Logical Interpolation and Projection onto State in the Duration Calculus
(Extended Abstract)

Dimitar P. Guelev*

June 6, 2003

Introduction

The classical interpolation theorem of Craig (cf. e.g. [ChK73]) states that if $\varphi \Rightarrow \psi$ is a valid first order predicate logic formula, then there exists a first order formula θ built using only non-logical symbols occurring in both φ and ψ and, possibly, equality, such that the formulas $\varphi \Rightarrow \theta$ and $\theta \Rightarrow \psi$ are valid too. A formula θ with this property is called an *interpolant* between φ and ψ. Similar statements apply to a variety of non-classical and modal logics. In [Gue01] it was shown that abstract time interval temporal logics admit a different kind of interpolation theorems. In these theorems the parts φ and ψ are written using a shared vocabulary of rigid symbols and disjoint copies of the same vocabulary of flexible symbols. Instead of having shared flexible symbols between φ and ψ, it is required that the pairs of corresponding symbols from the two copies of the flexible symbol vocabulary occurring in φ and ψ evaluate to the same predicates, functions and constants *within specified intervals of time*. Given that, it is shown that interpolants between φ and ψ can be restricted to specify properties of the considered intervals of time only too.

Let the languages built using the two copies of the flexible symbol vocabulary mentioned above be called $\mathbf{L}_1$ and $\mathbf{L}_2$, respectively. Let, given a formula α from $\mathbf{L}_1$, the result of replacing its flexible symbols except the flexible constant ℓ (see the definition of *ITL* in Section 1) by their counterparts from $\mathbf{L}_2$ be denoted by α'. Let Φ be a finite set of formulas from $\mathbf{L}_1$. Let φ and ψ be in $\mathbf{L}_1$ too. Let c_0, c_1 and c_2 be rigid constants, which are shared by $\mathbf{L}_1$ and $\mathbf{L}_2$. Then in the case of *ITL* the interpolated formula has the form

$$\left(\ell = c_1; \bigwedge_{\chi\in\Phi} \Box\forall(\chi \Leftrightarrow \chi'); \ell = c_2\right) \wedge \ell = c_0 \Rightarrow (\varphi \Rightarrow \psi'), \tag{1}$$

where $\forall$ denotes the universal closure of its argument formula. The antecedent of the implication (1) is to express that the formulas from Φ are equivalent to their counterparts from $\mathbf{L}_2$ within the interval defined using c_0, c_1 and c_2. In particular, Φ can be chosen to consist of atomic formulas so that the above antecedent would express the equality of the interpretations of the corresponding pairs of flexible symbols from $\mathbf{L}_1$ and $\mathbf{L}_2$ within this interval. According to a theorem from [Gue01], if (1) is valid in *ITL*, then there exists a formula θ in $\mathbf{L}_1$ such that the formulas

$$\left(\begin{array}{c}\ell = c_0 \wedge c_1 + c_2 \leq c_0 \wedge \\ 0 \leq c_1 \wedge 0 \leq c_2\end{array}\right) \Rightarrow (\varphi \Rightarrow (\ell = c_1; \theta; \ell = c_2)) \text{ and } (\ell = c_1; \theta; \ell = c_2) \wedge \ell = c_0 \Rightarrow \psi$$

*School of Computer Science, The University of Birmingham, UK, E-mail: D.P.Guelev@cs.bham.ac.uk; Institute of Mathematics and Informatics of the Bulgarian Academy of Sciences. Project funded by the Future and Emerging Technologies arm of the IST Programme FET Open Scheme. Research on the topic of this article was partially supported through Contract No. I-1102/2001 by the Ministry of Education and Science of the Republic of Bulgaria.

are valid too. Note that here the interpolant θ occurs in the same context as $\bigwedge\limits_{\chi\in\Phi} \Box\forall(\chi \Leftrightarrow \chi')$ in (1) which can be used to express the equality of interpretations of pairs of corresponding flexible symbols. This way θ is restricted to specify a property of the subinterval of the reference interval where this equality is supposed to hold. It can be additionally restricted to contain only non-logical symbols s occurring in both φ and ψ. The theorem as in [Gue01] does not include this restriction, but it can be very easily achieved using almost the same proofs. Thus the Craig and the interval-related conditions get combined.

Craig interpolation for *ITL* can be viewed as a special case of this form of interval-related interpolation. Let both φ and ψ be formulas from $\mathbf{L}_1$. Let Φ consist of the formulas of the kind $p(x_1,\dots,x_n)$, $f(x_1,\dots,x_n) = x_{n+1}$ and $d = x_1$, where p, f, and d stand for the flexible predicate symbols, flexible function symbols and the flexible constant symbols occurring in φ, respectively. Let the rigid constant c occur neither in φ nor in ψ. Then $\models_{ITL} \varphi \Rightarrow \psi$ is equivalent to

$$\models_{ITL} \left(\ell = 0; \bigwedge_{\chi\in\Phi} \Box\forall(\chi \Leftrightarrow \chi'); \ell = 0\right) \wedge \ell = c \Rightarrow (\varphi \Rightarrow \psi').$$

Interpolation can be understood in the context of formal verification as follows. Let $[\![S]\!]$ and $[\![S']\!]$ be formulas which describe the possible behaviours of some simultaneously running parts S and S' of a system in terms of their observable signals. Let α and β describe properties of the system's runs in terms of the signals of S and S', respectively. Then

$$\models ([\![S]\!] \wedge \alpha) \Rightarrow ([\![S']\!] \Rightarrow \beta)$$

means that the property α of the behaviour of S implies β about the behaviour of S', because of the two components' having shared signals. An interpolant for the above implication can be regarded as an explicit description of the interaction between S and S' which makes α cause β *in terms of the signals S and S' share*. In the case of interval-related interpolation we can additionally restrict S and S' to share signals within part of the considered runs only, and an interpolant would be a description of what S does when α holds to cause β *during that part* of the runs.

The modality $(.;.)$ of *ITL* is known as *introspective*, because it allows access to subintervals of the reference interval only. Neighbourhood Logic [ZH96], has operators to access intervals outside the reference interval and admits a form of interval-related interpolation where a corresponding antecedent specifies that certain formulas from $\mathbf{L}_1$ are equivalent to their $\mathbf{L}_2$-counterparts everywhere in the past relative to a certain time point. Accordingly, interpolants are also restricted to specify properties of the past relative to that time point. The precise formulation and proof of that theorem can be found in [Gue01]. Obviously, conjunctions of contexts of the kind $(\ell = c_1; [\,]; \ell = c_2) \wedge \ell = c_0$ can form more complex antecedents

$$\bigwedge_{j\in J} \left(\ell = c_{1,j}; \bigwedge_{\chi\in\Phi_j} \Box\forall(\chi \Leftrightarrow \chi'); \ell = c_{2,j}\right) \wedge \ell = c_0 \Rightarrow \dots$$

in the interpolated formulas, and, following the pattern of the proofs from [Gue01], it can be shown that implications with such antecedents can be interpolated by appropriate vector interpolants $\langle\theta_j : j \in J\rangle$, with a component formula θ_j to occur in each of the contexts $(\ell = c_{1,j}; [\,]; \ell = c_{2,j}) \wedge \ell = c_0$.

In this paper we present a more flexible generalisation of interval-related interpolation. In this new form the parts of the time domain where pairs of corresponding symbols of the languages $\mathbf{L}_1$ and $\mathbf{L}_2$ are required to be equal themselves are specified by a means of the interpretation of a distinguished flexible symbol. Thus it is achieved that the positions of these parts need not be fixed in the the formula to be interpolated. The kind of flexible non-logical symbols that we use to specify the above parts of time domains are Duration Calculus (*DC*, [ZHR91]) *state variables.* State variables are the basic ingredient used to extend *ITL* to *DC*. They are interpreted as boolean functions on *time points*, unlike the other flexible non-logical *ITL* symbols, which depend on *intervals* for their interpretations. State variables are restricted to evaluate to functions which have the *finite variability property.* This means that, given a state variable, every bounded interval of time, can be partitioned into finitely many subintervals within each of which the state variable evaluates to a constant. This makes state variables an appropriate means of specifying finite sequences of subintervals. *DC* allows the formation of boolean combinations of state variables called *state expressions*. State expressions evaluate to boolean functions of time, their values being the corresponding boolean combinations of the values of the involved state variables. The finite variability property extends from state variables to state expressions.

In [DVH99] *DC* was extended by a *projection* operator which, given a formula φ and a state expression S, returns the truth value of φ at the interval obtained by gluing the parts of the reference interval where S holds with the interpretations of the non-logical symbols occurring in φ transferred from that original reference interval. In this setting φ is called the *projected formula* and the interval it is evaluated at is called the *projected interval*. This kind of projection in *DC* can be viewed as analogous to the operator Π which was introduced to discrete time *ITL* in [HMM83]. It greatly facilitates the specification of interleaving. An extensive study of projection onto state in *DC* can be found in [GD02]. Another group of projection operators for *ITL* and *DC* take formulas instead of state expressions to define projected intervals [Mos86, Mos95, He99, BT00, Gue00a]. Formulas in the scope of a projection operator can obviously specify only properties of the parts of the reference interval which participate in the construction of the projected interval. This makes projection suitable as the context to restrict interpolants.

In this paper we first give brief formal definitions of abstract time *ITL* and abstract time *DC*. Next, we intoduce projection onto state in *DC*. We only define projection on the subset of *DC* where state variables are the only kind of flexible non-logical symbols. Projection can be defined on arbitrary *DC* vocabularies [GD02], but then it involves cases where the interpretation of some symbols in the projected interval depend on the interpretations of these symbols *outside* the reference interval, and therefore is not strictly introspective. We choose to avoid this phenomenon, because it is not consistent with the intended meaning of our result. We establish a specialised form of Craig interpolation for *DC* by translating *DC* into a suitable ω-theory in some appropriately extended *ITL* vocabulary. Projection-related interpolation in *DC* is a generalised form of interval-related interpolation where the role of the contexts $(\ell = c_{1,j}; [\ ]; \ell = c_{2,j}) \wedge \ell = c_0$ is taken by the operator of projection onto state in *DC*. Our proof of projection-related interpolation for *DC* is by reducing it to this specialised form of Craig interpolation. Finally, we show that interpolants can be found constructively in the $\lceil P \rceil$-subset of the extension DC^* of *DC* by iteration.

1 Preliminaries

1.1 Interval Temporal Logic

Interval Temporal Logic was first introduced for the case of discrete time in [Mos85]. The completeness of a Hilbert-style proof system for an abstract time version of *ITL* was first demonstrated in [Dut95]. Sequent systems for *ITL* [Ras02] cannot be cut-free, because the propositional subset of *ITL* is undecidable. Abstract time *ITL* is a classical first order modal logic with fixed domains and one normal binary modality $(.;.)$, known as *chop*.

Languages Given a first order vocabulary of *constant symbols* c, d, ..., *function symbols* f, g, ..., *predicate symbols* R, ..., and countably many *individual variables* x, y, ..., terms t and formulas φ in the corresponding *ITL* language are defined by the BNFs:

$t ::= c \mid x \mid f(t, \ldots, t) \qquad \varphi ::= \bot \mid R(t, \ldots, t) \mid \varphi \Rightarrow \varphi \mid (\varphi; \varphi) \mid \exists x \varphi$

Non-logical symbols are either *rigid* or *flexible*, depending on the type of their interpretation, as it becomes clear below. Every *ITL* vocabulary contains the rigid constant 0, the flexible constant ℓ, the rigid binary function symbol $+$ and equality $=$.

Frames, models and satisfaction A *time domain* is a linearly ordered set. Given a time domain $\langle T, \leq \rangle$, we denote the set $\{[\tau_1, \tau_2] : \tau_1, \tau_2 \in T, \tau_1 \leq \tau_2\}$ by $\mathbf{I}(T)$. A *duration domain* is a system of the kind $\langle D, 0^{(0)}, +^{(2)} \rangle$ which satisfies the following axioms:

$(D1)$	$x+(y+z)=(x+y)+z$	$(D4)$	$x+y=0 \Rightarrow x=0$
$(D2)$	$x+0=x,\ 0+x=x$	$(D5)$	$\exists z(x+z=y \vee y+z=x)$
$(D3)$	$x+y=x+z \Rightarrow y=z,\ x+z=y+z \Rightarrow x=y$		$\exists z(z+x=y \vee z+y=x)$

Given a time domain $\langle T, \leq\rangle$ and a duration domain $\langle D, 0, +\rangle$, a function $m : \mathbf{I}(T) \to D$ is called a *measure function*, if the following properties hold for all $\sigma, \sigma' \in \mathbf{I}(T)$:

$(M1)$ $\quad \min\sigma = \min\sigma' \wedge m(\sigma) = m(\sigma') \Rightarrow \max\sigma = \max\sigma'$

$(M2)$ $\quad \max\sigma = \min\sigma' \Rightarrow m(\sigma) + m(\sigma') = m(\sigma \cup \sigma')$

$(M3)$ $\quad m(\sigma) = x + y \Rightarrow \exists\tau \in \sigma\ m([\min\sigma, \tau]) = x$

An *ITL frame* is a tuple of the form $\langle\langle T, \leq\rangle, \langle D, 0, +\rangle, m\rangle$ where $\langle T, \leq\rangle$ is a time domain, $\langle D, 0, +\rangle$ is a duration domain and $m : \mathbf{I}(T) \to D$ is a measure function.

Given an *ITL* frame F with its components named as above, and an *ITL* language $\mathbf{L}$, a function I on the vocabulary of $\mathbf{L}$ is an *interpretation* of $\mathbf{L}$ into F, if it satisfies the following conditions:

$I(c) \in D$, $I(f) : D^n \to D$, $I(R) : D^n \to \{0,1\}$ for rigid c, n-ary rigid f, R;

$I(c) : \mathbf{I}(T) \to D$, $I(f) : \mathbf{I}(T) \times D^n \to D$, $I(R) : \mathbf{I}(T) \times D^n \to \{0,1\}$ for flexible c, n-ary flexible f, R;

$I(x) \in D$ for individual variables x; $\quad I(=)$ is $=$, $I(0) = 0$, $I(+) = +$, and $I(\ell) = m$.

Given an *ITL* language $\mathbf{L}$, an *ITL model for* $\mathbf{L}$ is a pair of the form $\langle F, I\rangle$ where F is an *ITL* frame and I is an interpretation of $\mathbf{L}$ into F. Given an *ITL* language $\mathbf{L}$, a model $M = \langle\langle T, \leq\rangle, \langle D, 0, +\rangle, m, I\rangle$ for it, and an interval $\sigma \in \mathbf{I}(T)$, the value $I_\sigma(t)$ of terms t in $\mathbf{L}$ is defined by induction on the construction of t as follows:

$I_\sigma(x)$	$=$	$I(x)$	for individual variables x
$I_\sigma(c)$	$=$	$I(c)$	for rigid constants c
$I_\sigma(c)$	$=$	$I(c)(\sigma)$	for flexible constants c
$I_\sigma(f(t_1, \ldots, t_n))$	$=$	$I(f)(I_\sigma(t_1), \ldots, I_\sigma(t_n))$	for rigid n-place f
$I_\sigma(f(t_1, \ldots, t_n))$	$=$	$I(f)(\sigma, I_\sigma(t_1), \ldots, I_\sigma(t_n))$	for flexible n-place f

Given an interpretation I of an *ITL* language $\mathbf{L}$ into a frame F, a symbol s from $\mathbf{L}$ and an object a of the type of s in F, we denote the interpretation which assigns a to s and is equal to I for all the other symbols from $\mathbf{L}$ by I_s^a. Given a time domain $\langle T, \leq\rangle$ and $\sigma_1, \sigma_2 \in \mathbf{I}(T)$ such that $\max\sigma_1 = \min\sigma_2$, we denote $\sigma_1 \cup \sigma_2$ by $\sigma_1;\sigma_2$. The relation $M, \sigma \models \varphi$ where $M = \langle F, I\rangle$ is an *ITL* model for some language $\mathbf{L}$ which contains φ and $\sigma \in \mathbf{I}(T)$ is defined by induction on the construction of φ as in the standard way (we assume that the components of F are named as above). The clause about $(.;.)$ is

$M, \sigma \models (\varphi;\psi)$ iff $M, \sigma_1 \models \varphi$ and $M, \sigma_2 \models \psi$ for some $\sigma_1, \sigma_2 \in \mathbf{I}(T)$ such that $\sigma_1;\sigma_2 = \sigma$

Given a frame F, we denote its components by $\langle T_F, \leq_F\rangle$, $\langle D_F, 0_F, +_F\rangle$ and m_F, respectively.

Abbreviations First order logic abbreviations and infix notation are used in *ITL* in the ordinary way. These include $\top$, $\neg$, $\wedge$, $\vee$, $\Leftrightarrow$ and $\forall$. The following abbreviations, are specific to $(.;.)$:

$\Diamond\varphi \rightleftharpoons (\top;\varphi;\top)$, $\Box\varphi \rightleftharpoons \neg\Diamond\neg\varphi$

$(\varphi_1;\varphi_2;\ldots;\varphi_n) \rightleftharpoons (\varphi_1;\ldots;(\varphi_{n-1};\varphi_n)\ldots)$

$\varphi^0 \rightleftharpoons \ell = 0$, $\varphi^{k+1} \rightleftharpoons (\varphi;\varphi^k)$

1.2 Abstract Time Duration Calculus

The Duration Calculus was first introduced for the case of real time in [ZHR91]. The relative completeness of a Hilbert-style proof system for *DC* with respect to real time was first demonstrated in [HZ92]. A comprehensive survey of *DC* can be found in [HZ97]. Abstract time *DC* was introduced and the ω-completeness of a Hilbert-style proof system for it was demonstrated in [Gue98]. That proof system contains an ω-rule, which can be regarded as an explicit description of the intended deductive power of the induction rules in the system from [HZ92].

Languages A *DC* vocabulary extends an *ITL* vocabulary by a set of *state variables* P, Q, ... State variables are used to construct *state expressions* S, which are defined by the BNF:

$S ::= \mathbf{0} \mid P \mid S \Rightarrow S$

The BNF for *DC* formulas is the same as that for *ITL* formulas. The BNF for *DC* terms extends that for *ITL* terms by including duration terms $\int S$, which are formed using state expressions:

$t ::= c \mid x \mid \int S \mid f(t, \ldots, t)$

Frames, models and satisfaction Abstract time *DC* frames are the same as abstract time *ITL* frames. Given a frame $\langle\langle T, \leq\rangle, \langle D, 0, +\rangle, m\rangle$ and a language $\mathbf{L}$ a *DC* interpretation I of the vocabulary of $\mathbf{L}$ into F is like an *ITL* interpretation on the *ITL* non-logical symbols in $\mathbf{L}$, and maps every state variable P to a function $I(P) : T \to \{0, 1\}$. $I(P)$ is required to have the following *finite variability property*:

> For every $\sigma \in \mathbf{I}(T)$ there exist $\sigma_1, \ldots, \sigma_n \in \mathbf{I}(T)$ such that $\sigma = \sigma_1; \ldots; \sigma_n$ and $I(P)$ is constant on $[\min \sigma_i, \max \sigma_i)$, $i = 1, \ldots, n$.

DC models are like *ITL* models, the only difference being that their second component is a *DC* interpretation and therefore assigns values to state variables too. The following equalities define the value $I_\tau(S)$ of a state expression S at time $\tau \in T$ under interpretation I:

$I_\tau(\mathbf{0}) = 0$, $I_\tau(P) = I(P)(\tau)$ for state variables P, $I_\tau(S_1 \Rightarrow S_2) = \max\{1 - I_\tau(S_1), I_\tau(S_2)\}$.

We use the following technical definition to extend I_σ to duration terms:

Definition 1 Let $h : T \to \{0, 1\}$ have the finite variability property and $\sigma \in \mathbf{I}(T)$. Given $\sigma_1, \ldots, \sigma_n \in \mathbf{I}(T)$ such that $\sigma = \sigma_1; \ldots; \sigma_n$ and h is constant on $[\min \sigma_i, \max \sigma_i)$, $i = 1, \ldots, n$, we put

$$\int\limits_{\min \sigma}^{\max \sigma} h(\tau) d\tau = \sum_{\substack{i=1,\ldots,n \\ I_{min\sigma_i}(S)=1}} m(\sigma_i)$$

Obviously finite variability holds for functions of the kind $\lambda\tau.I_\tau(S)$. We put

$$I_\sigma(\textstyle\int S) = \int\limits_{\min \sigma}^{\max \sigma} I(S)(\tau) d\tau.$$

The clauses about I_σ on other kinds of terms and those about $\models$ are as for *ITL*.

Abbreviations The connectives $\neg$, $\vee$, $\wedge$ and $\Leftrightarrow$ are used as abbreviations in state expressions in the usual way. The following abbreviations are also frequently used:

$\mathbf{1} \rightleftharpoons \mathbf{0} \Rightarrow \mathbf{0}$, $\lceil S \rceil \rightleftharpoons \int S = \ell \wedge \ell \neq 0$

Proof system Abstract time *DC* is a conservative extension to *ITL*. An ω-complete proof system for it was first presented in [Gue98]. It was obtained by adding several axioms and an ω-rule to the proof system for *ITL* known from [Dut95]. Various forms of the rule and the list of axioms can be chosen to obtain such an ω-complete system. In the rest of this paper we employ the following variant, which is more convenient to prove our results:

(ω') $\dfrac{\forall k < \omega\ [(\int S = \ell \vee \int S = 0)^k / R]\alpha}{[\top/R]\alpha}$

$(DC0)$ $\ell = 0 \Rightarrow \int S = 0$

$(DC1)$ $\int \mathbf{0} = 0$

$(DC2')$ $\int S_1 = 0 \vee \int S_2 = \ell \Rightarrow \int(S_1 \Rightarrow S_2) = \ell$

$(DC3')$ $\int S_1 = \ell \wedge \int S_2 = 0 \Rightarrow \int(S_1 \Rightarrow S_2) = 0$

$(DC4')$ $(\int S = x; \int S = y) \Rightarrow \int S = x + y$

In (ω') above, R denotes an arbitrary 0-place predicate symbol.

2 Projection onto State in Abstract Time *DC*

In this paper we present projection-related interpolation for *DC* vocabularies whose flexible symbols are ℓ and state variables only. The definition of projection onto state we adopt here is a simplified abstract time variant of that from [GD02]. *DC* projection formulas have the form (φ/S) where φ is a formula and S is a state expression. We need some technical definitions in order to define $\models$ on the new kind of formulas. Let $\langle F, I\rangle$ be a model for the *DC* language $\mathbf{L}$ whose only flexible symbols are ℓ and state variables.

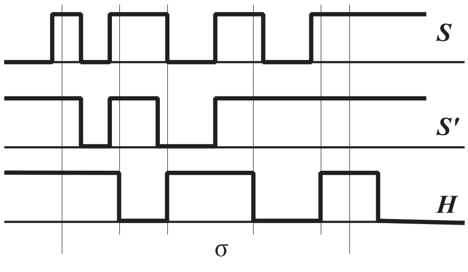

Figure 1: $\sigma \models (\Box\forall x((\int S = x) \Leftrightarrow (\int S' = x))/H)$, or, equivalently, $\sigma \models \ell = \int(H \Rightarrow (S \Leftrightarrow S'))$

Definition 2 Let $\sigma \in \mathbf{I}(T_F)$, $\tau \in \sigma$, and $h : T_F \to \{0,1\}$ have the finite variability property. Let $\gamma^{h,\sigma} : \sigma \to \sigma$ be defined by the equality $m([\min\sigma, \gamma^{h,\sigma}(\tau)]) = \int\limits_{\min\sigma}^{\tau} h(\tau')d\tau'$. Let $\sigma^h = [\min\sigma, \gamma^{h,\sigma}(\max\sigma)]$. Let $I^{\sigma,h}$ be an interpretation of $\mathbf{L}$ into F such that

$I^{\sigma,h}(s) = I(s)$ for all rigid symbols s; $\quad I^{\sigma,h}(\ell) = m_F$; $\quad I^{\sigma,h}(P)(\max\sigma^h) = I(P)(\max\sigma)$;
$I^{\sigma,h}(P)(\tau') = I(P)(\tau)$, for $\tau' \in [\min\sigma, \max\sigma^h)$, if $m([\min\sigma,\tau']) = \int\limits_{\min\sigma}^{\tau} h(\tau'')d\tau''$ and $h(\tau) = 1$.

We are not interested in defining $I^{\sigma,h}(P)$ outside σ^h. Informally σ^h is an initial subinterval of σ which has the duration of h in σ as its total duration, and the interpretation $I^{\sigma,h}$ on σ^h is obtained by "gluing" the parts of I on the subintervals of σ where h evaluates to 1. An exception is made for the case of $\int\limits_{\min\sigma}^{\max\sigma} h(\tau)d\tau = 0$, which, however, has no effect on the values of duration terms at subintervals of σ^h under $I^{\sigma,h}$. Given a formula φ and a state expression H, we put
$\langle F, I\rangle, \sigma \models (\varphi/H)$ iff $\langle F, I^{\sigma,\lambda\tau.I_\tau(H)}\rangle, \sigma^{\lambda\tau.I_\tau(H)} \models \varphi$.

3 Interpolation in Abstract Time *DC*

The main result in this paper is an interpolation theorem for formulas of the kind

$$\left(\left(\bigwedge_{\chi\in\Phi} \Box\forall(\chi \Leftrightarrow \chi')\right)/H\right) \wedge \ell = c \Rightarrow (\varphi \Rightarrow \psi') \tag{2}$$

where $(.)'$ stands for the result of replacing the flexible symbols of the vocabulary of φ, ψ and the formulas from Φ, except those occurring in the state expression H, by corresponding ones from a disjoint vocabulary, like in the introduction. To illustrate (2), Figure 3 shows the simultaneous runs of two systems with states S and S' respectively, which satisfy the instance $\Phi = \{\int S = x\}$ of the antecedent of (2) at the reference interval σ. Then $\varphi \Rightarrow \psi'$ can be the description of some influence exercised by one of the systems on the other, because of their interaction during H-subintervals of σ.

Following the pattern from earlier work given in the introduction, it would be natural to expect that for Φ, H, φ and ψ such that (2) is a valid formula in *DC* there exists a formula θ of some accordingly restricted form such that

$$\varphi \wedge \ell = c \Rightarrow \theta \text{ and } \theta \wedge \ell = c \Rightarrow \psi$$

are valid too. Yet the kind of restrictions on θ we want cannot be achieved, unless some more restrictions be imposed on φ and ψ, because the considered subset of *DC*, both in the abstract and in the real time case, does not have Craig's interpolation property, which would have been otherwise possible to derive by choosing H to be $\mathbf{1}$. Here follows a counterexample to Craig's interpolation. Let P be a state variable and consider the conjunction:

$A \rightleftharpoons (\ell = 1 \wedge \lceil P \rceil; \top) \wedge$	the reference interval σ has an initial subinterval of length 1 where P holds and
$\neg(\top; \ell = 1 \wedge \lceil P \rceil; \ell = 1 \wedge \neg\lceil \neg P \rceil; \top) \wedge$	if a $\lceil P \rceil$-subinterval of σ of length 1 ends at least 1 time unit earlier than σ itself, then it is followed by a $\lceil \neg P \rceil$-subinterval of σ of length 1 and
$\neg(\top; \ell = 1 \wedge \lceil \neg P \rceil; \ell = 1 \wedge \neg\lceil P \rceil; \top)$	if a $\lceil \neg P \rceil$-subinterval of σ of length 1 ends at least 1 time unit earlier than σ itself, then it is followed by a $\lceil P \rceil$-subinterval of σ of length 1.

Clearly $I, \sigma \models A \wedge (\top; \ell = 1 \wedge \lceil \neg P \rceil)$ only if $\max \sigma - \min \sigma$ is an even positive integer. Let Q be a state variable that is distinct from P. A direct check shows that:

$$\models_{DC} A \wedge (\top; \ell = 1 \wedge \lceil \neg P \rceil) \Rightarrow ([Q/P]A \Rightarrow (\top; \ell = 1 \wedge \lceil \neg Q \rceil))$$

To interpolate this valid formula, we can use no other flexible symbol but ℓ. However, no state variable-free DC formula θ satisfies both

$$\models_{DC} A \wedge (\top; \ell = 1 \wedge \lceil \neg P \rceil) \Rightarrow \theta \text{ and } \models_{DC} \theta \wedge A \Rightarrow (\top; \ell = 1 \wedge \lceil \neg P \rceil), \qquad (3)$$

nor variants of these written with, e.g., Q instead of P. This is so, because a formula built using ℓ only always defines a finite set of intervals of nonnegative reals as the set of the durations of the intervals that satisfy it.

Next in this section we is an appropriately restricted form of Craig interpolation theorem, which holds for DC with abstract time. Its proof follows the pattern used to prove Craig's theorem about first order classical predicate logic in [ChK73], which was used to prove interval-related interpolation theorems in [Gue01] too. We translate abstract time DC into a suitable ω-theory in ITL to do this. The translation is similar to the one involved in the proof of relative completeness of the finitary proof system for real time DC first presented in [HZ92]. The purpose of this translation is to simplify the terms involved in the considered languages. After that we show how the expressibility of projection in the chosen subset of DC enables a projection-related interpolation theorem to be proved as a corollary to Craig interpolation.

3.1 Abstract Time DC as an ω-theory in ITL

The translation of DC into an ITL theory is based on the possibility to treat the duration terms which can be built starting from a given vocabulary of state variables as a system of flexible constants with constraints on their values. The constraints follow from these symbols' being representations of the durations of state expressions and can be formulated as axioms and rules of the corresponding ITL theory, which can be obtained as the translations of the axioms $DC0$, $DC1$, $DC2'$, $DC3'$ and $DC3'$, and the rule ω' about duration terms from the proof system for DC from Section 1.2. We denote the considered translation by t. Here follows its definition:

Definition 3 Given a state expression S in a DC language $\mathbf{L}$, we denote the class of the state expressions S' in $\mathbf{L}$ which are propositionally equivalent to S by $[S]$. Let $\ell^{[S]}$ be a fresh flexible constant for every state expression S in $\mathbf{L}$. We denote the ITL language built using the flexible constants $\ell^{[S]}$, and all the non-logical symbols of $\mathbf{L}$, except the state variables, by $\mathbf{L}^{ITL}$.

Given a term t from $\mathbf{L}$, $\mathsf{t}(t)$ is the result of replacing every occurrence of a duration term $\int S$ in t by the corresponding flexible constant $\ell^{[S]}$. Similarly, given a formula φ from $\mathbf{L}$, $\mathsf{t}(\varphi)$ is obtained by replacing the occurrences of duration terms in φ by their corresponding flexible constants.

The translation t from $\mathbf{L}$ into $\mathbf{L}^{ITL}$ defined above is obviously invertible up to the propositional equivalence of state expressions. Now, given a DC language $\mathbf{L}$, let the theory $ITL_{\mathbf{L}}$ in $\mathbf{L}^{ITL}$ have the t-translations of the axioms $DC0$, $DC1$, $DC2'$, $DC3'$ and $DC4'$ as its axioms:

$(DC0_{ITL})$ $\ell = 0 \Rightarrow \ell^{[S]} = 0$

$(DC1_{ITL})$ $\ell^{[\mathbf{0}]} = 0$

$(DC2'_{ITL})$ $\ell^{[S_1]} = 0 \vee \ell^{[S_2]} = \ell \Rightarrow \ell^{[S_1 \Rightarrow S_2]} = \ell$

$(DC3'_{ITL})$ $\ell^{[S_1]} = \ell \wedge \ell^{[S_2]} = 0 \Rightarrow \ell^{[S_1 \Rightarrow S_2]} = 0$

$(DC4'_{ITL})$ $(\ell^{[S]} = x; \ell^{[S]} = y) \Rightarrow \ell^{[S]} = x + y$

where S, S_1 and S_2 range over the set of state expressions in $\mathbf{L}$. Let $ITL_{\mathbf{L}}$ be also closed under the t-translation of the rule ω' for every state expression S:

$$(\omega'_{ITL}) \quad \frac{\forall k < \omega \ [(\ell^{[S]} = \ell \vee \ell^{[S]} = 0)^k / R]\alpha}{[\top / R]\alpha}$$

Proposition 4 *A set of formulas* Γ *in* $\mathbf{L}$ *is satisfiable iff* $ITL_{\mathbf{L}} \cup \{\mathsf{t}(\varphi) : \varphi \in \Gamma\}$ *is satisfiable.*

Remark 5 Note that if $\mathbf{L}$ has only finitely many state variables, it can contain no more than finitely many pairwise non-equivalent state expressions. That is, for such $\mathbf{L}$ the corresponding language $\mathbf{L}^{ITL}$ contains finitely many flexible constants of the kind $\ell^{[S]}$, and $ITL_{\mathbf{L}}$ has finitely many instances of the axioms $DC0_{ITL}$, $DC1_{ITL}$, $DC2'_{ITL}$, $DC3'_{ITL}$ and $DC4'_{ITL}$ and rule ω'_{ITL}.

3.2 Craig Interpolation for the Abstract Time Duration Calculus

Now we formulate a specialised Craig interpolation property of the ω-theories $ITL_{\mathbf{L}}$. Let $\mathbf{L}_1$ and $\mathbf{L}_2$ be DC languages. Let $\mathbf{L}_0$ and $\mathbf{L}_3$ be the DC languages based on the intersection and on the union of the vocabularies of $\mathbf{L}_1$ and $\mathbf{L}_2$, respectively. In the following theorem we consider interpolation of implications with their antecedent being a formula in $\mathbf{L}_1^{ITL}$ and their succedent being a formula in $\mathbf{L}_2^{ITL}$. We may assume that $\mathbf{L}_1$ and $\mathbf{L}_2$ have finitely many state variables each. Then the sets $L_i = \{\ell^{[S]} : S \text{ is a state expression in } \mathbf{L}_i\}$, $i = 1, 2$, are finite by Remark 5.

Theorem 6 *Let* φ *and* ψ *be in* $\mathbf{L}_1^{ITL}$ *and* $\mathbf{L}_2^{ITL}$*, respectively. Let* $ITL_{\mathbf{L}_3} \models_{ITL} \varphi \Rightarrow \psi$*. Let* $k_v < \omega$*, for every* $v \in L_1 \cup L_2$*. Then there exists a formula* θ *in* $\mathbf{L}_0^{ITL}$ *such that*

$$ITL_{\mathbf{L}_1} \vdash_{ITL} \varphi \wedge \bigwedge_{v \in L_1} (v = \ell \vee v = 0)^{k_v} \Rightarrow \theta \text{ and } ITL_{\mathbf{L}_2} \vdash_{ITL} \theta \wedge \bigwedge_{v \in L_2} (v = \ell \vee v = 0)^{k_v} \Rightarrow \psi.$$

Corollary 7 (Craig Interpolation for Abstract Time DC) *Let* φ *and* ψ *be in* $\mathbf{L}_1$ *and* $\mathbf{L}_2$*, respectively. Let* $k_P < \omega$ *for every state variable* P *in* $\mathbf{L}_3$*. Let*

$$\models_{DC} \bigwedge_{P \text{ occurs in } \varphi, \psi} (\textstyle\int P = \ell \vee \int P = 0)^{k_P} \Rightarrow \varphi \Rightarrow \psi.$$

Then there exists a formula θ *in* $\mathbf{L}_0$ *such that*

$$\models_{DC} \bigwedge_{P \text{ occurs in } \varphi} (\textstyle\int P = \ell \vee \int P = 0)^{k_P} \wedge \varphi \Rightarrow \theta \text{ and } \models_{DC} \bigwedge_{P \text{ occurs in } \psi} (\textstyle\int P = \ell \vee \int P = 0)^{k_P} \wedge \theta \Rightarrow \psi.$$

3.3 Projection-related Interpolation in Abstract Time DC

We reduce projection-related interpolation to Craig interpolation using the expressibility of projection in DC formulas with no other flexible symbols but state variables and ℓ [DVH99].

Theorem 8 (Projection-related interpolation in abstract time DC) *Let (2) hold. Let* $k_P < \omega$ *for every state variable* P *in* $\mathbf{L}_3$*. Then there exists a formula* θ *of the form given by the BNF*

$\theta ::= \bot \mid t = t \mid (\alpha / H) \mid \theta \Rightarrow \theta \mid (\theta; \theta) \mid \exists x \theta$

where α *stands for an arbitrary formula and the terms* t *contain no other flexible symbols but* ℓ*, such that*

$$\models_{DC} \bigwedge_{P \text{ occurs in } \varphi} (\textstyle\int P = \ell \vee \int P = 0)^{k_P} \wedge \varphi \Rightarrow \theta \text{ and } \models_{DC} \bigwedge_{P \text{ occurs in } \psi} (\textstyle\int P = \ell \vee \int P = 0)^{k_P} \wedge \theta \Rightarrow \psi.$$

4 Projection-related Interpolation in the $\lceil P \rceil$-subset of DC^*

We recall Craig interpolation for this subset of DC^* without (other) restrictions from [Gue00b] and establish that projection-related interpolation holds without (other) restrictions as a corollary. We include the full proofs in this section, because they entail an algorithm to obtain interpolants.

Iteration in DC is defined by the clause:

$M, \sigma \models \varphi^*$ iff either $\min \sigma = \max \sigma$, or there exist $\sigma_1, \ldots, \sigma_n \in \mathbf{I}(T_M)$ such that $\sigma_1; \ldots; \sigma_n = \sigma$ and $M, \sigma_i \models \varphi$, $i = 1, \ldots, n$.

Positive iteration $(.)^+$ can be defined in terms of $(.)^*$ by putting $\varphi^+ \rightleftharpoons (\varphi; \varphi^*)$. The $\lceil P \rceil$-subset of DC^* is defined by the BNF:

$\varphi ::= \bot \mid \ell = 0 \mid \lceil S \rceil \mid \varphi \Rightarrow \varphi \mid (\varphi; \varphi) \mid \varphi^*$

4.1 Craig interpolation for the $\lceil P \rceil$-subset of DC^*

Consider the state variable binding quantifier which is defined in DC as follows:

$\langle F, I \rangle, \sigma \models \exists P \varphi$ iff $\langle F, I_P^f \rangle, \sigma \models \varphi$ for some $f : T_F \rightarrow \{0, 1\}$ with the finite variability property.

Proposition 9 *Let φ be a formula of the form:*

$\varphi ::= \bot \mid \ell = 0 \mid \lceil S \rceil \mid \varphi \vee \varphi \mid (\varphi; \varphi) \mid \varphi^*$

Then $\exists P \varphi$ is equivalent to a formula of the same form, which can be constructed from φ using the state variables of φ with the exception of P.

Proof: Induction on the construction of φ, using the DC^* equivalences:

$\models \exists P \bot \Leftrightarrow \bot$ $\quad$ $\models \exists P \lceil S \rceil \Leftrightarrow (\lceil [\mathbf{0}/P]S \rceil \vee \lceil [\mathbf{1}/P]S \rceil)^+$ $\quad$ $\models \exists P \ell = 0 \Leftrightarrow \ell = 0$

$\models \exists P \varphi^* \Leftrightarrow (\exists P \varphi)^*$ $\quad$ $\models \exists P(\varphi \vee \psi) \Leftrightarrow \exists P \varphi \vee \exists P \psi$ $\quad$ $\models \exists P(\varphi; \psi) \Leftrightarrow (\exists P \varphi; \exists P \psi)$

$\dashv$

Proposition 10 *Every formula in the $\lceil P \rceil$-subset of DC^* is equivalent to one in the form mentioned in Proposition 9.*

Proof: This is essentially a corollary to Lemma 9 from [ZHS93], where a decision procedure for the $\lceil P \rceil$-subset of DC^* is presented. Let the alphabet $\mathbf{A}(\varphi)$ consist of all the elementary conjunctions containing all the state variables of φ. That lemma states that given a formula φ in the $\lceil P \rceil$-subset of DC^*, there exists a regular language $\mathbf{L}(\varphi) \subseteq (\mathbf{A}(\varphi))^*$ such that $\langle F, I \rangle, \sigma \models \varphi$ iff $\alpha(\sigma) \in \mathbf{L}(\varphi)$, where $\alpha(\sigma)$ is defined as follows:

If $m_F(\sigma) = 0$, then $\alpha(\sigma)$ is the empty word.

Otherwise, $\alpha(\sigma) = a_1 \ldots a_n$ where $a_i \neq a_{i+1}$ for $i = 1, \ldots, n-1$, and $M, \sigma \models (\lceil a_1 \rceil; \ldots; \lceil a_n \rceil)$.

Let $R_{\mathbf{L}(\varphi)}$ be a regular expression for $\mathbf{L}(\varphi)$. Let us substitute every occurrence of $a \in \mathbf{A}(\varphi)$ in $R_{\mathbf{L}(\varphi)}$ by $\lceil a \rceil$. Let us also replace the regular operations $\circ$, $\cup$ and * by their DC^* counterparts $(.;.)$, $\vee$ and DC iteration in this expression, respectively. Let the obtained formula be ψ. Clearly, $\mathbf{L}(\varphi) = \mathbf{L}(\psi)$, whence $\models_{DC^*} \varphi \Leftrightarrow \psi$. $\dashv$

Theorem 11 (Craig interpolation for the $\lceil P \rceil$-subset of DC^*) *Let φ and ψ be in the $\lceil P \rceil$-subset of DC^*. Let $\models_{DC^*} \varphi \Rightarrow \psi$. Then a formula θ in the $\lceil P \rceil$-subset of DC^* can be constructed using state variables which occur in both φ and ψ only, such that*

$\models_{DC^*} \varphi \Rightarrow \theta$ *and* $\models_{DC^*} \theta \Rightarrow \psi$.

Proof: Let $P_1, \ldots, P_n$ be the state variables which occur in φ, but not in ψ. Then θ can be chosen to be a quantifier-free formula that is equivalent to $\exists P_1 \ldots \exists P_n \varphi$. Such a formula exists and can be constructed from φ by Propositions 9 and 10. $\dashv$

4.2 Projection-related interpolation for the $\lceil P \rceil$-subset of DC^*

Let the state variables in the vocabularies of the DC^* languages $\mathbf{L}_1$ and $\mathbf{L}_2$ be in some disjoint sets $\{P_1, \ldots, P_n\}$ and $\{P'_1, \ldots, P'_n\}$, respectively. Let φ and ψ be formulas in the $\lceil P \rceil$-subsets of $\mathbf{L}_1$ and $\mathbf{L}_2$, respectively. Consider the formula

$$\left(\left(\bigwedge_{i=1}^{n} \Box(\lceil P_i \Leftrightarrow P'_i \rceil) \right) / H \right) \Rightarrow (\varphi \Rightarrow \psi) \quad (4)$$

Theorem 12 (Projection-related interpolation for the $\lceil P \rceil$-subset of DC^*) *If (4) is valid in DC^*, then there exists a θ in the $\lceil P \rceil$-subset of DC^* such that*

$$\models_{DC^*} \varphi \Rightarrow \theta \text{ and } \models_{DC^*} [P'_i/P_i : i = 1, \ldots, n]\theta \Rightarrow \psi,$$

and all the state expressions S occurring in θ have the form defined by the BNF
$S ::= \mathbf{0} \mid Q \mid H \wedge P \mid S \Rightarrow S$
where Q stands for a state variable occurring in H, and P stands for one of $P_1, \ldots, P_n$.

Proof: Let $s_0 = [(H \wedge Q_i) \vee (\neg H \wedge P_i)/P_i, (H \wedge Q_i) \vee (\neg H \wedge P'_i)/P'_i : i = 1, \ldots, n]$ where $Q_1, \ldots, Q_n$, are some fresh state variables. If (4) is valid, then applying this substitution to (4) should produce a valid formula too. A direct check shows that $\models_{DC^*} s_0 \left(\left(\bigwedge_{i=1}^{n} \Box(\lceil P_i \Leftrightarrow P'_i \rceil) \right) / H \right)$. Hence the validity of (4) entails that $\models_{DC^*} s_0\varphi \Rightarrow s_0\psi$. The only state variables shared between $s_0\varphi$ and $s_0\psi$ are some of $Q_1, \ldots, Q_n$, and the variables occurring in H. Hence, by Theorem 11, there exists a formula θ_0 in the $\lceil P \rceil$-subset of the DC^* language built using these variables only and such that $\models_{DC^*} s_0\varphi \Rightarrow \theta_0$ and $\models_{DC^*} \theta_0 \Rightarrow s_0\psi$. Now consider the substitutions $s = [H \wedge P_i/Q_i : i = 1, \ldots, n]$ and $s' = [H \wedge P'_i/Q_i : i = 1, \ldots, n]$. A direct check shows that $\models_{DC^*} \varphi \Leftrightarrow ss_0\varphi$ and $\models_{DC^*} \psi \Leftrightarrow s's_0\psi$, whence $\models_{DC^*} \varphi \Rightarrow s\theta_0$ and $\models_{DC^*} s'\theta_0 \Rightarrow \psi$. Besides, obviously $[P'_i/P_i : i = 1, \ldots, n]s = s'$. Let θ be $s\theta_0$. Then all the occurrences of $P_1, \ldots, P_n$ in θ are in state expressions of the form $H \wedge P_i$, $i = 1, \ldots, n$, and the only state variables occurring in θ are $P_1, \ldots, P_n$ and those occurring in H. This concludes the proof. $\dashv$

Concluding remarks

The $\lceil P \rceil^*$-subset of DC has the same expressive power as that of the propositional subset of discrete time ITL with the truth values of the propositional variables restricted to depend on the beginnings of reference intervals, and to that of regular expressions. Proposition 9 can be reformulated in terms of appropriate homomorphisms on the regular languages involved in the decidability argument for the $\lceil P \rceil^*$-subset in [ZHS93]. That is why the constructivity of interpolation here is natural to expect. A careful examination of projection-related interpolation can lead to regular language theory counterparts too. This emphasises the gap between the decidable subsets of DC, where automata-theoretic decision procedures are readily available, and the full first order systems of ITL and DC, where abstraction can be handled using flexible constant, function and predicate symbols.

In the first order case, interpolation and its failure are interesting for the account of the correspondence between explicit and implicit definability they give, the way these notions are known from first order predicate logic, yet in the case of the temporal first order system ITL and its extension DC. The failure of interpolation means that implicit definitions in DC can be more expressive than explicit ones, unless somehow restricted. The counterexample to interpolation can be viewed as a *rigorous* motivation for resorting to the state variable binding quantifier $\exists$ in DC as known from [Pan95], because it obviously contributes to the possibility to interpolate and, consequently, define explicitly in DC.

The effect of most of the extending operators for ITL and DC on interpolation seems largely unknown. Providing a flexible formulation of a general interval-related interpolation theorem by means of projection is perhaps the most outstanding idea in this paper. Projection is relevant to specification by DC in various other ways too.

References

[BT00] Bowman, H. and S. Thompson. *A Complete Axiomatisation of Interval Temporal Logic with Projection.* T. R. 6-00, Computing Laboratory, University of Kent, 2000.

[ChK73] Chang, C. C. and H. J. Keisler. *Model Theory.* North Holland, Amsterdam, 1973.

[Dut95] Dutertre, B. *On First Order Interval Temporal Logic* Report no. CSD-TR-94-3 Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, England, 1995.

[DVH99] Dang Van Hung. *Projections: A Technique for Verifying Real-Time Programs in DC.* Technical Report 178, UNU/IIST, P.O. Box 3058, Macau, November 1999.

[Gue98] Guelev, D. P. *A Calculus of Durations on Abstract Domains: Completeness and Extensions.* Technical Report 139, UNU/IIST, P.O.Box 3058, Macau, May 1998.

[Gue00a] Guelev, D. P. *A Complete Proof System for First Order Interval Temporal Logic with Projection.* Technical Report 202, UNU/IIST, P.O. Box 3058, Macau, June 2000.

[Gue00b] Guelev, D. P. *Interpolation and Related Results on the $\lceil P\rceil$-fragment of DC with Iteration.* Technical Report 203, UNU/IIST, P.O. Box 3058, Macau, June 2000.

[Gue01] Guelev, D. P. Interval-related Interpolation in Interval Temporal Logics. *Logic Journal of the IGPL*, volume 9 issue 5, 2001, pp. 677-685.

[GD02] Guelev, D. P. and Dang Van Hung. Prefix and Projection onto State in Duration Calculus. *ENTCS*, Volume 65, Issue 6, Elsevier Science, 2002.

[He99] He Jifeng. A Behavioral Model for Co-design. *Proceedings of FM'99*, LNCS 1709, Springer, 1999, pp. 1420-1438.

[HMM83] Halpern, J., Z. Manna and B. Moszkowski. A Hardware Semantics Based on Temporal Intervals. *Proceedings of ICALP'83*, LNCS 154, Springer, 1983, pp. 278-291.

[HZ92] Hansen, M. R. and Zhou Chaochen. Semantics and Completeness of Duration Calculus. *Real-Time: Theory and Practice*, LNCS 600, Springer, 1992, pp. 209-225.

[HZ97] Hansen, M. R. and Zhou Chaochen. Duration Calculus: Logical Foundations. *Formal Aspects of Computing*, 9, 1997, pp. 283-330.

[Mos85] Moszkowski, B. Temporal Logic For Multilevel Reasoning About Hardware. *IEEE Computer*, 18(2), February 1985, pp. 10-19.

[Mos86] Moszkowski, B. *Executing Temporal Logic Programs.* Cambridge University Press, 1986.

[Mos95] Moszkowski, B. Compositional Reasoning about Projected and Infinite Time. *ICECCS'95*, IEEE Computer Society Press, 1995, pp. 238-245.

[Pan95] Pandya, P. K. Some extensions to Mean-Value Calculus: Expressiveness and Decidability. *Proceedings of CSL'95.*, LNCS 1092, 1995, pp. 434-451.

[Ras02] Rasmussen, T.M. *Interval Logic - Proof Theory and Theorem Proving.* Ph.D. Thesis, Technical University of Denmark, 2002.

[ZH96] Zhou Chaochen, and M. R. Hansen. An Adequate First Order Interval Logic. LNCS 1536, Springer, 1998, pp. 584-608.

[ZHR91] Zhou Chaochen, C. A. R. Hoare and A. P. Ravn. A Calculus of Durations. *Information Processing Letters*, 40(5), 1991, pp. 269-276.

[ZHS93] Zhou Chaochen, M. R. Hansen and P. Sestoft. Decidability and Undecidability Results for Duration Calculus. *STACS'93*, LNCS 665, Springer, 1993, pp. 58-68.