-MPC problem utilizing the proposed graph and time representations is described in [1].

References

[1] E. C. Balta, I. Kovalenko, I. A. Spiegel, D. M. Tilbury, and K. Barton, "Model Predictive Control of Priced Timed Automata Encoded with First-Order Logic," *IEEE Transactions on Control Systems Technology*, Conditionally accepted, 2020.

[2] D. Bhave, S. N. Krishna, and A. Trivedi, "On nonlinear prices in timed automata," in *Electronic Proceedings in Theoretical Computer Science*, vol. 232, 2016, pp. 65–78.