Appl. Math. and Comp. Sci., 1997, Vol.7, No.3, 689–700

TOWARDS SIMILARITIES OF INTERPRETATIONS OF TEMPORAL LOGIC FORMULAE

RADOSŁAW KLIMEK*

The paper deals with similarities of interpretations of temporal logic formulae. Two basic similarities are defined and some theorems towards these similarities are given. An application of similarities is presented based on the example of a typical car ferry. The analysis of the system employs the model checking method.

1. Introduction

Temporal logic is one of the most important formalisms used while inferring in software systems. The system properties are proved by the syntax or semantic methods. The syntax method assumes that the desired formulae are derived from a set of basic formulae through a well-known set of logic axioms and laws. On the other hand, the semantic method assumes examining whether a formula is satisfied in some legal interpretations. Every temporal logic formula may have infinitely many interpretations. Interpretations may differentiate between one another in a radical way but they may also show a similarity which could disregard the differences and then we can ascertain that some interpretations are identical in practice. In this way the whole process of proving and examining the system properties may become easier.

The aim of this paper is to define the basic similarities of temporal logic interpretations and to present their properties. The equivalence as defined in (Wolper, 1987) seems to be insufficient for a semantical analysis of systems and it is worthwile to define its generalization. It is possible to extend the class of interpretations which could be considered as similar ones. It also enables us to manipulate interpretations. The paper is organized as follows. After a short presentation of the temporal logic, two types of similarities of interpretations are defined. An origin of these similarities is also mentioned. The next section deals with some related theorems. A typical car ferry is considered as an example. The model checking method is employed. It seems that this method is the most appropriate one for the semantic examination of a system. The last section contains conclusions.

* Institute of Automatics, University of Mining and Metallurgy, al. Mickiewicza 30, 30–059 Kraków, Poland, e-mail: rklimek@ia.agh.edu.pl.

2. Preliminaries (Temporal Logic)

In this section the classical *temporal logic* is briefly presented. In order to present the *syntax* of the temporal logic, we have to introduce the notion of a set of acceptable symbols (alphabet) and the notion of inductive rules producing acceptable formulae (production rules).

Definition 1. The *alphabet* of temporal logic formulae consists of the following sets:

- A set of *atomic formulae* P;
- Classical logic symbols: **true**, **false**, $\neg$, $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$;
- Temporal logic operators: $\Box$, $\Diamond$, $\mathcal{U}$.

The power of the set of symbols in the case of both classical and temporal logic is greater than necessary. For example, in the case of the temporal logic the operator $\mathcal{U}$ is sufficient since the others can be introduced as its abbreviations. Besides the symbols listed above, it is also possible to add the parenthesis symbols in order to change a priority of calculations or to remove ambiguities.

Definition 2. The *production rules* of temporal logic formulae are the following sentences:

i. Every atomic formula is a formula;

ii. If p and q are formulae, then $\neg p$, $p \wedge q$, $p \vee q$, $p \Rightarrow q$, $p \Leftrightarrow q$ are formulae, too;

iii. If p and q are formulae, then $\Box p$, $\Diamond p$, $p \mathcal{U} q$ are formulae, too.

All unary operators have a higher priority than the others. The $\mathcal{U}$ operator has a higher priority than other binary operators.

In order to present the *semantics* of the temporal logic, we have to introduce the notions of the structure and the interpretation of a temporal formula.

Definition 3. The *structure* of a temporal formula over a set of atomic propositions P is a tuple $K = \langle S, \rho, v \rangle$, where:

S	—	a set of states;
$\rho : S \to S$	—	transition between states;
$v : S \to 2^P$	—	valuation, i.e. an assignment of the truth values to atomic formulae in each state.

The relation ρ provides a unique state. $\rho^i(s)$, where $i \geq 0$, denotes the i-th successor of s, i.e. if $\rho^0(s) = s$, then $\rho^i(s)$ is the i-th element of the sequence

$$s,\ \rho(s),\ \rho\big(\rho(s)\big),\ \rho\Big(\rho\big(\rho(s)\big)\Big) \ldots$$

The relation v stands for a selection of a subset of P so as all elements of this subset, and only these elements, have the truth value.

Definition 4. The *interpretation* of a temporal formula over a set of atomic propositions P is the tuple $I = \langle K, s_0 \rangle$, where:

K — the structure of a formula;

s_0 — a first state.

If an interpretation I satisfies a formula p, which is denoted by $I \models p$, then the formula p is called the *satisfied* one. If a class of interpretations C satisfies a formula p, it is denoted by $C \models p$. If all interpretations satisfy a formula p, what is denoted by $\models p$, then the formula p is called the *valid* one.

An interpretation of a formula which satisfies this formula is called the *model interpretation* of this formula, or briefly the *model*. Let us consider the models of the following formulae:

$\langle K, s \rangle \models p$	iff	$p \in v(s)$
$\langle K, s \rangle \models \neg p$	iff	$\neg \langle K, s \rangle \models p$
$\langle K, s \rangle \models p \wedge q$	iff	$\langle K, s \rangle \models p$ and $\langle K, s \rangle \models q$
$\langle K, s \rangle \models \Box p$	iff	$\langle K, \rho^i(s) \rangle \models p$ for every $i \geq 0$
$\langle K, s \rangle \models \Diamond p$	iff	$\langle K, \rho^i(s) \rangle \models p$ for some $i \geq 0$
$\langle K, s \rangle \models p \mathcal{U} q$	iff	$\langle K, \rho^j(s) \rangle \models q$ for some $j \geq 0$ and $\langle K, \rho^i(s) \rangle \models p$ for every $i < j$

The list of axioms and laws of temporal logic, together with some interesting properties of temporal operators, may be found e.g. in (Manna and Pnueli, 1981; Kröger, 1987). A shortened list may also be found in (Klimek,1992). Good surveys and initiations which have been prepared in different ways are the works (Emerson, 1990; Manna and Pnueli, 1992).

3. Basic Similarities

In this section, basic similarities of formulae interpretations are presented. Before the presentation the notion of bisimulation proposed in (Milner, 1980) should be recalled. The connection is justifiable but it should also be noticed that the bisimulation refers to a different kind of processes in comparision with interpretations of temporal logic formulae. The strict similarity is defined as equivalence in (Wolper, 1987), and then we try to generalize this notion. The strict similarity is joined to an isomorphism of structures and then we try to define their homomorphism.

Every temporal logic formula may have infinitely many legal interpretations which satisfy this formula. The valuation in temporal logic is quite different from the one used in classical logic. In succeeding states variables and propositions can have different values. The valuation of a temporal formula is in reality an inference

from a whole sequence of states (worlds). However, perhaps there are some differences between two various sequences of states together with valuations in these states which are not significant and therefore they could be disregarded. At first, let us consider the situation which seems most common, i.e. the situation when all succeeding states starting from the initial ones have identical valuations. Let us present it more formally.

Definition 5. Two interpretations I_1 and I_2 over a set P are *strictly similar*, which is denoted by $I_1 \simeq I_2$, if there exists a one-to-one relation $\pi_1 : S_1 \times S_2$, and the following conditions are satisfied:

i. $(s_{0_1}, s_{0_2}) \in \pi_1$

ii. $(s_1, s_2) \in \pi_1 \Rightarrow (\rho_1(s_1), \rho_2(s_2)) \in \pi_1$

iii. $(s_1, s_2) \in \pi_1 \Rightarrow v_1(s_1) = v_2(s_2)$

This definition states that any two interpretations are strictly similar if it is possible to construct a relation called π_1 such that the initial states belong to the relation, the transition relations of the interpretations preserve the relation and the valuations are identical in the states which match towards the relation.

It is possible to construct an algorithm to create the π_1 relation for any two interpretations provided that there exists a similarity between these interpretations. The algorithm is not difficult and is omitted here. The example of two strictly similar interpretations shown in Fig. 1 illustrates the idea of how the algorithm works.

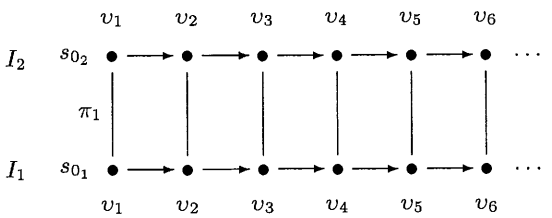


Fig. 1. An example of strictly similar interpretations.

It seems clear that for such a definition of strict similarity one can formulate the following theorem which renders it possible to make use of the definition.

Theorem 1. *If I_1 and I_2 are strictly similar interpretations, i.e. $I_1 \simeq I_2$, then for any temporal formula f it is satisfied that $I_1 \models f \Leftrightarrow I_2 \models f$.*

Proof. (sketch) If I_1 satisfies a formula, this means that the sequence of valuations of succeeding states satisfies this formula. The strict similarity assures the identity of the valuations for succeeding states after transition to I_2, which means that the formula is satisfied here as well. An inference in the reverse direction is possible as well because the similarity relation is a one-to-one relation. ■

Now let us consider another situation which could be recognized as an extension with reference to the strict similarity case. Perhaps there are some subsequences of states which are not essentially different, especially with reference to the valuations in succeeding states. Let us present it formally.

Definition 6. An interpretation I_1 is *generally similar* to an interpretation I_2, which is denoted by $I_1 \simeq\!\!> I_2$ or $I_2 <\!\!\simeq I_1$, if there exists a many-to-one relation $\pi_2 : S_1 \times S_2$, and the following conditions are satisfied:

i. $(s_{0_1}, s_{0_2}) \in \pi_2$

ii. $(s_1, s_2) \in \pi_2 \Rightarrow (\rho_1(s_1), s_2) \in \pi_2 \ \vee \ (\rho_1(s_1), \rho_2(s_2)) \in \pi_2$

iii. $(s_1, s_2) \in \pi_2 \Rightarrow v_1(s_1) = v_2(s_2)$

This definition states that any two interpretations are generally similar if it is possible to construct a relation called π_2 such that the initial states belong to the relation, a transition relation of one of these interpretations preserves the relation and the valuations are identical in the states which match towards the relation.

It is also possible to present an algorithm for construction of the general similarity relation. The algorithm is not difficult and is omitted here. The example of two generally similar interpretations of Fig. 2 shows the idea of how the algorithm works.

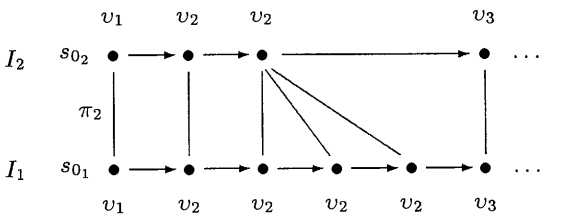


Fig. 2. An example of generally similar interpretations.

Just as in the case of a strict similarity relation, it is possible to formulate a theorem which makes use of the general similarity definition.

Theorem 2. *If an interpretation I_1 is generally similar to an interpretation I_2, i.e. $I_1 \simeq\!\!> I_2$, and for any temporal formula f $I_1 \models f$ is satisfied, then $I_2 \models f$ is also satisfied.*

Proof. (sketch) If I_1 satisfies a formula, this means that the sequence of the valuations of succeeding states satisfies this formula. The general similarity assures transition from any subsequence of states with identical valuation to a non-longer subsequence of I_2 with the same valuation, which means that the formula is satisfied here, too.

■

4. Properties of Similarities

In this section some theorems which could clarify the nature of both the similarities are formulated. At the beginning, let us view the similarities from a different perspective, i.e. from the side of the transition relation of a temporal structure.

Theorem 3. *If an interpretation I_1 is strictly similar to an interpretation I_2, i.e. $I_1 \simeq I_2$, then the following statements are satisfied:*

i. $(t_1, t_2) \in \rho_1 \Rightarrow \exists t_3 : ((\pi_1(t_1), t_3) \in \rho_2 \wedge \pi_1(t_2) = t_3)$

ii. $(t_1, t_2) \in \rho_1 \Rightarrow \exists t_3 : ((t_3, \pi_1(t_2)) \in \rho_2 \wedge \pi_1(t_1) = t_3)$

Proof. (sketch) The transition relations always determine a single state, and the strict similarity relation preserves both of the transitions. Therefore, for any state the successor (the predecessor) of its similar state is equivalent to the similar state of its successor (predecessor). ■

Theorem 4. *If an interpretation I_1 is generally similar to an interpretation I_2, i.e. $I_1 \simeq\!> I_2$, then the following statements are satisfied:*

i. $(t_1, t_2) \in \rho_1 \Rightarrow (\exists t_3 : ((\pi_2(t_1), t_3) \in \rho_2 \wedge \pi_2(t_2) = t_3)) \vee \pi_2(t_2) = \pi_2(t_1)$

ii. $(t_1, t_2) \in \rho_1 \Rightarrow (\exists t_3 : ((t_3, \pi_2(t_2)) \in \rho_2 \wedge \pi_2(t_1) = t_3)) \vee \pi_2(t_2) = \pi_2(t_1)$

Proof. (sketch) The transition relations always determine a single state, and the general similarity relation preserves both transitions. Therefore, for any state the successor (the predecessor) of its similar state, or exactly the similar state, is equivalent to the similar state of its successor (predecessor). ■

As was already mentioned, every interpretation can contain subsequences of states which do not differentiate between one another. Perhaps it would be possible to concentrate such a subsequence of states taking into account their valuations and to replace the subsequence with a single state. This could make further comparisons between interpretations easier.

Definition 7. An interpretation I is an *aggregated interpretation*, which is denoted by $\widehat{I}$, if and only if for every $i > 0$ it is satisfied that $v(s_i) \neq v(s_{i+1})$.

It is also possible to propose an algorithm for concentration of an arbitrary interpretation. The algorithm is not difficult and is omitted here.

Calling both definitions of similarity and getting a possibility to reduce the states of interpretation, we present now the following result.

Theorem 5. *For any interpretations I_1, I_2 and I_3 the following statements are satisfied:*

1. $I_1 \lessapprox I_2 \wedge I_1 \simeq\!> I_2 \Rightarrow I_1 \simeq I_2$

2. $I_1 \simeq\!\!> I_2 \wedge I_2 \simeq\!\!> I_3 \Rightarrow I_1 \simeq\!\!> I_3$
3. $I_1 \simeq I_2 \Rightarrow \widehat{I}_1 \simeq \widehat{I}_2$
4. $I_1 <\!\!\simeq I_2 \Rightarrow \widehat{I}_1 <\!\!\simeq I_2$
5. $I_1 <\!\!\simeq I_2 \Rightarrow \widehat{I}_1 \simeq \widehat{I}_2$

Proof.

1–2. The proof is immediate.

3–5. (Sketch) If the interpretation is concentrated, then for many subsequences of states it is possible to use a one-to-one instead of a many-to-one relation. ■

From a practical point of view the bound of a valuation seems to be useful for further considerations.

Definition 8. A *bound of a valuation* of an interpretation I by a set of atomic formulae A, which is denoted by I^{-A}, is the replacement of the valuation for the new valuation v which is specified as follows: $v_2(s) = v(s) \setminus A$.

5. Example

In this section a typical car ferry plying between two banks of a river is considered, cf. Fig. 3. There is a certain carrying capacity, i.e. a maximum number of cars which

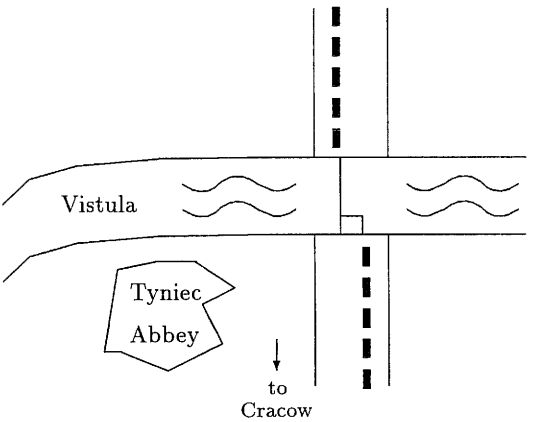


Fig. 3. A typical car ferry.

could be carried across during a single course. The ferry plies continuously under the stipulation that there is always a short stop (idle time) on each bank. The ferry does not run only when there is no car at any riverside. In this case the ferry is waiting for a car at the riverside where it has stopped the last time. If a new car stops at the riverside to which the ferry has been moored, it enters on board, but if a new

car stops on the opposite bank, the ferry should run there even though it is empty. The cars which stop on a bank are added to the end of the corresponding queue. The queues are infinite. From the point of view of the system both the queues are critical sections.

The system behaviour is described by temporal logic formulae. The set of atomic formulae P is the following: *newcar* (or briefly n) – a new car arrives at a bank; $queuecar(q)$ – the car which arrived is added to the end of the queue on a bank; $wait(w)$ – there is at least one car on a bank which waits for a carry; $at(a)$ – the ferry is moored to the bank; $full(f)$ – the ferry is full; $empty(m)$ – the ferry is empty; $entercar(e)$ – the first car of a queue enters on board; $time(t)$ – a short and obligatory stop (idle time) of the ferry is over; $crossing(c)$ – the ferry leaves a bank and is crossing the river; $reach(r)$ – the ferry moores to a bank; $exitcars(x)$ – all the cars on the board leave the ferry. The notation $\overline{bank}$ means the bank opposite to a given one, i.e. *bank*. We also write e.g. $\overline{wait}$, or briefly $\overline{w}$, instead of $wait(\overline{bank})$.

Suppose that the set of temporal formulae which describes the liveliness aspect of the system behaviour is the following:

$$newcar(bank) \Rightarrow \Diamond(queuecar(bank) \wedge wait(bank)) \tag{1}$$

$$at(bank) \wedge \neg full \wedge wait(bank) \Rightarrow \Diamond(entercar(bank) \wedge \neg empty) \tag{2}$$

$$time \wedge full \Rightarrow \Diamond crossing(bank) \tag{3}$$

$$time \wedge \neg empty \wedge \neg wait(bank) \Rightarrow \Diamond crossing(bank) \tag{4}$$

$$time \wedge \neg wait(bank) \wedge wait(\overline{bank}) \Rightarrow \Diamond crossing(bank) \tag{5}$$

$$crossing(bank) \Rightarrow \neg at(bank) \tag{6}$$

$$crossing(bank) \Rightarrow \Diamond reach(\overline{bank}) \tag{7}$$

$$reach(bank) \Rightarrow \Diamond(at(bank) \wedge \neg time \wedge exitcars(bank)) \tag{8}$$

$$exitcars(bank) \Rightarrow \Diamond(empty \wedge \neg full) \tag{9}$$

$$\neg time \Rightarrow \Diamond time \tag{10}$$

It is worth noticing that most of these formulae refer to the so-called total correctness of a system. Since the liveness property could be expressed in many different ways, it is also possible to prepare different sets of formulae. The safety aspect of the system is not included in these formulae since this property is not important in our considerations.

Whice verifying the system correctness, we can examine many different properties. For example, one of the most obvious ones is the liveness property which states that when a new car arrives at a bank, it is obvious that at a future moment the car will leave on the opposite bank. It is expressed by the formula:

$$newcar(bank) \Rightarrow \Diamond exitcars(\overline{bank}) \tag{11}$$

This requirement seems to be the most general among liveness properties and it includes seemingly other liveness properties. The formula (11) could be derived from the set of temporal formulae listed above. This kind of analysis is rather a syntax method based on transformations of a set of formulae while using well-known axioms and laws of temporal logic in order to obtain the desired formula. A semantic method is another method which examines whether formulae are satisfied in a set of legal interpretations. We are going to use this method in further considerations.

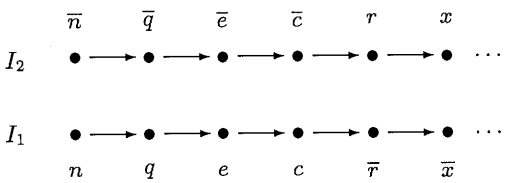
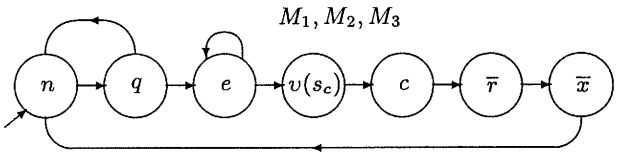


Fig. 4. Interpretations for two cars crossing the river in opposite directions.

Let us suppose that we have two interpretations, as is shown in Fig. 4. Both the interpretations concern the case when two cars cross the river in opposite directions. Each car arrives at a bank, joins the queue, enters the ferry and, after crossing the river, the ferry reaches a bank and the car leaves on the opposite side of the river. It is irrelevant to which bank of the river the ferry is moored first, but it seems more natural that it is moored to the bank which is connected with interpretation I_1. Both these interpretations are strictly similar, i.e. $I_1 \simeq I_2$, and the only remark is that the corresponding formulae, e.g. n and $\overline{n}$, are equivalent though they refer to the opposite banks. (By the way, for bounded interpretations $J_1 = I_1^{-(P \cap \{n,\overline{x}\})}$ and $J_2 = I_2^{-(P \cap \{n,\overline{x}\})}$, we can write $J_1 \simeq J_2$ and also $J_1 \models (n \Rightarrow \Diamond \overline{x}) \Leftrightarrow J_2 \models (n \Rightarrow \Diamond \overline{x})$.)

One of the well-known methods of verifying the program correctness is the method called *model checking*. It is based on a verification whether a program, which is described explicitly by a set of states and transitions, satisfies a specification and properties given by a set of formulae. This method is important because sometimes the model is already determined, the more so we are sometimes interested in formulae which are not satisfied in general, i.e. they are not valid, but they are satisfied for some models. In the latter situation we can say that some formulae are *relatively valid*. (Of course, if a formula is generally valid, it is valid for all models.)

Suppose that we have some different models while considering start conditions of crossing the river, as is shown in Fig. 5, cf. (3)–(5) and valuations in state s_c. Let $\mathcal{M} = \{M_1, M_2, M_3\}$. In these models there are some repetitive substructures and hence some considerations concerning interpretations are more complicated. However, we can write the following statement: $M_i, M_j, M_k \in \mathcal{M} \wedge i \neq j \neq k \Rightarrow \forall I_i(M_i) \exists I_j(M_j) \exists I_k(M_k) : I_i \simeq I_j \simeq I_k$. (The notation $I(M)$ means an arbitrary interpretation of model M.) If we bound interpretations $J_1 = I_1(M_1)^{-(P \cap \{n,\overline{x}\})}$, $J_2 = I_2(M_2)^{-(P \cap \{n,\overline{x}\})}$ and $J_3 = I_3(M_3)^{-(P \cap \{n,\overline{x}\})}$, then for concentrated interpretations we can write simply $\widehat{J_1} \simeq \widehat{J_2} \simeq \widehat{J_3}$.



$M_1 : \ v(s_c) = \{t, f\}$

$M_2 : \ v(s_c) = \{t, \neg m, \neg w\}$

$M_3 : \ v(s_c) = \{t, \neg w, \overline{w}\}$

Fig. 5. Models for different start conditions of crossing the river.

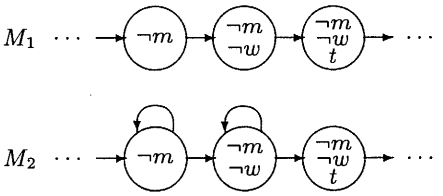


Fig. 6. More realistic models for a start condition of crossing the river.

Suppose that we have some fragments of different models, as shown in Fig. 6, which are related to (4). The situation when all appropriate atomic formulae are satisfied in a state at once, as described in Fig. 5, is unrealistic. Therefore it is assumed for model M_1 in Fig. 6 that cars from a queue enter the ferry (the first state), the queue becomes empty (the second state), and finally, when the idle period of the ferry is over, the start condition of crossing the river is satisfied (the third state). Model M_2 is a development of model M_1 since it seems to be more adequate, if it is assumed that the entry on the ferry, car by car, may last for some time (some states), and also, if there is already no car in the queue, it may last for some time (some states) until the idle time of the ferry is over. Now, we can write the following sentence: $\forall I_2(M_2)\forall I_1(M_1) : I_2 \simeq\!\!> I_1$. When we concentrate interpretations, we can also write $\widehat{I}_2 \simeq \widehat{I}_1$.

6. Conclusions

Two basic types of similarities of temporal logic interpretations have been presented. The algorithms for construction of the similarity relations have been omitted here and they may be found in (Klimek, 1996). Some theorems for the introduced similarities have also been presented. A typical car ferry plying between two banks of a river has been used as an example and then a semantic analysis of some selected models through examining the similarities of the generated interpretations has been carried out.

It seems that the similarities of temporal logic interpretations defined as an isomorphism together with a homomorphic mapping have not been considered. Similarities of interpretations are important since they enable us to understand better the nature of interpretations, as well as to make the analysis of a system easier. The analysis becomes easier because it is possible to distinguish between the situations when the interpretations differentiate between one another in a radical way and the situations when the differences between the interpretations could be disregarded. It also seems that the definitions of the formulae similarities could be based on the similarities of interpretations, cf. some comments in (Klimek, 1996). An interesting question arises here about the range of the differences which could be disregarded: Does the similarity of interpretations or formulae make it possible to replace one interpretation or formula with another, or does it rather allow that one formula is included in another which describes the reality in a more general and declarative form? The problem of distribution of the analysis as a result of decomposition of the whole developing process into a sequence of small steps is another important issue. The most important and popular method for this kind of software specification (and verification) is the method called the step-wise refinement. However, all these issues mentioned above will constitute the next step in the research.

Acknowledgements

The author wishes to express his thanks to Prof. Tomasz Szmuc and an anonymous reviewer for their comments which helped him to improve the final version of the paper. This research was supported by the State Committee for Scientific Research (KBN) under grant No.8T11A01308 (*Formal Methods and Tools Supporting Design and Analysis of Real-Time Software for Control*).

References

Emerson E.A. (1990): *Temporal and modal logic*, In: Handbook of Theoretical Computer Science. Vol.B: Formal Models and Semantics (J. van Leeuwen, Ed.). — Amsterdam: Elsevier, Cambridge: The MIT Press, pp.995–1072.

Klimek R. (1992): *Temporal logic as a tool for program correctness analysis.* — Appl. Math. Comp. Sci., Vol.2, No.2, pp.217–235.

Klimek R. (1996): *Similarities of interpretations of temporal logic formulae.* — Res. Rep. No.55, Institute of Automatics University of Mining Metallurgy, Cracow, Poland.

Kröger F. (1987): *Temporal Logics of Programs.* — EATCS Monographs on Theoretical Computer Science, Berlin: Springer-Verlag.

Manna Z. and Pnueli A. (1981): *Verification of concurrent programs: Temporal framework*, In: The Correctness Problem in Computer Science. Int. Lecture Series in Computer Science (R.S. Boyer, J.S. Moore, Eds.). — London: Academic Press, pp.215–272.

Manna Z. and Pnueli A. (1992): *The Temporal Logic of Reactive and Concurrent Systems.* — New York: Academic Press.

Milner R. (1980): *A Calculus of Communicating Systems.* — Lecture Notes in Computer Science, 92, Berlin: Springer-Verlag.

Wolper P. (1989): *On the relation of programs and computations to models of temporal logic*, In: Proc. Temporal Logic in Specification (B. Banieqbal, H. Barringer and A. Pnueli, Eds.); Altrincham, UK, April 8–10, 1987, Lecture Notes in Computer Science, No.398, Berlin: Springer-Verlag, pp.75–123.

Received: 17 July 1996
Revised: 20 February 1997