Abstract—In this paper, we continue to investigate timed alternating finite automata (TAFA), in particular we generalize the existing theory to the case of extended timed alternating finite automata (ETAFA). We define a framework extension of TAFA, study their power and properties. We develop an algebraic treatment of such ETAFA, along the lines of the algebraic treatment of systems of equations based on timed alternating finite automata. We present an equational language representation for ETAFA which parallels that of languages equations for TAFA. The power of these machines is discussed, as well as some of their fundamental properties. Moreover, we consider timed $\epsilon$-transitions and clock precisions, and discuss their interpretations in ETAFA.

I. INTRODUCTION

The power of timed alternating finite automata (TAFA) lies in its natural alternation between existential and universal transitions during the course of a computation. Their power lies not only exclusively in automata theory, but they become an efficient framework in many applications, ranging from proving properties of real time systems and specifying their behaviors to verifying and model checking software systems. Moreover, they provide a succinct representation for regular languages, but are double-exponentially more succinct than deterministic finite automata. Time alternating finite automata have been independently introduced in [8], [10] and thoroughly investigated in literature. Since their introduction, these models have been largely investigated under several theoretical and practical aspects. TAFA have been extended with clock variables, in almost the same way that time finite automata (TFA) [4].

Classical automata are traditionally untimed or synchronous models of computation in which only the ordering of events, not the time at which events occurs, would affect the result of a computation. Timed automata received their first seminal treatment in [4], since then they have become a powerful canonical model for describing time to model and verify embedded systems with real-time constraint computations. In timed automata the value of a clock depends on the path taken by the automaton are determined by transition relations. Clocks were also extended to alternating finite automata to justify timed transitions and sequences in these models. A comprehensive analysis of the theory of timed alternating finite automata (TAFA) based upon a hybrid combination of alternating finite automata and timed automata models were proposed in [8], [10], [11]. In addition, deterministic timed alternating finite automata (DTAFA) were introduced in [12] as the first determinizable subclass of alternating timed automata by restricting the use of clocks. Moreover, they have been shown to be more expressive and powerful than timed automata and TAFA. Unlike timed automata model, the key for the determinization of TAFA is the property that each computation step, all clock values are determined only by the input timed word. DTAFA are characterized by a fixed, predefined association between the clocks and the symbols of the input alphabet.

The aim of this paper is to propose a formalism which is sufficiently general for modeling all variants of timed alternating finite automata for which the Boolean operations can be effectively defined. In this paper, we introduce extended timed alternating finite automata (ETAFA), a general class of automata that includes all restricted versions of timed AFA. Moreover, we present a general equational language representation for ETAFA which parallels that of timed languages for TAFA. We also give some fundamental properties of ETAFA.

This paper is organized as follows. Section II is devoted to notations and preliminaries. Section III introduces extended timed alternating finite automata (ETAFA) – a general framework for several types of timed alternating finite automata. In addition in Section IV, we develop an algebraic treatment of such ETAFA, along the lines of the algebraic treatment of systems of equations based on timed alternating finite automata. The solutions for such equations over time languages parallel that of language equations for TAFA. Section V describes the transformation between an equational language representation and ETAFA. Closure properties and some results of ETAFA are stated in Sections VI, VII, and VIII. In Section IX, we consider the clock precision and discuss timed $\epsilon$-transitions. Finally, in Section X we draw some concluding remarks.

II. PRELIMINARIES

We denote by $\mathbb{R}$ the set of all non-negative reals including 0. The cardinality of a finite set $A$ is $|A|$. An alphabet $\Delta$ is a finite, nonempty set whose elements are called symbols or letters. A timed word, $w_t$ over $\Delta$ is a finite sequence $w_t = (a_1, t_1)(a_2, t_2)\cdots(a_i, t_i)$ where the $a_i$s are symbols of $\Delta$ and the $t_i$s are in $\mathbb{R}$ such that for all $i \geq 1$, $t_i < t_{i+1}$. The first element, $a'_1$s, of each pair are the input symbols, and the second element, $t'_i$s, are the time elapsed with respect to the $a'_i$s since the previous symbol reading. We assume that $t_1 = 0$. Thus, $t_1 \cdots t_i$ is a finite monotonically non-decreasing time sequence of $\mathbb{R}$. The time language $(\Delta \times \mathbb{R})^*$
is the set of all timed words over $\Delta$ where $\lambda$ denotes the empty timed word. Recall that classical words over $\Delta$ form the free monoid $(\Delta^*, \cdot, \lambda)$ generated by $\Delta$, where $\cdot$ is the classical concatenation operator (we write either $ab$ or $a \cdot b$ for the concatenation). For any timed language $L_t \subseteq (\Delta \times \mathbb{R})^*$, $\overline{L}_t = \Delta^\ast \backslash L_t$, is the complement of $L_t$ with respect to $\Delta^\ast$. For languages $L_{t1}$ and $L_{t2}$ over $\Delta$, the union and intersection are denoted by $L_{t1} \cup L_{t2}$ and $L_{t1} \cap L_{t2}$, respectively.

Given a finite set $X$ of clock variables, a clock constraint $\psi$ over $X$ on a given input symbol $a \in \Delta$ can be generated by the following grammar:

$$\psi ::= x \leq c \mid x < c \mid e \leq x \mid e < x \mid \psi_1 \lor \psi_2 \mid \psi_1 \land \psi_2$$

where $x$ is any clock in $X$ and $c \in \mathbb{R}$ such that $c \geq 0$. The operators $\lor$ and $\land$ stands for the logical-or and logical-and, respectively. A clock interpretation (valuation) $\nu$ for $X$ is a mapping from $X$ to $\mathbb{R}$. That is, $\nu$ assigns to each clock $x \in X$ the value $\nu(x)$. A clock interpretation represents the values of all clocks in $X$ at a given snapshot in time. The length of a timed word $w_t$, denoted by $|w_t|$, is the total number of symbols in $w_t$.

### III. Extended Timed Alternating Finite Automata

Let $\mathbb{B}$ denote the two-element Boolean algebra $\mathbb{B} = \{0, 1\}$, $\lor$, $\land$, $\neg$ denote the or, and not, respectively. Le $X$ is a vector with $|X|$ elements referring to all the Boolean functions from $Q$ to $\mathbb{B}$ where $\lor$, $\land$, $\neg$ (interchangeably $\neg$) denote the “or”, “and” and “not”, respectively. Let $X = \mathbb{R}$ be a vector over $\mathbb{R}$ with $|X|$ elements referring to all real functions from $X$ to $\mathbb{R}$. For notation convenience, let $X$ denote a vector over $\mathbb{R}^+$.

**Definition 3.1:** An extended timed alternating finite automaton (ETFA) is a sept-tuple $A = (Q, \Delta, s, X, g, h, F)$, where (a) $Q$ is a finite set, the set of states, (b) $\Delta$ is the alphabet, the input alphabet, (c) $s \in Q$ is the starting state, (d) $X$ is a finite set, the set of clocks, (e) $h$ is the time transition function, $h : (\mathbb{B}^Q)^{X} \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$; (f) $g$ is the letter transition function from $(Q \times \mathbb{R}^X)$ into the set of all functions $(\mathbb{B}^Q)^{\mathbb{R}^X}$ into $\mathbb{B}$. (g) $F \subseteq Q$ is the set of final states.

The function $h$ is specifically defined as:

$$h((x_1, x_2, \ldots, x_n), t) = ((x_1 + t, x_2 + t, \ldots, x_n + t))$$

where $x_i \in X$ for $1 \leq i \leq |X|$ and $t \in \mathbb{R}$ such that $t \geq 0$.

For convenience, the definition of the function $h$ can be rewritten as $h(x, t) = x'$, where for all $x \in X$, $x' = x + t$ where $t$ is the time associated with the word that has been read.

For each state $q \in Q$ and with each vector $X \in \mathbb{R}^X$, $g(q, X)$ is a function from $(\mathbb{B}^Q)^{\mathbb{R}^X} \times (\Delta \times \mathbb{R}) \rightarrow \mathbb{B}$, which we denote as $g_q(X)$. For each $q \in Q$ with each $X \in \mathbb{R}^X$ and for each $a \in (\Delta \times \mathbb{R})$, we define $g_q(X)(a)$ to be the Boolean function $(\mathbb{B}^Q)^{\mathbb{R}^X} \rightarrow \mathbb{B}$ such that:

$$g_q(X)(a)(u) = g_q(X)(a, u)$$

where $u \in (\mathbb{B}^Q)^{\mathbb{R}^X}$. Thus, for any $u \in (\mathbb{B}^Q)^{\mathbb{R}^X}$, the value of $g_q(X)(a)(u)$, also $g_q(X)(a, u)$ is either 1 or 0.

We define the function $g_{Q \times X^2} : (\mathbb{B}^Q)^{\mathbb{R}^X} \times (\Delta \times \mathbb{R}) \rightarrow \mathbb{B}$ by putting together the $|Q \times \mathbb{R}^X|$ functions $g_q(X) : (\mathbb{B}^Q)^{\mathbb{R}^X} \times (\Delta \times \mathbb{R}) \rightarrow \mathbb{B}$ for each $q \in Q$ and with each $X \in \mathbb{R}^X$ as follows:

For $a \in (\Delta \times \mathbb{R})$ and $u, v \in (\mathbb{B}^Q)^{\mathbb{R}^X}$, $g_{Q \times X^2}(u, a) = v$ if and only if $g_q(X)(u, a) = v_{(q, x)}$, for each $q \in Q$ and with each $X \in \mathbb{R}^X$, where $v_{(q, x)}$ indexes the $q$ element of the $X$ vector in $(\mathbb{B}^Q)^{\mathbb{R}^X}$. For convenience, we write $g(u, q)$ instead of $g_{Q \times X^2}$ when there is no confusion.

Now, we extend $g$ to a function of $Q \times \mathbb{R}^X$ into the set of all functions $(\mathbb{B}^Q)^{\mathbb{R}^X} \times (\Delta \times \mathbb{R})^* \rightarrow \mathbb{B}$ as follows:

$$g_q(X)(u, w_t) = \begin{cases} u(q, x) & \text{if } w_t = \epsilon \\ g_q(X)((u, w_t'), a) & \text{if } w_t = aw_t' \end{cases}$$

where $a \in (\Delta \times \mathbb{R}), w_t, w_t' \in (\Delta \times \mathbb{R})^*$ and $u \in (\mathbb{B}^Q)^{\mathbb{R}^X}$. Note that $\epsilon \in (\Delta \times \mathbb{R})$ is the timed null word where $\epsilon = (0, \lambda)$ such that $w_t \cdot \epsilon = w_t \epsilon = w_t$.

For all $q \in Q$ and with each $X \in \mathbb{R}^X$, we define the characteristic vector $f \in (\mathbb{B}^Q)^{\mathbb{R}^X}$ of $F$ as follows:

$$f(q, X) = 1 \iff q \in F$$

$$f(q, X) = 0 \iff q \in Q \setminus F$$

**Definition 3.2:** A word $w_t \in (\Delta \times \mathbb{R})^*$ is accepted by a ETFA $M$ if and only if $g_h(h(0, t))(f, w_t) = 1$, where $s$ is the starting state, $f$ is the characteristic vector of $F$, $t$ is the time associated with the word that has been read, and $0 \in \mathbb{R}^X$ is the zero-valued vector (i.e., $0 = 0$ for all $x \in X$).

The language accepted by a ETFA $A = (Q, \Delta, s, X, g, h, F)$ is defined as follows:

$$L_t(A) = \{w_t \in (\Delta \times \mathbb{R})^* \mid g_s(h(0, t))(f, w_t) = 1\}$$

Note that $w_t = (a, t)w_t'$ for some $a \in \Delta, t \in \mathbb{R}$, and $w_t' \in (\Delta \times \mathbb{R})^*$.

**Example 3.1:** Consider the following ETFA $A = (Q, \Delta, q_0, X, g, h, F)$ where $Q = \{q_0, q_1, q_2\}$, $\Delta = \{a\}$, $h$ is given as in the ETFA’s definition, $X = \{x\}$, and $F_2 = \{q_2\}$. Thus, the characteristic vector $f$ and the function $g$ are as follows:

$$f = (0, 0, 1), (0, 0, 1), \ldots, (0, 0, 1) = (0, 0, 1)^{\mathbb{R}^X} = (0, 0, 1)$$

<table>
<thead>
<tr>
<th>$a$</th>
<th>$g_a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0 \lor (q_1[x := 0])$</td>
<td></td>
</tr>
<tr>
<td>$q_1[(x \neq 1) \land q_1] \lor ((x = 1) \land q_2)$</td>
<td></td>
</tr>
<tr>
<td>$q_2</td>
<td>q_2$</td>
</tr>
</tbody>
</table>

Analyzing the definition of the function $g$ in the ETFA $A$, we can see how to trace one of the $|Q \times \mathbb{R}^X|$ functions for every $g_q(X)$ transition that is made while tracing the word. In addition the notation $[x := 0]$ denotes a “reset” of the clock $x$ and $(x = 1)$ indicates a comparison.
For example, tracing the timed word
\[ w_t = < a, 1/2 > < a, 0 > < a, 1 > \] assuming \( \epsilon = < \lambda, 0 > \) is:
\[
\begin{align*}
g_{g_0}/(h(0,1/2)) &= (0,0,1), < a, 1/2 > < a, 0 > < a, 1 > \\
g_{g_0}/(h(1/2)) &= (0,0,1), < a, 1/2 > < a, 0 > < a, 1 > \\
g_{g_0}/(h(1/2,0)) &= (0,0,1), < a, 0 > < a, 1 > \\
g_{g_0}/(h(0,0)) &= (0,0,1), < a, 0 > < a, 1 > \\
g_{g_0}/(h(1,0)) &= (0,0,1), < a, 0 > < a, 1 > \\
g_{g_0}/(h(1)) &= (0,0,1), < a, 0 > < a, 1 > \\
g_{g_0}/(h(1/2)) &= (0,0,1), < a, 0 > < a, 1 > \\
g_{g_0}/(h(1/2,0)) &= (0,0,1), < a, 0 > < a, 1 > \\
g_{g_0}/(h(0,0)) &= (0,0,1), < a, 0 > < a, 1 > \\
g_{g_0}/(h(1,0)) &= (0,0,1), < a, 0 > < a, 1 > \\
g_{g_0}/(h(1)) &= (0,0,1), < a, 0 > < a, 1 > \\
&= 0 \lor 0 \lor 1 \lor 1 \\
&= 1.
\end{align*}
\]

Therefore the timed word \( w_t = < a, 1/2 > < a, 0 > < a, 1 > \) is accepted.

Because ETAFA only considers passage of time between two input symbols (with the \( h \) function) by adding the elapsed time to the current clock values, a change in either the ETAFA, or the format of an input word is required. This can be accomplished in one of the following two methods: (1) Adding a universal clock to the ETAFA. This is a clock that never resets and contains the value of the total elapsed time since the start of a word. (2) Choosing the representation of a timed word so that each time value of the \( (a, t) \in (\Delta \times \mathbb{R}) \) would be the time since the last symbol was read as opposed to the time since the start of the word. We have chosen the second method in the previous tracing example. Either method will work fine as there would be no change to the current ETAFA definition.

IV. EQUATIONAL LANGUAGE REPRESENTATION OF ETAFA

The theory of classical language equations received its formal treatment in the seminal paper of [6]. Most of the subsequent research focused on restricted various types of language equations and their different solution techniques. (See for example, [10], [12], [13]). However, language equation solutions still suffer from a lack of generality. In this section, we define a general framework class of language equations and relate them to extended alternating timed finite automata.

Let \( A = (Q, \Delta, s, X, g, h, F) \) be a ETAFA. For all \( q \in Q \), \( A \) can be represented by the following system of equations.

\[
\hat{X}_{(q,x)} = \left\{ \sum_{a \in \Delta} a \cdot g_q(\lambda) \right\}(\hat{X}^q, a) + \hat{\epsilon}_{(q,x)}
\]

for all \( X \in \mathbb{R}^x \) and where \( "\cdot" \) is the concatenation operation. \( \hat{X} \) is a vector of \( |Q \times \mathbb{R}^x| \) variables \( \hat{X}_{(q,x)} \) indexed in/cm.\f in/share/texmf- texlive/fonts/type1/public/amsfonts/cm/cmr7.pfb¿¡ by the states \( q \in Q \) and vectors \( X \in \mathbb{R}^x \) and

\[
\hat{\epsilon}_{(q,x)} = \begin{cases} 
\lambda & \text{if } q \in F \\
0 & \text{otherwise}
\end{cases}
\]

for each \( q \in Q \) with each \( X \in \mathbb{R}^x \).

Note all terms of the form \( a, x \), \( a \in \Delta \) can be omitted as can the term \( \hat{\epsilon}_{(q,x)} \) if \( q \in Q \backslash F \).

The following theorems are the most important results relating timed alternating finite automata and their equational representations. The second theorem gives a sufficient condition for the uniqueness of the solution of the system of equations (1).

\begin{center}
\textbf{Theorem 4.1:} [11] Let \( A = (Q, \Delta, s, X, g, h, F) \) be a ETAFA represented by the system of language equations \( \hat{X}_{(q,x)} \) of the form (1) whose solution is \( \{\hat{X}_q\}_q \in Q \). Let \( s \) be the starting state of \( A \). Then \( L_1(A) = \hat{X}_s \).
\end{center}

\begin{center}
\textbf{Theorem 4.2:} [11], [12] Any system of language equations of the form of (1) has a unique solution for each \( \{\hat{X}_q\}_q \in Q \). Furthermore, the solution for each \( \{\hat{X}_q\} \) is regular.
\end{center}

The following system of equations represents the ETAFA \( A \) given in Example 3.1.

\[
\hat{X}_{g_0} = a \cdot (\hat{X}_{g_0} \lor (\hat{X}_{q_1}[x := 0])) \\
\hat{X}_{q_1} = a \cdot (x \neq 1 \land \hat{X}_{q_1}) \lor ((x = 1) \land \hat{X}_{q_2}) \\
\hat{X}_{q_2} = a \cdot \hat{X}_{q_2} + \lambda
\]

Note that \( \hat{X}_{(g_0,x)} \) for all \( X \in \mathbb{R}^x \) can be written as \( \hat{X}_{g_0} \) for convenience.

\begin{center}
\textbf{Lemma 4.1:} Let \( \hat{X}_{(q,x)} \) be an equational representation of a given ETAFA \( A = (Q, \Delta, s, X, g, h, F) \), then \( \hat{X}_{(q,x)} \) and \( A \) are equivalent in terms of language acceptance.
\end{center}

\begin{center}
\textbf{Proposition 1:} The family of languages accepted by \( \text{ETAFA} \) is the class of timed regular languages.
\end{center}

V. FROM THE EQUATIONAL REPRESENTATION TO ETAFA

Given the equational representation of ETAFA, we should be able to construct the corresponding ETAFA. For example, given the following system of equations:

\[
\hat{X}_{g_0} = a \cdot \hat{X}_{g_0} + b \cdot (\hat{X}_{q_1} \land -\hat{X}_{q_2}[y := 0]) \\
\hat{X}_{q_1} = a \cdot (x \leq 2) \land \hat{X}_{q_1} + b \cdot \hat{X}_{q_2} + \lambda \\
\hat{X}_{q_2} = a + b \cdot (y = 1/2)
\]

We will assume that \( g_0 \) is the starting state, but this may not always be the case. We also assume that \( h \) is defined the usual way unless explicitly told so. X Thus, the ETAFA \( A = (Q, \Delta, g_0, g, h, X, F) \) where \( Q = \{g_0, q_1, q_2\} \), \( \Delta = \{a, b\} \), \( X = \{x, y\} \), \( F = \{q_1\} \), and \( g \) is given as:
is a time transition function of the form $(q, \phi, \rho, a, q')$ where $q, q' \in Q'$, $a \in \Delta' \cup \{\epsilon\}$, $\rho \in X'$, $\phi$ is the transition guard, it is a boolean combination of the form $X \in I$ for some clock $X$ and some bound interval $I$. $q$ is the current state, $q'$ is the next state, $a$ is the letter read, $\rho$ is the set of clocks to be reset.

**Corollary 7.1:** Let $A' = (Q', \Delta', s', X', \delta', F')$ be a TFA, we can construct a ETAFA $A = (Q, \Delta, s, X, g, h, F)$ where $Q = Q'$, $\Delta = \Delta'$, $s = s'$, $h$ is defined the usual way, $X = X'$, $F = F'$, and $g$ is given by the boolean language representation $X$ as follows: For each $(q, \phi, \rho, a, q') \in \delta'$ where $a \neq \epsilon$,

$$
\tilde{X}_q := \tilde{X}_q \lor (\emptyset \land (\tilde{X}_q[p := 0]))
$$

for all $p \in \mathcal{P}(X)$.

The proof is straightforward. Simply perform an “or” operator with the current boolean expression terms and $(\emptyset \land (\tilde{X}_q[p := 0]))$.

**Example 7.1:** Consider $A' = (Q', \Delta', s', X', \delta', F')$ be a TFA where $Q' = \{q_0, q_1, q_2\}$, $\Delta' = \{a, b\}$, $s' = \{q_0\}$, $X' = \{x, y\}$, $\delta' = \{\{q_0, (x, y)\}, \{a, q_1\}, \{q_1, (x, y)\}, \{0, b, q_1\}, \{q_1, x \in [2, 2] \lor y \in \mathbb{R}, 0, b, q_2\}, \{q_2, (x, y)\}, \{x, q_2\}, \{q_2, (x, y)\}, \{0, b, q_0\}\}$

The ETAFA equivalent is $A = (\Delta, s, X, g, h, F)$

The TFA can be transformed to a ETAFA. Formally, we have the following theorem:

**Theorem 8.1:** For every $\epsilon$-transitions $TFA$ $M'$ there exists a ETAFA $M$ such that $L_\epsilon(M') = L_\epsilon(M)$.

**Proof:** Due to space constraint we present a constructive proof. Consider each transition $(q, x \in \mathbb{R}, X', \epsilon, q')$ for every transition $(q', x \in I, X^\rho, a, q)$ for some interval $I$, some $a \in \Sigma$, some $q' \in Q$, and $X^\rho \subseteq X$, add the edge $(q, \phi, X^\rho \cup X', a, q')$. After all transitions have been added, delete the edge $(q, x \in \mathbb{R}, |x|, \epsilon, q')$. The TFA can be transformed to a ETAFA. Formally, we have the following theorem:
then we notice that we cannot reduce this yet, but the ETAFA equation \( \hat{X}_q \) for the starting state \( q \) is as follows:

\[
\hat{X}_{q(X)} = \left\{ \sum_{a \in \Delta} \alpha \cdot g_{q(X)}(\hat{X}, a) \lor g_{q(X)}(\hat{X}, a) \right\} q \in Q
\]

\[
\hat{X}_{q(X)} = \left\{ \sum_{a \in \Delta} \alpha \cdot g_{q(X)}(\hat{X}, a) \lor g_{q(X)}(\hat{X}, a) \right\} q \in Q
\]

\(
X \]

IX. DISCRETE CLOCK AND CLOCK PRECISION

In the previous section we discuss timed \( \epsilon \)-transitions, in this section we consider another clock metric, clock precision. Let \( X \) be the finite set of clocks and \( x \in X \). We may ask the following question what is the smallest possible values of \( x \in X \) that satisfies \( x > 2 \)? If \( x \in \mathbb{R} \), then we can only say that \( x \) approaches \( 2 \) from the positive side, but this is a poorly defined way of describing the clock value. Instead, we will define the precision as some positive number in \( \mathbb{R} \) such that a clock value may only be multiples of this value., i.e., if the precision is \( 10^{-3} \), then the smallest value of \( x \) for \( x > 2 \) is \( 2.001 \).

To simulate an \( \epsilon \)-transition, we must redefine \( g_q(X)(a, u) \) for \( a \in \Delta \) as defined in Section III to \( g_q(X)(a, u) \) for \( a \in \Delta \cup \{ \lambda \} \) and its extension over \((\Delta \times \mathbb{R})^* \) becomes \((\Delta \cup \{ \lambda \} \times \mathbb{R})^* \) as some positive number in \( \mathbb{R} \) such that a clock value may only be multiples of this value., i.e., if the precision is \( 10^{-3} \), then the smallest value of \( x \) for \( x > 2 \) is \( 2.001 \).

However, since we are now considering \( \lambda \), a readable letter, we don’t want to collapse the \( g_q \) function to \( u_{q,X} \) because we might not be at the end of the word. Therefore, we define \( \tau = (X, 0) \) to denote the word terminator that is read at the end of each word and collapse \( g \) to \( u_{q,X} \) if \( a = \tau \) is read. Now, the equation becomes how to interpret these new \( \epsilon \)-transition in ETAFA. This is best illustrated by the following example:

\[
\hat{X}_{q(X)} = \alpha \cdot (X_{q(X)}[y := 0]) + \lambda \cdot ((X_{q(X)}[y := 0]) \land (y > 2))
\]

Note that \( X \) indicates that \( q \) is a final state. Now if we consider the expression \( g_q(h((0, 1), 3))(a, a) \) for clock \( X = (x, y) = (0, 1) \), we notice that the time expression function \( h((0, 1), 3) \) will result in \( y > 2 \) being satisfied. In this case we chose the minimum value in the range \([0, 3] = \mu \) that will satisfy \( 1 + \mu > 2 \) since \( \mu \in \mathbb{R} \), then there is no adequate way to properly write what \( \mu \) is. However, if we consider the precision \((i.e., 10^{-3})\), then \( 1.001 \) will work, therefore \( 1 + [1, 001] > 2 \) is the least value that we can choose for this example.

X. CONCLUSION

We investigated extension of timed alternating finite automata with \( \epsilon \)-transitions and presented is presented a general framework for this class of automata. We further extended the equational representation of TAFA to represent ETAFA and explore solutions and properties for such equations over time languages. Timed \( \epsilon \)-transitions and clock precisions need further investigation. There are several future directions worth mentioning, including the relationship between extended timed regular expressions and extended timed alternating finite automata which are being investigated.