Additional Definitions and Examples of the Paper

Symbolic State Estimation in Bounded Timed Labeled Petri Nets[1]

Yifan Dong, Naiqi Wu, and Zhiwu Li

Institute of Systems Engineering, Macau University of Science and Technology

In this note, we show some definitions with respect to the operations about MDDs and matrix diagrams that are critical for the paper entitled "Symbolic State Estimation in Bounded Timed Labeled Petri Nets". Then, some examples about those operations for the timed labeled Petri net (TLPN) are presented.

1 Operations about MDDs and matrix diagrams

Before the definitions of union of two MDDs (Definition 1), union of two matrix diagrams (Definition 2), and relational product of an MDD and a matrix diagrams (Definition 3), we introduce some notations pertaining to labeling function δ_t. Let $F = (Q, D, q_0, q_t, q_f, \delta_t)$ be an MDD. Given a non-terminal vertex $q_i \in Q_i$ $(i = 1, 2, \ldots, r)$ in an MDD and a label $d \in D$, $q' = \delta_t(q_i, d) \in Q$ is said to be the child vertex of q_i with respect to d, denoted as $q_i[d]$, i.e., $q' = q_i[d]$. If $\delta_t(q_i, d) = q_{i-1}$ ($i = 1$ implies $q_{i-1} = q_t$), the vertex q_i is said to be extensible with respect to a true child vertex q_{i-1}, denoted as $q_i[d]_t$; otherwise, $\delta_t(q_i, d) = q_f$, and q_i is said to be extensible with respect to the false child vertex q_f, denoted as $q_i[d]_f$. Specifically, for the terminal vertexes q_t and q_f, their child vertexes are themselves, i.e., $q_t = q_t[d]$ and $q_f = q_f[d]$.

Analogously, in a matrix diagram $H = (Q, \mathcal{D}, q_0, q_t, q_f, \delta_t)$, we use the notation $q_i[(d_u, d_v)]$ ($(d_u, d_v) \in \mathcal{D}$) to denote the child vertex of q_i with respect to (d_u, d_v). Also, we write $q_i[(d_u, d_v)]_t$ and $q_i[(d_u, d_v)]_f$ to denote that a non-terminal vertex q_i is extensible with respect to a true child vertex and the false child vertex, respectively.

Definition 1 Let $F' = (Q', D, q'_0, q'_t, q'_f, \delta'_t)$ and $F'' = (Q'', D, q''_0, q''_t, q''_f, \delta''_t)$ be two MDDs with the same label set D and the same number of levels ($r+1$ levels). The union of F' and F'' (denoted as $F' \cup F''$) is defined as $F_u = (Q,\ D,\ (q'_0, q''_0),\ (q'_t, q''_t),\ (q'_f, q''_f),\ \delta_t)$ such that the set of vertexes Q is $Q \subseteq Q' \times Q'' = Q_r \cup Q_{r-1} \cup \cdots \cup Q_0$, where $Q_r = \{(q'_0, q''_0)\}$,

$$Q_{i-1} = \begin{cases} \cup\{(q'_i[d], q''_i[d])\} & \text{if } (\forall d \in D)\ \neg(q'_i[d]_f\ \&\ q''_i[d]_f); \\ \emptyset & \text{otherwise,} \end{cases}$$

and

$$\delta_t((q'_i, q''_i), d) = \begin{cases} (q'_i[d], q''_i[d]) & \text{if } (\forall d \in D)\ \neg(q'_i[d]_f\ \&\ q''_i[d]_f); \\ (q'_f, q''_f) & \text{otherwise,} \end{cases}$$

($i = r, r-1, \ldots, 1$, $(q'_t, q''_f) = (q'_f, q''_t) = (q'_t, q''_t)$, and if $i = 1$, $Q_{i-1} = \{(q'_t, q''_t), (q'_f, q''_f)\}$). $\Diamond$

[1] To cite this note, please refer to Y. F. Dong, N. Q. Wu, and Z. W. Li, "State Estimation in TLPN," Available: https://github.com/dongyifan199/Opacity-on-TLPN, Jul. 2022.

For the union of two MDDs, i.e., F_u, its non-terminal vertexes in $Q = Q_r \cup Q_{r-1} \cup \cdots \cup Q_1$ are recursively defined from Q_r to Q_1. We now explain the details of the union of two MDDs F' and F''. Given two non-terminal vertexes $q'_i \in Q'_i$ and $q''_i \in Q''_i$, if for all $d \in D$, the child vertex of q'_i or q''_i with respect to d is extensible with respect to a true child vertex, i.e., $q'_i[d]_t$ or $q''_i[d]_t$, the vertex $(q'_i[d], q''_i[d])$ is added to Q_{i-1}.

As for the labeling function δ_t, if for all $d \in D$, a vertex $(q'_i[d], q''_i[d])$ is added to set Q_{i-1}, $\delta_t((q'_i, q''_i), d) = (q'_i[d], q''_i[d])$ is defined; otherwise, $\delta_t((q'_i, q''_i), d) = (q'_f, q''_f)$ is defined. At the level 1 of the recursive definition, if $q'_1[d]$ or $q''_1[d]$ is extensible with respect to the terminal vertex q_t, i.e., $q'_1[d]_t$ or $q''_1[d]_t$, we have $\delta_t((q'_1, q''_1), d) = (q'_t, q''_t)$.

Definition 2 Let $H' = (Q', \mathcal{D}, q'_0, q'_t, q'_f, \delta'_t)$ and $H'' = (Q'', \mathcal{D}, q''_0, q''_t, q''_f, \delta''_t)$ be two matrix diagrams with the same label pair set $\mathcal{D}$ and the same number of levels ($r+1$ levels). The union of H' and H'' (denoted as $H' \cup H''$) is defined as $H_u = (Q, \mathcal{D}, (q'_0, q''_0), (q'_t, q''_t), (q'_f, q''_f), \delta_t)$ such that the set of vertexes Q is $Q \subseteq Q' \times Q'' = Q_r \cup Q_{r-1} \cup \cdots \cup Q_0$, where $Q_r = \{(q'_0, q''_0)\}$,

$$Q_{i-1} = \begin{cases} \cup\{(q'_i[(d,d')], q''_i[(d,d')])\} & \text{if } (\forall (d,d') \in \mathcal{D})\ \neg(q'_i[(d,d')]_f \ \&\ q''_i[(d,d')]_f); \\ \emptyset & \text{otherwise,} \end{cases}$$

and

$$\delta_t((q'_i, q''_i), (d,d')) = \begin{cases} (q'_i[(d,d')], q''_i[(d,d')]) & \text{if } (\forall (d,d') \in \mathcal{D})\ \neg(q'_i[(d,d')]_f \ \&\ q''_i[(d,d')]_f); \\ (q'_f, q''_f) & \text{otherwise,} \end{cases}$$

$(i = r, r-1, \ldots, 1,\ (q'_t, q''_f) = (q'_f, q''_t) = (q'_t, q''_t)$, and if $i = 1$, $Q_{i-1} = \{(q'_t, q''_t), (q'_f, q''_f)\})$. $\Diamond$

Definition 3 Let $F' = (Q', D, q'_0, q'_t, q'_f, \delta'_t)$ be an MDD and $H'' = (Q'', \mathcal{D}, q''_0, q''_t, q''_f, \delta''_t)$ be a matrix diagram ($\mathcal{D} = D \times D$) with the same number of levels ($r+1$ levels). The relational product of F' and H'' (denoted as $F' \otimes H''$) is defined as $F_r = (Q, D, (q'_0, q''_0), (q'_t, q''_t), (q'_f, q''_f), \delta_t)$ such that the set of vertexes Q is $Q \subseteq Q' \times Q'' = Q_r \cup Q_{r-1} \cup \cdots \cup Q_0$, where $Q_r = \{(q'_0, q''_0)\}$,

$$Q_{i-1} = \begin{cases} \cup\{(q'_i[d], q''_i[(d,d')])\} & \text{if } (\forall d \in D)\ (\forall (d,d') \in \mathcal{D})\ q'_i[d]_t \ \&\ q''_i[(d,d')]_t; \\ \emptyset & \text{otherwise,} \end{cases}$$

and

$$\delta_t((q'_i, q''_i), d') = \begin{cases} (q'_i[d], q''_i[(d,d')]) & \text{if } (\forall d \in D)\ (\forall (d,d') \in \mathcal{D})\ q'_i[d]_t \ \&\ q''_i[(d,d')]_t; \\ (q'_f, q''_f) & \text{otherwise,} \end{cases}$$

$(i = r, r-1, \ldots, 1,\ (q'_t, q''_f) = (q'_f, q''_t) = (q'_f, q''_f)$, and if $i = 1$, $Q_{i-1} = \{(q'_t, q''_t), (q'_f, q''_f)\})$.

Delete the vertexes $q_n \in Q_n$ such that (for all $d \in D$) $\delta_t(q_n, d) = q_f$ (when such a vertex is deleted, its out-edges are removed as well, and the labeling function $\delta_t(q'_n, d') = q_n$ is changed to $\delta_t(q'_n, d') = q_f$, where $q'_n \in {}^\bullet q_n$). If there exist two vertexes $q^1_{i-1} \in Q_{i-1}$ and $q^2_{i-1} \in Q_{i-1}$ that are the child vertexes of $q_i \in Q_i$ where $\delta_t(q_i, d) = q^1_{i-1}$ and $\delta_t(q_i, d') = q^2_{i-1}$, such that $d = d'$ holds, merge vertexes q^1_{i-1} and q^2_{i-1}, and keep $\delta_t(q_i, d) = q^1_{i-1}$ or $\delta_t(q_i, d') = q^2_{i-1}$. $\Diamond$

Definition 3 describes the behavior of a timed next-state function. Namely, given a set of states represented by an MDD, under the transition relations represented by a matrix diagram, an MDD that represents the output set of states of the timed next-state function is generated.

2 Examples of operations

Examples in this section are derived from the TLPN in Fig. 1. We show the examples of operations about MDDs and matrix diagrams.

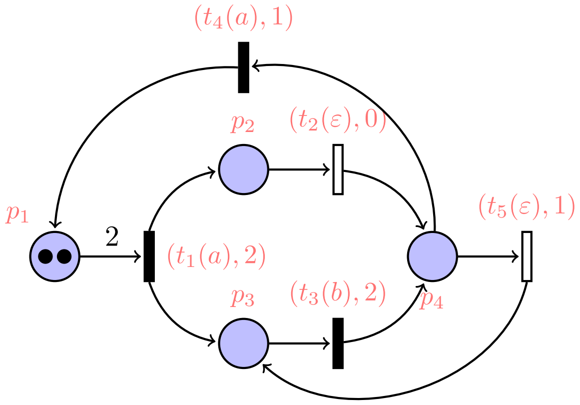

Figure 1: A TLPN.

Example 1 Fig. 2(a) shows an MDD $F = (Q,\, D,\, q_0,\, q_t,\, q_f,\, \delta_t)$, where $Q = \{q_0,\, q_4^0,\, q_4^1,\, q_3^0,\, q_3^1,\, q_3^2,\, q_2^0,\, q_2^1,\, q_2^2,\, q_1^0,\, q_1^1,\, q_t, q_f\}$ is a set of vertexes, $D = \{0, 1, 2\}$ is a set of labels, vertexes q_0, q_t, and q_f are the root vertex, terminal vertex valued with **True**, and terminal vertex valued with **False**, respectively. The labeling function δ_t is defined by $\delta_t(q_0,\, 0) = q_4^0$, $\delta_t(q_0,\, 1) = q_4^1$, $\delta_t(q_0,\, 2) = q_f$, $\ldots$, $\delta_t(q_1^1, 1) = q_t$, $\delta_t(q_1^1, 0) = q_f$, and $\delta_t(q_1^1, 2) = q_f$. For the sake of brevity, when we graphically represent an MDD (or a matrix diagram), the out-edges directed to the terminal vertex q_f, labels on these edges, and the vertex q_f are omitted. The simplified form of the MDD in Fig. 2(a) is portrayed in Fig. 2(b).

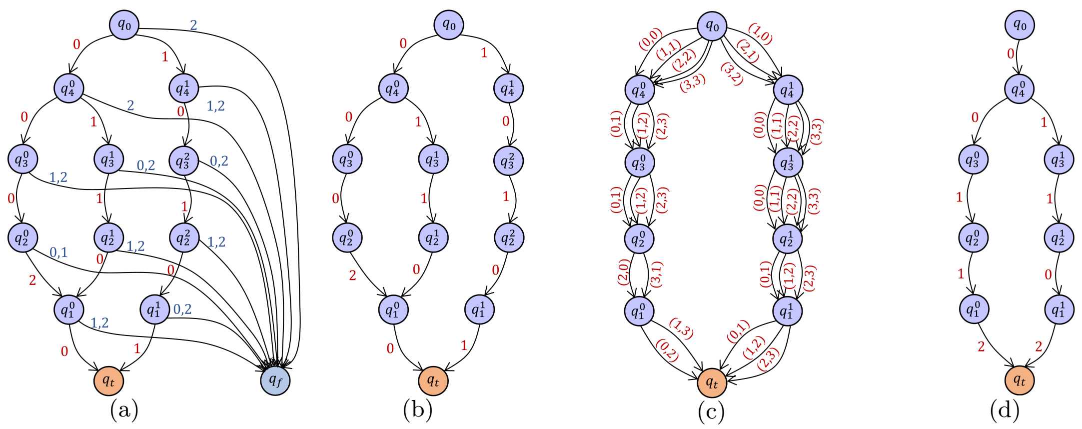

Figure 2: (a) An MDD, (b) the simplified representation of the MDD in Fig. 2(a), (c) a matrix diagram, and (d) the MDD generated from the MDD in Fig. 2(b) and the matrix diagram in Fig. 2(c) under the relational product operation.

Fig. 2(c) shows a matrix diagram $H = (Q, \mathcal{D}, q_0, q_t, q_f, \delta_t)$, where $Q = \{q_0,\, q_4^0,\, q_4^1,\, q_3^0,\, q_3^1,\, q_2^0,\, q_2^1,\, q_1^0,\, q_1^1,\, q_t,\, q_f\}$ (note that vertex q_f is omitted graphically) is a set of vertexes, $\mathcal{D} = D \times D$ ($D = \{0, 1, 2, 3\}$) is a set of label pairs, vertexes q_0, q_t, and q_f are the root vertex, terminal vertex valued with **True**, and terminal vertex valued with **False**, respectively. The function δ_t is defined by $\delta_t(q_0, (0, 0)) = q_4^0, \delta_t(q_0, (0, 1)) = q_f, \delta_t(q_0, (0, 2)) = q_f, \delta_t(q_0, (0, 3)) = q_f, \delta_t(q_0, (1, 0)) = q_4^1, \delta_t(q_0, (1, 1)) = q_4^0, \ldots, \delta_t(q_1^1, (0, 0)) = q_f, \delta_t(q_1^1, (0, 1)) = q_t, \ldots, \delta_t(q_1^1, (3, 2)) = q_f$, and $\delta_t(q_1^1, (3, 3)) = q_f$. □

Example 2 Consider the TLPN G in Fig. 1, where transitions t_1, t_3, and t_4 are observable and labeled with a, b, and a, respectively, while transitions t_2 and t_5 are unobservable. The timed function π is defined by $\pi(t_1) = 2, \pi(t_2) = 0, \pi(t_3) = 2, \pi(t_4) = 1$, and $\pi(t_5) = 1$. The reachable markings of G_n with the initial marking $M_0 = [2, 0, 0, 0]^T$ are listed in Table 1. We assume that there exists a set of states $\mathcal{M}_t = \{([2, 0, 0, 0]^T, 0), ([0, 1, 1, 0]^T, 0), ([0, 1, 0, 1]^T, 1)\}$. The state set can be represented by the MDD portrayed in Fig. 2(b). There are three top-bottom paths ending with q_t, i.e., $\xi_{t1} = q_0 0 q_4^0 0 q_3^0 0 q_2^0 2 q_1^0 0 q_t$, $\xi_{t2} = q_0 0 q_4^0 1 q_3^1 1 q_2^1 0 q_1^0 0 q_t$, and $\xi_{t3} = q_0 1 q_4^1 0 q_3^2 1 q_2^2 0 q_1^1 1 q_t$. The label sequences of ξ_{t1}, ξ_{t2}, and ξ_{t3} are $\varrho_1 = 00020$, $\varrho_2 = 01100$, and $\varrho_3 = 10101$ that represent states $([2, 0, 0, 0]^T, 0)$, $([0, 1, 1, 0]^T, 0)$ and $([0, 1, 0, 1]^T, 1)$, respectively. $\square$

Table 1: Reachable markings of the TLPN shown in Fig. 1 with the initial marking $M_0 = [2, 0, 0, 0]^T$.

$M_0 = [2, 0, 0, 0]^T$	$M_1 = [0, 1, 1, 0]^T$
$M_2 = [0, 0, 1, 1]^T$	$M_3 = [0, 1, 0, 1]^T$
$M_4 = [0, 0, 0, 2]^T$	$M_5 = [1, 0, 1, 0]^T$
$M_6 = [0, 0, 2, 0]^T$	$M_7 = [1, 1, 0, 0]^T$
$M_8 = [1, 0, 0, 1]^T$	-

Example 3 Let us consider the TLPN G in Fig. 1 and a set of transition-time pairs $\mathcal{T}_t = \{(t_1, 2), (t_4, 1)\}$. Given a set of states $\mathcal{M}_t = \{([2, 0, 0, 0]^T, 0), ([0, 1, 1, 0]^T, 0), ([0, 1, 0, 1]^T, 1)\}$, we have $\mathcal{M}'_t = \mathcal{N}(\mathcal{M}_t, \mathcal{T}_t) = \{([1,\ 1,\ 0,\ 0]^T,\ 2), ([0, 1, 1, 0]^T, 2)\}$. The set of states $\mathcal{M}_t$ is represented by the MDD $\hat{\mathcal{M}}_t$ shown in Fig. 2(b) (here $D = \{0, 1, 2, 3\}$, and the out-edges labeled with 3 are directed to q_f, i.e., $\delta_t(q_n, 3) = q_f$ $(q_n \in Q_n)$, which are omitted graphically), and the matrix diagram H decided by $\mathcal{T}_t$ is portrayed in Fig. 2(c). Then, under the relational product of $\hat{\mathcal{M}}_t$ and H, we have $\hat{\mathcal{M}}'_t = \hat{\mathcal{M}}_t \otimes H$. The MDD $\hat{\mathcal{M}}'_t$ that represents the set of states $\mathcal{M}'_t$ is illustrated in Fig. 2(d). $\square$

3 Time semantics

In this section, we detail the time semantics used in this paper. We adopt the single-server policy, which implies that a transition t_i at a state (M, τ) may only fire once at a time regardless of the current enabling degree, i.e., the largest integer number x such that $M \geq x \cdot Pre(\cdot, t_i)$. For the choice policy, the transitions enabled at a state have the same priority to fire. Namely, at a state, any enabled transition can fire and generate a new state. In our setting, we assume that transitions share a unique clock, and when a transition t becomes enabled, the clock begins to count time. The clock is reset to 0 only when it reaches to $\pi(t)$. During the process of the firing of a transition, another enabled transition cannot fire unless the clock is reset to 0. Therefore, for the memory policy, the memory for the firing of each transition is independent.