Logic of Estimates for Fuzzy Logics Sentences

Gerald Plesniewicz[a]

[a] *National Research University MPEI, 14 Krasnokazarmennaya str., Moscow, Russian Federation*

Abstract

Let **L** be any fuzzy logic. An *estimate* for **L** sentence (formula) φ is an expression of the form $\varphi \ge a$ or $\varphi \le a$ where a is a number from the unit interval [0,1]. Estimates are interpreted naturally: for any fuzzy interpretation “ •” of **L** formulas, we set the estimates $\varphi \ge a$ and $\varphi \le a$ to be true (i.e., “$\varphi \ge a$” = 1 and “$\varphi \le a$ “ = 1) if and only if “φ” $\ge a$ and “φ “ $\le a$. Let **B** be the set of all Boolean combinations of estimates. We define the *logic of estimates* **E-L** assuming that **B** is the set of its sentences, and extending interpretations “•” to **B** in accordance with the meaning of Boolean operations. Thus, we may consider **E-L** as a crisp logic that is a metalogic for the logic **L**. In this paper we build for the logics **E-Z** and **E-FLTL** sound and complete inference methods based on analytical tableaux. Here **Z** denotes the Zadeh's fuzzy propositional logic and **FLTL** denotes the fuzzy linear temporal logic. Also we describe a query answering method over knowledge bases written in the logics **E-Z** and **E-FLTL**.

Keywords 1

Fuzzy logics, inference methods, analytical tableaux, knowledge bases, fuzzy ontologies and fact bases, query answering

1. Introduction. Main Definitions

Let **L** be any fuzzy logic. An *estimate* for **L** sentence (formula) φ is an expression of the form $\varphi \ge a$ or $\varphi \le a$ where a is a number from the unit interval [0, 1]. Estimates are interpreted naturally: for any fuzzy interpretation “•” of **L** formulas, we set the estimates $\varphi \ge a$ and $\varphi \le a$ to be true (i.e., “$\varphi \ge a$” = 1 and “$\varphi \le a$ “ = 1) if and only if “φ” $\ge a$ and “φ “ $\le a$. Let **B** be the set of all Boolean combinations of estimates, i.e., (*i*) every estimate belong to **B**; (*ii*) $\sim\alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha{\to}\beta \in$ **B** if $\alpha, \beta \in$ **B**.

We define the *logic of estimates* **E-L** assuming that **B** is the set of its sentences, and extending interpretations “•” of **L** formulas to **B** in accordance with the meaning of Boolean operations: “$\sim\alpha$” = $\sim$“α”, “$\alpha \wedge \beta$” = “α”$\wedge$“β”, “$\alpha \vee \beta$” = “α”$\vee$“β”, “$\alpha{\to}\beta$” = “α”$\to$“β”.

As each logic, **E-L** induces the relation ‘|=’ of logical consequence. Let E be a set of **E-L** sentences and α be a **E-L** sentence. Then α is the *logical consequence* of the set E ($E \models \alpha$) if and only if there is no interpretation “•” such that α is false in this interpretation and all sentences from E are true (“β” = 0 for all $\beta \in E$).

In this paper we consider two fuzzy logic **L**: the Zadeh's fuzzy propositional logic **Z** and the fuzzy linear temporal logic **FLTL**. We will build a complete and sound inference system for the logics of estimates **E-Z** and will describe briefly how to build such a system for the logic **E-FLTL**. The inference systems consist of the rules acting in the style of analytic tableaux [1]. Let us remind that an inference system is *complete* if the logical consequence relation ‘|=’ implies the inference relation ‘|–’. An inference system is *sound* if ‘|–’ implies ‘|=’

Remark 1. By Zadeh' logic we mean the fuzzy proposition logic with the Boolean operations and with the usual meaning of these operations: ‘$\sim x$’ = $1-x$, ‘$x \wedge y$’ = $\min\{x,y\}$, ‘$x \vee y$’ = $\max\{x,y\}$, ‘$x \to y$’ = $\max\{1-x, y\}$. Zadeh's logic relates to fuzzy sets in the same way Boolean logic relates to crisp sets [7].

Russian Advances in Fuzzy Systems and Soft Computing: selected contributions to the 8-th International Conference on Fuzzy Systems, Soft Computing and Intelligent Technologies (FSSCIT-2020), June 29 – July 1, 2020, Smolensk, Russia

EMAIL: salve777@mail.ru

Remark 2. Earlier, the logic of estimates for Zadeh's logic has been considered by J. Chen and S. Kundu [2, 4]. They have built a sound and complete inference system for this logic. The system of Chen-Kundu is based on generalized resolution.

One of the applications of logics of estimates **E-L** is related to query answering over knowledge bases. Estimates of the forms $p \ge a$ and $p \le a$ with propositional variables p are called *elementary.* Elementary estimates are considered as *facts.* A finite set ***F*** of facts is called *fact base.* A finite set ***O*** of non-elementary estimates from **E-Z** is called *ontology*. A finite set ***F*** of facts is called a *fact base.* A *knowledge base* consists in an ontology plus a fact base: ***Kb*** = ***O*** ∪***F***.

We will consider *queries* to a knowledge base. We write a query to ***Kb*** in the following form: ? (max x, min y) -- $x \le \varphi \le y$, where φ is **E-Z** formula The query reads as follows: "Find the maximum value of the variable x and the minimum value of the variable y such that the estimates $\varphi \ge x$ and $\varphi \le y$ follow from ***Kb***". We show, by example, how to evaluate queries to a knowledge base written in the logic **E-Z**. For this the inference system for the logic **E-Z** is used.

2. Inference in the logic E-Z

The following lemma contains basic equivalences of the logic **E-Z**. They will be used in proving soundness of the inference system for the logic **E-Z**.

Lemma 1. Let φ and ψ are sentences of the logic **E-Z**. and $a, b \in [0,1]$. The following equivalences are true :

$\sim\varphi \ge a \equiv \varphi \le 1-a,$

$\sim\varphi \le a \equiv \varphi \ge 1-a,$ (2.1)

$\varphi \wedge \psi \ge a \equiv (\varphi \ge a) \wedge (\psi \ge a),$

$\varphi \wedge \psi \le a \equiv (\varphi \ge a) \vee (\psi \ge a),$

$\varphi \vee \psi \ge a \equiv (\varphi \ge a) \vee (\psi \ge a),$

$\varphi \vee \psi \le a \equiv (\varphi \le a) \wedge (\psi \le a),$ (2.2)

$(\varphi \ge a) \wedge (\varphi \ge b) \equiv \varphi \ge \max\{a,b\},$ (2.3)

$(\varphi \le a) \wedge (\varphi \le b) \equiv \varphi \le \min\{a,b\},$

Here are the proofs of the equivalences (2.1), (2.2) и (2.3):

- "$\sim\varphi \le a$" $= 1 \Leftrightarrow$ "$\sim\varphi$" $\le a \Leftrightarrow 1-$"φ" $\le a \Leftrightarrow$ "φ" $\ge 1-a \Leftrightarrow$ "$\varphi \ge 1-a$" $= 1$ for all interpretation "•". Hence $\sim\varphi \le a \equiv \varphi \ge 1-a$.
- "$\varphi \vee \psi \le a$" $= 1 \Leftrightarrow$ "$\varphi \vee \psi$" $\le a \Leftrightarrow$ "φ" ∨ "ψ" $\le a \Leftrightarrow$ "φ" $\le a$ and "ψ" $\le a$ (since $\max\{x,y\} \le a \Leftrightarrow x \le a$ and $y \le b$) $\Leftrightarrow$ "$\varphi \le a$" $= 1$ and "$\psi \le b$" $= 1 \Leftrightarrow$ "$\varphi \le a$" ∧ "$\psi \le b$" $= 1 \Leftrightarrow$ "$\varphi \le a$" ∧ "$\psi \le b$" $\Leftrightarrow$ "$(\varphi \le a) \wedge (\psi \le a)$" $= 1$ for all interpretation "•". Hence $\varphi \vee \psi \le a \equiv (\varphi \le a) \wedge (\psi \le a)$.
- "$(\varphi \ge a) \wedge (\varphi \ge b)$" $= 1 \Leftrightarrow$ "$\varphi \ge a$" $= 1$ and "$\varphi \ge b$" $= 1 \Leftrightarrow$ "φ" $\ge a$ and "φ" $\ge b \Leftrightarrow$ "φ" $\ge \max\{a, b\} \Leftrightarrow$ "$\varphi \ge \max\{a, b\}$" $= 1$ for all interpretation "•". Hence $(\varphi \ge a) \wedge (\varphi \ge b) \equiv \varphi \ge \max\{a,b\}$,

The tables Table 1, Table 2 and Table 3 show the inference rules by analytic tableaux method for the logic **E-Z**.

Propositional connectives in **E-Z** sentences are on two levels: external and internal. Table 1 gives the rules for dealing with connectives on the external level. In the rules of Table 2, the connectives are at the internal level. At the external level, formulas are supplied with '+' or '–' signs. When a formula is marked with a "+", this means that it is true (in the intended interpretation); the "–" sign indicates its falsity.

Table 1

Inference rules for signed sentences

Rule number	Antecedent	Consequents
1	$+ \sim \varphi$	$- \varphi$
2	$- \sim \varphi$	$+ \varphi$
3	$+ \varphi \wedge \psi$	$+ \varphi$ and $+ \psi$
4	$- \varphi \wedge \psi$	$- \varphi$ or $- \psi$
5	$+ \varphi \vee \psi$	$+ \varphi$ or $+ \psi$
6	$- \varphi \vee \psi$	$- \varphi$ and $- \psi$
7	$+ \varphi \rightarrow \psi$	$- \varphi$ or $+ \psi$

8	$-\varphi \rightarrow \psi$	$+\varphi$ and $-\psi$
9	$+\rho$	$\rho \geq 1$
10	$-\rho$	$\rho \leq 0$
11	$+\lambda$	λ
12	$-\lambda$	$\sim \lambda$

φ and ψ are **E-Z** sentences, λ is an estimate, ρ is a **Z** sentence

Table 2
Inference rules for sentences with inequlities signs

Rule number	Antecedent	Consequents
1	$\sim \varphi \geq a$	$\varphi \leq 1-a$
2	$\sim \varphi \leq a$	$\varphi \geq 1-a$
3	$\varphi \wedge \psi \geq a$	$\varphi \geq a$ and $\psi \geq a$
4	$\varphi \wedge \psi \leq a$	$\varphi \leq a$ or $\psi \leq a$
5	$\varphi \vee \psi \geq a$	$\varphi \geq a$ or $\psi \geq a$
6	$\varphi \vee \psi \leq a$	$\varphi \leq a$ and $\psi \leq a$
7	$\varphi \rightarrow \psi \geq a$	$\varphi \leq 1-a$ or $\psi \geq a$
8	$\varphi \rightarrow \psi \leq a$	$\varphi \leq 1-a$ and $\psi \geq a$

Table 3
Inference rules for sentences with negations

Rule number	Antecedent	Consequents
1	$\sim (\sim \varphi \geq a)$	$\sim (\varphi \leq 1-a)$
2	$\sim (\sim \varphi \leq a)$	$\sim (\varphi \geq 1-a)$
3	$\sim (\varphi \wedge \psi \geq a)$	$\sim (\varphi \geq a)$ or $\sim (\psi \geq a)$
4	$\sim (\varphi \wedge \psi \leq a)$	$\sim (\varphi \leq a)$ and $\sim (\psi \leq a)$
5	$\sim (\varphi \vee \psi \geq a)$	$\sim (\varphi \geq a)$ and $\sim (\psi \geq a)$
6	$\sim (\varphi \vee \psi \leq a)$	$\sim (\varphi \leq a)$ or $\sim (\psi \leq a)$
7	$\sim (\varphi \rightarrow \psi \geq a)$	$\sim (\varphi \leq 1-a)$ and $\sim (\psi \geq a)$
8	$\sim (\varphi \rightarrow \psi \leq a)$	

Table 4
Binary inference rules

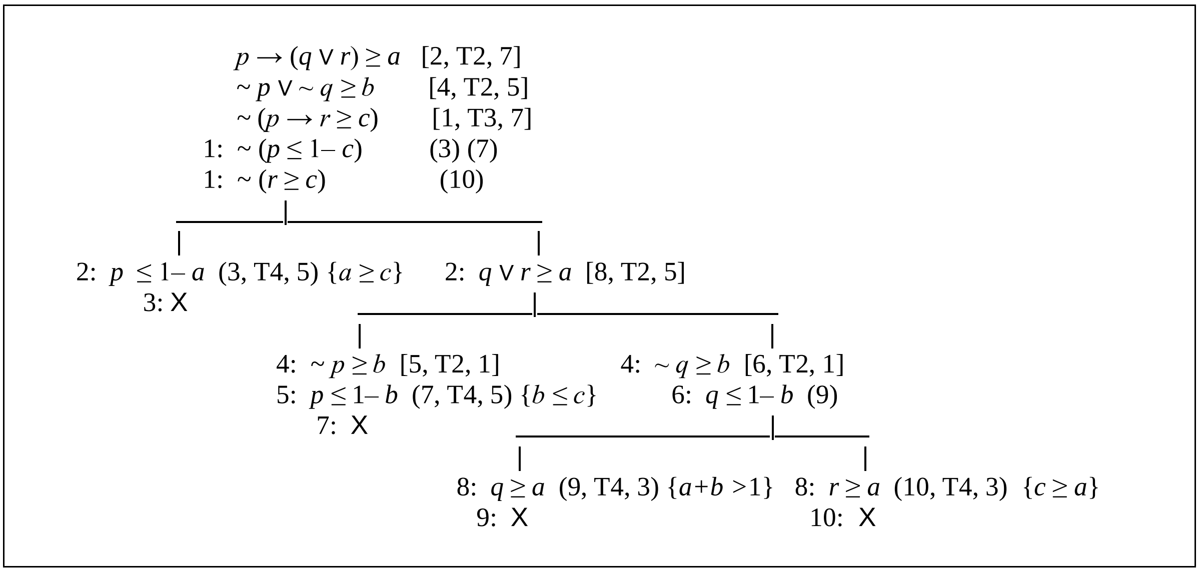
Rule number	Antecedent	Consequents
1	$+\varphi,\ -\varphi$	X
2	$\lambda,\ \sim \lambda$	X
3	$\varphi \geq a,\ \varphi \leq b$ if $a > b$	X
4	$\varphi \geq a,\ \sim (\varphi \geq b)$ if $b \leq a$	X
5	$\sim (\varphi \leq a),\ \varphi \leq b$ if $b \leq a$	X
6	$\sim (\varphi \leq a),\ \sim (\varphi \geq b)$ if $b \leq a$	X
7	$\varphi \leq a,\ \varphi \leq b$	$\varphi \leq \min\{a,b\}$
8	$\varphi \leq a,\ \varphi \leq b$	$\varphi \geq \max\{a,b\}$

Semantics of the rules from Table 1 corresponds to the usual meaning of logical propositional connectives. Each rule is sound in the sense that in any interpretation, if the antecedent of the rule is true, then its consequents are also true in that interpretation. The inference rules from Table 1 coincide with the standard rules of the analytical tableaux method for propositional logic, if we consider φ and ψ to be formulas of classical propositional logic [1, 4]. But it is assumed that φ and ψ in Table 1 are sentences of the logic **E-Z**.

When applying the method of analytical tableaux, inferences are presented in the form of trees, the vertices of which are logical sentences with labels, and the edges are determined from the applied inference rules.

Example 1. This is an example of building an inference tree for recognizing logical consequences from knowledge bases.

Consider the knowledge base ***Kb*** = $\{p \to (q \vee r) \geq a,\ {\sim} p \vee {\sim} q \geq b\}$ and the estimate $p \to r \geq c$ with unknowns a, b and c. Let us decide the problem of finding such relations between these unknowns that there is a logical consequence ***Kb*** $\models p \to r \geq c$. This is the case if and only if the extended knowledge base ***Kb′*** = ***Kb*** $\cup \{{\sim} (p \to r \geq c)\}$ is inconsistent.



$p \to (q \vee r) \geq a$ [2, T2, 7]
${\sim} p \vee {\sim} q \geq b$ [4, T2, 5]
${\sim} (p \to r \geq c)$ [1, T3, 7]
1: ${\sim} (p \leq 1- c)$ (3) (7)
1: ${\sim} (r \geq c)$ (10)
2: $p \leq 1- a$ (3, T4, 5) $\{a \geq c\}$
3: X
2: $q \vee r \geq a$ [8, T2, 5]
4: ${\sim} p \geq b$ [5, T2, 1]
5: $p \leq 1- b$ (7, T4, 5) $\{b \leq c\}$
7: X
4: ${\sim} q \geq b$ [6, T2, 1]
6: $q \leq 1- b$ (9)
8: $q \geq a$ (9, T4, 3) $\{a+b >1\}$
9: X
8: $r \geq a$ (10, T4, 3) $\{c \geq a\}$
10: X

Figure 1: Inference tree from Example 1

We begin to build the inference tree with an initial branch been the sequence of estimates from E (see Figure 1). Then we may to apply some inference rule to any of member from the initial branch. We take the estimate ${\sim} (p \to r) \geq c$ (the third estimate in the initial branch) to which we add the label [1, T3, 7] at right. The label indicates that at step 1, the rule number 7 from Table 3 was applied. As a result of this application, two vertices with estimates ${\sim} (p \leq 1- c)$ and ${\sim} (r \geq c)$ are added to the initial branch, one after the other. These vertices are supplied by the left label '1:' which indicate that the estimates were obtained at the step 1. Our choice of the third estimate for rule application due to the fact that the rule 7 is conjunctive (i.e. its consequents are linked by the connective "and") while the disjunctive rules are applicable to other vertices of the initial branch. (When building inference trees, the priority of disjunctive rules is used, since this leads to more economical trees.)

Consider the vertex '2: $p \leq 1- a$ (3, T4, 5)'. It contains the right label (3, T4, 5) which indicates that the binary rule 5 of Table 4 was applied to the estimate $p \leq 1- a$ and to other estimate (since rule 5 is binary). This other estimate is ${\sim} (p \leq 1- c)$ with right label (3). The result of the rule application is 'X' which means contradiction and is attached to the first current branch of the inference tree. Thus, all four branches of the tree are inconsistent sets. Hence, the set E is inconsistent (due to soundness of the inference rules of the tables (besides the rules 1– 6 of Table 4).

There is also the second form of inference trees (see Fig.2). For example, the second form of inference trees is used in the book [3] for exposition of tableaux methods.

A branch of an inference tree is *closed* if it ends with the symbol X. An inference tree is *closed* if all its branches are closed. A branch is *completed* if it closed or if some inference rule was applied to each of non-atomic sentence of the branch. A tree is *completed* if all its branches are *completed.*

Theorem 1. The system of inference rules for the logic **E-Z** is sound and complete. The following statements hold:

- If a knowledge base is inconsistent then every completed inference tree that for this knowledge base is closed;
- A knowledge base is inconsistent if it has a closed inference tree built for this knowledge base.

The proof of this theorem is similar to the proof of the theorem about soundness and completeness of the standard tableaux system for propositional logic [4]. In particular, the proof uses structural inductions and Hintikka's sets [2].

3. Query answering over knowledge bases in E-Z

Let's look at an example of how you can find by the method analytical tableaux answers to queries to the knowledge base written in the logic **E-Z.**

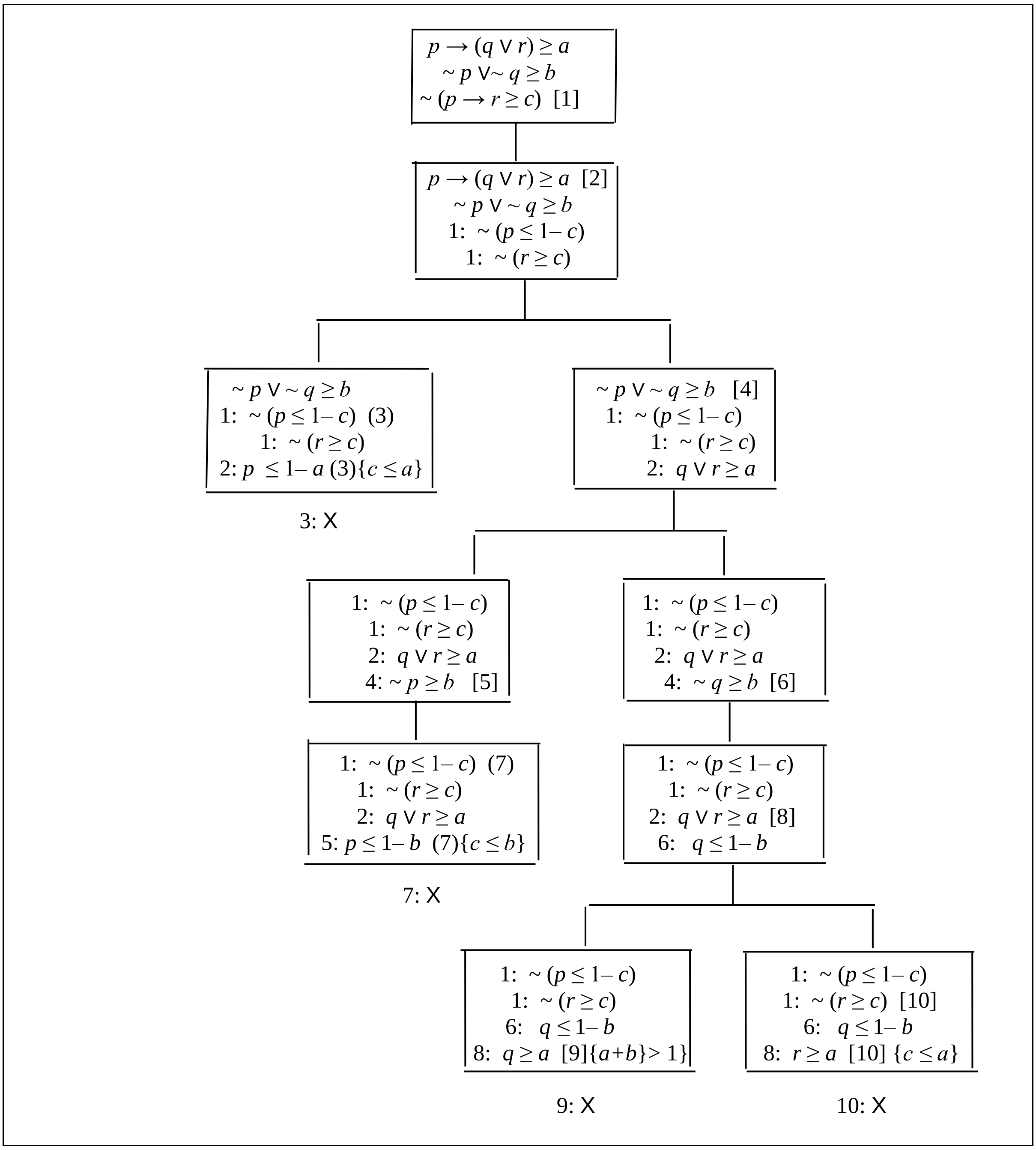


Figure 2: The second form of inference tree

Example 2. Consider the problem of medical diagnosis in a situation where there are 2 diseases q_1, q_2 and 3 symptoms p_1, p_2, p_3. Suppose the following knowledge establishes the relation between these diseases and symptoms:

(a) if the disease q_2 occurs, then with a degree of confidence ≤ 0.5 there can be no disease q_1.

(b) the disease q_1 is uniquely determined with the confidence degree ≥ 0.6 by the presence of the symptoms p_3, p_4, and by the absence of symptom p_1;

(c) if the disease q_2 occurs, then the symptom p_1 is observed but there is no symptom p_3.

Suppose symptom p_1 is observed. We are interested in the following queries:

(d1) Find the best lower and upper estimates for disease q_1;

(d2) Find the best lower and upper estimates for disease q_1.

In the logic of **E-Z**, this knowledge and queries are represented by the following **E-Z** sentences:

- $\sim q_3 \rightarrow (\sim q_1 \geq 0.5)$ for (a);
- $q_1 \rightarrow \sim p_1 \wedge p_3 \wedge p_4 \geq 0.6$ and $\sim p_1 \wedge p_3 \wedge p_4 \rightarrow q_2 \geq 0.6$ for (b);
- $q_2 \rightarrow p_1 \wedge \ \sim p_2 \geq 0.3$ for (c);
- Q1: ? (max x, min y) – $x \ \leq q_1 \leq y$ for (d1);
- Q2: ? (max x, min y) – $x \ \leq q_2 \leq y$ for (d2).

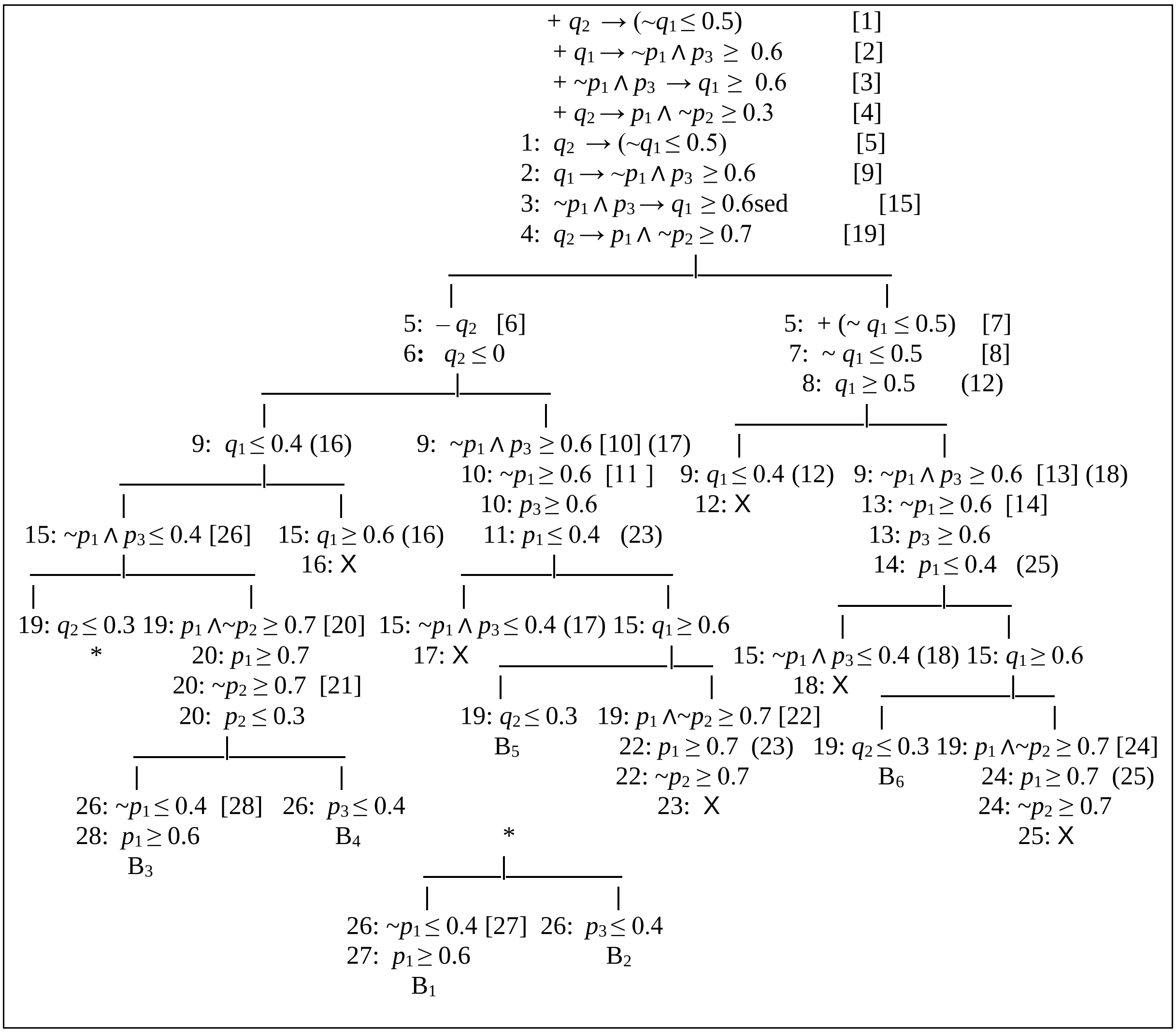
It is obvious that this two queries can be replace on fourth queries:

Q11: ? max x – $q_1 \geq x$, Q12: ? min y – $q_2 \leq y$, Q21: ? min x – $q_2 \geq x$, we Q12: ? min y – $q_2 \leq y$.

So, we have the knowledge base

Kb = $\{\sim q_3 \rightarrow \sim q_1 \geq 0.5,\ q_1 \rightarrow \sim p_1 \wedge p_3 \wedge p_4 \geq 0.6,\ \sim p_1 \wedge p_3 \wedge p_4 \rightarrow q_2 \geq 0.6,\ q_2 \rightarrow p_1 \wedge \ \sim p_2 \geq 0.3\}$

Suppose we observed the symptom p_1. Then we have the fact $p_1 \geq 1$, and we f the fact base $\boldsymbol{F} = \{p_1 \geq 1\}$. If we address to ***Kb*** $\cup$ ***F*** with the query Q11 then the answer to that query will be the estimate $q_1 \geq a$ such that ***Kb*** $\cup$ ***F*** **|=** $q_1 \geq a$ and ***Kb*** $\cup$ ***F*** $|\neq q_1 \geq b$ if $b > a$. To obtain this answer, we build the inference tree for the set +***Kb*** = {+ α **|** α ∈ ***Kb***} (see Figure 3).



$+ q_2 \rightarrow (\sim q_1 \leq 0.5)$ [1]
$+ q_1 \rightarrow \sim p_1 \wedge p_3 \geq 0.6$ [2]
$+ \sim p_1 \wedge p_3 \rightarrow q_1 \geq 0.6$ [3]
$+ q_2 \rightarrow p_1 \wedge \sim p_2 \geq 0.3$ [4]
1: $q_2 \rightarrow (\sim q_1 \leq 0.5)$ [5]
2: $q_1 \rightarrow \sim p_1 \wedge p_3 \geq 0.6$ [9]
3: $\sim p_1 \wedge p_3 \rightarrow q_1 \geq 0.6$sed [15]
4: $q_2 \rightarrow p_1 \wedge \sim p_2 \geq 0.7$ [19]
5: $- q_2$ [6]
6: $q_2 \leq 0$
5: $+ (\sim q_1 \leq 0.5)$ [7]
7: $\sim q_1 \leq 0.5$ [8]
8: $q_1 \geq 0.5$ (12)
9: $q_1 \leq 0.4$ (16)
9: $\sim p_1 \wedge p_3 \geq 0.6$ [10] (17)
10: $\sim p_1 \geq 0.6$ [11]
10: $p_3 \geq 0.6$
11: $p_1 \leq 0.4$ (23)
9: $q_1 \leq 0.4$ (12)
12: X
9: $\sim p_1 \wedge p_3 \geq 0.6$ [13] (18)
13: $\sim p_1 \geq 0.6$ [14]
13: $p_3 \geq 0.6$
14: $p_1 \leq 0.4$ (25)
15: $\sim p_1 \wedge p_3 \leq 0.4$ [26]
15: $q_1 \geq 0.6$ (16)
16: X
19: $q_2 \leq 0.3$
*
19: $p_1 \wedge \sim p_2 \geq 0.7$ [20]
20: $p_1 \geq 0.7$
20: $\sim p_2 \geq 0.7$ [21]
20: $p_2 \leq 0.3$
26: $\sim p_1 \leq 0.4$ [28]
28: $p_1 \geq 0.6$
B$_3$
26: $p_3 \leq 0.4$
B$_4$
15: $\sim p_1 \wedge p_3 \leq 0.4$ (17)
17: X
15: $q_1 \geq 0.6$
19: $q_2 \leq 0.3$
B$_5$
19: $p_1 \wedge \sim p_2 \geq 0.7$ [22]
22: $p_1 \geq 0.7$ (23)
22: $\sim p_2 \geq 0.7$
23: X
*
26: $\sim p_1 \leq 0.4$ [27]
27: $p_1 \geq 0.6$
B$_1$
26: $p_3 \leq 0.4$
B$_2$
15: $\sim p_1 \wedge p_3 \leq 0.4$ (18)
18: X
15: $q_1 \geq 0.6$
19: $q_2 \leq 0.3$
B$_6$
19: $p_1 \wedge \sim p_2 \geq 0.7$ [24]
24: $p_1 \geq 0.7$ (25)
24: $\sim p_2 \geq 0.7$
25: X

Figure 3: Inference tree from Example 1

The tree has 5 closed branches and 6 open branches: B$_i$ ($1 \le i \le 6$). From each branch B$_i$ we write out all elementary estimates and form the conjuncts:

$C_1 = (p_1 \ge 0.6) \wedge (q_1 \le 0.4) \wedge (q_2 \le 0)$,
$C_2 = (p_3 \le 0.4) \wedge (q_1 \le 0.4) \wedge (q_2 \le 0)$,
$C_3 = (p_1 \ge$ then $0.7) \wedge (p_2 \le 0.3) \wedge (q_1 \le 0.4) \wedge (q_2 \le 0)$,
$C_4 = (p_1 \ge 0.7) \wedge (p_2 \le 0.3) \wedge (p_3 \le 0.4) \wedge (q_1 \le 0.4) \wedge (q_2 \le 0)$,
$C_5 = (p_1 \le 0.4) \wedge (p_3 \ge 0.6) \wedge (q_1 \ge 0.6) \wedge (q_2 \le 0.3)$,
$C_6 = (p_1 \ge 0.4) \wedge (p_3 \le 0.4) \wedge (q_1 \ge 0.6) \wedge (q_2 \le 0.3)$.

For any set E of sentences, we denote E^ the conjunction of all members of E: $E\hat{} = \bigwedge\{\alpha \mid \alpha \in E\}$. It is easy to understand that the **E-Z** sentence ***Kb***^ is equivalent to DNF $C_1 \vee C_2 \vee C_3 \vee C_4 \vee C_5 \vee C_6$. Since $C_1 \vee C_3 \equiv C_1$ and $C_2 \vee C_4 \equiv C_2$ we may delete C_3 and C_4 from the DNF. So, ***Kb***^ is equivalent to DNF $C_1 \vee C_2 \vee C_5 \vee C_6$. If we take the fact base $\boldsymbol{F} = \{p_1 \le 0, p_3 \ge 1\}$ then we will have for q_1

$$(\boldsymbol{Kb} \cup \boldsymbol{F})\hat{} = \boldsymbol{Kb}\hat{} \wedge \boldsymbol{F}\hat{} = (C_1 \vee C_2 \vee C_5 \vee C_6) \wedge (p_1 \le 0) \wedge (p_3 \ge 1) = C_5 \wedge (p_1 \le 0) \wedge (p_3 \ge 1) =$$
$$(p_1 \le 0.4) \wedge (p_3 \ge 0.6) \wedge (q_1 \ge 0.6) \wedge (q_2 \le 0.3) \wedge (p_1 \le 0) \wedge (p_3 \ge 1),$$
$$\boldsymbol{Kb} \cup \boldsymbol{F} \models q_1 \ge x \Leftrightarrow (\boldsymbol{Kb} \cup \boldsymbol{F})\hat{} \models q_1 \ge x \Leftrightarrow (\boldsymbol{Kb} \cup \boldsymbol{F})\hat{} \wedge \sim (q_1 \ge x) \equiv 0 \Leftrightarrow$$
$$(p_1 \le 0.4) \wedge (p_3 \ge 0.6) \wedge (q_1 \ge 0.6) \wedge (q_2 \le 0.3) \wedge (p_1 \le 0) \wedge (p_3 \ge 1) \wedge (q_1 < x) \equiv 0 \Leftrightarrow$$
$$(q_1 \ge 0.6) \wedge (q_1 < x) \equiv 0.$$

Hence, $\max\{x \mid \boldsymbol{Kb} \cup \boldsymbol{F} \models q_1 \ge x\} = \max\{ \mid (q_1 \ge 0.6) \wedge (q_1 < x) \equiv 0\} = 0.6$. Also we have

$$\boldsymbol{Kb} \cup \boldsymbol{F} \models q_1 \le y \Leftrightarrow (\boldsymbol{Kb} \cup \boldsymbol{F})\hat{} \models q_1 \le y \Leftrightarrow (\boldsymbol{Kb} \cup \boldsymbol{F})\hat{} \wedge \sim (q_1 \le y) \equiv 0 \Leftrightarrow$$
$$(p_1 \le 0.4) \wedge (p_3 \ge 0.6) \wedge (q_1 \ge 0.6) \wedge (q_2 \le 0.3) \wedge (p_1 \le 0) \wedge (p_3 \ge 1) \wedge (q_1 > y) \equiv 0.$$

The last expression is false for any $y \in [0, 1]$. Hence, $\min\{y \mid \boldsymbol{Kb} \cup \boldsymbol{F} \models q_1 \le y\} = \min\{x \mid 0\} = \min \emptyset = 1$.

Similarly, we obtain for q_2: $\max\{x \mid \boldsymbol{Kb} \cup \boldsymbol{F} \models q_2 \ge x\} = 0$ and $\min\{y \mid \boldsymbol{Kb} \cup \boldsymbol{F} \models q_1 \le y\} = 0.3$.

Thus, the answer to the query Q1 is $0.6 \le q_1 \le 1$, and answer to the query Q2 is $0 \le q_2 \le 0.3$.

4. Inference in the logic E-FLTL

Linear temporal logic **LTL** is the basic temporal logic intended for representing and analysis of state transition systems [5]. LTL has founded extensive applications in many domains, especially, in computer science.

4.1. Syntax of LTL

The *syntax* of **LTL** is defined like the syntax of propositional logic except for addition of three unary operators ɳ, ɑ and e which are read "next", "always" and "eventually" (correspondingly), and binary operator $\mathcal{U}$ which is read "until".

LTL sentences are defines from *atoms* which are propositional variables and constants:

- **At** ⊆ **LTL**;
- $\sim\varphi$, ɳφ, eφ, ɑφ ∈ **LTL** if φ ∈ **LTL**
- $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, $\varphi \,\mathcal{U}\, \psi$ ∈ **LTL** if φ, ψ ∈ **LTL**.

Here **At** denotes a set of atoms and **LTL** denotes the set of definable sentences of the logic **LTL**. Also we denote by **Lit** the set of corresponding literals: **Lit** = $\{\alpha, \sim\alpha \mid \alpha \in$ **LTL**$\}$. If E is any set of **LTL** sentences then the **At**(E) and **Lit**(E) denote the sets of atoms and literals entered the sentences of E.

Interpretations of **LTL** sentences are based on state transition diagrams. The intuitive meaning is that each state represents a "*world* " (or "*state of affairs*") and a sentence can have different truth values in different worlds. Transitions represent changes from world to another.

A *state* (for inferences from E) is any subset of **Lit**(E). For a state s and a literal λ, we write $s(\lambda) = 1$ if $\lambda \in s$ and $s(\lambda) = 0$ if $\lambda \in s$. A *state transition graph* is a directed graph Γ whose vertices are states and edges are pairs of states denoting transitions. A *track* σ is an infinite path in graph Γ: $\sigma = s_0 s_1 s_2\ldots$. (Tracks exist in Γ if and only if there are cycles in Γ.) For any track σ and integer $i \ge 0$, we denote $\sigma_i = s_i s_{i+1} s_{i+2}\ldots$. Let **T**(Γ) denote the set of all tracks in the graph Γ. We call a *box* any finite set of **LTL** sentences. (We choose the name ‘box’ due its use in pictures of inference trees as in Figure 2.) Thus, states are particular cases of boxes.

A *valuation* is a function “•”: **LTL** X T(Γ) →{0,1}. *Interpretations* are defined as an extension of valuations:

- “φ, σ” = s_0(φ) if φ is an atom, φ ∈ **At**;
- “~φ, σ” = ~“φ”;
- “ɳ φ, σ” = “φ, $σ_1$”;
- “℮ φ, σ” = ∃*i*. “φ, $σ_i$ ”;
- “ɑ φ, σ” = ∀*i*. “φ, $σ_i$ ”;
- “φ ∧ ψ, σ” = “φ, σ” ∧ “ψ, σ”;
- “φ ∨ ψ, σ” = “φ, σ” ∨ “ψ, σ”;
- “φ → ψ, σ” = “φ, σ” → “ψ, σ”;
- “φ 𝓤 ψ, σ” = sup[“ψ, $σ_i$” ∧ (∀*k*: 0 ≤ *k* < *i*) “φ, $σ_k$”]

The inference rules for **LTL** are those in Table 1 but without ‘+’ and ‘–’ signs, and five rules with number 1 – 5 in Table 5 . The soundness of the rules 1-5 are follows from the equivalences:

ɳ ɑ φ ≡ φ ∧ ɳ ɑ φ, ℮ φ ≡ φ ∨ ɳ ℮ φ, φ 𝓤 ψ ≡ ψ ∨ (φ ∧ ɳ (φ 𝓤 ψ)).

It easy to show that these equivalences are true.

There is a peculiarity of applying the first and second rules from Table 5 (i.e., the rules with the sign ‘ɳ’). Formulas of the form ɳ *p* and ~ ɳ *p* (where *p* is a propositional variable from **At**) are called *X*-formulas. If the rule 1 is applied to the formula ɳ *p* when it belongs to some box α in the inference tree such that (α \ {~ ɳ *p*}) ∪ {*p*} is coincide with other box β in the current inference tree, then we connect α with β by an edge.

Remark. The described inference method proposed by Ben-Free in the book [2].

Table 5

Inference rules for sentences with the operations ɑ, ℮ and U

Rule number	Antecedent	Consequents
1	ɳ φ	φ
2	~ ɳ φ	~ φ
1	ɑ φ	φ and ɳ ɑ φ
2	~ ɑ φ	~ φ or ~ ɳ ɑ φ
3	℮ φ	φ or ɳ ℮ φ
4	~ ℮ φ	~ φ and ~ ɳ ℮ φ
5	φ 𝓤 ψ	ψ or φ ∧ ɳ (φ 𝓤 ψ)
6	~ φ 𝓤 ψ	~ ψ and ~ φ ∨ ~ ɳ (φ 𝓤 ψ)
7	ɑ φ ≥ *a*	φ ≥ *a* and ɳ ɑ φ ≥ *a*
8	ɑ φ ≤ *a*	φ ≤ *a* or ɳ ɑ φ ≤ *a*
9	℮ φ ≥ *a*	φ ≥ *a* or ɳ ℮ φ ≥ *a*
10	℮ φ ≤ *a*	φ ≤ *a* and ɳ ℮ φ≤ *a*
11	φ 𝓤 ψ ≥ *a*	ψ ≥ *a* or φ ∧ ɳ (φ 𝓤 ψ) ≥ *a*
12	φ 𝓤 ψ ≤ *a*	ψ ≤ *a* and φ ∧ ɳ (φ 𝓤 ψ) ≤ *a*

4.2. Semantics of FLTL

Fuzzy linear temporal logic **FLTL** has the same syntax as **LTL**. Interpretations in **FLTL** differs from ones in **LTL** in that {0,1} is replaced by [0,1]. Their definitions are the following:

- “φ, σ” = s_0(φ) if φ is a literal, φ ∈ **Lit**;
- “~φ, σ” = 1– “φ, σ”;

- “ɳ φ, σ” = “φ, σ_1”;
- “e φ, σ” = sup{“φ, σ_i” | $i = 0, 1, 2,\ldots$};
- “e φ, σ” = inf{“φ, σ_i” | $i = 0, 1, 2,\ldots$};
- “φ ∧ ψ, σ” = min{“φ, σ”, “ψ, σ”};
- “φ ∨ ψ, σ” = max{“φ, σ”, “ψ, σ”};
- “φ → ψ, σ” = max{1– “φ, σ”, “ψ, σ”};
- “φ $\mathcal{U}$ ψ, σ” = sup{min{“ψ, σ_i”, ($\forall k$: $0 \leq k < i$) “φ, σ_k”}}.

4.3. Inference rules for the logic E-FLTL

The inference rules for **E-FLTL** are presented in Table 1 – Table 4 and Table 5 (rules 7-12).

Theorem 2. The system of inference rules for the logic **E-FLTL** is sound and complete. The following statements hold:

- If a knowledge base is inconsistent then every completed inference tree that is bult for this knowledge base is closed;
- A knowledge base is inconsistent if it has a closed inference tree built for this knowledge base.

5. Conclusions

Logics **E-L** of estimates are important because experts who use a fuzzy logic **L** in the development of knowledge-based systems often do not know the exact values of the degrees of fuzziness for modelled concepts but be able only to indicate some boundaries of these values. The logic **E-L** can be used for manipulating the estimates to get new estimates for logically derived concepts. In this paper we described the sound and complete inference systems for the logics **E-Z** and **FLTL** where **Z** is the Zadeh’s fuzzy propositional logic and **FLTL** is the fuzzy linear temporal logic.

6. Acknowledgment

This work was supported by Russian Foundation for Basic Research (projects 20-07-00615, 18-29-03088 and 20-07-00770).

7. References

[1] M. Agostino, D. Gabbay, R. Hahnle, J. Posegga (Eds.), Handbook of tableaux methods. Springer, 2001.

[2] M. Ben-Ari, Mathematical logic for computer science, Third edition, Springer, 2012.

[3] J. Chen, S. Kundu, A sound and complete fuzzy logic system using Zadeh’s implication operator, in Z. W. Ras, M. Maciek (Eds.), Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, 1996, 233-242.

[4] M. Fitting, First-order logic and automated theorem proving. Springer, 1996.

[5] F. Kroger, S. Merz, Temporal logic and state systems, Springer, 2008.

[6] S. Kundu, An Improved Method for Fuzzy-Inferencing Using Zadeh’s implication operator, in Proceedings of IJCAL Workshop on Fuzzy Logic in Artificial Intelligence, 1995, 117-125.

[7] L. A. Zadeh, Fuzzy sets, Information and Control, 8 (1965) 338-353.