Slowdown Invariance of Timed Regular Expressions

Ingram Bondin
Dept. of Computer Science
University of Malta
ibon0001@um.edu.mt

Gordon J. Pace
Dept. of Computer Science
University of Malta
gordon.pace@um.edu.mt

Christian Colombo
Dept. of Computer Science
University of Malta
christian.colombo@um.edu.mt

Abstract

In critical systems, it is frequently essential to know whether the system satisfies a number of real-time constraints, usually specified in a real-time logic such as timed regular expressions. However, after having verified a system correct, changes in its environment may slow it down or speed it up, possibly invalidating the properties.

Colombo et al. [1] have presented a theory of slowdown and speedup invariance to determine which specifications are safe with respect to system retiming, and applied the approach to duration calculus. In this paper we build upon their approach, applying it to timed regular expressions. We hence identify a fragment of the logic which is invariant under the speedup or slowdown of a system, enabling more resilient verification of properties written in the logic.

1. Introduction

In the field of verification, one often wants to check whether a system's behaviour satisfies some particular constraint. The behaviour of a system can be envisaged as being the set of traces that it is able to execute. Traces can take various forms; In a well structured system, for example, a trace can be represented as a string of method calls.

Constraints on the behaviour of systems are often referred to as *properties*. A property can also be envisaged as characterising a set of traces, namely those traces which satisfy the constraint it represents. Properties are written using the formulas of some logic. Each formula acts as a concise way to identify a certain type of behaviour. As an example, consider the property expressed by the regular expression $(login \circ request \circ logout)^*$. This expression accepts all traces in which the user logs in before making a request, and in which he/she subsequently logs out. Due to the star operation, a user can make any number of these transactions without breaking the property. Any system trace which does not subscribe to the above pattern would be deemed invalid. Sometimes, properties also need to impose constraints on the actual duration of the events within the system. In order to express such properties, one needs special logics which take into account this element of time, known as *real-time logics*.

When the properties being used involve the element of time, certain problematic issues can arise [1]. These problems occur because systems can slow down or speed up due to changes in their environment. For example, a server which is being subjected to a heavy load will mean that the applications running on it will be less responsive. On the other hand, upgrading the hardware on which a program is running will speed that program up. Yet another scenario is that of runtime verification [2]. In this case, instrumenting a monitor within the system means that more code needs to be executed, slowing the system down. On the other hand if the monitor is removed after one has enough confidence in the system, the system will speed up.

Any process which can slow a system down or speed it up carries with it the risk of breaking real-time properties which previously held for that system. For example suppose that an industrial process needs a flame to burn for between 2 and 3 seconds. We might implement a program to control this operation, and we might have verified that it does satisfy the property. If the program slows down for some reason, the guarantees we have provided will be broken.

One way of reasoning about when properties can be broken and when they are safe from the effects of system retiming, is the theory of slowdown and speedup [1]. This theory allows one to prove which operators of a logic are unaffected by this phenomenon. Properties written using only these operators, are there-

fore guaranteed to hold even when the system is retimed. Although this theory can be easily applied to logics whose semantics are defined in terms of duration calculus [3] interpretations, a different strategy is needed to deal with logics whose semantics are defined in terms of some other model. This is the case for *timed regular expressions* (TRE) [4], which use an underlying model known as *signals*.

In this work we present such an approach which involves giving an interpretation based semantics to TRE, and then proving that these semantics are sound with respect to one another. Once soundness is proven, one can apply Colombo et. al's theory directly to the interpretation semantics to derive the behaviour of TRE under system retiming. Figure 1 illustrates the approach pictorially. The circles with the solid borders show the definitions in our possession. The circle with the broken border shows what has to be defined; whilst the broken lines show the proofs to be completed.

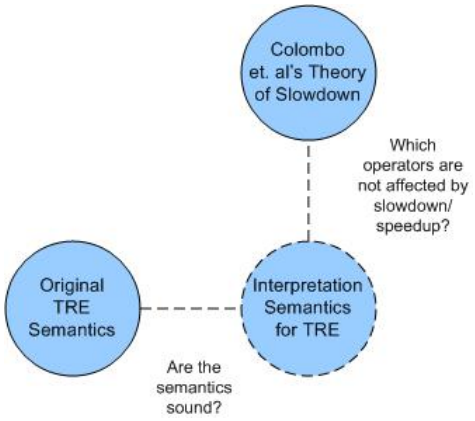

Figure 1. Using the Theory for Other Logics

This work is organised as follows. Section 2 discusses the signal model of time, timed regular expressions (TRE), and provides an outline of Colombo et al's theory. Following this, section 3 describes the aforementioned approach. Section 4 provides and discusses the resulting slowdown/speedup results for TRE. Section 5 outlines issues with the approach in section 3 and illustrates a different approach to the problem. Some issues with the second approach are also considered and future directions for research are suggested. Finally, section 6 concludes the report.

2. Background

2.1. Interpretations and Signals

We shall use the term *model of time* to denote some notion that enables us to describe the behaviour of a system as time passes. A model for the behaviour of a system is often defined over some finite set of symbols Σ. These symbols can be used to represent events, such as method calls. A duration calculus *interpretation* is one such model. It is essentially a total function $\Sigma \to \mathbb{T} \to \{true, false\}$ which given a symbol[1] and a time, will tell us whether that symbol is active at that point in time. We define $\mathbb{T}$, the set of all time points as being R_0^+. In duration calculus interpretations, multiple symbols can be active at the same time. An alternative model is that of a signal [4]. A signal is also defined over a set of symbols Σ, but at each point in time, one and only one event can be occurring. In a signal, the emphasis is on the fact that the system is performing a sequence of events, one after the other, with each having its own duration. For example, the signal $a^1 b^{7.5}$ tells us that the system has taken 1 unit of time executing method a and 7.5 units of time executing the the second. On the otherhand, in an interpretation, the emphasis is on what is happening at each point in time. Figure 2 shows a signal on the left and a duration calculus interpretation on the right over the alphabet $\Sigma = \{a, b, c\}$.

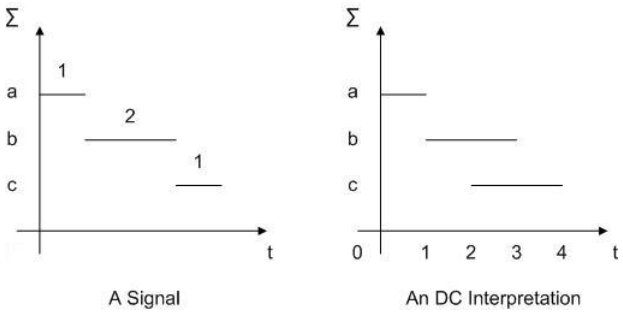

Figure 2. Signals and Interpretations

Signals have a finite length, and can be appended to one another. We denote the length of a signal ξ by $|\xi|$ and the concatenation of two signals ξ_1 and ξ_2 by $\xi_1 \circ \xi_2$. We shall now describe the semantics given to the temporal logic called timed regular expressions in [4], which operates in terms of signals.

2.2. Timed Regular Expressions

Timed regular expressions are a simple logic for describing the behaviour of timed systems. They are similar to regular expressions, with the addition of some operators related to reasoning about the time domain. Each timed regular expression is defined over some alphabet and characterises a set of signals. We

1. Known in duration calculus as a variable.

shall denote the set of signals given by an expression ψ by $|[\psi]|$. The most basic expression is a symbol from the alphabet. This yields the set of signals, in which the event represented by that symbol happens and lasts for any duration.

$$|[a]| = \{a^r \mid r \in \mathbb{R}^+\}$$

From such expressions, more complex ones can be built. For example, the concatenation of two symbols $\circ$ will generate the set of signals for each of the symbols, and will then concatenate each signal from the first set to each signal of the second set.

$$|[\psi_1 \circ \psi_2]| = \{\xi_1 \circ \xi_2 \mid \xi_1 \in |[\psi_1]| \wedge \xi_2 \in |[\psi_2]|\}$$

There are other operators which work in a similar way to regular expressions, such as union ($\cup$) and kleene star (*). One operator missing from regular expressions is the interval restriction operator $\langle\rangle_{[a,b]}$. Using this operator, one can force the signals resulting from an expression to be of a certain length.

$$|[\langle\psi\rangle_{[a,b]}]| = |[\psi]| \cap \{\xi : Signal \mid |\xi| \in [a,b]\}$$

Timed regular expressions also feature intersection ($\cap$), since this is necessary for the expression of certain desirable timed constraints. We shall now review important elements of the theory of slowdown and speedup.

2.3. The Theory of Slowdown and Speedup

In Colombo et al.'s model, when a system is slowed down, system events start taking longer to occur and last for a longer time. When a system is speeded up, events seem to take a shorter time before occurring, and also last for a shorter time. If one were to visualise the operation of a system as an interpretation, a slowed down version of the function would look like the original version but stretched in the time domain. Similarly a speeded up interpretation would look like the original, but compressed in the time domain. This is shown in Figure 3. The relationship between an interpretation and its slowed down/speeded up version is characterised by the notion of a time transform.

2.4. Time Transforms

A time transform *s* is a total continuous function $\mathbb{T} \rightarrow \mathbb{T}$, (where $\mathbb{T}$ is the set of all time points). Such a function will remap each point in time to some other point. This remaping has to satisfy certain constraints, the most important of which is to preserve the original ordering of the points. Time transforms can

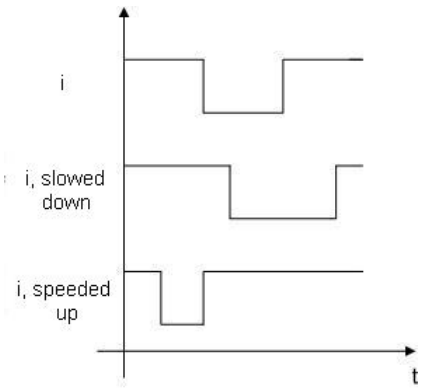

Figure 3. Time Transforms and Interpretations

be applied to interpretations. This represents the effect of modifying the time at which system events occur and the duration for which they last. If a time transform stretches an interpretation, then it is called a *time stretch transform*. If it compresses it, it is called a *time compress transform*. If i is an interpretation, and s a time transform, we denote a transformed interpretation by i_s. In order to model system slowdown/speedup as described earlier, if an event is active at some time, then it must also be active at the new time in the retimed system. This relationship between an interpretation and its transformed version is thus expressed as the following law (where $i(a,t)$ tells us whether an event a is active at time t in an interpretation i).

$$\forall t : \mathbb{T}, a : \Sigma.\ i(a,t) = i_s(a, s(t))$$

The next important contribution of the theory of slowdown and speedup is to give a precise meaning to the statement *a property is not affected by system slowdown/speedup*.

2.5. Stretch/Compress Truth Preservation

In section 1 it was said that a formula of some logic could be used to denote a property which some system would then be required to satisfy. It was also claimed that certain properties did not run the risk of being broken by slowdown/speedup. One can say that a formula representing a property is *interval stretch truth preserving* if whenever the system satisfies the property, slowing the system down by any amount cannot break the property. As an example consider the property *once event X has happened, event Y cannot happen before 3 seconds have elapsed*. This property is stretch truth preserving. This occurs because if a system satisfies this property, slowing it down means that it will take longer for event Y to occur, such that

the property cannot be broken. Expressed more formally, we say that a formula ψ representing a property is interval stretch truth preserving (written $istr_t(\psi)$), if for every interpretation, when that interpretation satisfies the property over some interval of time [b,e] (written $i \vDash_{[b,e]} \psi$), any slowed down interpretation also satisfies it over the stretched interval [s(b),s(e)].

$$istr_t(\psi) \triangleq \forall i, s.\ i \vDash_{[b,e]} \psi \Rightarrow i_s \vDash_{[s(b),s(e)]} \psi$$

A similar concept holds for system speedup, called *interval compress truth preservation* $(icom_t)$. The reader is referred to [1] for more information. If one can identify a subset of the operators of the logic which are stretch truth preserving, then any properties written using only these operators are also guaranteed to be stretch truth preserving. We shall now see how to apply the theory for logics that do not work using interpretations.

3. Using the Theory with TRE

In the previous section have seen that the theory of slowdown and speedup works with logics which use interpretations as their model of time. However, in section 2.2 we determined that TRE use a different model, that of signals. In the introduction we have seen that one solution is to give a new semantics to TRE based on interpretations. Once this is done, one would then have to prove that these two semantics are sound with respect to each other. We shall outline this procedure below. For a full account of how this approach was applied to TRE, including the actual proofs, consult [5].

3.1. Preliminary Steps

The original semantics of TRE allow a formula to characterise a number of signals. The new semantics should allow TRE to characterise the collection of interpretations equivalent to those signals. To this end, one needs two functions *stoi* and *itos* which given a signal will yield an interpretation and vice-versa. These functions encode the information present in one model into the other model. Now, the signal model is less powerful than the interpretation model, since its symbols must be mutually exclusive. Since TRE only have the power to characterise signals, the functions will only need to convert between signals and a restricted version of interpretations (one in which symbols also occur with mutual exclusion).

To make sure that the constructions are sane, one should prove that these functions are inverses of one another. This also makes the functions bijective, guaranteeing that for each signal there is one and only one (restricted) interpretation, and vice-versa.

3.2. Defining an Interpretation Semantics

After this has been done, an interpretation semantics is given to the logic in question. In such a semantics, formulas of the logic will characterise a number of interpretations through the notion of *satisfaction over an interval*. An interpretation will satisfy a formula between the times b and e (written $i \vDash_{[b,e]} \psi$) if it is of a certain form between those times. For example, an interpretation could satisfy the TRE a under these semantics, if for the duration of the interval, the interpretation attains the value of a.

$$i \vDash_{[b,e]} a \triangleq \forall t : \mathbb{T}.\ b \leq t < e \Rightarrow i(a,t)$$

When defining interpretation semantics, it is important to keep in mind that in order for the semantics to be consistent with the original ones, an interpretation has to satisfy a certain expression whenever its corresponding signal encoding is in the set characterised by that expression in the original semantics.

3.3. Soundness

If the above is done correctly, it becomes possible to prove that the semantics are in fact sound with one another. For example, if proving that signal semantics are sound with interpretation semantics, one would need to show that each time a signal is in the set characterised by a TRE, its corresponding interpretation is characterised by the same expression under the interpretation semantics. Naïvely, we could try to express this as:

$$\forall \xi, \psi.\ \xi \in |[\psi]| \Rightarrow stoi(\xi) \vDash \psi$$

The problem with this however, is that while the interpretation semantics uses satisfaction *over an interval*, this concept is missing from the signal based semantics. In this case, what we really need to express is that, if the part of the signal between the times b and e is in the set $|[\psi]|$, then the corresponding interpretation also satisfies ψ over [b,e]. This can be written as:

$$\forall \xi, \psi.\ slice(\xi)_{[b,e]} \in |[\psi]| \Rightarrow stoi(\xi) \vDash_{[b,e]} \psi$$

where the $slice$ function gives us a way to obtain the part of the signal which lies within the interval. Defining such a function however, might not always

be straightforward. Once all this work has been completed, slowdown and speedup results can be derived for the operators found in the logic. In the next section, we show the results obtained for timed regular expressions.

4. Slowdown/Speedup Results for TRE

Table 1 shows which operators are stretch/compress truth preserving.

	$istr_t$	$icom_t$
a $\in \Sigma$	✓	✓
$\psi_1 \circ \psi_2$	✓	✓
$\psi_1 \cup \psi_2$	✓	✓
$\psi_1 \cap \psi_2$	✓	✓
ψ^*	✓	✓
$\langle\psi\rangle_{[b,\infty]}$ (l.b.)	✓	×
$\langle\psi\rangle_{[0,e]}$ (u.b.)	×	✓
$\langle\psi\rangle_{[b,e]}$	×	×

Table 1. Slowdown and Speedup Results for TRE

As we said before, properties built from a combination of stretch/compress truth preserving operators only, will inherit this trait. For example, consider the property $(\langle a\rangle_{[3,\infty]} \circ b)^*$. Here the allowed behaviour consists of an occurrence of an event a followed by event b for any number of times, where a must last for at least 3 seconds. This property is stretch truth preserving. On the other hand consider $(\langle a\rangle_{[3,\infty]} \circ b)^* \cup \langle c\rangle_{[0,2]}$, in which we also allow behaviours where event c must last for not more than 2 seconds. In this case, a term which is not stretch truth preserving was added to the expression, meaning that no slowdown guarantees can be given for this property.

What is important to note is that any expression which does not contain the interval restriction operator, is unaffected by the problems of slowdown and speedup. For those properties that do contain timing constraints, the following observations can be made. If the time constraint contains an upper bound (u.b.) only, then, it is safe from the effects of system speedup. If it contains a lower bound (l.b.) only, then it is safe from the effects of system slowdown. If it contains both, then it is safe from neither. To this end, when using TRE the best one can hope for is to be able to write properties using just lower bounds or just upper bounds, if one wants to have an immunity with respect to either slowdown or speedup.

5. Discussion and Future Work

We shall now discuss some issues involved in applying the theory to logics which do not have an interpretation based semantics. First of all, one needs to define the functions for moving between models of time. These functions may not be always simple to define. Essentially they are constructions showing how one can translate from one model of time to another. For example in TRE the function for converting from interpretations to signals has to consider the fact that interpretations are piecewise continuous functions whilst signals are a list of symbols and their durations. The function *itos* works by mapping each 'piece' of the function to a signal symbol. This entails a discretisation step, in which the points where the interpretation changes value are found and the segments between these points extracted. This process complicates the sanity proof because it necessitates the proof of additional lemmas about the behaviour of the discretisation step.

The second issue to consider is that in the approach presented, the interpretation semantics have to be chosen with care so that they will be consistent with the signal semantics. This is necessary since otherwise soundness cannot be proven. The third point to consider is that proving soundness, which in the case of TRE was done by structural induction, can take a substantial effort. This was especially evident for the proof of the base case of the logic. Besides this fact, as we have seen in section 3.3, the expressions for soundness will contain a *slice* function. In order for this to be manipulated in the proofs, several lemmas will need to be proven regarding its behaviour.

An observation worthy of note is that, in this approach, giving a new semantics and proving soundness has to be done for each new logic; little can be reused. The above is not the only possible approach. A second approach (see Figure 4) involves defining a new theory of slowdown and speedup which uses the same underlying model as the new logic. This theory must then be proven to be sound with respect to Colombo et al.'s theory. This approach has the advantage that it can be reused. This means that once a new theory of slowdown and speedup is devised with a certain underlying model, it can be reused for all logics whose semantics are given in terms of that model. In our case, this would mean that it could be applied to logics whose semantics are grounded in signals.

The major element which has to be defined in the new theory is a way in which to apply time transforms to the new model, such as signals. In doing so, a difficulty is encountered since time transforms operate on absolute time points, while signals only illustrate each symbol and its duration. To transform a signal, we need to transform the durations of its symbols. For each symbol, we need to find its start and end point,

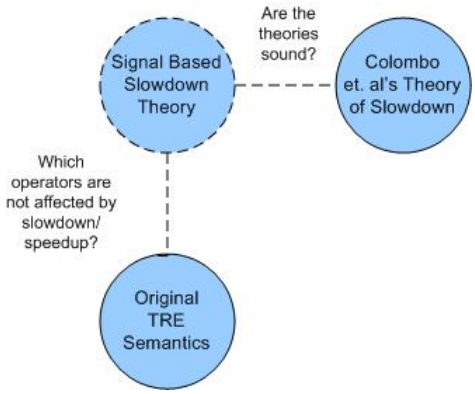

Figure 4. An Alternative Approach

transform them both, and use their difference as the transformed duration. If s is a time transform, we can express the application of a time transform on a signal as follows.

$$(\sigma_1^{d_1}\sigma_2^{d_2}\dots\sigma_n^{d_n})_s = \sigma_1^{s(S_1)}\sigma_2^{s(S_2)-s(S_1)}\dots\sigma_n^{s(S_n)-s(S_{n-1})}$$

where $S_n = \sum_{i=1}^{n} d_i$. The term S_n will locate the endpoint of the n^{th} symbol.

For the soundness proof, one wants to show that using Colombo et. al's time transforms on interpretations has the same effect as using the new time transform on the corresponding signals and vice-versa. This can be shown by proving that if one starts with a signal, transforms it (using time transforms on signals), and converts it to an interpretation (via *stoi*), then this interpretation is the same one obtained by first converting the signal (via *itos*), and then transforming it (using time transforms on interpretations). The other direction for soundness is similar. Formally we can express the above as $stoi(\xi_s) = xtoi(\xi)_s$ and $itos(i_s) = itos(i)_s$

Thus in the second approach, one has to give a new definition for time transforms and to prove that time transforms distribute through the functions for converting between models of time. This approach seems to be simpler; recall that in the first approach, soundness needs to be proven for each base expression and operator of the logic, besides having to craft the interpretation semantics correctly. On the other hand, the second approach only needs two proofs. However, the difficulty involved in these proofs depends heavily on the complexity of the functions *stoi* and *itos*. In fact when working with the second approach for TRE it was noted that the time transforms did not distribute cleanly through the internals of these functions.

Once the groundwork was completed for the first approach, proving which operators were slowdown/speedup truth preserving and which were not, was relatively easy. If the second approach was applied to derive these results for TRE, it would act as an independent confirmation of these results. More usefully, it would give a measure of how complex it is to apply the second approach to logics such as TRE. As it is, it remains to be seen whether the simplicity gained with the second approach would then be matched by a corresponding increase in complexity during the slowdown/speedup proofs.

If the second approach proves to be feasible, then it should be tried out on other logics which use the signal model for the definition of their semantics. This would substantiate the claim that new theories of slowdown and speedup can be effectively reused for other logics sharing the same model.

6. Conclusion

In this paper we have outlined how one can apply Colombo et al.'s theory of slowdown and speedup to other logics, such as TRE. The results obtained for TRE showed that immunity from slowdown/speedup problems depends on the presence of the interval restriction operator in a property, as well as on the bounds used with this operator. It was found that when using time constraints, one can either provide immunity from slowdown by using lower bounds only, or immunity from speedup by using upper bounds only. To this end it would be useful for a study to be performed in order to determine whether and what useful properties can be written under these constraints.

References

[1] C. Colombo, G. J. Pace, and G. Schneider, "Safe runtime verification of real-time properties," in *FORMATS*, ser. Lecture Notes in Computer Science, vol. 5596, 2009, pp. 103–117.

[2] S. Colin and L. Mariani, "Run-time verification," in *Model-Based Testing of Reactive Systems*, 2004, pp. 525–555.

[3] Z. ChaoChen, C. Hoare, and A. Ravn, "A calculus of durations," *Information Processing Letters*, vol. 40, no. 5, pp. 269–276, 1991.

[4] E. Asarin, P. Caspi, and O. Maler, "A kleene theorem for timed automata," in *LICS '97: Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science*. Washington, DC, USA: IEEE Computer Society, 1997, p. 160.

[5] I. Bondin, "Real-time logics and slowdown invariance for runtime verification," B.Sc. Final Year Project, University of Malta, May 2009.