Specifying message passing systems requires extending temporal logic

Specifying Message Passing Systems Requires Extending Temporal Logic

by

Specifying Message Passing Systems Requires Extending Temporal Logic

by

Ron Koymans

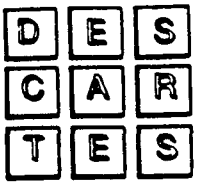
January 1987

COMPUTING SCIENCE NOTES

This is a series of notes of the Computing Science Section of the Department of Mathematics and Computing Science of Eindhoven University of Technology.
Since many of these notes are preliminary versions or may be published elsewhere, they have a limited distribution only and are not for review.
Copies of these notes are available from the author or the editor.

Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O. Box 513
5600 MB EINDHOVEN
The Netherlands

editor: F.A.J. van Neerven



European Strategic Programme of Research and Development in Information Technology

Project 937 : Debugging and Specification of Ada Real-Time Embedded Systems
Package 4 : Formal Semantics and Proof Systems for Real-Time Languages

Mail to : C. Bonnet
Doc. No. : PE.02
Type : PE
Title : Specifying Message Passing Systems requires extending Temporal Logic (extended abstract)
Author : R. Koymans
Date : 6-01-87 Version : 0
Replaces:

Document Status : Submitted
Confidentiality Level : Public Domain

GSI-TECSI
SYSTEAM KG
FOXBORO Netherlands NV
ELECTRONIQUE SERGE DASSAULT
EINDHOVEN UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF STIRLING
ADCAD Ltd

SPECIFYING MESSAGE PASSING SYSTEMS REQUIRES EXTENDING TEMPORAL LOGIC

(Extended Abstract)
(Revised)

Ron Koymans

Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O.Box 513
5600 MB Eindhoven
The Netherlands

January 12, 1987

Abstract

We prove that it is impossible to express asynchronous message passing within the framework of first-order temporal logic with both future and past operators (as studied by Kamp). This is an extension of a result of Sistla et al. that unbounded buffers cannot be expressed in linear time temporal logic. Although strengthening Kamp's logic by adding counting and quantification over occurrences of propositions enables the expression of most message passing systems, we argue that order preserving systems which may loose messages still remain inexpressible. This is caused by the impossibility to *couple each message that is delivered by a message passing system to a* **unique** *message accepted by that system.* These results seem to necessitate the enrichment of TL-based formalisms, e.g. with auxiliary data structures or histories as done, respectively, by Lamport and Hailpern. Observe that Lamport employs a hybrid formalism (TL + Data Structures), and that in Hailpern's method similar systems, such as FIFO and LIFO, do not have similar specifications. We shall prove that no such enrichment is logically required. This is done by introducing an assumption which makes the unique coupling mentioned above explicit as an additional axiom within TL. In this way, no extraneous formalisms are introduced, and both FIFO and LIFO are expressible with equal ease.

1. Introduction

The need of a general specification methodology for the formal reasoning about computerized systems is now beyond doubt. Not that evident are the properties such a methodology should satisfy to be of practical use. Three such properties that we consider essential are:

1. it is built on a simple and well-known mathematical basis,
2. it supports hierarchical development (i.e. the refinement of a higher level module towards a lower level) and compositional reasoning (i.e. the specification of the whole system is a function of the specifications of its components),
3. abstractness: systems are specified in a black box fashion, that is only in terms of their (observable) interfaces with the environment (this implies the absence of any implementation bias whatsoever).

Two further desirable properties are in our opinion:

4. generality: similar systems have similar specifications,
5. uniformity: the methodology is based on a single formalism covering all aspects of a specification.

In this paper we concentrate on message passing systems. The motivation for this choice is supplied by their manifold appearances in practice: (asynchronous) message passing is one of the most important means of interprocess communication in distributed systems, either on a high level (e.g. in telecommunication applications where programming could be done in a high-level concurrent language with asynchronous message passing such as CHILL [CHILL]) or on a lower level (such as in implementations of synchronous languages for distributed computing like Ada [Ada]).

Since the introduction of (linear time) temporal logic in the area of program verification ([P]), it has proved to be a most versatile tool for the specification and verification of concurrent systems. It can be used as the basis for a specification methodology fulfilling the five requirements listed above and a lot more as shown in the work of Manna & Pnueli, Lamport, Barringer & Kuiper, Moszkowski and many others. So it seems that linear time temporal logic is an excellent candidate for the basis of a general specification methodology.

However, as Sistla et al. indicated, temporal logic has its limitations, too. They proved that certain types of unbounded buffers cannot be specified in linear time temporal logic (although bounded buffers can be specified). Our first result is the generalization of this to more expressive logics studied by Kamp. The systems to which the result can be applied can also be considerably extended: many practical message passing systems cannot be specified in these logics. The result is first proved for the propositional versions. In that case, the result could be expected since infinite objects cannot be specified propositionally. Not obvious is that this result can be immediately strengthened to the first-order case. Next we show that many systems (including unbounded

buffers) *can* be specified once we are allowed to reason about the n-th occurrence of a proposition and quantification over such numbers is added. (This extension of temporal logic agrees with a suggestion recently made by Mark Trakhtenbrot ([Tr]) to enrich Harel's statecharts formalism ([Har]).) However, we present strong arguments supporting a second inexpressiveness result, stating that this addition does not solve the problem for order preserving message passing systems which may loose messages. This is serious, for reliable transmission over unreliable media is what most protocols are about, and this should therefore be specifiable in any proper specification methodology. In both cases, in our analysis the source of this inexpressiveness is the impossibility to correlate a message that is delivered by the system with a *unique* message accepted (earlier) by the system.

These limitations give a theoretical foundation for the fact that researchers using linear time temporal logic use to enrich their formalisms to specify such systems, e.g. by adding certain data structures (queues etc.) or by using auxiliary variables (such as histories). We review three of such proposed extensions. The first two of these add supplementary formalisms to temporal logic, thus violating the generality/uniformity requirements (see points 4 and 5 above). The third one is an attempt to remain completely within the temporal logic domain, by introducing an additional axiom which makes the coupling of a delivered message to a unique accepted message explicit, thus removing the trouble spot.

The paper is organized as follows. In section 2 we define the syntax and semantics of Kamp's logic and give the definitions of the message passing systems considered. In section 3 we present our inexpressiveness results and their consequences for the specification of message passing systems. In section 4 we review three possible solutions to overcome the previous logical limitations. At last, in section 5 we draw some conclusions and indicate future work.

2. Temporal Logic and Message Passing Systems

We first define the syntax of Kamp's logic.

Definition: For I an arbitrary set, $L_I(U, S)$ is the language with

vocabulary: atomic propositions $P_i (i \in I)$

logical operators $\neg$, $\wedge$, U , S

formulae: $P_i (i \in I)$

$\neg f_1$, $f_1 \wedge f_2$, $f_1 U f_2$ and $f_1 S f_2$ (f_1, f_2 formulae).

We now give the semantics of $L_I(U, S)$. A state is a mapping from I to $\{True, False\}$. Σ is the set of all states. A model M is a triple $<T, <, D>$ where $<$ is a linear order on T (the time domain) and D a function from T to Σ. An interpretation is a pair $<M, t>$ where M is a model

and $t \in T$. Truth of a formula $f \in L_I(U, S)$ in an interpretation $<M, t>$, notation $M, t \models f$, is inductively defined as follows:

$M, t \models P_i := D(t)(i) = True \quad (i \in I)$
$M, t \models \neg f_1 := \quad not \quad M, t \models f_1$
$M, t \models f_1 \wedge f_2 := M, t \models f_1$ and $M, t \models f_2$
$M, t \models f_1 U f_2 :=$ there exists a $t' \in T$ such that $t < t'$ and $M, t' \models f_2$ and for all $t'' \in T$:
$(t < t''$ and $t'' < t')$ implies $M, t'' \models f_1$
$M, t \models f_1 S f_2 :=$ there exists a $t' \in T$ such that $t' < t$ and $M, t' \models f_2$ and for all $t'' \in T$:
$(t' < t''$ and $t'' < t)$ implies $M, t'' \models f_1$.

Concerning the expressive power of Kamp's logic: in [K] it is proved that $L_I(U, S)$ with I the natural numbers is expressively complete with respect to the class of complete linear orders. For the class of ω-models (obtained by taking $<T, <>$ isomorphic with the natural numbers with its usual ordering) it is shown in [GPSS] that only U as temporal operator already suffices for expressive completeness.

Next we turn to several types of message passing systems. Let *Messages* be a non-empty set of messages, the message alphabet. A schematic picture of a message passing system could be

MPS = Message Passing System

where $m \in$ *Messages* and

in (m) corresponds to the acceptance (from the environment) of message m by the MPS, and

out (m) corresponds to the delivery (to the environment) of message m by the MPS.

The MPS can be a simple buffer or transmission medium but also a complex communication network. *in* (m) and *out* (m) constitute the interface with the environment and *out* (m) is considered to be the system reaction on the environment action *in* (m). Of course, the above picture should be supplemented by restrictions on the functions *in* and *out*, dependent on the particular type of message passing system considered. For all types we take the following restrictions as basic assumptions:

BA1. the acceptance and delivery of messages can be viewed as instantaneous actions (in the sense that always a unique moment of time can be identified at which a message can be said to be accepted, respectively delivered), which are always possible,

BA2. at any moment of time, at most *one* message can be accepted (respectively delivered),

BA3. the MPS does not create messages by itself (in other words: the bag of delivered messages is always *some part* of the bag of accepted messages),

BA4. the speed of the MPS is finite, i.e. there is a positive (maybe infinite) delay between the acceptance of a message and its delivery.

Additionally, we distinguish the following restrictions:

P. the system does not loose messages (all accepted messages are eventually delivered),

IP. the system delivers all accepted messages unless it crashes at some point (and then does not deliver any messages anymore),

FP. if a finite number of messages is accepted, they will all be delivered (but not necessarily for an infinite number),

EL. the system always looses messages after a while.

P (perfect) and IP (initially perfect) correspond to unbounded buffers (respectively with and without liveness property in the terminology of Sistla et al.). An example of a MPS with the FP (finitely perfect) property is a system with a fixed period in which it looks into the bag of hitherto accepted but not yet delivered messages and chooses randomly one of these to be delivered (unless the bag is empty, of course). Note that P is part of both IP and FP but that IP and FP are incomparable: FP guarantees that all messages will be delivered whenever a *finite* number is *accepted* whereas in contrast IP guarantees this whenever an *infinite* number of messages is *delivered*. EL abbreviates Eventual Loss. An example of a MPS often occurring in practice that is subject to restrictions BA1-BA4 only is a transmission medium with a probability between zero and one of a successful transmission. Such a MPS exhibits all behaviors allowed by BA1-BA4 although the probability of the occurrence of certain behaviors may differ.

A further distinction of message passing systems can be made by the order in which accepted messages are delivered. This can be FIFO (first-in first-out, like queues), LIFO (last-in first-out, like stacks) or unordered (like bags), that is in no order at all (as in communication networks in which each message is sent on to an arbitrary node in the network until it arrives at the destination node).

Since, ideally, message passing systems operate over an infinite time period, we henceforth assume that the time domain T of our logics is infinite.

3. Inexpressiveness results

The first inexpressiveness result concerns types of message passing systems that cannot be characterized in Kamp's logic.

Definition: Let $f \in L_J(U, S)$, M be a model, $t \in T$.
Define $[t]_{M,f} := \{g \in SF(f) \mid M, t \models g\}$ where $SF(f)$ is the set of subformulae of f (including f itself).

Definition: Let M be a model and $t_1, t_2 \in T$ such that $t_1 \leqslant t_2$.
Then $M^{t_2}_{t_1}$ is the reduction of M to $T^{t_2}_{t_1} := \{ t \in T \mid t \leqslant t_1 \vee t_2 < t \}$.

Theorem: Let $f \in L_I(U, S)$, M be a model and $t_1, t_2 \in T$ such that $t_1 \leqslant t_2$ and $[t_1]_{M,f} = [t_2]_{M,f}$.
Then for all $t \in T^{t_2}_{t_1}$:
$M, t \models f$ if and only if $M^{t_2}_{t_1}, t \models f$.

Proof: By structural induction on f. The details are given in the full paper. We prove the theorem for one of the interesting cases.
Let $f \equiv f_1 U f_2$, M be a model and $t_1, t_2 \in T$ such that $t_1 \leqslant t_2$.
Assume
(i) $[t_1]_{M,f} = [t_2]_{M,f}$.
We are going to show that $M, t \models f$ implies $M^{t_2}_{t_1}, t \models f$ for $t \leqslant t_1$.
Hence assuming
(ii) $t \leqslant t_1$ and
(iii) $M, t \models f_1 U f_2$,
we prove that $M^{t_2}_{t_1}, t \models f_1 U f_2$.
From (i) and the induction hypothesis we deduce
(iv) $M, t \models f_1$ implies $M^{t_2}_{t_1}, t \models f_1$ for all $t \in T^{t_2}_{t_1}$,
(v) $M, t \models f_2$ implies $M^{t_2}_{t_1}, t \models f_2$ for all $t \in T^{t_2}_{t_1}$.
From (iii) it follows that
(vi) there exists a $t_0 \in T$ such that $t < t_0$ and $M, t_0 \models f_2$ and $M, t' \models f_1$ for all $t' \in T$ such that $t < t'$ and $t' < t_0$.
Distinguish between two cases:
(a) $t_0 \leqslant t_1$: The result follows in this case immediately from (iv),(v) and (vi)
(b) $t_1 < t_0$:In this case by (ii),(vi) we get also $M, t_1 \models f_1 U f_2$.
By (i) it follows that $M, t_2 \models f_1 U f_2$. Hence
(vii) there exists a $t_3 \in T$ such that $t_2 < t_3$ and $M, t_3 \models f_2$ and $M, t' \models f_1$ for all $t' \in T$ such that $t_2 < t'$ and $t' < t_3$.
Because of $t_1 < t_0$ and (vi) we have also
(viii) $M, t' \models f_1$ for all $t' \in T$ such that $t < t'$ and $t' \leqslant t_1$.
Then $M^{t_2}_{t_1}, t \models f_1 U f_2$ by (vii) and (viii). ■

Remark: The result of Sistla et al. is obtained by taking I finite and considering only ω-models (see section 2) and noting that their operators next-time, until, last-time and since are all expressible in terms of U and S.

Corollary: The following types of message passing systems cannot be specified by Kamp's logic:

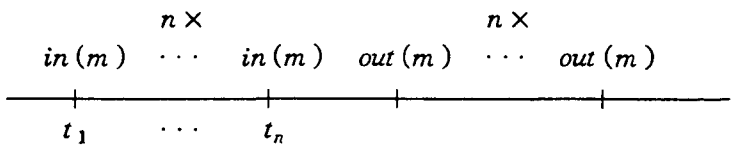
(i) satisfying only BA1-BA4

(ii) satisfying BA1-BA4 + P

(iii) satisfying BA1-BA4 + IP

(iv) satisfying BA1-BA4 + FP.

Proof: Suppose there exists a formula f characterizing one of these four types. The number of subformulae of f is bounded by $2^{|f|}$ where $|f|$ is the length of f. Now choose $n > 2^{|f|}$ and consider the following model M:



where $m \in Messages$.

This is a possible behavior for all these four types. Hence f is satisfied in M. Because $n > 2^{|f|}$ there are i, j such that $1 \leq i < j \leq n$ and $[t_i]_{M,f} = [t_j]_{M,f}$. Applying the theorem we conclude that f is also satisfied in a model with less than n inputs and exactly n outputs. This violates our basic assumption BA3 about message passing systems in section 2. Hence such a f characterizing one of these four types cannot exist. ■

Remark 1: Although the types (ii), (iii) and (iv) are contained in type (i), the result for (i) in itself need not imply the result for the others. In fact, the type that looses all messages is contained in (i) but can be specified indeed. It only happens to be the case that the model M in the proof above is a possible behavior for all four types.

Remark 2: The model M uses only one message and hence the same argument is also valid for all types where we add a particular ordering such as FIFO or LIFO to one of the four types above.

Remark 3: Because the model M uses only a finite number of different messages (in this case 1), allowing quantification over the message alphabet (which is here the underlying domain of data) will not help; hence the result can be generalized to the first-order variant of Kamp's logic.

Remark 4: The above argument does not work for the type satisfying BA1-BA4 + EL because it is not the case that the model M will always (for all n) be a possible behavior of this type. For this type we can use a dual argument now using the other direction of the if and only if of the theorem and concentrating on outputs instead of inputs:
For n large enough, the model M above is not a possible behavior of this type, but all models with n inputs and less than n outputs are. Hence a formula f characterizing this type would according to the theorem also be satisfied in M. A contradiction with the assumption that f characterizes this type.

We now show that we can specify all the four types of the corollary when we add counting of occurrences of propositions, notation P_i^n ($i \in I$, $n > 0$), and allow quantification over them. The intended semantics of P_i^n is that it is only true at the moment of time when P_i is true for the n-th time. Below we show that for each (fixed) n, P_i^n is expressible in Kamp's logic.

We first define some derived operators. The familiar temporal logic operators F (eventually) and its dual G (henceforth) can be defined by

$Ff := f \vee \mathbf{true}\ U\ f$ where $\mathbf{true} \equiv \neg (P_i \wedge \neg P_i)$ for some $i \in I$,
$Gf := \neg F \neg f$.

Both F and G include the present moment as part of the future. A past operator similar to F but not including the present as part of the past is defined by

$Pf := \mathbf{true}\ S\ f$.

Intuitively, Pf asserts that f was true some moment in the past. Using the operator P we can express P_i^n for each fixed n, e.g.

$P_i^3 \equiv P_i \wedge P\,(P_i \wedge P\,P_i) \wedge \neg P\,(P_i \wedge P\,(P_i \wedge P\,P_i))$, or alternatively

$P_i^3 \equiv P_i \wedge \neg P_i\ S\ (P_i \wedge \neg P_i\ S\ (P_i \wedge \neg P\ P_i))$.

Now BA1-BA4 can be specified as follows:

BA 2 $G\ \forall m \forall m' \forall n \forall n'\ [((in\,(m)^n \wedge in\,(m')^{n'}) \vee (out\,(m)^n \wedge out\,(m')^{n'})) \rightarrow m = m']$

BA3,4 $G\ \forall m\ \forall n\ [out\,(m)^n \rightarrow P\ in\,(m)^n]$.

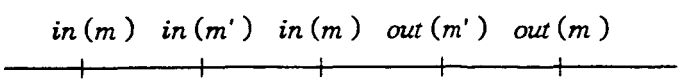
There is no need to specify BA1 because this is already fulfilled by the nature of the formalization: $in\,(m)$ and $out\,(m)$ are propositions which can be true at any moment. Additional restrictions can be specified by an appropriate axiom such as

$G\ \forall m\ \forall n\ [in\,(m)^n \rightarrow F\ out\,(m)^n]$

to specify perfect message passing systems.
If we also demand FIFO-ordering of messages this can be expressed by

$G\ \forall m\ \forall m'\ \forall n\ \forall n'\ [(out\,(m)^n \wedge P\ out\,(m')^{n'}) \rightarrow P\ (in\,(m)^n \wedge P\ in\,(m')^{n'})]$.

This specification, however, depends essentially on the assumption that the system is perfect. If this is not the case, the specification of FIFO is not possible anymore. For example, if messages may get lost, the above axiom would disallow the behavior

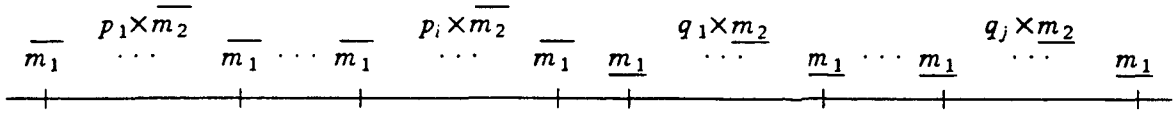


while this is a legal behavior of a FIFO message passing system which has lost the first m.

So, our next aim is to show that even with the addition of counting and quantifying over occurrences of propositions we still cannot specify systems satisfying BA1-BA4 + FIFO. The basic idea is the following one. Below we describe classes $C_{i,j}$ of models with the property that a formula distinguishing models in $C_{i,j}$ which satisfy BA1-BA4 + FIFO from those which don't, requires at least j independent parameters to be determined. Assuming that one cannot characterize these j parameters in a uniform way, this means that such a formula cannot exist, for it

should be of infinite length.

The classes $C_{i,j}$ use two different messages m_1 and m_2, and have exactly $i+1$ occurrences of $in(m_1)$ and $j+1$ occurrences of $out(m_1)$. Furthermore, between each two consecutive occurrences of $in(m_1)$ there are an arbitrary number of occurrences of $in(m_2)$ and similarly an arbitrary number of occurrences of $out(m_2)$ between each two consecutive occurrences of $out(m_1)$. So, by abuse of notation, let $\overline{m_1}$ and $\overline{m_2}$ denote $in(m_1)$ resp. $in(m_2)$, and $\underline{m_1}$ and $\underline{m_2}$ denote $out(m_1)$ resp. $out(m_2)$, then a model in $C_{i,j}$ looks like



The intention is that some occurrences of messages m_1 and m_2 may get lost but that order remains preserved, as in accordance with requirements BA1-BA4 + FIFO.

Now, a model in $C_{i,j}$ satisfies BA1-BA4 + FIFO if and only if

$$\exists k_1 \cdots \exists k_{j+1} \, [1 \leqslant k_1 < \cdots < k_{j+1} \leqslant i+1 \; \wedge \; \forall r \; [1 \leqslant r \leqslant j \rightarrow q_r \leqslant \sum_{l=k_r}^{k_{r+1}-1} p_l \;]].$$

Intuitively, this asserts that the s-th $(1 \leqslant s \leqslant j+1)$ occurrence of $out(m_1)$ corresponds to the k_s-th occurrence of $in(m_1)$, i.e. $k_1, \ldots, k_{j+1}$ are exactly the occurrences of m_1 that are delivered. The only thing left to be checked then is that $q_r (1 \leqslant r \leqslant j)$, the number of occurrences of $out(m_2)$ between occurrence r and $r+1$ of $out(m_1)$, is at most the total number of occurrences of $in(m_2)$ between occurrence k_r and k_{r+1} of $in(m_1)$. We conclude that knowing the parameters $k_2, \ldots, k_j$ (without loss of generality one can take $k_1 = 1$ and $k_{j+1} = i+1$) is essential to distinguish models in $C_{i,j}$ with respect to their satisfaction of BA1-BA4 + FIFO.

At present, we have no rigorous proof why there couldn't be a uniform characterization of these parameters. Apart from the above straightforward attempt to prove that BA1-BA4 + FIFO is not characterizable, another possibility for such a proof is based on the connections between temporal logics and formal language and automata theory (see e.g. [Th]). For example, pure propositional temporal logic is equivalent in expressive power both with ω-star-free ω-languages and with counter-free ω-automata, and hence is less expressive than ω-regular ω-languages. This motivated Wolper to extend temporal logic to become expressively equivalent with the class of ω-regular ω-languages ([W]). However, the fact that the addition of counting and quantifying over occurrences of propositions enables the specification of BA1-BA4 + FIFO, implies the definability of a language that is not even context-free: consider only models where a finite number of inputs preceed a finite number of outputs (this corresponds language theoretically to intersection with the regular language $\{ in(m) \mid m \in Messages \}^* \; \{ out(m) \mid m \in Messages \}^*$), then the class of models satisfying BA1-BA4 + P + FIFO corresponds to the language $\{ ww' \mid w \in \{ in(m) \mid m \in Messages \}^*, w' \in \{ out(m) \mid m \in Messages \}^*, w' = w[out/in] \}$ which is not context-free. Applying the same restriction to models satisfying BA1-BA4 + FIFO we get the language $\{ ww' \mid w \in \{ in(m) \mid m \in Messages \}^*, w' \in \{ out(m) \mid m \in Messages \}^*, w'[in/out]$ is a substring of $w \}$. Again this language is not context-free.

On the other hand, both languages above are recognizable by a deterministic queue automaton (a push-down automaton with as memory a queue instead of a stack). E.g., for the latter language, this automaton operates as follows. First, for each $in(m)$ encountered, it puts m in the queue. Then, for each $out(m')$, it empties the queue up till and including m'. The given model satisfies BA1-BA4 + FIFO if and only if it is always possible to find m in the queue for each $out(m)$ encountered. In fact, a similar procedure works for recognition of all models (also with possible mixtures of *in* and *out*) satisfying BA1-BA4 + FIFO.

The above remarks indicate limits for the expressive power of the addition of counting and quantifying over occurrences of propositions to temporal logic. For a lower bound one can pose the question whether all ω-regular ω-languages are definable with this addition and for an upper bound one can ask whether this addition can be captured within the class of deterministic queue automata on infinite words. These matters should be investigated further.

The essential problem in both inexpressiveness cases is that we need both quantification (to account for a possibly infinite message alphabet) and, more importantly, the coupling of a reaction to the unique action that caused this reaction (to account for the counting of an unbounded number of inputs of the same message). Hence, in the first case we could not demand that to each $out(m)$ in a row of n there corresponded a unique $in(m)$. In the second case, messages could get lost, and hence it was not clear anymore to which $in(m)$ an $out(m)$ corresponded (in other words: several choices for the instances of m that were lost could be made).

4. Extensions of Temporal Logic

In this section we consider three extensions of linear time temporal logic to overcome the logical limitations of section 3.

One possibility is the addition of special data structures to characterize the internal behavior of a system, e.g. queues for FIFO-behavior, stacks for LIFO-behavior etcetera. One advocate of this approach is Lamport (see e.g. [L]). We note the following problems with this approach:

1. using an additional internal data structure is implementation biased and as such violates the abstractness requirement (see point 3 in section 1),
2. the behavior of the additional component is described by an additional formalism such as abstract data types, and hence the method looses its uniformity (point 5 in section 1),
3. for different applications we have to plug in different additional components which is in conflict with the generality requirement (see point 4 in section 1).

A second approach is to add special auxiliary variables and operations on them with fixed interpretations. One example of this is history variables with the prefix relation as in the work of Hailpern (see e.g. [Hai]). The main problem with this approach is that it is biased towards certain behaviors; for specifying FIFO this method is well suited, but awkward for other ordering

disciplines such as LIFO. In general one then has to use projections on histories to access the individual elements. What one would like to have is a set of operations on histories such that one can specify each application in terms of this set (such as done for specifying safety properties in [ZRE]). Again this is in conflict with the generality requirement.

Note that in these approaches incoming messages are implicitly made unique by their place in the data structure, respectively, the history. This resolves the coupling of a reaction to a unique action. In [KR] a third approach can be found in which the unique identification of incoming messages is explicitly assumed on beforehand, e.g. by means of *conceptual* time stamps. The advantages of doing this are threefold:

1. uniformity: the specifications remain purely temporal,
2. abstractness: the only propositions are $in(m)$ and $out(m)$ for all $m \in Messages$,
3. generality: in [KR] it is demonstrated that by slight changes of the specification we can describe different properties of systems (e.g. whether it can loose messages or not, whether the ordering is FIFO or LIFO etcetera, see below).

As a consequence of our decision to describe the relation between events in a purely temporal way, the resulting specifications can become rather elaborate. This might be alleviated by modularizing the specification of a system into groups of axioms describing a particular aspect (e.g. subcomponent) of this system.

We illustrate the method of [KR] by specifying FIFO and LIFO message passing systems, i.e. systems satisfying BA1-BA4 + FIFO/LIFO. First we formulate our assumption about the uniqueness of *incoming* messages as an axiom within our logic:

$G\ \forall m\ \neg\ (\ in(m)\ \wedge\ P\ in(m)\ )$.

For the specification of BA2-BA4 we can more or less mimic the specification using occurrences in section 3 (again BA1 is fulfilled by the nature of the formalization):

BA 2 $G\ \forall m\ \forall m'\ [((in(m)\ \wedge\ in(m'))\ \vee\ (out(m)\ \wedge\ out(m')))\ \rightarrow\ m = m'\ ]$

BA3',4 $G\ \forall m\ [\ out(m)\ \rightarrow\ P\ in(m)\ ]$

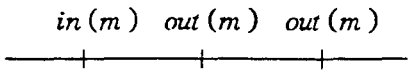
BA3'' $G\ \forall m\ \neg\ (\ out(m)\ \wedge\ P\ out(m)\ )$.

Notice that we split requirement BA3 (no creation of messages) into the following two cases:

BA3' no creation of altogether *new* messages,

BA3'' no *multiplication* of messages already present.

Axiom BA3',4 does not cover requirement BA3'' as is shown by the BA3''-illegal behavior



which is allowed by this axiom. Therefore we need a separate axiom BA3''. In section 3, axiom BA3,4 *did* cover both BA3' and BA3'' since it stated the correspondence between the n-th delivery of a message m and its n-th acceptance earlier on.

Next we specify FIFO, respectively LIFO.

FIFO $G\ \forall m\ \forall m'\ [(out(m)\ \wedge\ P\ out(m'))\ \rightarrow\ P\ (in(m)\ \wedge\ P\ in(m'))]$

LIFO $G\,\forall m\,\forall m'\,[(out\,(m)\wedge P\;out\,(m'))\rightarrow(P(in\,(m')\wedge P\;in\,(m))\vee P(out\,(m')\wedge\neg P\;in\,(m)))]$.

The specification of FIFO mimics the corresponding axiom in section 3, but is in this case *independent* of the perfectness of the system. The intuition behind the specification of LIFO (stack-like behavior) is as follows. If m' is earlier taken from the stack than m, then *either* m' was put on the stack when m was already there (the first disjunctive clause (because of BA3" we do not additionally need to require that m was not yet delivered at the moment of putting m' on the stack)) *or* m' was already taken from the stack before m was put on it (the second disjunctive clause). Note that the axioms for FIFO and LIFO become equivalent when it is additionally assumed that the capacity of the message passing system to store messages is 1 (since in that case the first disjunctive clause of LIFO is impossible). It is easy to check that the axiom for either FIFO or LIFO together with the axiom about the uniqueness of incoming messages imply the axiom for BA3".

Intuitively, all the formalized properties above are safety properties. It is nice to notice that all axioms above use only the temporal operators G and P and hence are safety properties according to the syntactical characterization of temporal formulae into safety and liveness properties of [LPZ]. When we want to formalize a typical liveness property such as being perfect the corresponding axiom uses the liveness operator F:

$G\;\forall m\;[\,in\,(m)\rightarrow F\;out\,(m)\,]$.

5. Conclusions

We proved several limitations of temporal logics for the specification of message passing systems. The counterexamples indicate that a necessary ingredient for such a specification is the ability to trace back (in time) every delivered message to its unique moment of acceptance. With this in mind one can take one of two directions. Either one argues that, because it is not expressive enough, temporal logic should be enriched with an additional formalism for reasoning about such systems, *or*, having identified the trouble spot, one makes some general assumptions about these systems that are strong enough to enable a purely temporal specification. The first course is taken by most researchers in the field. This might be caused by lack of recognition of the essential missing ingredients. The second course is attractive since the general assumption about message passing systems, viz. that incoming messages can be uniquely identified, can be translated into an axiom of the logic and hence can be reasoned with inside the formalism itself. One might view this axiom as representing an assumption about the environment of the system. From this viewpoint, the other axioms of the specification are then commitments of the system.

As to directions for future research, it would be interesting to find for each type of message passing system a temporal logic that is sufficient to specify merely this type. In this way one would get a correspondence between certain properties of message passing systems and the essential ingredients needed for (reasoning about) their temporal formalization.

Acknowledgements

The author wishes to thank Willem-Paul de Roever for his critical reading and suggested improvements, my other colleagues in the theoretical computer science group of the Eindhoven University of Technology (especially my roommate Ruurd Kuiper) for helpful discussions, and Wolfgang Thomas for providing information about the relationship between temporal logics and several other theories.

References

[Ada] *The programming language Ada. Reference manual*, LNCS 155, 1983.

[CHILL] *CHILL Recommendation Z.200 (CHILL Language Definition)*, C.C.I.T.T. Study Group XI, 1980.

[DHJR] T.Denvir, W.Harwood, M.Jackson, M.Ray, *The Analysis of Concurrent Systems*, Proceedings of a Tutorial and Workshop, Cambridge University, September 1983, LNCS 207, 1985.

[GPSS] D.Gabbay, A.Pnueli, S.Shelah, J.Stavi, *On the Temporal Analysis of Fairness*, 7th ACM POPL, pp. 163-173, 1980.

[Hai] B.T.Hailpern, *Verifying Concurrent Processes Using Temporal Logic*, Ph.D. Thesis, Stanford University, 1980.

[Har] D.Harel, *Statecharts: A Visual Approach to Complex Systems (Revised)*, Weizmann Institute of Science, CS 86-02, March 1986.

[K] J.A.W.Kamp, *Tense Logic and the Theory of Linear Order*, Ph.D. Thesis, University of California, Los Angeles, 1968.

[KR] R.Koymans, W.P. de Roever, *Examples of a Real-Time Temporal Logic Specification*, in [DHJR], pp. 231-251.

[L] L.Lamport, *STL/SERC Problems*, in [DHJR], pp. 252-270.

[LPZ] O.Lichtenstein, A.Pnueli, L.Zuck, *The Glory of The Past*, Logics of Programs '85, LNCS 193, pp. 196-218, 1985.

[P] A.Pnueli, *The Temporal Logic of Programs*, 18th FOCS, pp. 46-57, 1977.

[SCFG] A.P.Sistla, E.M.Clarke, N.Francez, Y.Gurevich, *Can Message Buffers Be Characterized in Linear Temporal Logic?*, 1st ACM PODC, pp. 148-156, 1982.

[SCFM] A.P.Sistla, E.M.Clarke, N.Francez, A.R.Meyer, *Can Message Buffers Be Axiomatized in Linear Temporal Logic?*, Information and Control 63, pp. 88-112, 1984.

[Th] W.Thomas, *Safety- and Liveness-Properties in Propositional Temporal Logic: Characterization and Decidability*, Rheinisch-Westfälische Technische Hochschule Aachen, April 1986.

[Tr] M.Trakhtenbrot, *Expression of Real Time Constraints within Statelan and Temporal Logic*, Review Report for Deliverable D6-1-1 of ESPRIT project 937: Debugging and Specification of Ada Real-Time Embedded Systems (DESCARTES), August 1986.

[W] P.Wolper, *Temporal logic can be more expressive*, Information and Control 56, pp. 72-99, 1983.

[ZRE] J.Zwiers, W.P. de Roever, P. van Emde Boas, *Compositionality and Concurrent Networks: Soundness and Completeness of a Proofsystem*, 12th ICALP, LNCS 194, pp. 509-519, 1985.

COMPUTING SCIENCE NOTES

In this series appeared :

No.	Author(s)	Title
85/01	R.H. Mak	The formal specification and derivation of CMOS-circuits
85/02	W.M.C.J. van Overveld	On arithmetic operations with M-out-of-N-codes
85/03	W.J.M. Lemmens	Use of a computer for evaluation of flow films
85/04	T. Verhoeff H.M.J.L. Schols	Delay insensitive directed trace structures satisfy the foam rubber wrapper postulate
86/01	R. Koymans	Specifying message passing and real-time systems
86/02	G.A. Bussing K.M. van Hee M. Voorhoeve	ELISA, A language for formal specifications of information systems
86/03	Rob Hoogerwoord	Some reflections on the implementation of trace structures
86/04	G.J. Houben J. Paredaens K.M. van Hee	The partition of an information system in several parallel systems
86/05	Jan L.G. Dietz Kees M. van Hee	A framework for the conceptual modeling of discrete dynamic systems
86/06	Tom Verhoeff	Nondeterminism and divergence created by concealment in CSP
86/07	R. Gerth L. Shira	On proving communication closedness of distributed layers

86/08	R. Koymans R.K. Shyamasundar W.P. de Roever R. Gerth S. Arun Kumar	Compositional semantics for real-time distributed computing (Inf.&Control 1987)
86/09	C. Huizing R. Gerth W.P. de Roever	Full abstraction of a real-time denotational semantics for an OCCAM-like language
86/10	J. Hooman	A compositional proof theory for real-time distributed message passing
86/11	W.P. de Roever	Questions to Robin Milner - A responder's commentary (IFIP86)
86/12	A. Boucher R. Gerth	A timed failure semantics for communicating processes
86/13	R. Gerth W.P. de Roever	Proving monitors revisited: a first step towards verifying object oriented systems (Fund. Informatica IX-4)
86/14	R. Koymans	Specifying passing systems requires extending temporal logic
87/01	R. Gerth	On the existence of sound and complete axiomatizations of the monitor concept
87/02	Simon J. Klaver Chris F.M. Verberne	Federatieve Databases
87/03	G.J. Houben J.Paredaens	A formal approach distributed information systems
87/04	T.Verhoeff	Delay-insensitive codes - An overview

Available Reports from the Theoretical Computing Science Group

	Author(s)	Title	Classification EUT	Classification DESCARTES
TIR83.1	R. Koymans, J. Vytopil, W.P. de Roever	Real-Time Programming and Synchronous Message passing (2nd ACM PODC)		
TIR84.1	R. Gerth, W.P. de Roever	A Proof System for Concurrent Ada Programs (SCP4)		
TIR84.2	R. Gerth	Transition Logic - how to reason about temporal properties in a compositional way (16th ACM FOCS)		
TIR85.1	W.P. de Roever	The Quest for Compositionality - a survey of assertion-based proof systems for concurrent progams, Part I: Concurrency based on shared variables (IFIP85)		
TIR85.2	O. Grünberg, N. Francez, J. Makowsky, W.P. de Roever	A proof-rule for fair termination of guarded commands (Inf.& Control 1986)		
TIR85.3	F.A. Stomp, W.P. de Roever, R. Gerth	The μ-calculus as an assertion language for fairness arguments (Inf.& Control 1987)		
TIR85.4	R. Koymans, W.P. de Roever	Examples of a Real-Time Temporal Logic Specification (LNCS207)		
TIR86.1	R. Koymans	Specifying Message Passing and Real-Time Systems (extended abstract)	CSN86/01	
TIR86.2	J. Hooman, W.P. de Roever	The Quest goes on: A Survey of Proof Systems for Partial Correctness of CSP (LNCS227)	EUT-Report 86-WSK-01	

TIR86.3	R. Gerth, L. Shira	On Proving Communication Closedness of Distributed Layers (LNCS236)	CSN86/07	
TIR86.4	R. Koymans, R.K. Shyamasundar, W.P. de Roever, R. Gerth, S. Arun Kumar	Compositional Semantics for Real-Time Distributed Computing (Inf.&Control 1987)	CSN86/08	
TIR86.5	C. Huizing, R. Gerth, W.P. de Roever	Full Abstraction of a Real-Time Denotational Semantics for an OCCAM-like Language	CSN86/09	PE.01
TIR86.6	J. Hooman	A Compositional Proof Theory for Real-Time Distributed Message Passing	CSN86/10	TR.4-1-1(1)
TIR86.7	W.P. de Roever	Questions to Robin Milner - A Responder's Commentary (IFIP86)	CSN86/11	
TIR86.8	A. Boucher, R. Gerth	A Timed Failure Semantics for Communicating Processes	CSN86/12	TR.4-4(1)
TIR86.9	R. Gerth, W.P. de Roever	Proving Monitors Revisited: a first step towards verifying object oriented systems (Fund. Informatica IX-4)	CSN86/13	
TIR86.10	R.Koymans	Specifying Passing Systems Requires Extending Temporal Logic	CSN86/14	PE.02
TIR87.1	R. Gerth	On the existence of sound and complete axiomatizations of the monitor concept	CSN87/01	