

# A calculus of durations \*

Zhou Chaochen \*\*, C.A.R. Hoare and Anders P. Ravn \*\*\*

*Oxford University Computing Laboratory, Programming Research Group, 11 Keble Road, Oxford OX1 3QD, United Kingdom*

Communicated by J. Misra  
Received 4 February 1991  
Revised 8 July 1991

## Abstract

Zhou Chaochen, C.A.R. Hoare and A.P. Ravn, A calculus of durations, Information Processing Letters 40 (1991) 269–276.

The purpose of the calculus of durations is to reason about designs and requirements for time-critical systems, without explicit mention of absolute time. Its distinctive feature is reasoning about integrals of the durations of different states within any given interval. The first section introduces the running example, of leakage in a gas burner. The second section defines and axiomatises the proposed calculus as an extension of interval temporal logic. The third section applies it to the problem described in the introduction. The fourth section briefly surveys alternative calculi.

*Keywords:* Real-time systems, software engineering, specification languages, temporal logic

## 1. Introduction

A central step in development of software for real-time, embedded systems is to formalize and reason about real-time requirements. The duration calculus uses the integrated duration of states within a given interval of time to describe such requirements. Let for instance *Leak* denote an undesirable but unavoidable state of some system, perhaps a flow of unlit gas from the nozzle of a gas burner. A safety engineer may calculate that the ventilation required for normal combustion would prevent dangerous accumulation of gas, provided that the proportion of time spent in the leak state is not more than one twentieth of the elapsed time. The interval over which the system is observed should be at least one minute long—otherwise the requirement would be violated immediately on the start of a leak.

It is convenient to represent a state of a real-time system as a function from reals (representing *time*) to  $\{0, 1\}$ , where 1 denotes that the system is in the state, and 0 denotes that the system is not in the state. The observation interval is taken to be bounded, and conventionally we let  $b$  stand for its start and  $e$  stand for its end. So we formulize the safety requirement as:

$$\text{Req-1} \quad e - b \geq 60 \text{ sec.} \Rightarrow 20 \int_b^e \text{Leak}(t) dt \leq (e - b).$$

An engineer with expert knowledge about gas combustion should be able to certify a statement of requirements expressed at this level of abstraction [13,14].

\* Work partially supported by the Commission of the European Communities (CEC) under the ESPRIT programme in the field of Basic Research Action proj. no. 3104: "ProCoS: Provably Correct Systems".

\*\* Permanent address: Software Institute, Academia Sinica, Beijing 100080, China.

\*\*\* Permanent address: Dept. of Computer Science, Building 344, Technical University Denmark, DK-2800 Lyngby, Denmark.

Turning next to the task of design, certain decisions must be taken about how the agreed requirements are to be met. For example, a leak should be detectable and stoppable within one second; and then it is acceptable to wait thirty seconds before risking another leak by switching the gas on again. Using  $c$  and  $f$  to denote the end points of an arbitrary subinterval of the observation interval  $b$  to  $e$ , these two decisions can be formalized:

$$\text{Des-1} \quad \int_c^f \text{Leak}(t) dt = (f - c) \Rightarrow f - c \leq 1 \text{ sec.}$$

$$\text{Des-2} \quad \text{Leak}(c) \wedge \text{Leak}(f) \wedge (\exists t \cdot c < t < f \wedge \neg \text{Leak}(t)) \Rightarrow 30 \text{ sec.} \leq f - c.$$

Clearly the conjunction of these two formulae implies the original requirement, a fact which should be proved and certified by the designing engineer before implementation proceeds. The objective of this paper is to provide a simple notation for expressing such theorems, and a simple calculus for proving them.

## 2. The duration calculus

The simplicity of temporal logic derives from elision of time variables. In interval temporal logic [9,4] the variables  $b$  and  $e$ , denoting the end points of an arbitrary observation interval, are elided from expressions such as **Req-1**. Its main operator is "chop", here denoted  $\frown$ , in terms of which the usual modalities  $\diamond$  and  $\square$  can be defined. The duration calculus adopts this as primitive and introduces integrals of states over intervals as variables in the interval temporal logic.

### 2.1. States

Simple states are Boolean variables, defined for every instant of time, and of finite variability. A state is finitely varying on a (bounded) interval when the interval can be partitioned into a finite number of subintervals in which the state is constant almost everywhere [1]. Finite variability implies integrability.

A state is an expression formed from simple states by the usual Boolean operators, for example  $\neg P$  which is an abbreviation of  $1 - P$ . Finite variability is preserved by finite application of the Boolean operators.

### 2.2. Durations

**Definition 1.** For an arbitrary state  $P$  and an arbitrary observation interval  $[b, e]$ , the *duration of  $P$*  is defined

$$\int P \triangleq \int_b^e P(t) dt.$$

It follows that  $\int 1 = e - b$ . Thus our requirement **Req-1** can be encoded more succinctly as:

$$\int 1 \geq 60 \text{ sec.} \Rightarrow 20 \int \text{Leak} \leq \int 1.$$

We now begin to list some simple theorems of analysis, which are sufficiently useful to be taken as axioms in a calculus of durations.

**Axiom 1.**  $\int 0 = 0$ .

**Axiom 2.** For an arbitrary state  $P$ ,

$$\int P \geq 0.$$

**Axiom 3.** For arbitrary states  $P$  and  $Q$ ,

$$\int P + \int Q = \int (P \vee Q) + \int (P \wedge Q).$$

Using these axioms, we prove readily properties like:

**Theorem 2.** For an arbitrary state  $P$ ,

- (a)  $\int P + \int \neg P = \int 1$ ,
- (b)  $\int P \leq \int 1$ .

2.3. Predicates

A state  $P$  can be lifted to a simple predicate  $[P]$  in interval temporal logic.

**Definition 3.**  $[P] \triangleq (\int P = \int 1) \wedge (\int 1 > 0)$ .

This means that  $P$  holds almost everywhere in a proper observation interval. We use  $[ \ ]$  to denote the predicate which is true only for point intervals.

**Definition 4.**  $[ \ ] \triangleq (\int 1 = 0)$ .

In the following, we use the conventional logical connectives on predicates as well as states. This does not lead to ambiguities because a state will occur only after the duration symbol ( $\int$ ) or within predicate brackets ( $[ \ ]$ ).

It is easy to prove that an observation interval is either a proper interval or a point interval:

**Theorem 5.**  $[1] \vee [ \ ]$ .

**Theorem 6.** For any state  $P$ ,

- (a)  $[P] \Rightarrow (\int \neg P = 0)$ ,
- (b)  $[ \ ] \Rightarrow (\int P = 0)$ .

The following theorem expresses monotonicity of  $\int$ :

**Theorem 7.** For states  $P$  and  $Q$ ,

$$[P \Rightarrow Q] \Rightarrow (\int P \leq \int Q).$$

The basic interval temporal logic operator is "chop" ( $\frown$ ).

$\Gamma P \Rightarrow Q \Gamma$   
 $\Rightarrow [P \vee Q]$   
 $\Rightarrow \int P \vee Q = \int 1 \wedge \int 1 > 0$   
 $\Rightarrow \int P + \int Q - \int P \wedge Q = \int 1$   
 $\Rightarrow \int 1 - \int P + \int Q - \int (P \wedge Q) = \int 1$   
 $\Rightarrow \int P = \int Q - \int (P \wedge Q)$   
 $\Rightarrow \int P = \int Q - \epsilon \quad \epsilon > 0$   
 $\Rightarrow \int P \leq \int Q$   
□

**Definition 8.** Let  $B$  and  $C$  be predicates with the special free variables  $b$  and  $e$

$$(B \wedge C) \hat{=} \exists m \cdot (b \leq m \leq e) \wedge B[m/e] \wedge C[m/b].$$

As usual  $[m/e]$  denotes substitution of  $m$  for  $e$ , and  $[m/b]$  substitution of  $m$  for  $b$  in the predicates. Thus  $B \wedge C$  is true of any interval which can be split into two subintervals, of which the first satisfies  $B$  and the second satisfies  $C$ .

The definition of  $\wedge$  is exactly analogous to that of relational composition [2], and has the same properties; for example it is associative, has the point interval ( $[]$ ) as unit, *false* as zero, and distributes through disjunction ( $\vee$ ), but not through conjunction ( $\wedge$ ).

The conventional modal operators can also be defined:

**Definition 9.** For a predicate  $B$ ,

$$\diamond B \hat{=} true \wedge B \wedge true.$$

This is true of any interval in which  $B$  holds for some subinterval.

**Definition 10.** For a predicate  $B$ ,

$$\square B \hat{=} \neg \diamond \neg B.$$

This is true of any interval when  $B$  holds in every subinterval of it.

Monotonicity of  $\wedge$  is a general property of interval temporal logic; that is, for predicates  $B, C$  and  $D$ ,

$$\square (B \Rightarrow C) \Rightarrow \square ((B \wedge D \Rightarrow C \wedge D) \wedge (D \wedge B \Rightarrow D \wedge C)).$$

The basic axiom relating  $\wedge$  and  $\int$  states that the duration of a state in an interval is the sum of its duration in each partition into subintervals.

**Axiom 4.** Let  $P$  be a state and  $r, s$  nonnegative reals

$$\left( \int P = r + s \right) \Leftrightarrow \left( \int P = r \right) \wedge \left( \int P = s \right).$$

Accepting this axiom, we can prove properties like the following:

**Theorem 11.** For a state  $P$  and nonnegative reals  $r, s, t$  and  $u$

$$\left( r \leq \int P \leq s \right) \wedge \left( t \leq \int P \leq u \right) \Rightarrow \left( r + t \leq \int P \leq s + u \right).$$

The next theorem states arbitrary divisibility of intervals; i.e., density of time.

**Theorem 12.** For a state  $P$ ,

$$[P] \wedge [P] \Leftrightarrow [P].$$

*since we associate IR with Time we can*

It is useful to have an induction rule which extends a hypothesis over adjacent subintervals. Such a rule relies on the finite variability of states and on the finitude of the intervals, that any interval can be split into a *finite* alternation of state  $P$  and state  $\neg P$ .

**Induction rule.** Let  $X$  denote a formula letter occurring in the formula  $R(X)$  and let  $P$  be a state. If  $[[]]$  holds, and  $R(X \vee X \wedge [P] \vee X \wedge [\neg P])$  is provable from  $R(X)$ , then  $R(true)$  holds.

A symmetric rule uses  $[P] \wedge X$  for  $X \wedge [P]$  and  $[\neg P] \wedge X$  for  $X \wedge [\neg P]$ . Induction can be used to prove that a proper interval ends with either  $P$  or  $\neg P$ .

**Theorem 13.** For a state  $P$ ,

$$(true \wedge [P]) \vee (true \wedge [\neg P]) \vee [ ]$$

As induction hypothesis, the proof uses  $R(X) \triangleq X \Rightarrow (true \wedge [P]) \vee (true \wedge [\neg P]) \vee [ ]$ .

The four axioms and the induction rules can be shown to constitute a sound and (relative) complete formal system of durations [15].

### 3. Example

Using the notations introduced above, the requirement and design decisions for the Gas Burner of Section 1 can be coded in the duration calculus.

**Req-1**  $\int 1 \geq 60 \text{ sec.} \Rightarrow \text{Safe},$

where  $\text{Safe} \triangleq 20 \int \text{Leak} \leq \int 1$ .

**Des-1**  $\square ([\text{Leak}] \Rightarrow \int 1 \leq 1 \text{ sec.})$

**Des-2**  $\square ((\diamond [\text{Leak}]) \wedge (\diamond [\neg \text{Leak}]) \wedge (\diamond [\text{Leak}]) \Rightarrow \int 1 \geq 30 \text{ sec.})$

or  $\square ([\text{Leak}] \wedge [\neg \text{Leak}] \wedge [\text{Leak}] \Rightarrow \int 1 \geq 30)$

An application of the calculus is a proof that the design decisions imply the requirement. The proof rests on two lemmas.

**Lemma 14.**  $\text{Des-1} \wedge \text{Des-2} \Rightarrow \square (\text{Triple} \Rightarrow \text{Safe})$ , where

$$\text{Triple} \triangleq ([\text{Leak}] \wedge \int 1 \leq 0.5 \text{ sec.}) \wedge [\neg \text{Leak}] \wedge [\text{Leak}]$$

**Lemma 15.**  $[\text{Leak}] \Rightarrow [\text{Leak}] \wedge ([\text{Leak}] \wedge \int 1 \leq 0.5 \text{ sec.})$

The proof of Lemma 14 is

$$\begin{aligned} &\square (\text{Triple} \Rightarrow \text{Safe}) && (\wedge\text{-monotonicity, Theorems 2, 6, Des-1}) \\ &\Rightarrow (\diamond [\text{Leak}] \wedge (\diamond [\neg \text{Leak}]) \wedge (\diamond [\text{Leak}]) && \\ &\quad \wedge (\int \text{Leak} \leq 0.5) \wedge (\int \text{Leak} = 0) \wedge (\int \text{Leak} \leq 1) && (\text{Des-2, Theorem 11}) \\ &\Rightarrow (\int 1 \geq 30) \wedge (\int \text{Leak} \leq 1.5) && (\text{arithmetic}) \\ &\Rightarrow \text{Safe} \end{aligned}$$

The proof of Lemma 15 is

$$\begin{aligned} &[\text{Leak}] && (\text{Definition 3}) \\ &\Rightarrow \int \text{Leak} = \int 1 > 0 && (\text{arithmetic}) \\ &\Rightarrow \exists r, u > 0 \cdot u \leq 0.5 \wedge \int \text{Leak} = \int 1 = r + u && (\text{Axioma 4}) \\ &\Rightarrow \exists r, u > 0 \cdot u \leq 0.5 \wedge (\int \text{Leak} = \int 1 = r) && \\ &\quad \wedge (\int \text{Leak} = \int 1 = u) && (\text{Definition 3}) \\ &\Rightarrow \exists r, u > 0 \cdot [\text{Leak}] \wedge ([\text{Leak}] \wedge \int 1 \leq 0.5) && (\text{predicate logic}) \\ &\Rightarrow [\text{Leak}] \wedge ([\text{Leak}] \wedge \int 1 \leq 0.5) \end{aligned}$$

The main proof:  $(\text{Des-1} \wedge \text{Des-2} \wedge f1 \geq 60) \Rightarrow \text{Safe}$  uses Theorem 13 to divide the proof into three cases:  $\lceil \ \rceil \vee (\text{true} \wedge \lceil \text{Leak} \rceil) \vee (\text{true} \wedge \lceil \neg \text{Leak} \rceil)$ . The first case  $\lceil \ \rceil$  is trivial, because the premiss  $f1 \geq 60$  is not satisfied by a point interval. The two other cases are proven using the induction rule with  $R(X)$  defined by the predicate

$$\begin{aligned} & \text{Des-1} \wedge \text{Des-2} \wedge f1 \geq 60 \\ & \Rightarrow \\ & (X \wedge \lceil \text{Leak} \rceil \Rightarrow \text{Safe}) \\ & \wedge (X \wedge \lceil \neg \text{Leak} \rceil \Rightarrow \text{Safe}) \\ & \wedge (X \wedge \lceil \neg \text{Leak} \rceil \Rightarrow (\lceil \neg \text{Leak} \rceil \vee \text{Safe} \wedge (\lceil \text{Leak} \rceil \wedge f1 \leq 0.5) \wedge \lceil \neg \text{Leak} \rceil)) \end{aligned}$$

The base case  $R(\lceil \ \rceil)$  holds because  $\lceil \text{Leak} \rceil$  and **Des-1** do not satisfy the premiss  $f1 \geq 60$ , while  $\lceil \neg \text{Leak} \rceil$  satisfies *Safe* trivially. As  $R(X)$  distributes over disjunction, the induction step can be separated into a proof of  $R(X \wedge \lceil \text{Leak} \rceil)$  and  $R(X \wedge \lceil \neg \text{Leak} \rceil)$ .

The induction step with  $R(X \wedge \lceil \text{Leak} \rceil)$  has three parts:

$$\begin{aligned} & X \wedge \lceil \text{Leak} \rceil \wedge \lceil \text{Leak} \rceil \quad (\text{Theorem 12}) \\ & \Rightarrow X \wedge \lceil \text{Leak} \rceil \quad (R(X)) \\ & \Rightarrow \text{Safe} \\ & X \wedge \lceil \text{Leak} \rceil \wedge \lceil \neg \text{Leak} \rceil \quad (\text{Theorem 6}) \\ & \Rightarrow (X \wedge \lceil \text{Leak} \rceil) \wedge (f\text{Leak} = 0) \quad (R(X)) \\ & \Rightarrow \text{Safe} \wedge (f\text{Leak} = 0) \quad (\text{Axioma 4, arithmetic}) \\ & \Rightarrow \text{Safe} \end{aligned}$$

and

$$\begin{aligned} & X \wedge \lceil \text{Leak} \rceil \wedge \lceil \neg \text{Leak} \rceil \quad (\text{Lemma 15}) \\ & \Rightarrow X \wedge \lceil \text{Leak} \rceil \wedge (\lceil \text{Leak} \rceil \wedge f1 \leq 0.5) \wedge \lceil \neg \text{Leak} \rceil \quad (R(X)) \\ & \Rightarrow \text{Safe} \wedge (\lceil \text{Leak} \rceil \wedge f1 \leq 0.5) \wedge \lceil \neg \text{Leak} \rceil \end{aligned}$$

The induction step with  $R(X \wedge \lceil \neg \text{Leak} \rceil)$  has two parts:

$$\begin{aligned} & X \wedge \lceil \neg \text{Leak} \rceil \wedge \lceil \text{Leak} \rceil \quad (R(X)) \\ & \Rightarrow (\lceil \neg \text{Leak} \rceil \vee \text{Safe} \wedge (\lceil \text{Leak} \rceil \wedge f1 \leq 0.5) \wedge \lceil \neg \text{Leak} \rceil) \wedge \lceil \text{Leak} \rceil \quad (\wedge\text{-distributivity}) \\ & \Rightarrow \lceil \neg \text{Leak} \rceil \wedge \lceil \text{Leak} \rceil \vee \text{Safe} \wedge \text{Triple} \quad (\text{Theorem 6, Des-1, Lemma 14}) \\ & \Rightarrow (f\text{Leak} = 0) \wedge (f\text{Leak} \leq 1) \vee \text{Safe} \wedge \text{Safe} \quad (f1 \geq 60, \text{Axioma 4}) \\ & \Rightarrow \text{Safe} \end{aligned}$$

and

$$\begin{aligned} & X \wedge \lceil \neg \text{Leak} \rceil \wedge \lceil \neg \text{Leak} \rceil \quad (\text{Theorem 12}) \\ & \Rightarrow X \wedge \lceil \neg \text{Leak} \rceil \quad (R(X)) \\ & \Rightarrow \text{Safe} \wedge (\lceil \neg \text{Leak} \rceil \vee \text{Safe} \wedge (\lceil \text{Leak} \rceil \wedge f1 \leq 0.5) \wedge \lceil \neg \text{Leak} \rceil) \end{aligned}$$

Thus  $R(\text{true})$ , which completes the proof.

#### 4. Other approaches

A specification in timed CSP [12,3] uses  $s$  to stand for an arbitrary trace of *timed events*. For simplicity let the events  $\uparrow \text{Leak}$  and  $\downarrow \text{Leak}$  be the only events occurring in  $s$ ; the design decisions can then be encoded

$$\forall t, u: \mathbf{R} \cdot \langle (t, \uparrow \text{Leak}), (u, \downarrow \text{Leak}) \rangle \text{ in } s \Rightarrow (u - t) \leq 1 \text{ sec.}$$

and

$$\forall t, u: \mathbf{R} \cdot \langle (t, \downarrow Leak), (u, \uparrow Leak) \rangle \text{ in } s \Rightarrow (u - t) \geq 30 \text{ sec.}$$

Timed CSP has a specification oriented semantics which permits proof that a program is correct. It has a very high expressive power, and it might be possible to interpret duration formulae as abstraction of formulae like the ones above.

In Real Time Logic [6], the  $i$ th occurrence of an event  $a$  is denoted by  $@(a, i)$ . Assuming the initial state is  $\neg Leak$ , we can specify our design:

$$\forall i \cdot @( \downarrow Leak, i ) - @( \uparrow Leak, i ) \leq 1 \text{ sec.}$$

and

$$\forall i \cdot @( \uparrow Leak, i + 1 ) - @( \downarrow Leak, i ) \geq 30 \text{ sec.}$$

RTL has good expressive power, and is quite convenient to describe systems where there is a simple correlation (e.g. identity) between the event numbers of related events.

Pnueli and Harel [10] suggest use of an increasing (temporal) variable  $T$  to stand for the current time, and a finite number of global variables  $x, y, z, \dots$  to denote time instants. So the design decisions can be expressed:

$$(T = x \wedge Leak) \Rightarrow \diamond((T \leq x + 1) \wedge \neg Leak)$$

and

$$\neg(T = x \wedge Leak \wedge \diamond(\neg Leak \wedge \diamond((T < x + 30) \wedge Leak))).$$

Koyman's metric temporal logic [8], used by Hooman and Widom [5] for program specifications, has timed until operators  $\mathcal{U}_{\leq t}$  and  $\mathcal{U}_{\geq t}$ .  $P\mathcal{U}_{\leq t}Q$  ( $P\mathcal{U}_{\geq t}Q$ ) means that  $P$  holds from the current time  $c$ , and continually until  $Q$  holds at some time  $t'$ , where  $t' \leq c + t$  ( $t' \geq c + t$ ). The first design decision is easily expressed

$$Leak \mathcal{U}_{\leq 1} \neg Leak$$

and it is also possible to express the second one

$$Leak \mathcal{U}((\neg Leak) \mathcal{U} Leak) \Rightarrow Leak \mathcal{U}((\neg Leak) \mathcal{U}_{\geq 30} Leak)$$

In all these approaches time is observed at the moments when state transitions take place. Durations are then given by arithmetic expressions in time variables.

## 5. Conclusion

The most distinctive characteristic of the Duration Calculus is the simple way of describing proportional requirements such as **Req-1**. Secondly, it is easy to express invariant properties independent of initial states or initialisation transients. Thirdly there is complete symmetry between past and future time. The last two are properties of interval temporal logic.

The Calculus has been used to specify requirements and designs for a few real-time systems, e.g. [11]; but no full analysis has been made of its expressive power. A proof that a delivered program meets the design decisions which constitute its specification has not been attempted. For this, the behaviour of the program should be expressible in the same notation as its specification, so that correctness is established by simple implication. We will therefore need a specification-oriented semantics for the programming language, a topic which will be the subject of a companion paper [7].

### Acknowledgment

The authors are grateful to Ron Koymans and the referees for excellent advice on the presentation of this paper.

### References

- [1] H. Barringer, R. Kuiper and A. Pnueli, A really abstract concurrent model and its temporal logic, in: *Proc. 13th ACM Symp. on the Principles of Programming Languages* (1986) 173–183.
- [2] S.M. Brien, A relational calculus of intervals, M.Sc. Thesis, PRG, Oxford University Computing Laboratory, September 1990.
- [3] S. Schneider, Correctness and communication of real-time systems, Ph.D. Thesis 1989, Techn. Monograph PRG-84, Oxford University Computing Laboratory, March 1990.
- [4] R.W.S. Hale, Programming in temporal logic, Ph.D. Thesis, University of Cambridge, October 1988.
- [5] J. Hooman and J. Widom, A temporal-logic based compositional proof system for real-time message passing, in: *PARLE '89*, Lecture Notes in Computer Science 366 (Springer, Berlin).
- [6] F. Jahanian and A.K.-L. Mok, Safety analysis of timing properties in real-time systems, *IEEE Trans. Software Eng.* 12 (9) (1986) 890–904.
- [7] Jifeng He, Specification-oriented semantics for ProCoS programming language  $PL^{time}$ , ProCoS Tech. Rept. PRG/OU HJF7, ESPRIT BRA 3104, May 1991.
- [8] R. Koymans, Specifying real-time properties with metric temporal logic, *J. Real-time Systems* 2 (1990).
- [9] B. Moszkowski, A temporal logic for multi-level reasoning about hardware, *IEEE Comput.* 18 (2) (1985) 10–19.
- [10] A. Pnueli and E. Harel, Applications of temporal logic to the specification of real time systems, in: M. Joseph, ed., *Proc. Symp. Formal Techn. in Real-Time and Fault-Tolerant Systems*, Lecture Notes in Computer Science 331 (Springer, Berlin, 1988) 84–98.
- [11] A.P. Ravn and H. Rischel, Requirements capture for embedded real-time systems, in: *Proc. of IMACS MCTS 91 Symp., Vol. 2*, Villeneuve-d'Asq, France (1991) 147–152.
- [12] G.M. Reed and A.W. Roscoe, Metric spaces as models for real-time concurrency, in: *Mathematical Foundations of Programming*, Lecture Notes in Computer Science 298 (Springer, Berlin, 1987) 331–343.
- [13] E.V. Sørensen, A.P. Ravn and H. Rischel, Control program for a gas burner: Part 1: Informal requirements, ProCoS Case Study 1, Tech. Rept. ID/DTH EVS2, ESPRIT BRA 3104, March 1990.
- [14] E.V. Sørensen, On the specification of dependability requirements involving time integration, ProCoS Working Paper, October 1989.
- [15] Zhou Chaochen and M.R. Hansen, A note on completeness of the Duration Calculus, ProCoS Working Paper, February 1991.