Fuzzy Branching Temporal Logic

Seong-ick Moon, Kwang H. Lee, and Doheon Lee

***Abstract*—Intelligent systems require a systematic way to represent and handle temporal information containing uncertainty. In particular, a logical framework is needed that can represent uncertain temporal information and its relationships with logical formulae. Fuzzy linear temporal logic (FLTL), a generalization of propositional linear temporal logic (PLTL) with fuzzy temporal events and fuzzy temporal states defined on a linear time model, was previously proposed for this purpose. However, many systems are best represented by branching time models in which each state can have more than one possible future path. In this paper, fuzzy branching temporal logic (FBTL) is proposed to address this problem. FBTL adopts and generalizes concurrent tree logic (CTL*), which is a classical branching temporal logic. The temporal model of FBTL is capable of representing fuzzy temporal events and fuzzy temporal states, and the order relation among them is represented as a directed graph. The utility of FBTL is demonstrated using a fuzzy job shop scheduling problem as an example.**

***Index Terms*—Branching temporal logic, CTL*, fuzzy logic, temporal logic.**

I. Introduction

INTELLIGENT systems should be able to represent and handle temporal information. They must be able to represent the times at which activities take place and the causal relationships among those activities. However, like other kinds of information, temporal information can have a degree of uncertainty, creating the need for a framework that can represent uncertain time information and the relationships among that information. Furthermore, the logic underlying those relationships should be represented with possible inference mechanisms. If this need is to be satisfied, we must be able to represent temporal information primitives and their logical binding in a well-defined manner.

The time model for a system can be a linear model or a branching model. In linear models, the order of the events and states is already known, but the event times are unknown. In branching models, the order of the events and states, as well as the event times, are only partially known. The branched nature of these models means that a given current state may have several possible futures. Moreover, logically identical events may occur in several different system states, and there are an infinite number of possible execution paths. Previously, we proposed a formalism referred to as fuzzy linear temporal logic (FLTL) for modeling systems with linear time [1]. This logic subsumes propositional linear temporal logic (PLTL) [2], and is capable of representing fuzzy events and states. However, FLTL is only suitable for modeling systems with linear time and cannot represent concurrent and/or parallel systems, which require a branching time model [3].

Here we propose a temporal model and logic for representing systems with branching time and uncertain temporal information. In the proposed formalism, the uncertainty in the logical information is handled using fuzzy logic, and the uncertainty in the temporal information is represented with fuzzy temporal primitives. For temporal information without uncertainty, a time instant is the natural choice for the temporal primitive. However, three distinct types of temporal primitive have been previously proposed for representing uncertain temporal information: a time interval [4], an interval with possibility measures [5], and a fuzzy representation of the time instant [6]. In the present work we chose a fuzzy representation of the time instant as the temporal primitive, as we did previously in FLTL [1]. This primitive is used to define fuzzy events and fuzzy states, which are then used to extend and generalize the temporal model of concurrent tree logic* (CTL*). The resulting logic, referred to as fuzzy branching temporal logic (FBTL), can describe a branching temporal model with uncertain temporal and logical information.

This paper is organized as follows. In Section II, we give a summary of related studies. Section III presents a description of FLTL to assist in the understanding of FBTL, and it is demonstrated that FLTL subsumes PLTL. FBTL is then described in Section IV; it is shown that FBTL subsumes CTL*. Finally, in Section V, FBTL is applied to a fuzzy job shop scheduling problem as an example of the proposed method.

II. Related Works

To manage uncertain temporal information, we must be able both to represent temporal information primitives in a suitable manner and to represent the relationships among them. Various definitions of the temporal primitive have been employed previously for systems with uncertain temporal information. For example, Allen used an interval representation [4], Dutta added fuzzy membership values to interval calculus [5], and Dubois used generic fuzzy numbers [6]. Among these approaches, the interval representation of Allen is the simplest method, and the interval algebra proposed by Dutta also represents the relationships among intervals. However, the interval representation is hampered by difficulties in the determination of the membership of the border time point of two adjoining intervals, a shortcoming known as the Divided Instant Problem [7]. Although Dutta incorporated fuzzy membership to improve uncertainty management, his method still suffers from the divided instant problem.

The fuzzy instant representation of Dubois uses fuzzy numbers and fuzzy intervals to represent temporal information.

Manuscript received April 16, 2002; revised April 1, 2003. This work was supported by KOSEF through AITrc. This work was recommended by Associate Editor A. Gomez-Skarmeta.

The authors are with the Department of Biosystems, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: khlee@bioif.kaist.ac.kr).

Digital Object Identifier 10.1109/TSMCB.2003.819485

Since fuzzy arithmetic can be used with fuzzy numbers, temporal primitives represented as fuzzy numbers can be manipulated with considerable flexibility and power. Fuzzy numbers have been used as the temporal primitive in several previous studies involving uncertain temporal information has been used in several studies [8]–[10]. We adopted this representation in the present work and used it to represent events and related states of the fuzzy temporal model.

The relationships among temporal entities have been represented in numerous ways. When the relationships among temporal entities are only partially known, it is necessary to find one or more feasible relations between every pair of entities. Algorithms that model the relationships among temporal entities can be applied in various contexts, including natural language processing, planning and knowledge based systems. Time Map [11] is a database of uncertain temporal information developed by Dean and McDermott. They used intervals to represent uncertain temporal information and provided an algorithm to manage and resolve conflicts among the information. Time Map is a network with nodes representing uncertain time points, which can be a certain time point or an interval. The relationship between two time points is represented by the time distance, which is also uncertain. Dean and McDermott used a LISP-like language to describe the reasoning system. This system, which is an extension of the classical predicate calculus system, handles temporal reasoning for Time Map, such as the resolution of conflicts with new temporal information and answering queries for temporal information. The temporal constraint satisfaction problem (TCSP) was explored by Dechter *et al.* [12]. In this problem, relationships among time intervals are represented by a temporal constraint network (TCN). The goal of TCSP is the sequence of the intervals which most satisfies the relationships represented by TCN. Decther *et al.* showed that a network with at most one interval for each edge can be solved in polynomial time, while the more general case is NP-hard. Beek proposed efficient solutions for resolving relationships among time points and intervals [13]. He improved on previously developed algorithms through the use of point algebra and a subset of the interval algebra.

Vila and Godo proposed the fuzzy temporal constraint network (FTCN) [14]. FTCN is able to express precise or imprecise temporal information with fuzzy temporal entities. FTCN also provides a reasoning mechanism that is capable of checking its consistency and answering temporal queries. Barro *et al.* proposed the fuzzy temporal constraint satisfaction network (FTCSN), which is able to resolve relationships among fuzzy temporal entities [8]. The entities used by Barro *et al.* are similar to those proposed by Dubois, and FTCSN is able to resolve incomplete knowledge about the relationships among fuzzy temporal entities. This work was extended again by Viedma *et al.* to give fuzzy temporal constraint logic, a formalism that can represent and resolve fuzzy temporal constraints [15]. Chen proposed the fuzzy timed causal-like net (FTCLN), which can describe causal relationships among fuzzy temporal entities [9]. He extended interval algebra to uncertain temporal entities and provided an algorithm to evaluate each FTCLN node.

Since the introduction of temporal logic in 1977 by Pnueli [16], it has come to be widely used to specify the dynamic behavior of computer programs and to describe their properties. In the area of Artificial Intelligence, three kinds of temporal logic frameworks have been used [7]. The simplest one is temporal arguments (TA) [17]. This is a first-order logic with temporal information parameters in functions and predicates. However, because TA does not give any special status to temporal information, it is hard to recognize the well-formedness of temporal formulae within this logic framework. In addition, the expressive power of TA is relatively weak [7]. The second type of temporal logic framework is called reified logic [18]. This logic encapsulates nontemporal formulae with temporal notations and quantifications. Reified logic can be efficiently applied to theorem provers for first-order logic. Additionally it can express temporal knowledge such as "effects cannot precede their causes," which cannot be easily represented using TA. The third type of temporal logic framework is modal temporal logic [2]. Classic modal logic has long been used in the description of computer programs and their properties. Although this logic has not been widely used in intelligent systems on account of its complexity, it has rich expressive power, and its temporal model can explicitly express the temporal states of a system. Examples of formalisms based on modal temporal logic include propositional linear temporal logic (PLTL) and computational tree logic (CTL*) [2]. These logics can represent sequences of states and their associated propositions.

An additional factor that must be considered is that, even though a software system may be inherently crisp and discrete, modeling a real-time system can introduce uncertainty arising from a lack of sufficient knowledge of the system [19]. This problem can be particularly acute during the early development stages. Barringer suggested that modeling a program abstractly can only be done on a continuous time domain [20]. Several studies have incorporated fuzzy logic into modal temporal logic. Kim proposed a temporal logic on a linear time domain consisting of a set of adjacent intervals, each with a fuzzy membership value [21], [22]. However, this logic lacked common temporal operators such as "until" and "next." Thiele proposed a modal temporal logic on the integer time domain [23]. It provided the usual temporal operators, including "always," "eventually," and "until." However, its temporal model was defined on a discrete time domain, hence temporal uncertainty was not easy to represent. A number of other studies have incorporated a representation of uncertain temporal information into a logical framework. For example, van Le proposed model-theoretic semantics for fuzzy event simulation [24], Bae proposed fuzzy duration logic and used it to analyze a design of the software system for a nuclear power plant [19], and Dubois reified possibilistic logic (a type of multivalued logic distinct from fuzzy logic) [25].

III. Fuzzy Linear Temporal Logic (FLTL)

Before introducing FBTL, we first consider its linear version, FLTL. The version of FLTL described below is the revised formalism introduced in our previous paper [1]. FLTL is a fuzzy modal temporal logic defined on a single, 1-D time domain. In contrast, FBTL can include many "paths" of such time domains.

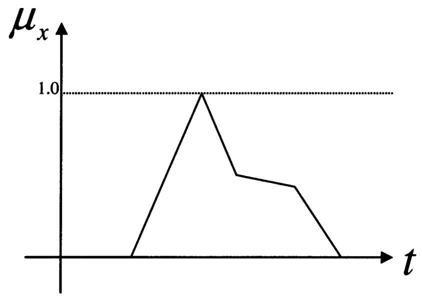

Fig. 1. Typical fuzzy set x, whose membership function $\mu_x(t)$ is defined on T.

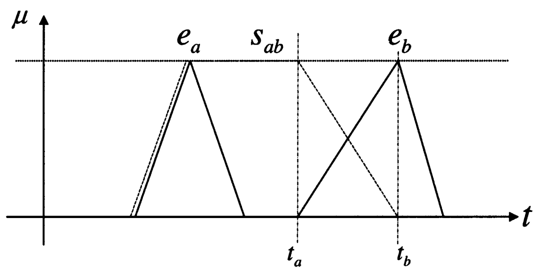

Fig. 2. Membership functions of two adjacent fuzzy temporal events and a fuzzy temporal state.

A. Definition of FLTL

The temporal model of FLTL resembles that of PLTL inasmuch as it is an absolute and linear time model; however, PLTL is based on a discrete temporal model whereas the temporal model of FLTL is continuous. On this continuous time domain T, *fuzzy temporal events* are defined as fuzzy numbers. A fuzzy number is a fuzzy set, which is defined by its membership function. For example, Fig. 1 shows a fuzzy number x defined on the time domain T. In this plot, the membership function $\mu_x(t)$ determines the *possibility distribution* of x. A fuzzy number is a fuzzy set whose membership function is convex and normalized, i.e., $\max \mu_x(t) = 1$. We denote a continuous fuzzy set x defined on T as $x = \int \mu_x(t)/t, t \in T$. This means that a continuous membership function of x is defined on the domain of $t \in T$. In the case of a discrete membership function, it is denoted as $x = \sum \mu_x(t)/t$.

In the case of a fuzzy temporal event e, its membership function $\mu_e(t)$ gives the possibility of the event occurring at time t. Each *fuzzy temporal state* is defined as a half-open interval between two neighboring events. In Fig. 2, a fuzzy state s_{ab} is defined as the half-open interval $[e_a, e_b)$ between the fuzzy events e_a and e_b. Since it is a half-open interval, $\mu_{s_{ab}}(t)$ reaches its peak with $\mu_{e_a}(t)$ and after a while reaches 0 just as $\mu_{e_b}(t)$ reaches 1. In other words, for $t_a \leq t \leq t_b$, $\mu_{s_{ab}}(t) = 1 - \mu_{e_b}(t)$. Now let us define the FLTL model, which consists of fuzzy events and states as outlined below.

Definition 1: The model of FLTL is a tuple $M = (T, E, S, P)$, where

- T is the time domain, $\mathcal{R}^+$;
- E is the set of fuzzy events, $E = \{e | e = \int \mu_e(t)/t, t \in T\}$, where e is convex and normalized;
- S is the set of fuzzy temporal states, each of which is a half-open interval between two fuzzy events $S = \{s | s = [e_i, e_f), e_i, e_f \in E\}$;
- P is the set of fuzzy propositions, with truth values for each state. $P = \{p | p = \sum \mu_p(s)/s, s \in S\}$.

■

An FLTL formula consists of atomic propositions in P, logical operators, and temporal operators. The logical operators include $\vee$ (logical or), $\wedge$ (logical and), and $\neg$ (logical negation), and the temporal operators include $\mathbf{U}$ (until) and $\mathbf{X}$ (delay). Now let us define the syntax of FLTL.

Definition 2: With respect to an FLTL model M, a well-formed FLTL formula is defined inductively as follows:

- $p \in P$ is a well-formed formula (wff);
- if p and q are wffs, then $\neg p$ and $p \wedge q$ are also wffs;
- if p and q are wffs, then $p\mathbf{U}q$ is also a wff;
- if p is a wff, then $\mathbf{X}_{Rd}p$ is also a wff, where $R \in \{\leq, \geq, =, <, >\}$, and d is a normalized fuzzy duration, $d \in R^+$.

■

The temporal operator $\mathbf{X}$ takes the truth value of the following formula after a time duration d. The duration $d \in \mathcal{R}^+$ can be a *fuzzy duration* or a crisp duration. The expression $\mathbf{X}_{=d}$ means "after exactly d;" $\mathbf{X}_{<d}$ represents "before d has passed;" and $\mathbf{X}_{>d}$ is "after d has passed." The temporal operator $\mathbf{U}$ is the common "until" operator. Hence $p\mathbf{U}q$ means "p is true until q is true." The shorthand notations customarily used in PLTL can also be used in FLTL. For example

$$p \vee q = \neg(\neg p \wedge \neg q), \tag{1}$$

$$\mathbf{F}p = \text{true}\mathbf{U}p \tag{2}$$

where $\mathbf{F}$ means "sometimes" or "eventually." From this operator, another operator $\mathbf{G}$ can be defined as

$$\mathbf{G}p = \neg\mathbf{F}\neg p \tag{3}$$

which means "always." We assume the priority among the operators to be the same as that given in [2]. Highest binding power is given to the temporal operators $\mathbf{F}$ and $\mathbf{G}$ followed by $\mathbf{X}$ and $\mathbf{U}$. The logical operator $\neg$ is next, followed by $\wedge$ and then $\vee$.

B. Interpretation of FLTL

Now let us define the truth value of a given FLTL formula. Since the interpretation of an FLTL formula varies over time, we denote the truth value of a formula p at time t according to the model M as $\nu_M p(t)$.

Definition 3: For a model M, the truth value of an FLTL formula p at time t, $\nu_M p(t)$, is defined inductively as

$$\text{if } p \in P, \nu_M p(t) = \sup_{s \in S} \min(\mu_p(s), \mu_s(t)); \tag{4}$$

$$\nu_M \neg p(t) = 1 - \nu_M p(t); \tag{5}$$

$$\nu_M p \wedge q(t) = \min(\nu_M p(t), \nu_M q(t)); \tag{6}$$

$$\nu_M p\mathbf{U}q(t) = \max(\nu_M q(t), \sup_{t<t_m} \inf_{t \leq t_n < t_m} \min(\nu_M p(t_n), \nu_M q(t_m))); \tag{7}$$

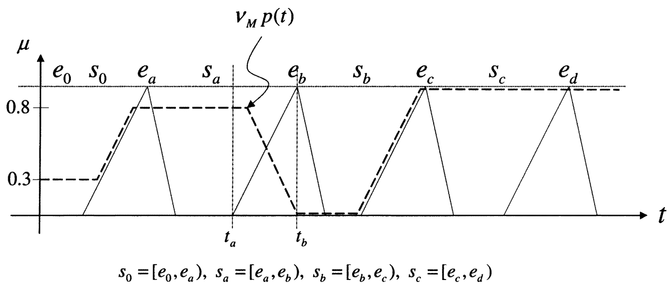

Fig. 3. Truth value change of an FLTL atomic proposition p.

$$\nu_M \mathbf{X}_{Rd} p(t) = \sup_{t_d > t} \min(\mu_R(t_d, t+d), \nu_M p(t_d)). \quad (8)$$

■

The function $\mu_R(a, b)$ in (8) is the satisfaction function on a continuous domain (SFC) $S(a\ R\ b)$ proposed by Lee [26] with respect to the comparison operator R. The notation is changed to prevent confusion with S in the model M.

The interpretation of FLTL is similar to that of the fuzzy temporal logic of Thiele [23], except FLTL is defined on a continuous time domain. The truth value of each atomic proposition is determined by the set P, with respect to the fuzzy temporal states. The membership value of each state is determined by $s \in S$ at a particular time t, as in Fig. 2. The truth and membership values are composed by sup-min composition to create the truth value of an atomic proposition. In Fig. 3, the truth value $\nu_M p(t)$ of an atomic proposition p is depicted. This figure shows how $\nu_M p(t)$ changes given $\mu_p(s_0) = 0.3$, $\mu_p(s_a) = 0.8$, $\mu_p(s_b) = 0$ and $\mu_p(s_c) = 1$. Since a state is a half-open interval between two events, by sup-min composition $\nu_M p(t)$ changes just before each event boundary. For example, $\mu_{s_a}(t) = 1 - \mu_{e_b}(t)$ for $t_a \le t \le t_b$, and since $\mu_{e_b}(t) < 0.2$ even after t_a, $\nu_M p(t)$ maintains a value of 0.8. Soon after, however, $\nu_M p(t)$ goes to 0 as $\mu_{e_b}(t)$ rises to 1 before t_b. Combining the truth values of these propositions with operators, we can describe the properties of the system at each temporal state.

Logical operators such as $\neg$ and $\wedge$ follow the usual fuzzy logic definitions. The "until" operator is interpreted as sup-inf composition for every possible "meet" time point t_m where formulae p and q meet. This operator $\mathbf{U}$ is a *strong until* operator. Unless there is a time point t_m at which $\nu_M q(t_m) > 0$, $\nu_M p\mathbf{U}q$ always yields 0. A weak until operator $\mathbf{U}_w$ can be defined as $p\mathbf{U}_w q \equiv p\mathbf{U}q \vee \mathbf{G}p$.

The "delay" operator $\mathbf{X}$ in (8) combines the truth value of the subformula at time $t_d > t$ with $\mu_R(t_d, t+d)$, just as the truth value of an atomic proposition is interpreted by reference to $\mu_s(t)$. SFC μ_R compares a crisp value t_d with a (possibly) fuzzy value $t + d$ and the result is in [0, 1]. For example, Fig. 4 shows the values of $\nu_M \mathbf{X}_{=d} p(a)$ and $\nu_M \mathbf{X}_{>d} p(a)$ for a given truth value $\nu_M p(t)$. Since d is a positive crisp value in this case, $\mu_=(t_d, t+d)$ in (8) is 1 iff $t_d = t+d$, and hence $\nu_M \mathbf{X}_{=d} p(a) = \nu_M p(a+d)$. On the other hand, since $\mu_>(t_d, a+d) = 1$ iff $t_d > a+d$, $\nu_M \mathbf{X}_{>d} p(a)$ takes the supreme value of $\nu_M p(t_d)$ with respect to $t_d > a + d$, which is also represented in the figure.

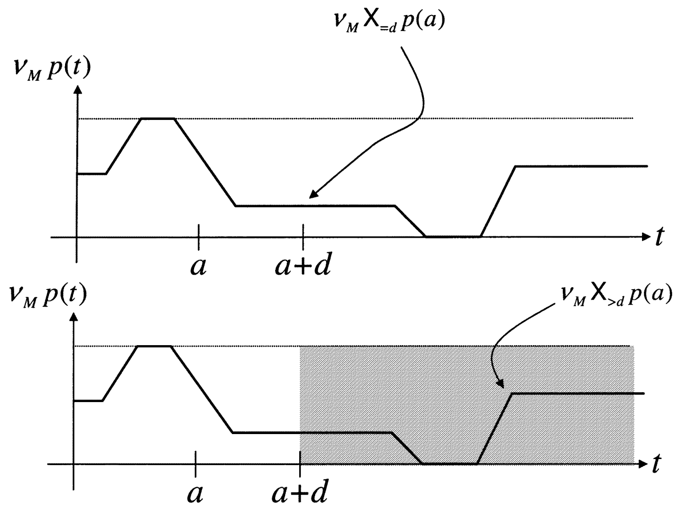

Fig. 4. Interpretation of the temporal operator $\mathbf{X}$.

C. FLTL and PLTL

FLTL is a fuzzy generalization of PLTL. Moreover, as will be shown below, FLTL is more expressive than PLTL. We use the notation employed in [2] for describing PLTL models and formulae.

A PLTL model is a tuple $M = (S, x, L)$ where

- S is a set of states;
- $x : \mathcal{N} \to S$ is an infinite sequence of states;
- $L : S \to Powerset(P)$ is a labeling of each state for which the set of atomic propositions in P is true.

When a PLTL model M satisfies a formula p we denote this as $M \models p$, and otherwise we write $M \not\models p$. A suffix of sequence x starting from the i-th state is denoted as x^i. Now we show that an FLTL model can represent a PLTL model.

Theorem 1: The FLTL model subsumes the PLTL model.

Proof: A PLTL model $M = (S, x, L)$ can be represented by the FLTL model $M' = (T', E', S', P')$ in which

$$T' = \mathcal{R}^+ \quad (9)$$
$$E' = \{e_i | 1 \le i \le n, e_i = i\} \quad (10)$$
$$S' = S, s_i = [e_i, e_{i+1}), i \ge 0 \quad (11)$$
$$P' = \{p | \mu_p(s) = 1 \text{ iff } p \in L(s)\}. \quad (12)$$

■

With respect to M', a PLTL formula p can be translated to an FLTL formula p' by substituting $\mathbf{X}$ with $\mathbf{X}_{=1}$. To show that FLTL subsumes PLTL, we note that, for $i = \lfloor t \rfloor$, $\nu_{M'} p'(t)$ equals 1 iff $x^i \models p$, and equals 0 otherwise. When this is satisfied, we say that p' *is equivalent to* p. Below we show by induction that an FLTL formula p' is equivalent to the PLTL formula p.

Theorem 2: An FLTL formula p' that is translated from the PLTL formula p is equivalent to p with respect to the model M' that represents the PLTL model M.

Proof:

- If $p' \in P'$, from (4), $\mu_{p'}(s) = 1$ iff $p \in L(s_i)$, and $\mu_s(t) = 1$ iff $\lfloor t \rfloor = i$. Hence, $\nu_{M'} p(t) = 1$ iff $x^i \models p$.
- If FLTL formulae p' and q' are equivalent to PLTL formulae p and q, respectively, it is clear from (5) and (6) that $\neg p'$ and $p' \wedge q'$ are equivalent to $\neg p$ and $p \wedge q$, respectively.

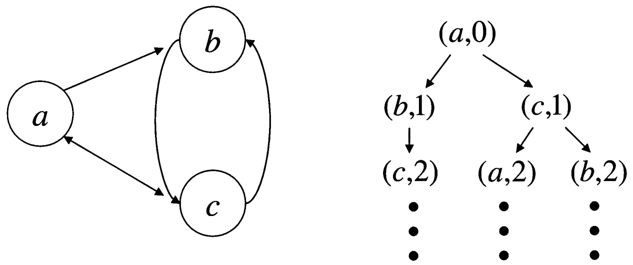

Fig. 5. CTL* temporal model.

- If $x^i \models p\mathbf{U}q$, $\exists j \geq i$, $x^j \models q$ and $\forall k, k < j$ implies $x^k \models p$. If $j > i$, from (7), for $t_m = j$, $\nu_{M'}p'\mathbf{U}q'(t) = 1$; If $j = i$, then $x^i \models q'$, and hence $\nu_{M'}q'(t) = 1$, which makes $\nu_{M'}p'\mathbf{U}q'(t) = 1$. And if $x^i \not\models p\mathbf{U}q$, either $\mathbf{G}\neg q$ or $\forall j$ such that $x^j \models q$, there exists $k, k < j$ such that $x^k \not\models p$. Hence, likewise for the true case, $\nu_{M'}p'\mathbf{U}q'(t) = 0$. On the other hand, if $\nu_{M'}p'\mathbf{U}q'(t) = 1$, there exists a t_m at which the value of the expression is maximized. Since $\nu_{M'}q'(\lfloor t_m \rfloor) = \nu_{M'}q'(t_m)$, for $j = \lfloor t_m \rfloor$, $x^i \models p\mathbf{U}q$. Likewise, if $\nu_{M'}p'\mathbf{U}q'(t) = 0$, $x^i \not\models p\mathbf{U}q$.
- It is clear that $\mathbf{X}_{=1}p'$ is equivalent to $\mathbf{X}p$, since from (8), $\mu_{=}(t_d, t+1) = 1$ iff $t_d = t + 1$.

Therefore we can conclude that the FLTL formula p' is equivalent to the PLTL formula p. ■

Given that FLTL can represent the PLTL model and formula, and can additionally represent temporal and logical uncertainty, we conclude that FLTL is more expressive than PLTL.

IV. Fuzzy Branching Temporal Logic: FBTL

Although FLTL can represent fuzzy events and states, it cannot describe a branching time model. To overcome this shortcoming, we herein develop FBTL. The temporal model of FBTL has the same fuzzy events and states as FLTL, but, in contrast to FLTL, the relationship among events and states is not a total order. In FBTL, the order relation is defined as a directed graph, as is the case in the temporal model of CTL*. Similar to PLTL, the boolean value of each proposition in the directed graph of CTL* changes according to the state. This graph can be expanded to a tree, as shown in Fig. 5. In this figure, a graph with three nodes a, b and c is transformed into a tree.

A. Definition of FBTL

The model of FBTL is a graph similar to the CTL* model. But in contrast to the CTL* model, each arc in the FBTL graph represents a transition between states due to the occurrence of an associated fuzzy event. Because there are many possible paths to follow, the temporal possibility distribution of each event is not fixed on an absolute time scale. These distributions are defined on a relative time domain, τ. We define the model of FBTL as follows.

Definition 4: The model of FBTL is a tuple (τ, E, S, P, A, B), in which

- τ is the *relative* time domain, $\tau = \mathcal{R}^+$;
- E is the set of fuzzy events, $E = \{e | e = \int \mu_e(t)/t, t \in \tau\}$, where e is convex and normalized;
- S is the set of states, $S = s_0, s_1, s_2, \ldots$;
- P is the set of fuzzy propositions $P = \{p | p = \sum \mu_p(s)/s, s \in S\}$;
- $A \subset S \times S$ is the set of directed arcs between two states;
- $B : A \rightarrow E$ is the mapping associating each arc with an event, $\forall a, b \in A, B(a) \neq B(b)$ if a and b have the same originating state.

■

Although the name of each state is defined in the model, the temporal possibility distributions of the states are yet to be determined. The state-proposition relationship is fixed, but the state-time relationship may vary. In fact, a state s can have different temporal possibility distributions for the same set of proposition values. The system starts in the *initial state* s_0 and changes its state when one of the possible events occurs, that is, when it follows one of the outgoing arcs of the current state.

We now define the syntax of the FBTL formula. Similar to CTL*, there are two types of FBTL formula, known as *state formulae* and *path formulae*. State formulae are the legitimate FBTL formulae, and path formulae are subformulae that define the operators applicable on a determined path. Path formulae define the temporal operators used in FLTL, $\mathbf{U}$ and $\mathbf{X}$. State formulae introduce new operators, $\mathbf{E}$ and $\mathbf{A}$, which mean "at least in one path" and "for all paths," respectively.

Definition 5: Fuzzy Branching Temporal Logic (FBTL) has *path formulae* and *state formulae* defined inductively as follows. The state formula is defined as

- if $p \in P$, p is a state formula;
- if p and q are state formulae, then $\neg p$ and $p \wedge q$ are also state formulae;
- if p is a path formula, then $\mathbf{E}p$ and $\mathbf{A}q$ are state formulae.

And the path formula is defined as

- each state formula is also a path formula;
- if p and q are path formulae, then $\neg p$ and $p \wedge q$ are also path formulae;
- if p and q are path formulae, then $p\mathbf{U}q$ is also a path formula;
- if p is a path formula, then $\mathbf{X}_{Rd}p$ is also a path formula, where $R \in \{\leq, \geq, =, <, >\}$, and d is a normalized fuzzy duration on τ;

and the state formulae are the well-formed formula of FBTL. ■

Operator priority in FBTL is the same as in FLTL, except $\mathbf{E}$ and $\mathbf{A}$ have the highest binding.

B. Interpretation of FBTL

Because the temporal possibility distributions of the events are defined on a relative time domain, we must map the events to an absolute time domain before the truth value of an FBTL formula is determined. This is called *event mapping*. The first step in event mapping is to establish the sequence of the states. From this sequence and the set A, B in the FBTL model, we can determine the starting and finishing events of each state in the sequence. A *fullpath* is the sequence of *mapped states*, which is a tuple $s' = (s, t_i, e_i, e_f)$, where $s \in S$ is the name of the state, $t_i \in T$ is the time at which the state is initiated, and $e_i, e_f \in E$ are the initiating and finishing events for the state,

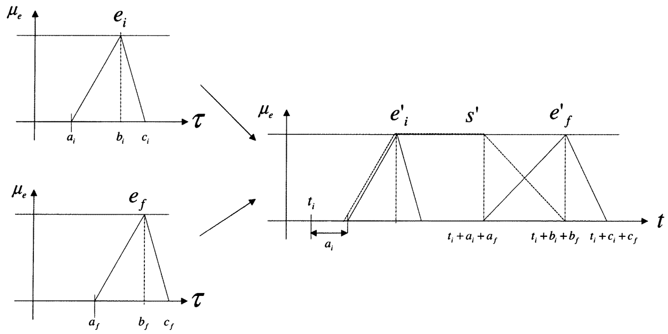

Fig. 6. Mapping events to an absolute time domain.

respectively. To map a state to the absolute time domain T, we first map events e_i, e_f to T. Let us denote the mapped events as e'_i, e'_f. Then

$$e'_i = e_i + t_i \tag{13}$$
$$e'_f = e_f + e'_i. \tag{14}$$

For example, in Fig. 6, while e_i is transposed right by a crisp t_i value (signified by a_i), e_f is moved right by the amount of e'_i and increased its uncertainty, since e'_i is a fuzzy value. Let us denote t_i for state s as $t_i(s)$. Then, for the initial state s_0, $t_i(s_0) = 0$, and for the following states s_i, $t_i(s_i) = e'_f(s_{i-1})$, where $e_f(s)$ is the finishing event of state s. The possibility distribution of the mapped state s' on the absolute time domain T is defined as the fuzzy half-open time interval $[e'_s, e'_t)$. The sequence of the mapped states makes a fullpath. For the initial state, which has no starting event, we assume an initial event that occurs at time 0.

In some cases, the first part of a fullpath is sufficient to interpret a formula, which is called the *prefix* of a fullpath. The prefix $x.s_n$ of a (possibly infinite) fullpath $x = s_0, s_1, s_2 \ldots$ is a finite subsequence $s_0, s_1, \ldots, s_n$. The *possibility of a prefix* is defined as $\mu_{x.s_n} = \inf^n_{s_i \in x} \sup \mu_{s_i}, i = 1, 2, \ldots, n$.

Like the syntax definition, the interpretation of an FBTL formula is also defined in two parts: a *state formula* and a *path formula*. As defined in Section IV-A, a well-formed FBTL formula is a state formula, which may contain path formulae as subformulae. From the model of FBTL, a state formula is defined on a state, and a path formula is defined on a fullpath. The interpretation starts from $s_0 \in S$, and the starting state or fullpath of each subformula must be determined to interpret the subformulae. To represent this, the truth value of a state formula p at time t with respect to the model M on state s is denoted $\nu_{M,s}p(t)$, whereas the truth value of a path formula on path x is denoted $\nu_{M,x}p(t)$.

Definition 6: For a model M, the *truth value of an FBTL formula* is defined inductively as follows. For the *state formula*

$$\text{if } p \in P, \quad \nu_{M,s}p(t) = \min(\mu_p(s), \mu_s(t)) \tag{15}$$
$$\nu_{M,s}\neg p(t) = 1 - \nu_{M,s}p(t) \tag{16}$$
$$\nu_{M,s}p \wedge q(t) = \min(\nu_{M,s}p(t), \nu_{M,s}q(t)) \tag{17}$$
$$\nu_{M,s}\mathbf{E}p(t) = \sup_{\forall x} \nu_{M,x}p(t); \text{where } x \text{ starts with } s \tag{18}$$
$$\nu_{M,s}\mathbf{A}p(t) = \inf_{\forall x} \nu_{M,x}p(t) \text{where } x \text{ starts with } s. \tag{19}$$

And for the *path formula*

$$\nu_{M,x}p(t) = \sup_{s \in x} \min(\nu_{M,s}p(t), \mu_{x.s}, \mu_s(t)) \quad \text{when } p \text{ is a state formula} \tag{20}$$
$$\nu_{M,x}\neg p(t) = 1 - \nu_{M,x}p(t) \tag{21}$$
$$\nu_{M,x}p \wedge q(t) = \min(\nu_{M,x}p(t), \nu_{M,x}q(t)) \tag{22}$$
$$\nu_{M,x}p\mathbf{U}q(t) = \max(\nu_{M,x}q(t), \sup_{t<t_m} \inf_{t \le t_n < t_m} \min(\nu_{M,x}p(t_n), \nu_{M,x}q(t_m))) \tag{23}$$
$$\nu_{M,x}\mathbf{X}_{Rd}p(t) = \sup_{t_d > t} \min(\mu_R(t_d, t+d), \nu_{M,x}p(t_d)). \tag{24}$$

■

Once a path is determined, the interpretation of FBTL becomes very similar to that of FLTL. However, unlike FLTL, FBTL requires two parameters (s and t) to determine the current state of the modeled system. The parameter s represents the current state of the system, and is used to determine the fullpath. The parameter t specifies the time point at which the truth value of a formula is to be evaluated. For example, (20) takes the sup-min composition for every state in the current fullpath x and the state formula takes the minimum of $\mu_s(t)$; this process leads eventually to the selection of the s with the highest $\mu_s(t)$.

Although FBTL syntax and interpretation subsumes FLTL, in general it is impossible to represent an FLTL model with an FBTL model. This is because the fuzzy addition operator is used in mapping the events. In contrast to the calculus of crisp numbers, the fact that fuzzy numbers a, b and c obey the relation $a + b = c$ does not generally mean that $c - b = a$ holds [27]. Hence, given an FLTL model, generally it is not possible to construct a FBTL model that represents the same fullpath. Of course, it is possible to subsume a class of the FLTL (especially PLTL) with FBTL.

C. FBTL and CTL*

In this section it is shown that FBTL is more expressive than CTL*. From a CTL* model and formula, it is possible to build an FBTL model and formula with an equivalent interpretation. In the description of CTL* presented here, we use the notation given in [2]. A model of CTL* is $M = (S, R, L)$ where

- S is the set of states;
- R is a total binary relation, $R \subset S \times S$;
- $L : S \to Powerset(P)$ is a labeling that associates with each state s an interpretation $L(s)$ of all atomic proposition symbols at state s.

The proof that a CTL* model can be represented by an FBTL model is similar to proof presented above that a PLTL model can be represented by an FLTL model (Theorem 1).

Theorem 3: A CTL* model can be represented by an FBTL model.

Proof: For a CTL* model $M = (S, R, L)$, we create an FBTL model $M' = (\tau, E, S', P, A, B)$ in which

- $\tau = \mathcal{R}^+$;
- $E = \{e_0, e_1, \ldots, e_n\}$ where $n = |R|$, $\mu_{e_i}(1) = 1$ and 0 otherwise, for $0 \leq i \leq n$;
- $S' = S$;
- $P = \{p | \mu_p(s) = 1 \text{ iff } p \in L(s)\}$;
- $A = R$;
- B maps each element of A to E.

■

In M', the possibility distribution of each mapped state is a crisp interval of unit length. Let us denote the interval of state s as $[t_i(s), t_f(s))$. We also introduce the notation that for $t \in [t_i(s), t_f(s))$, $s = s_M(t)$.

A CTL* formula p is translated to an FBTL formula p' by substituting $\mathbf{X}$ for $\mathbf{X}_{=1}$, in the same manner as a PLTL formula is translated to an FLTL formula in Section III-C. An FBTL state formula p' is *equivalent* to a CTL* state formula p when $\nu_{M',s} p'(t_i(s))$ is 1 iff $M, s \models p$, and 0 otherwise. An FBTL path formula p' is *equivalent* to a CTL* path formula p when $\nu_{M',x} p'(t)$ is 1 iff $M, x^{\lfloor t \rfloor} \models p$, and 0 otherwise.

We now show by induction that, for a CTL* formula p, the translated FBTL formula p' is equivalent to p.

Theorem 4: FBTL formula p' translated from a CTL* formula p is equivalent with respect to the model M' representing the CTL* model M.

Proof:

- If $p' \in P, \nu_{M',s} p'(t_i(s)) = \mu_{p'}(s)$, which is equivalent to p.
- If FBTL formulae p' and q' are equivalent to CTL* formulae p and q, respectively, then by (17) and (22), it is trivial to show that $p' \wedge q'$ is equivalent to $p \wedge q$.
- If an FBTL path formula p' is equivalent to p, then by (18) and (19), it is clear that $\mathbf{E}p'$ and $\mathbf{A}p'$ are equivalent to $\mathbf{E}p$ and $\mathbf{A}p$, respectively.
- If an FBTL path formula p' is also a state formula, since $\mu_s(t) = 1$ iff $t \in [t_i(s), t_f(s))$ and 0 otherwise, from (20), $\nu_{M',x} p'(t) = \nu_{M', s_M(t)} p'(t)$. Hence, $\nu_{M_f,x} p'(t)$ is 1 iff $M_c, x^{\lfloor t \rfloor} \models p$ and 0 otherwise, which is equivalent to the CTL* formula p.
- If FBTL path formulae p' and q' are equivalent to CTL* path formulae p and q, respectively, if $x^i \models p\mathbf{U}q, \exists j \geq i, x^j \models q$ and $\forall k, k < j$ implies $x^k \models p$. If $j > i$, from (23), for $t_m = j$, $\nu_{M',x} p'\mathbf{U}q'(t) = 1$; if $j = i$, then $x^i \models q'$, and hence $\nu_{M',x} q'(t) = 1$, which makes $\nu_{M',x} p'\mathbf{U}q'(t) = 1$. And if $x^i \not\models p\mathbf{U}q$, either $\mathbf{G}\neg q$ or $\forall j$ such that $x^j \models q$, there exists $k, k < j$ such that $x^k \not\models p$. Hence, likewise for the true case, $\nu_{M',x} p'\mathbf{U}q'(t) = 0$. On the other hand, if $\nu_{M',x} p'\mathbf{U}q'(t) = 1$, there exists a t_m that maximizes the value of the expression. Since $\nu_{M',x} q'(\lfloor t_m \rfloor) = \nu_{M',x} q'(t_m)$ by the construction of M', for $j = \lfloor t_m \rfloor$, $x^i \models p\mathbf{U}q$. Likewise, if $\nu_{M',x} p'\mathbf{U}q'(t) = 0$, $x^i \not\models p\mathbf{U}q$. Hence, $p'\mathbf{U}q'$ is equivalent to $p\mathbf{U}q$.
- If an FBTL formula p' is equivalent to a CTL* formula p, it is clear that $\mathbf{X}_{=1}p'$ is equivalent to $\mathbf{X}p$, since from (24), $\mu_{=}(t_d, t+1) = 1$ iff $t_d = t+1$.

Therefore we can conclude that the FBTL formula p' is equivalent to the CTL* formula p. ■

Hence, given that FBTL can represent a CTL* model and formula, and can additionally represent temporal and logical uncertainty, we conclude that FBTL is more expressive than CTL*.

D. Deductive System for FBTL

A classical deductive system for a temporal logic consists of a set of axiom schemes and inference rules. A formula p is said to be *provable*, denoted $\vdash p$, if the deductive process can show that p follows from the axioms and rules. CTL, which is a subset of CTL*, has a complete deductive system [2]. However it cannot be applied to FBTL directly, mainly due to differences between classical and fuzzy logic rather than to differences between the temporal models.

The deductive system for *fuzzy propositional logic* is defined in [28], [29] and [30]. A *signed fuzzy formula* is a tuple (p, ν) where p is a formula, and $\nu \in [0, 1]$ is an *assigned truth value*. This means that for any interpretation, p is true *at least* to the degree ν. A *Fuzzy Deductive System* is a pair $\mathcal{T} = (a, R)$ where a is a fuzzy set of fuzzy logical axioms, and R is a set of fuzzy inference rules. A *fuzzy inference rule* is a pair $r = (r_f, r_\nu)$ where r_f is a *syntactical component* that operates on formulae, and r_ν is a *valuation component* that operates on truth values to calculate how the conclusion depends on the truth values of the premises. A rule r is usually written as

$$\frac{f_1, f_2, \ldots, f_n}{r_f(f_1, f_2, \ldots, f_n)}, \frac{\nu_1, \nu_2, \ldots, \nu_n}{r_\nu(\nu_1, \nu_2, \ldots \nu_n)}.$$

This expression means that if the formulae $f_1, f_2, \ldots, f_n$ are known to be true *at least* to the degree $\nu_1, \nu_2, \ldots, \nu_n$, respectively, then $r_f(f_1, f_2, \ldots, f_n)$ is true at least to the degree $r_\nu(\nu_1, \nu_2, \ldots, \nu_n)$. A fuzzy inference rule is *sound* if $\nu_M r_f \geq r_v(\nu_M f_1, \nu_M f_2, \ldots, \nu_M f_n)$. In the case of FBTL, the valuation components ν_i and r_ν are functions of $t \in T$. However, since the definition of assigned truth value is that the formula is at least true to the degree for any interpretation on *any* temporal model, t can be omitted for notational simplicity.

For FBTL, all axioms for fuzzy propositional logic hold as described in [30]. Additionally, some axioms follow from the definition of FBTL operators

$$\mathbf{F}p \equiv \text{true}\mathbf{U}p \tag{25}$$
$$\mathbf{G}p \equiv \neg\mathbf{F}\neg p \tag{26}$$
$$\mathbf{A}p \equiv \neg\mathbf{E}\neg p \tag{27}$$
$$\mathbf{E}(p \vee q) \equiv \mathbf{E}p \vee \mathbf{E}q \tag{28}$$
$$\mathbf{F}(p \vee q) \equiv \mathbf{F}p \vee \mathbf{F}q. \tag{29}$$

Axioms (25) and (26) are from their definitions, (2) and (3). Axiom (27) follows from (19) and (18). Axiom (28) follows from (18), and axiom (29) follows from (25) and (23), which implies

$$\nu_{M,x}\mathbf{F}p(t) = \sup_{t_m \geq t} \nu_{M,x} p(t_m). \tag{30}$$

From the axioms above, other equivalences can be drawn. From (26) and axiom (29), we can write

$$\mathbf{G}(p \wedge q) \equiv \mathbf{G}p \wedge \mathbf{G}q. \tag{31}$$

And from axioms (27) and (28), we can write

$$\mathbf{A}(p \wedge q) \equiv \mathbf{A}p \wedge \mathbf{A}q. \tag{32}$$

The inference rules for fuzzy propositional logic can be applied, as the *Fuzzy Modus Ponens* (R_{FMP}) and the *Rule of Conjunction* (R_C) [30]

$$R_{FMP}: \frac{p, (p \rightarrow q)}{q}, \frac{a, b}{\min(a,b)} \tag{33}$$

$$R_C: \frac{p, q}{p \wedge q}, \frac{a, b}{\min(a,b)}. \tag{34}$$

In addition, there are four kinds of *Generalization Rule*

$$R_{GR}: \frac{p}{\mathbf{AG}p}, \frac{a}{a} \tag{35}$$

$$R_{GRF}: \frac{\mathbf{G}p}{\mathbf{F}p}, \frac{a}{a} \tag{36}$$

$$R_{GRE}: \frac{\mathbf{A}p}{\mathbf{E}p}, \frac{a}{a} \tag{37}$$

$$R_{GRX}: \frac{p}{\mathbf{X}_{Rd}p}, \frac{a}{a}. \tag{38}$$

R_{GR} is adopted from the deductive system of CTL. This rule is sound by the definition of the assigned truth value, which states that p is at least true to the degree a for any time point of any fullpath. R_{GRF} follows from (30), which states implies that $\nu_{M,x}\mathbf{F}p(t) \geq \nu_{M,x}\mathbf{G}p(t)$. R_{GRE} follows from (19) and (18), which implies that $\nu_{M,s}\mathbf{A}p(t) \leq \nu_{M,s}\mathbf{E}p(t)$. For R_{GRX}, R is a comparison operator, which is the same as the one in (8).

The soundness of other rules and axioms can also be shown. Additional rules such as the Rule of Disjunction R_D and Rule of Simplification R_S can be derived from R_C

$$R_D: \frac{p, q}{p \vee q}, \frac{a, b}{\max(a,b)} \tag{39}$$

$$R_S: \frac{p \wedge q}{p}, \frac{a}{a}. \tag{40}$$

Theorem 5: The inference rules in (33)–(40) are sound.

Proof:

- For two fuzzy formulae p and q with minimum truth values μ_p and μ_q, respectively, $(p \rightarrow q)$ has minimum truth value $\min(1-\mu_p, \mu_q)$. Now the conclusion of R_{FMP} has truth value of $\min(\mu_p, \min(1-\mu_p, \mu_q))$, certainly less than μ_q. Hence, R_{FMP} is sound.
- R_C is trivial.
- If, by definition of the FBTL deductive system, an FBTL formula p has minimum truth value a regardless of the model, $\mathbf{AG}p$ of course has the same truth value a. Hence R_{GR} is sound.
- By (26) and (30), for any path FBTL formula p, $\nu_{M,x}\mathbf{F}p(t) \geq \nu_{M,x}\mathbf{G}p(t)$ regardless of M and t. Hence, R_{GRF} is sound.
- Similarly, by (18) and (19), for any FBTL formula p, $\nu_{M,s}\mathbf{E}q(t) \geq \nu_{M,s}\mathbf{A}q(t)$ regardless of M and t. Hence, R_{GRE} is sound.
- Similar to R_{GR}, if p has minimum truth value a regardless of model, from (24) and the fact that d is normalized, $\nu_{M,x}\mathbf{X}_{Rd}p \geq a$ regardless of comparison operator R and delay value d. Hence R_{GRX} is sound.
- From R_C and the fact that $p \vee q \equiv \neg(\neg p \wedge \neg q)$, R_D is sound.
- From R_C, R_S is trivially sound.

Hence, the inference rules in (33)–(40) are sound. ■

Although this deductive system is not yet complete, it can be applied to prove simple theorems and formulae.

For example, the theorem

$$\frac{p \rightarrow q, \mathbf{AG}p}{\mathbf{AG}q}, \frac{a, b}{\min(a,b)}$$

can be proved as below. First we develop a trivial rule

$$(p \rightarrow q) \wedge p \;\; a \tag{41}$$

$$p \rightarrow q \;\; a \text{ by } R_S \tag{42}$$

$$p \;\; a \text{ from Eq. (41), by } R_S \tag{43}$$

$$q \;\; a \text{ from Eqs. (42) and (43), by } R_{FMP}. \tag{44}$$

Now we can show that

$$\begin{aligned} &p \rightarrow q, \mathbf{AG}p \quad a, b \\ &\mathbf{AG}(p \rightarrow q), \mathbf{AG}p \quad a, b \text{ by } R_{GR} \\ &\mathbf{AG}(p \rightarrow q) \wedge \mathbf{AG}p \;\; \min(a,b) \text{ by } R_{BC} \\ &\mathbf{AG}((p \rightarrow q) \wedge p) \;\; \min(a,b) \text{by Eqs. (32) and (31)} \\ &\mathbf{AG}q \;\; \min(a,b) \text{ by Eqs. (41)} - \text{(44)}. \end{aligned}$$

As stated above, this deductive system is not yet complete, although it covers all the operators used in FBTL: for example, $\mathbf{EF}p \wedge (p \rightarrow q) \rightarrow \mathbf{EF}q$ cannot be proven by this deductive system.

A complete deductive system for CTL is introduced in [2], but additional complexity of CTL* makes it difficult to build a complete deductive system. Since FBTL subsumes CTL*, it is supposed to be a difficult task to build a complete deductive system for FBTL. To find a complete deductive system for FBTL and prove its completeness, we suspect that more understanding of the model of FBTL is needed. Especially, the solution to the satisfiability testing problem would be helpful, as the completeness of the CTL deductive system was shown in [2].

V. Fuzzy Timed Job Shop Scheduling Problem

To illustrate the usefulness of FBTL, we describe a Fuzzy Timed Job Shop Scheduling problem within the FBTL framework. Although this representation does not solve the scheduling problem itself, it provides an example of the system modeling and constraint satisfaction analysis. Several previous studies have considered Fuzzy Job Shop Scheduling [31]–[33]. The crisp job shop scheduling problem is defined with exact time parameters; however, this is often not possible in real-world situations, especially when human factors are involved. To represent the temporal uncertainties commonly

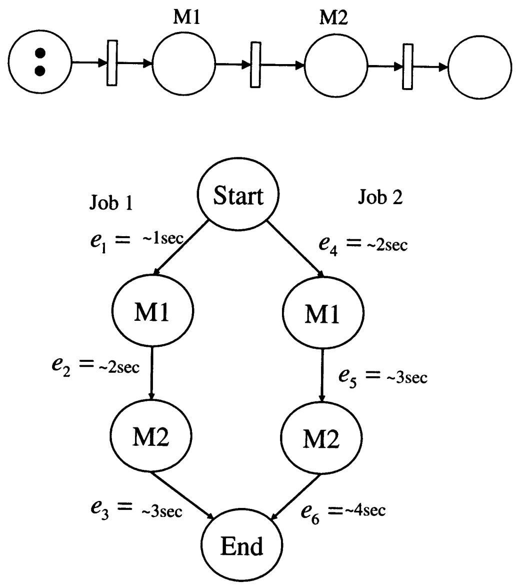

Fig. 7. Fuzzy timed job shop problem example.

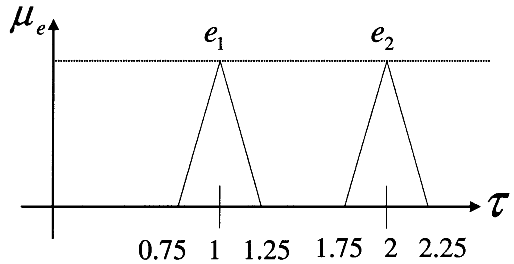

Fig. 8. Fuzzy temporal distributions "about 1 s" and "about 2 s" represented as fuzzy numbers.

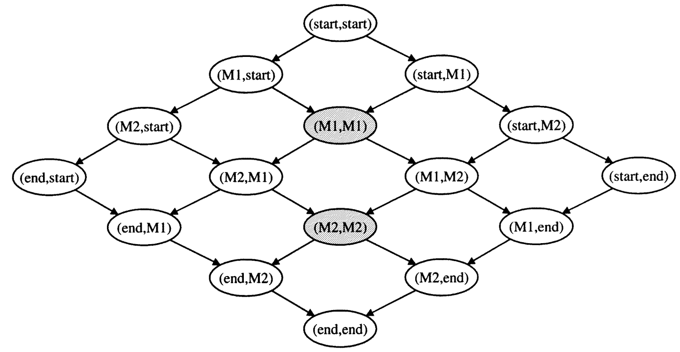

Fig. 9. FBTL model of the fuzzy timed job shop.

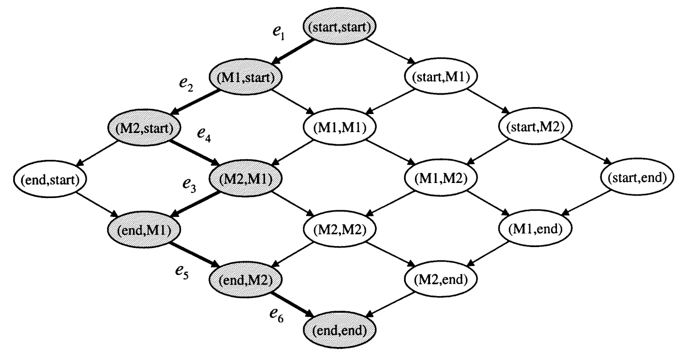

Fig. 10. Selected fullpath.

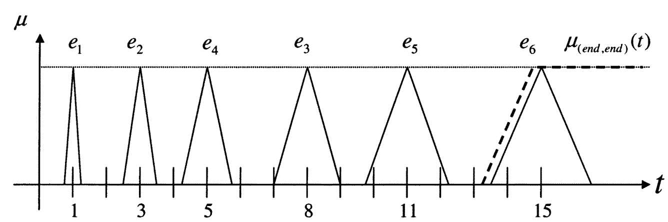

Fig. 11. Fullpath with mapped events.

encountered in the real world, a fuzzy job shop scheduling problem is defined [31]. In this formalism, the processing time for each job is represented by a fuzzy number rather than a real number.

The scheduling problem considered here has two jobs to be completed, as shown in Fig. 7. In this figure, the two jobs are represented by tokens in a Petri Net. There are two machines to use, and each job spends different time durations in each machine. Job 1 spends about 1 s before starting the processing, 2 s in Machine 1, and about 3 s in Machine 2. Job 2 spends about 2 s before processing, about 3 s in Machine 1, and about 4 s in Machine 2. Because the durations spent in each machine are uncertain, the events have fuzzy temporal distributions as depicted in Fig. 8.

The FBTL model for this problem can be constructed as outlined in Fig. 9. Note that this model includes two states in which the two jobs use the same machine, represented as grey states. The identification of such forbidden states enables us to express forbidden conditions in the scheduling requirements explicitly in FBTL. In Fig. 9, each state is labeled as (s_1, s_2), where s_1 and s_2 are the states of jobs 1 and 2, respectively. Because each job runs independently of the other and each job has 4 states, there are a total of $4^2 = 16$ FBTL states. Note that two grey states are included in these 16 states. These conflicting states must be included in the model too, because they have fuzzy time boundaries [34]–[38].

Each possible schedule will be represented as a fullpath, and the FBTL model represents every possible schedule. With this model, scheduling constraints can be expressed in terms of the FBTL formulae. For example, the constraint that one machine shouldn't be occupied by more than one job can be specified as a path formula $\mathbf{G}\neg(p \vee q)$, provided that $\mu_{(M1,M1)}(p) = 1$ and $\mu_{(M2,M2)}(q) = 1$. Hence the wff is $\mathbf{AG}\neg(p \vee q)$, which applies the constraint to every schedule. On the other hand, the deadline condition can be expressed as $\mathbf{X}_{<d}p_t$ with the deadline d and $\mu_{(\text{end},\text{end})}(p_t) = 1$. Since this is a path formula, a well-formed formula is $\mathbf{EX}_{<d}p_t$,, which means there is at least one possible schedule.

For example, let us consider the path marked in grey in Fig. 10. The state (end,end) is reached at about 15 s. The temporal distribution of this path is depicted in Fig. 11. Hence for $d \geq 15$, $\nu_{M,(\text{start},\text{start})}\mathbf{EX}_{<d}p_t(0)$ is evaluated to be 1. Meanwhile, when $d = 14$ as in Fig. 12, it will be less than 1 indicating that it is not fully satisfied, although not totally false; this represents the flexible due-date. Through such calculations, FBTL can be used to control the degree of satisfaction of a given set of constraints such as flexible due dates and termination conditions.

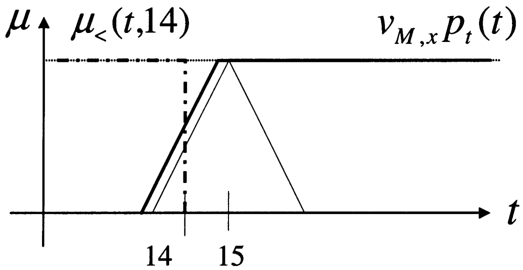

Fig. 12. Fullpath in Fig. 11, magnified around 15 s.

The job shop scheduling example presented above demonstrates the ability of FBTL to model systems with multiple execution paths. Compared to CTL*, FBTL has the advantage that it can represent temporal uncertainty. Moreover, the degree of satisfaction of the constraints on a system can be represented as the truth value of FBTL formula. Furthermore, although an FLTL formula represents the temporal properties of a single schedule solution, FBTL can describe the temporal properties of multiple schedules.

VI. Conclusion

In this paper, fuzzy branching temporal logic (FBTL) is proposed. This temporal logic can model dynamic systems with uncertain temporal information and a branching time model. It has fuzzy events and fuzzy states in its temporal model, and is able to express fuzzy logical formulae and fuzzy temporal relationships among those formulae. In addition it subsumes CTL*, which has previously been used to model state-based concurrent systems. A deductive system for FBTL is also discussed.

FBTL is applicable to systems in which the durations of, or intervals between events are not known exactly. To demonstrate the utility of the proposed method, the temporal model of a fuzzy timed job shop problem was constructed and the formulae for various conditions were derived. This example clearly demonstrated how FBTL enables us to control the satisfaction degree for each constraint under uncertain due dates and termination conditions.

Currently we are developing an FBTL-based analysis framework for fuzzy timed Statecharts. As further works, an algorithm for testing satisfiability of a formula is considered.

References

[1] S.-I. Moon and K.H. Lee, "Fuzzy linear temporal logic," in *Proc. 2nd Int. Symp. Adv. Intell. Syst. (ISAIS)*, Daejeon, Korea, Aug. 24–25, 2001, pp. 184–188.

[2] E. A. Emerson, "Temporal and modal logic," in *Handbook of Theoretical Computer Science*, j. Van Leeuwen, Ed. New York: Elsevier, 1990, vol. 2, ch. 16.

[3] P. Bellini, R. Mattolini, and P. Nesi, "Temporal logics for real-time system specification," *ACM Comput. Surveys (CSUR)*, vol. 32, no. 1, pp. 12–42, 2000.

[4] J. F. Allen, "Maintaining knowledge about temporal intervals," *Commun. ACM*, vol. 26, no. 1, pp. 832–843, 1983.

[5] S. Dutta, "An event-based fuzzy temporal logic," *Proc. 18th IEEE Intl. Symp. Multiple-Valued Logic*, pp. 64–71, 1988.

[6] D. Dubois and H. Prade, "Processing fuzzy temporal knowledge," *IEEE Trans. Syst., Man Cybern.*, vol. 19, pp. 729–744, July/Aug. 1989.

[7] L. Vil, "A survey on temporal reasoning in artificial intelligence," *AI Commun.*, vol. 7, no. 1, pp. 4–28, 1994.

[8] S. Barro, R. Marín, J. Mira, and A. R. Patón, "A model and a language for the fuzzy representation and handling of time," *Fuzzy Set Syst.*, vol. 61, pp. 153–175, 1994.

[9] Z. Chen, "Fuzzy temporal reasoning for process supervision," *Expert Syst.*, vol. 12, no. 2, pp. 123–137, 1995.

[10] G. Arroyo-Figueroa, E. Solis, A. Villavicencio, and L. E. Sucar, "SADEP — A fuzzy diagnostic system shell—An application to fossil power plant operation," *Expert Syst. Applicat.*, vol. 14, pp. 43–52, 1998.

[11] T. L. Dean and D. V. McDermott, "Temporal data base management," in *Artif. Intell.*. New York: Elsevier, 1987, vol. 32, pp. 1–55.

[12] R. Dechter, I. Meiri, and J. Pearl, "Temporal constraint networks," in *Artif. Intell.*. New York: Elsevier, 1991, vol. 49, pp. 61–95.

[13] P. van Beek, "Reasoning about qualitative temporal information," in *Proc. 8th (US) Nat. Conf. Artif. Intell. (AAAI-90)*, Boston, MA, 8 1990, pp. 728–734.

[14] L. Vila and L. Godo, "On fuzzy temporal constraint networks," *Mathware Soft Comput.*, vol. 3, pp. 315–334, 1994.

[15] M. A. C. Viedma, R. M. Morales, and I. N. Sánchez, "Fuzzy temporal constraint logic: A valid resolution principle," in *Fuzzy Sets Systems*. New York: Elsevier, 2001, vol. 117, pp. 231–250.

[16] A. Pnueli, "The temporal logic of programs," in *Proc. 18th IEEE Symp. Foundation Comput. Sci.*, 1977, pp. 46–57.

[17] B. A. Haugh, "Non-standard semantics for the method of temporal arguments," in *Proc. IJCAI'87*, 1987, pp. 449–454.

[18] H. Reichgelt, "Semantics for reified temporal logic," in *Advances in AI*, J. Hallam and C. Mellish, Eds. New York: Wiley, 1987, pp. 49–61.

[19] H. S. Bae, "Validation of Timing and Communication Constraints in Real-Time Parallel Programs," Ph.D. dissertation, Korea Adv. Inst. Sci. Technol., Daejon, 1999.

[20] H. Barringer, R. Kuiper, and A. Pnueli, "A really abstract concurrent model and its temporal logic," in *Proc. 13th Annu. ACM Symp. Principles Programming Languages*, 1986, pp. 173–183.

[21] H.-C. Kim and K.-W. Oh, "A fuzzy temporal logic—FTL," in *Proc. 1991 Workshop N. Armer. Fuzzy Inform. Processing Soc.*, May 1991, pp. 83–86.

[22] ——, "A new representation model in uncertain temporal knowledge," in *Proc. IFSA'91 Conf.*, 1991, pp. 113–116.

[23] H. Thiele and S. Kalenka, "On fuzzy temporal logic," in *Proc. 2nd IEEE Int. Conf. Fuzzy Syst.*, 3 1993, pp. 1027–1032.

[24] T. van Le, "A fuzzy temporal logic-based framework for fuzzy discrete-event simulation," in *Proc. EUROSIM'95 Cofn.*, F. Breitenecker and I. Husinsky, Eds., 1995, pp. 1125–1130.

[25] D. Dubois, J. Lang, and H. Prade, "Timed possibilistic logic," *Fundamenta Informat.*, vol. 15, no. 3–4, pp. 211–234, 1991.

[26] J.-H. Lee and H. Lee-Kwang, "Comparison of fuzzy values on a continuous domain," *Fuzzy Sets Syst.*, vol. 118, no. 3, pp. 419–428, 3 2001.

[27] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*. New York: Academic, 1980.

[28] G. Gerla, "Fuzzy control as a fuzzy deduction system," *Fuzzy Set Syst.*, vol. 121, pp. 409–425, 2001.

[29] F. Klawonn and V. Novák, "The relation between inference and interpolation in the framework of fuzzy systems," *Fuzzy Sets Syst.*, vol. 81, pp. 331–354 , 1996.

[30] E. Turunen, "Advances in soft computing," in *Mathmatics Behind Fuzzy Logic*. Heidelberg, Germany: Physica-Verlag, 1999.

[31] M. Sakawa and T. Mori, "An efficient genetic algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy due-date," *Comput. Ind. Eng.*, vol. 36, pp. 325–341, 1999.

[32] K. H. Kim, J. Y. Song, and K. H. Wang, "A negotiation based scheduling for items with flexible process plans," *Comput. Ind. Eng.*, vol. 33, no. 3–4, pp. 785–788, 1997.

[33] T. Murata, M. Gen, and H. Ishibuchi, "Multi-objective scheduling with fuzzy due-date," *Comput. Ind. Eng.*, vol. 35, no. 3–4, pp. 439–442, 1998.

[34] T. Alsinet, L. Odo, and S. Sandri, "On the semantics and automated deduction for PLFC, a logic of possibilistic uncertainty and fuzziness," in *Proc. 15th Annu. Conf. Uncertainty Artificial Intell. (UAI–99)*, San Francisco, CA, 1999, pp. 3–12.

[35] H. Lee-Kwang, K. A. Seong, and K.-M. Lee, "Hierarchical partition of nonstructured concurrent systems," *IEEE Trans. Syst., Man Cybern. B*, vol. 27, pp. 105–108, Feb. 1997.

[36] H. Lee-Kwang and J.-H. Lee, "A method for ranking fuzzy numbers and its application to decision making," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 677–685, Dec. 1999.

[37] D. Dubois and H. Prade, *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. New York: Plenum, 1986.

[38] P. Hájek, *Metamathmatics of Fuzzy Logic*. Boston, MA: Kluwer, 1998.

Seong-ick Moon was born in Korea in 1973. He received the B.E. and M.E. degrees in computer science from the Korea Advanced Institute of Science and Technology, Daejon, in 1994 and 1996, respectively, where he is currently pursuing the Ph.D. degree.

His research interests include artificial intelligence, fuzzy theory and logic, temporal logic and reasoning.

Kwang H. Lee was born in Korea in 1954. He received the B.S. degree in industrial engineering from Seoul National University, Seoul, Korea, in 1978, the M.S. degree in industrial engineering from the Korea Advanced Institute of Science and Technology (KAIST), Seoul, in 1980, the M.S. and Dr.Ing. degrees in computer science from the Institute National des Sciences Appliquees de Lyon, France, in 1982 and 1985, respectively, and the Dr.Sci. degree from Lyon I University, Lyon, France, in 1988.

He joined KAIST, Seoul, as an Assistant Professor in 1985, where he is now the Mirae Chair Professor since 2000. He was also an International Fellow with the Stanford Research Institute, Stanford, CA, from 1995 to 1996, Visiting Professor with the Tokyo Institute of Technology, Tokyo, Japan, from December 1998 to February 1999. His research interests include Bioinformatics, Fuzzy Intelligent Systems and Artificial Intelligence.

Dr. Lee is a member of the Korea Academy of Science, the President of the Korean Fuzzy and Intelligent Systems Society, and Vice President of the Korean Society for BioInformatics. He is also the General Chair of International Symposium on Andanced Intelligent systems (4th ISIS) in 2003.

Doheon Lee received the B. S., M. S., and Ph. D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, in 1990, 1992, and 1995, respectively.

He was a Visiting researcher at the University of Texas, Austin, and the National Institutes of Health, Bethesda, MD, in 1999 and 2002, respectively. He is now an Associate Professor in the Department of BioSystems, and a Vice-Director of the Bioinformatics Research Center, KAIST. His research interests include data mining, database, and bioinformatics. He has been an Associate Editor for ACM TRANSACTIONS ON INTERNET TECHNOLOGY since 2001

Dr. Lee is a member of ACM.