A Complete Fragment of Higher-Order Duration μ-Calculus

Dimitar P. Guelev

International Institute for Software Technology
of the United Nations University
(UNU/IIST), Macau, P.O.Box 3058.
Institute of Mathematics and Informatics
Bulgarian Academy of Sciences Bl. 8, Akad G. Bonchev blvd., Sofia.
E-mail: gelevdp@bgnet.bg, dg@iist.unu.edu

Abstract. The paper presents an extension μHDC of Higher-order Duration Calculus (HDC,[ZGZ99]) by a polyadic least fixed point (μ) operator and a class of non-logical symbols with a *finite variability* restriction on their interpretations, which classifies these symbols as intermediate between *rigid* symbols and *flexible* symbols as known in DC. The μ operator and the new kind of symbols enable straightforward specification of recursion and data manipulation by HDC. The paper contains a completeness theorem about an extension of the proof system for HDC by axioms about μ and symbols of finite variability for a class of *simple μHDC formulas*. The completeness theorem is proved by the method of local elimination of the extending operator μ, which was earlier used for a similar purpose in [Gue98].

Introduction

Duration calculus(DC, [ZHR91]) has been proved to be a suitable formal system for the specification of the semantics of concurrent real-time programming languages[SX98,ZH00]. The introduction of a least fixed point operator to DC was motivated by the need to specify recursive programming constructs simply and straightforwardly. Recursive control structures as available in procedural programming languages are typically approximated through translation into iterative ones with explicit special storage (stacks). This blurs intuition and can add a significant overhead to the complexity of deductive verification. It is also an abandonment of the principle of abbreviating away routine elements of proof in specialised notations. That is why it is worth having an immediate way not only to specify but also to be able to reason about this style of recursion as it appears in high level programming languages.

Recently, an extension of DC by quantifiers which bind state variables (boolean valued functions of time) was introduced[ZGZ99]. Systematic studies regarding the application of this sort of quantification in DC had gained speed earlier, cf. [Pan95]; HDC allowed the integration of some advanced features of DC, such as *super-dense chop* [ZH96,HX99], into a single general system, called Higher-order

Duration Calculus (HDC), and enabled the specification of the semantics of temporal specification and programming languages such as Verilog and Timed RAISE[ZH00,LH99] by DC. The kind of completeness of the proof system of HDC addressed in [ZGZ99], which is ω-completeness, allowed to conclude the study of the expressive power of some axioms about the state quantifier.

In this paper we present some axioms about the least fixed point operator in HDC and show that adding them to a proof system for HDC yields a complete proof system for a fragment of the extension of HDC with this operator, μHDC.

The axioms we study are obtained by paraphrasing of the inference rules known about the propositional modal μ-calculus(cf. [Koz83,Wal93]), which were first introduced to DC in [PR95]. The novelty in our approach is the way we use the expressive power of the axioms about the μ-operator in our completeness argument, because, unlike the propositional μ-calculus, μHDC is a first-order logic with a binary modal operator.

Our method was first developed and applied in [Gue98] to so-called simple DC^* formulas which were introduced in [DW94] as a DC counterpart of a class of finite timed automata. That class was later significantly extended in [DG99,Gue00]. In this paper we show the completeness of an extension of a proof system for HDC for a corresponding class of simple μHDC formulas.

Our method of proof significantly relies on the exact form of the completeness of the proof system for HDC, which underlies the extension in focus. The completeness theorem about the original proof system for DC[HZ92] applies to the derivability of individual formulas only, and we need to have equivalence between the satisfiability of the infinite sets of instances of our new axioms and the consistency of these sets together with some other formulas, i.e. we need an ω-complete proof system for HDC. That is why we use a modification of the system from [ZGZ99], which is ω-complete with respect to a semantics for HDC, shaped after the abstract semantics of ITL, as presented in [Dut95]. Material to suggest an ω-completeness proof for this modification can be found starting from completion of Peano arithmetics by an ω-rule (cf. e.g. [Men64]) to [ZNJ99]. The completeness result presumed in this paper applies to the class of abstract HDC frames with their duration domains satisfying the principle of Archimedes. Informally, this principle states that there are no infinitely small positive durations and it holds for the real-time based frame.

The purpose of the modification of HDC here is to make a form of finite variability which is preserved under logical operations explicitly appear in this system. The choice to work with Archimedean duration domains is just to provide the convenience to axiomatise this kind of finite variability (axiom $HDC5$ below).

The fragment of μHDC language that our completeness result applies to is sufficient to provide convenience of the targetted kind for the design and use of HDC semantics of practically significant timed languages which admit recursive procedure invocations.

1 Preliminaries on *HDC* with abstract semantics

In this section we briefly introduce a version of HDC with abstract semantics[ZGZ99], which closely follows the abstract semantics for ITL given in [Dut95]. It slightly differs from the one presented in [ZGZ99]. Along with quantification over state, we allow quantifiers to bind so-called temporal variables and temporal propositional letters with the finite variability property.

1.1 Languages

A language for HDC is built starting from some given sets of *constant symbols* a, b, c, ..., *function symbols* f, g, ..., *relation symbols* R, S, ..., *individual variables* x, y, ... and *state variables* P, Q, ... Function symbols and relation symbols have *arity* to indicate the number of arguments they take in terms and formulas. Relation symbols and function symbols of arity 0 are also called *temporal propositional letters* and *temporal variables* respectively. Constant symbols, function symbols and relation symbols can be either *rigid* or *flexible*. Flexible symbols can be either symbols of *finite variability* (*fv* symbols) or not. Rigid symbols, fv symbols and (general) flexible and symbols are subjected to different restrictions on their interpretations. Every HDC language contains countable sets of individual variables, fv temporal propositional letters and fv temporal variables, the rigid constant symbol 0, the flexible constant symbol ℓ, the rigid binary function symbol + and the rigid binary relation symbol =. Given the sets of symbols, *state expressions* S, *terms* t and *formulas* φ in a HDC language are defined by the BNFs:

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

$t ::= c|\int S|f(t,\ldots,t)|\overleftarrow{t}|\overrightarrow{t}$

$\varphi ::= \bot|R(t,\ldots,t)|\varphi \Rightarrow \varphi|(\varphi;\varphi)|\exists x\varphi|\exists v\varphi|\exists P\varphi$

In BNFs for formulas here and below v stands for a fv temporal variable or a fv temporal propositional letter.

Terms and formulas which contain no flexible symbols are called *rigid*. Terms and formulas which contain only fv flexible symbols, rigid symbols and subformulas of the kind $\int S = \ell$ are called *fv terms* and *fv formulas* respectively. Terms of the kinds $\overleftarrow{t}$ and $\overrightarrow{t}$ are well-formed only if t is a fv term. We call individual variables, temporal variables, temporal propositional letters and state variables just variables, in case the exact kind of the symbol is not significant.

1.2 Frames, models and satisfaction

Definition 1. *A* time domain *is a linearly ordered set with no end points. Given a time domain* $\langle T, \leq\rangle$*, we denote the set* $\{[\tau_1,\tau_2] : \tau_1,\tau_2 \in T, \tau_1 \leq \tau_2\}$ *of intervals in* T *by* $\mathbf{I}(T)$*. Given* $\sigma_1,\sigma_2 \in \mathbf{I}(T)$*, where* $\langle T, \leq\rangle$ *is a time domain, we denote* $\sigma_1 \cup \sigma_2$ *by* $\sigma_1;\sigma_2$*, in case* $\max\sigma_1 = \min\sigma_2$*. A* duration domain *is a system of the type* $\langle D, 0^{(0)}, +^{(2)}, \leq^{(2)}\rangle$ *which satisfies the following axioms*

$(D1)\ x + (y + z) = (x + y) + z$	$(D6)\ \ x \le x$
$(D2)\ x + 0 = x$	$(D7)\ \ x \le y \wedge y \le x \Rightarrow x = y$
$(D3)\ x + y = x + z \Rightarrow y = z$	$(D8)\ \ x \le y \wedge y \le z \Rightarrow x \le z$
$(D4)\ \exists z(x + z = y)$	$(D9)\ \ x \le y \Leftrightarrow \exists z(x + z = y \wedge 0 \le z)$
$(D5)\ x + y = y + x$	$(D10)\ x \le y \vee y \le x$

Given a time domain $\langle T, \le\rangle$, *and a duration domain* $\langle D, 0, +, \le\rangle$, $m : \mathbf{I}(T) \to D$ *is a* measure *if*

$(M0)\ x \ge 0 \Leftrightarrow \exists\sigma(m(\sigma) = x)$
$(M1)\ \min\sigma = \min\sigma' \wedge m(\sigma) = m(\sigma') \Rightarrow \max\sigma = \max\sigma'$
$(M2)\ \max\sigma = \min\sigma' \Rightarrow m(\sigma) + m(\sigma') = m(\sigma \cup \sigma')$
$(M3)\ 0 \le x \wedge 0 \le y \wedge m(\sigma) = x + y \Rightarrow \exists\tau \in \sigma\ m([\min\sigma, \tau]) = x.$

Definition 2. *A HDC* frame *is a tuple of the kind* $\langle\langle T, \le\rangle, \langle D, 0, +, \le\rangle, m\rangle$, *where* $\langle T, \le\rangle$ *is a time domain,* $\langle D, 0, +, \le\rangle$ *is a duration domain, and* $m : \mathbf{I}(T) \to D$ *is a measure.*

Definition 3. *Given a HDC frame* $F = \langle\langle T, \le\rangle, \langle D, 0, +, \le\rangle, m\rangle$ *and a HDC language* $\mathbf{L}$*, a function* I *which is defined on the set of the non-logical symbols of* $\mathbf{L}$ *is called* interpretation *of* $\mathbf{L}$ *into* F*, if*

- $\circ$ $I(c), I(x) \in D$ *for constant symbols* c *and individual variables* x
- $\circ$ $I(f) : D^n \to D$ *for rigid* n*-place function symbols* f
- $\circ$ $I(f) : \mathbf{I}(T) \times D^n \to D$ *for flexible* n*-place function symbols* f
- $\circ$ $I(R) : D^n \to \{0, 1\}$ *for rigid* n*-place relation symbols* R
- $\circ$ $I(R) : \mathbf{I}(T) \times D^n \to \{0, 1\}$ *for flexible* n*-place relation symbols* R
- $\circ$ $I(P) : T \to \{0, 1\}$ *for state variables* P
- $\circ$ $I(0) = 0$, $I(\ell) = m$, $I(+) = +$ *and* $I(=)$ *is* $=$.

The following finite variability *condition is imposed on interpretations of state variables* P*:*

> *Every* $\sigma \in \mathbf{I}(T)$ *can be represented in the form* $\sigma_1; \ldots; \sigma_m$ *so that* $I(P)$ *is constant on* $[\min\sigma_i, \max\sigma_i)$, $i = 1, \ldots, m$.

A similar condition is imposed on the interpretations of fv symbols s*. Given a frame* F *and an interpretation* I *as above, and* $\sigma \in \mathbf{I}(T)$*, a function (predicate)* A *on* $\mathbf{I}(T) \times D^n$ *is called* fv in F, I with respect to $\sigma_1, \ldots, \sigma_m \in \mathbf{I}(T)$ *iff* $\sigma = \sigma_1; \ldots; \sigma_m$ *for some interval* σ *and for all* $d_1, \ldots, d_n \in D$, $i, j \le m$, $i \le j$, $\sigma' \in \mathbf{I}(T)$*:*

- $\circ$ *if* $\min\sigma' \in (\min\sigma_i, \max\sigma_i)$ *and* $\max\sigma' \in (\min\sigma_j, \max\sigma_j)$, $A(\sigma', d_1, \ldots, d_n)$ *is determined by* $d_1, \ldots, d_n$ i *and* j *only;*
- $\circ$ *if* $\min\sigma' = \min\sigma_i$ *and* $\max\sigma' \in (\min\sigma_j, \max\sigma_j)$, $A(\sigma', d_1, \ldots, d_n)$ *is determined by* $d_1, \ldots, d_n$ i *and* j *only, possibly in a different way;*
- $\circ$ *if* $\min\sigma' \in (\min\sigma_i, \max\sigma_i)$ *and* $\max\sigma' = \min\sigma_j$, $A(\sigma', d_1, \ldots, d_n)$ *is determined by* $d_1, \ldots, d_n$ i *and* j *only, possibly in a different way;*
- $\circ$ *if* $\min\sigma' = \min\sigma_i$ *and* $\max\sigma' = \min\sigma_j$, $A(\sigma', d_1, \ldots, d_n)$ *is determined by* $d_1, \ldots, d_n$ i *and* j *only, possibly in a different way.*

A symbol s is fv with respect to $\sigma_1, \ldots, \sigma_m$ in some F, I *as above, if* $I(s)$ *has the corresponding property. Given a fv symbol* s*, for every* $\sigma \in \mathbf{I}(T)$ *there should be* $\sigma_1, \ldots, \sigma_m \in \mathbf{I}(T)$ *such that* s *is fv with respect to* $\sigma_1, \ldots, \sigma_m$ *in* F, I.

Given a language $\mathbf{L}$*, a pair* $\langle F, I\rangle$ *is a* model *for* $\mathbf{L}$ *if* F *is a frame and* I *is an interpretation of* $\mathbf{L}$ *into* F*.*

Interpretations I and J of language $\mathbf{L}$ into frame F are said to s-agree, if they assign the same values to all non-logical symbols from $\mathbf{L}$, but possibly s.

Given a frame F (model M) we denote its components by $\langle T_F, \leq_F\rangle$, $\langle D_F, 0_F, +_F, \leq_F\rangle$ and m_F ($\langle T_M, \leq_M\rangle$, $\langle D_M, 0_M, +_M, \leq_M\rangle$ and m_M) respectively. We denote the frame and the interpretation of a given model M by I_M and F_M respectively.

Definition 4. *Given a model* $M = \langle F, I\rangle$ *for the language* $\mathbf{L}$*,* $\tau \in T_M$ *and* $\sigma \in \mathbf{I}(T_M)$ *the values* $I_\tau(S)$ *and* $I_\sigma(t)$ *of state expressions* S *and terms* t *and the satisfaction of formulas* φ *are defined by induction on their construction as follows:*

$I_\tau(\mathbf{0}) = 0$
$I_\tau(P) = I(P)(\tau)$
$I_\tau(S_1 \Rightarrow S_2) = \max\{1 - I_\tau(S_1), I_\tau(S_2)\}.$
$I_\sigma(c) = I(c)$ *for rigid* c
$I_\sigma(c) = I(c)(\sigma)$ *for flexible* c
$I_\sigma(\int S) = \int_{\min\sigma}^{\max\sigma} I_\tau(S)d\tau$
$I_\sigma(f(t_1, \ldots, t_n)) = I(f)(I_\sigma(t_1), \ldots, I_\sigma(t_n))$ *for rigid* f
$I_\sigma(f(t_1, \ldots, t_n)) = I(f)(\sigma, I_\sigma(t_1), \ldots, I_\sigma(t_n))$ *for flexible* f
$I_\sigma(\overleftarrow{t}) = d$ *if* $I_{\sigma'}(t) = d$ *for some* $\tau < \min\sigma$ *and all* $\sigma' \subset (\tau, \min\sigma)$
$I_\sigma(\overrightarrow{t}) = d$ *if* $I_{\sigma'}(t) = d$ *for some* $\tau > \max\sigma$ *and all* $\sigma' \subset (\max\sigma, \tau)$
$M, \sigma \not\models \bot$
$M, \sigma \models R(t_1, \ldots, t_n)$ *iff* $I(R)(I_\sigma(t_1), \ldots, I_\sigma(t_n)) = 1$ *for rigid* R
$M, \sigma \models R(t_1, \ldots, t_n)$ *iff* $I(R)(\sigma, I_\sigma(t_1), \ldots, I_\sigma(t_n)) = 1$ *for flexible* R
$M, \sigma \models \varphi \Rightarrow \psi$ *iff either* $M, \sigma \models \psi$ *or* $M, \sigma \not\models \varphi$
$M, \sigma \models (\varphi;\psi)$ *iff there exist* $\sigma_1, \sigma_2 \in \mathbf{I}(T_F)$ *such that* $\sigma = \sigma_1;\sigma_2$*,* $M, \sigma_1 \models \varphi$ *and* $M, \sigma_2 \models \psi$
$M, \sigma \models \exists x\varphi$ *iff* $\langle F, J\rangle, \sigma \models \varphi$ *for some* J *which* x*-agrees with* I

Note that discrete time domains, which make the above definitions of $\overleftarrow{t}$ and $\overrightarrow{t}$ incorrect, also render any "corrected" definition for these operators grossly non-introspective, and therefore these operators should be disregarded in the case of discrete domains. In the clause about $\exists x$ above x stands for variable of an arbitrary kind, temporal variables and propositional temporal letters included. The integral used to define values of terms of the kind $\int S$ above is defined as follows. Given σ and S, there exist $\sigma_1, \ldots, \sigma_n \in \mathbf{I}(T_F)$ such that $\sigma = \sigma_1; \ldots; \sigma_n$ and $I_\tau(S)$ is constant in $[\min\sigma_i; \max\sigma_i)$, $i = 1, \ldots, n$. Given such a partitition $\sigma_1, \ldots, \sigma_n$ of σ, we put:

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

Clearly, the value thus defined does not depend on the choice of $\sigma_1, \ldots, \sigma_n$.

1.3 Abbreviations

Infix notation and propositional constant $\top$, connectives $\neg$, $\wedge$, $\vee$ and $\Leftrightarrow$ and quantifier $\forall$ are introduced as abbreviations in the usual way. $\mathbf{1}$ stands for $\mathbf{0} \Rightarrow \mathbf{0}$ in state expressions. The relation symbol $\leq$ is defined by the axiom $x \leq y \Leftrightarrow \exists z(x+z=y)$. The related symbols $\geq$, $<$ and $>$ are introduced in the usual way. We use the following DC-specific abbreviations:

$$\lceil S \rceil \rightleftharpoons \textstyle\int S = \ell \wedge \ell \neq 0,\ \ \Diamond\varphi \rightleftharpoons ((\top;\varphi);\top),\ \ \Box\varphi \rightleftharpoons \neg\Diamond\neg\varphi,\ \ n.t \rightleftharpoons \underbrace{t + \ldots + t}_{n \text{ times}}.$$

$$\Diamond_i\varphi \rightleftharpoons ((\ell \neq 0;\varphi);\ell \neq 0),\ \Box_i\varphi \rightleftharpoons \neg\Diamond_i\neg\varphi\ \ \ \xi_{t_1,t_2}(\varphi) \rightleftharpoons ((\ell = t_1;\varphi) \wedge \ell = t_2;\top).$$

1.4 Proof system

Results in the rest of this paper hold for the class of DC models which satisfy the principle of Archimedes. It states that given positive durations d_1 and d_2, there exists a natural number n such that $n.d_1 \geq d_2$.

Here follows a proof system for HDC which is ω-complete with respect to the class of HDC models which satisfy the principle of Archimedes:

$(A1_l)$ $(\varphi;\psi) \wedge \neg(\chi;\psi) \Rightarrow (\varphi \wedge \neg\chi;\psi)$
$(A1_r)$ $(\varphi;\psi) \wedge \neg(\varphi;\chi) \Rightarrow (\varphi;\psi \wedge \neg\chi)$
$(A2)$ $((\varphi;\psi);\chi) \Leftrightarrow (\varphi;(\psi;\chi))$
(R_l) $(\varphi;\psi) \Rightarrow \varphi$ if φ is rigid
(R_r) $(\varphi;\psi) \Rightarrow \psi$ if ψ is rigid
(B_l) $(\exists x\varphi;\psi) \Rightarrow \exists x(\varphi;\psi)$ if $x \notin FV(\varphi)$
(B_r) $(\varphi;\exists x\psi) \Rightarrow \exists x(\varphi;\psi)$ if $x \notin FV(\psi)$
$(L1_l)$ $(\ell = x;\varphi) \Rightarrow \neg(\ell = x;\neg\varphi)$
$(L1_r)$ $(\varphi;\ell = x) \Rightarrow \neg(\neg\varphi;\ell = x)$
$(L2)$ $\ell = x + y \Leftrightarrow (\ell = x;\ell = y)$
$(L3_l)$ $\varphi \Rightarrow (\ell = 0;\varphi)$
$(L3_r)$ $\varphi \Rightarrow (\varphi;\ell = 0)$

(MP) $\dfrac{\varphi\ \ \varphi \Rightarrow \psi}{\psi}$ $\qquad$ (G) $\dfrac{\varphi}{\forall x\varphi}$

(N_l) $\dfrac{\varphi}{\neg(\neg\varphi;\psi)}$ $\qquad$ (N_r) $\dfrac{\varphi}{\neg(\psi;\neg\varphi)}$

$(Mono_l)$ $\dfrac{\varphi \Rightarrow \psi}{(\varphi;\chi) \Rightarrow (\psi;\chi)}$

$(Mono_r)$ $\dfrac{\varphi \Rightarrow \psi}{(\chi;\varphi) \Rightarrow (\chi;\psi)}$

(ω) $\dfrac{\forall k < \omega\ [(\ell = 0 \vee \lceil S \rceil \vee \lceil \neg S \rceil)^k/R]\varphi}{[\top/R]\varphi}$

$(Arch)$ $\dfrac{\forall n < \omega\ \varphi \Rightarrow n.x \leq y}{\varphi \Rightarrow x \leq 0}$

$(DC0)$ $\ell = 0 \Rightarrow \int S = 0$
$(DC1)$ $\int \mathbf{0} = 0$
$(DC2)$ $\lceil \mathbf{1} \rceil \vee \ell = 0$
$(DC3)$ $(\int S = x;\lceil S \rceil \wedge \ell = y) \Rightarrow \int S = x + y$
$(DC4)$ $(\int S = x;\lceil \neg S \rceil) \Rightarrow \int S = x$
$(DC5)$ $\lceil S_1 \rceil \wedge \lceil S_2 \rceil \Leftrightarrow \lceil S_1 \wedge S_2 \rceil$
$(DC6)$ $\lceil S_1 \rceil \Leftrightarrow \lceil S_2 \rceil$, if $\vdash_{PC} S_1 \Leftrightarrow S_2$.
$(DC7)$ $\lceil S \rceil \Rightarrow \Box(\lceil S \rceil \vee \ell = 0)$

$(PV1)$ $(\ell \neq 0;\overleftarrow{t} = x \wedge \ell = y) \Leftrightarrow (\top;(\Box_i(t = x) \wedge \ell \neq 0;\ell = y))$
$(PV2)$ $(\overrightarrow{t} = x \wedge \ell = y;\ell \neq 0) \Leftrightarrow ((\ell = y;\Box_i(t = x) \wedge \ell \neq 0);\top)$

(NL) $\dfrac{((\ell = a;\varphi);\ell = b) \Rightarrow ((\ell = a;\psi);\ell = b)}{\varphi \Rightarrow \psi}$

$(\exists_v)$ $[t/v]\varphi \Rightarrow \exists v\varphi$ for fv-terms t and temporal variables v;
$(\exists_p)$ $[\psi/p]\varphi \Rightarrow \exists p\varphi$ for fv-formulas ψ and temporal propositional letters p;
$(HDC1)$ $\exists v(\overleftarrow{v} = x)$
$(HDC2)$ $\exists v(\overrightarrow{v} = x)$
$(HDC3)$ $(\exists S\varphi;\exists S\psi) \Leftrightarrow \exists S(\varphi;\psi)$

$(HDC3_{v,l})$ $x \leq \ell \Rightarrow \exists v \forall y_1 \forall y_2 (\overleftarrow{v} = \overleftarrow{t_1} \wedge \overrightarrow{v} = \overrightarrow{t_2} \wedge$
$\wedge (y_1 \leq x \wedge y_2 \leq x \wedge y_1 \leq y_2 \Rightarrow \xi_{y_1,y_2}(v = t_1)) \wedge$
$\wedge (y_1 > x \wedge y_2 > x \wedge y_1 \leq y_2 \wedge y_2 \leq \ell \Rightarrow \xi_{y_1,y_2}(v = t_2)) \wedge$
$\wedge (y_1 \leq x \wedge y_2 > x \wedge y_2 \leq \ell \Rightarrow \xi_{y_1,y_2}(v = t_3)))$

$(HDC3_{v,r})$ $x \leq \ell \Rightarrow \exists v \forall y_1 \forall y_2 (\overleftarrow{v} = \overleftarrow{t_1} \wedge \overrightarrow{v} = \overrightarrow{t_2} \wedge$
$\wedge (y_1 < x \wedge y_2 < x \wedge y_1 \leq y_2 \Rightarrow \xi_{y_1,y_2}(v = t_1)) \wedge$
$\wedge (y_1 \geq x \wedge y_2 \geq x \wedge y_1 \leq y_2 \wedge y_2 \leq \ell \Rightarrow \xi_{y_1,y_2}(v = t_2)) \wedge$
$\wedge (y_1 < x \wedge y_2 \geq x \wedge y_2 \leq \ell \Rightarrow \xi_{y_1,y_2}(v = t_3)))$

$(HDC3_{p,l})$ $x \leq \ell \Rightarrow \exists p \forall y_1 \forall y_2 ($
$(y_1 \leq x \wedge y_2 \leq x \wedge y_1 \leq y_2 \Rightarrow \xi_{y_1,y_2}(p \Leftrightarrow \psi_1) \wedge$
$\wedge (y_1 > x \wedge y_2 > x \wedge y_1 \leq y_2 \wedge y_2 \leq \ell \Rightarrow \xi_{y_1,y_2}(p \Leftrightarrow \psi_2)) \wedge$
$\wedge (y_1 \leq x \wedge y_2 > x \wedge y_2 \leq \ell \Rightarrow \xi_{y_1,y_2}(p \Leftrightarrow \psi_3)))$

$(HDC3_{p,r})$ $x \leq \ell \Rightarrow \exists p \forall y_1 \forall y_2 ($
$(y_1 < x \wedge y_2 < x \wedge y_1 \leq y_2 \Rightarrow \xi_{y_1,y_2}(p \Leftrightarrow \psi_1)) \wedge$
$\wedge (y_1 \geq x \wedge y_2 \geq x \wedge y_1 \leq y_2 \wedge y_2 \leq \ell \Rightarrow \xi_{y_1,y_2}(p \Leftrightarrow \psi_2)) \wedge$
$\wedge (y_1 < x \wedge y_2 \geq x \wedge y_2 \leq \ell \Rightarrow \xi_{y_1,y_2}(p \Leftrightarrow \psi_3)))$

$(HDC4)$ $\forall x \forall y ((\varphi \wedge \ell = x; \psi) \wedge \neg(\varphi \wedge \ell = y; \psi) \Rightarrow x < y) \Rightarrow$
$\Rightarrow \exists x (\forall y ((\varphi \wedge \ell = y; \psi) \Leftrightarrow y < x) \vee \exists x (\forall y ((\varphi \wedge \ell = y; \psi) \Leftrightarrow y \leq x)$

$(HDC5)$ $\ell \neq 0 \Rightarrow \exists x (x \neq 0 \wedge$
$\forall y \Box (\varphi \wedge \Diamond(\neg\varphi \wedge \Diamond(\varphi \wedge \Diamond(\neg\varphi \wedge \Diamond(\varphi \wedge \ell = y)))) \Rightarrow \ell \leq x + y))$

The symbol x denotes a variable of an arbitrary kind in the rule G and the axioms B_l and B_r. Instances of $HDC3_*$, $HDC4$ and $HDC5$ are valid only if $v, p, x, y, y_1, y_2 \notin FV(t_1), FV(t_2), FV(t_3), FV(\psi_1)$, $FV(\psi_2), FV(\psi_3)$, $FV(\varphi)$, $FV(\psi)$ and t_1, t_2, t_3, ψ_1, ψ_2, ψ_3, φ and ψ are fv terms and formulas respectively.

The proof system also includes the axioms $D1$-$D10$ for duration domains, first order axioms and equality axioms. Substitution $[t/x]\varphi$ of variable x by term t in formula φ is allowed in proofs only if either t is rigid, or x is not in the scope of a modal operator.

Note that this proof system is slightly different from the original HDC one, as fv symbols are not considered in HDC as in [ZGZ99]. Nevertheless, its ω-completeness can be shown in way that is similar to the one taken in [ZNJ99].

The meaning of the new axioms $HDC1$, $HDC2$ and $HDC3_*$ is to enable the construction of fv functions and predicates on the set of intervals of the given model (from simpler ones). Given that a language $\mathbf{L}$ has rigid constants to name all the durations in a model M for it, as in the case of canonical models which are used in the completeness argument for this system, the existence of every fv function and predicate on $\mathbf{I}(T_M)$ can be shown using these axioms. The axioms $HDC4$ and $HDC5$ express the restrictions on the interpretations of fv formulas, and hence - the fv symbols occurring in them. The following ω-completeness theorem holds about this proof system:

Theorem 1. *Let Γ be a consistent set of formulas from the language $\mathbf{L}$ of HDC. Then there exists a model M for $\mathbf{L}$ and an interval $\sigma \in \mathbf{I}(T_M)$ such that $M, \sigma \models \varphi$ for all $\varphi \in \Gamma$.*

2 μHDC

In this section we briefly introduce the extension of HDC by a least fixed point operator.

2.1 Languages of μHDC

A language of μHDC is built using the same sets of symbols as for HDC languages and a distinguished countable set of propositional variables X, Y, ... Terms are defined as in HDC. The BNF for formulas is extended to allow fixed point operator formulas as follows:

$\varphi ::= \bot | X | R(t, \dots, t) | \varphi \Rightarrow \varphi | (\varphi; \varphi) | \mu_i X \dots X.\varphi, \dots, \varphi | \exists x \varphi | \exists v \varphi | \exists P \varphi$

Formulas of the kind $\mu_i X_1 \dots X_m.\varphi_1, \dots, \varphi_n$ are well-formed only if $m = n$, all the occurrences of the variables $X_1, \dots, X_n$ in $\varphi_1, \dots, \varphi_n$ are *positive*, i.e. each of these occurrences is in the scope of an even number of negations, $X_1, \dots, X_n$ are distinct variables and $i \in \{1, \dots, n\}$. Formulas which contain μ are not regarded as fv. Note that we work with a vector form of the least fixed point operator. This has some technical advantages, because it enables elimination of nested occurrences of μ under some additional conditions.

2.2 Frames, models and satisfaction

Frames and models for μHDC languages are as for HDC languages. The only relative novelty is the extension of the satisfaction relation $\models$, which captures μ-formulas too.

Let $M = \langle F, I \rangle$ be a model for the (μHDC) language $\mathbf{L}$. Let $\tilde{I}(\varphi)$ denote the set $\{\sigma \in \mathbf{I}(T_F) : M, \sigma \models \varphi\}$ for an arbitrary formula φ from $\mathbf{L}$. Let s be a non-logical symbol in $\mathbf{L}$ and a be a constant, function or predicate of the type of s. We denote the interpretation of $\mathbf{L}$ into F which s-agrees with I and assigns a to s by I_s^a. Given a set $A \subseteq \mathbf{I}(T_M)$, we define the function $\chi_A : \mathbf{I}(T_M) \rightarrow \{0, 1\}$ by putting $\chi_A(\sigma) = 1$ iff $\sigma \in A$.

Now assume that the propositional variables $X_1, \dots, X_n$ occur in φ. We define the function $f_\varphi : \left(2^{\mathbf{I}(T_F)}\right)^n \rightarrow 2^{\mathbf{I}(T_F)}$ by the equality $f_\varphi(A_1, \dots, A_n) = (\widetilde{I_{X_1, \dots, X_n}^{\chi_{A_1}, \dots, \chi_{A_n}}})(\varphi)$. Assume that the variables $X_1, \dots, X_n$ have only positive occurrences in φ. Then f_φ is monotone on each of its arguments, i.e. $A_i \subseteq A'_i$ implies $f_\varphi(A_1, \dots, A_i, \dots, A_n) \subseteq f_\varphi(A_1, \dots, A'_i, \dots, A_n)$.

Now consider a sequence of n formulas, $\varphi_1, \dots, \varphi_n$, which have only positive occurrences of the variables $X_1, \dots, X_n$ in them. Then the system of inclusions

$$f_{\varphi_i}(A_1, \dots, A_n) \subseteq A_i,\ i = 1, \dots, n$$

has a least solution, which is also a least fixed point of the operator

$$\lambda A_1 \dots A_n.\langle f_{\varphi_1}(A_1, \dots, A_n), \dots, f_{\varphi_n}(A_1, \dots, A_n)\rangle.$$

Let this solution be $\langle B_1, \dots, B_n \rangle$, $B_i \subseteq \mathbf{I}(T_F)$. We define the satisfaction relation for
$\mu_i X_1 \dots X_n.\varphi_1, \dots, \varphi_n$ by putting:

$M, \sigma \models \mu_i X_1 \dots X_n.\varphi_1, \dots, \varphi_n$ iff $\sigma \in B_i$.

3 Simple μHDC formulas

The class of formulas which we call *simple* in this paper is a straightforward extension to the class of simple DC^* formulas considered in [Gue98]. We extend that class by allowing μ instead of iteration, positive formulas built up of fv symbols and existential quantification over the variables which occur in these formulas.

3.1 Super-dense chop

The *super-dense chop* operator $(. \circ .)$ was introduced in [ZH96] to enable the expression of sequential computation steps which consume negligible time, yet occur in some specified causal order, by DC. Given that $v_1, \dots, v_n$ are all the free temporal variables of formulas φ and ψ, $(\varphi \circ \psi)$ is equivalent to

$$\exists v'_1 \dots \exists v'_n \exists v''_1 \dots \exists v''_n \exists x_1 \dots \exists x_n \left(\begin{array}{l} [v'_1/v_1, \dots, v'_n/v_n]\varphi \wedge \bigwedge\limits_{i=1}^{n} \left(\begin{array}{c} \overleftarrow{v'_i} = \overleftarrow{v_i} \wedge \\ \overrightarrow{v'_i} = x_i \wedge \\ \Box v'_i = v_i \end{array} \right) ; \\ ; ([v''_1/v_1, \dots, v''_n/v_n]\psi \wedge \bigwedge\limits_{i=1}^{n} \left(\begin{array}{c} \overrightarrow{v''_i} = \overrightarrow{v_i} \wedge \\ \overleftarrow{v''_i} = x_i \wedge \\ \Box v''_i = v_i \end{array} \right) \end{array} \right)$$

3.2 Simple formulas

Definition 1. *Let* $\mathbf{L}$ *be a language for* μHDC *as above. We call* μHDC *formulas* γ *which can be defined by the BNF*

$\gamma ::= \bot | R(t, \dots, t) | X | (\gamma \wedge \gamma) | \gamma \vee \gamma | \neg\gamma | (\gamma ; \gamma) | (\gamma \circ \gamma) | \mu_i X \dots X.\gamma, \dots, \gamma$

where R *and* t *stand for either rigid or fv relation symbols and terms respectively,* open fv formulas. *We call an open fv formula* strictly positive *if it has no occurrences of propositional variables in the scope of* $\neg$. *An open fv formula is* propositionally closed *if it has no free occurrences of propositional variables.* Simple μHDC *formulas are defined by the BNF*

$\varphi ::= \ell = 0 | X | \lceil S \rceil | \lceil S \rceil \wedge \ell \prec a | \lceil S \rceil \wedge \ell \succ a | \lceil S \rceil \wedge \ell \prec a \wedge \ell \succ b |$
$\quad \varphi \vee \varphi | (\varphi ; \varphi) | (\varphi \circ \varphi) | \varphi \wedge \gamma | \mu_i X \dots X.\varphi, \dots, \varphi | \exists x \varphi | \exists v \varphi$

where a *and* b *denote rigid constants,* γ *denotes a a propositionally closed strictly positive open fv formula,* x *denotes a variable of arbitrary kind,* $\prec \in \{\leq, <\}$ *and* $\succ \in \{\geq, >\}$. *Additionally, a simple formula should not have subformulas of the kind* $\exists x \varphi$ *where* x *has a free occurrence in the scope of a* μ*-operator in* φ.

4 A complete proof system for the simple fragment of μHDC

In this section we show the completeness of a proof system for the fragment of μHDC where the application of μ is limited to simple formulas. We add the following axioms and rule to the proof system for HDC with abstract semantics:

(μ_1) $\Box(\mu_i X_1 \dots X_n.\varphi_1, \dots, \varphi_n \Leftrightarrow$
$[\mu_1 X_1 \dots X_n \varphi_1, \dots, \varphi_n / X_1, \dots, \mu_n X_1 \dots X_n \varphi_1, \dots, \varphi_n / X_n]\varphi_i)$

(μ_2) $\bigwedge_{i=1}^{n} \Box([\psi_1/X_1, \dots \psi_n/X_n]\varphi_i \Rightarrow \psi_i) \Rightarrow \Box(\mu_i X_1 \dots X_n.\varphi_1, \dots, \varphi_n \Rightarrow \psi_i)$ The

(μ_3) $\mu_i X_1 \dots X_m.\varphi_1, \dots, [\mu Z_1 \dots Z_n.\psi_1, \dots, \psi_n / Y]\varphi_k, \dots, \varphi_m \Leftrightarrow$
$\Leftrightarrow \mu_i X_1 \dots X_n Y Z_1 \dots Z_n \varphi_1, \dots, \varphi_m, \psi_1, \dots, \psi_n$

variable Y should not have negative free occurrences in φ_k in the instances of μ_3.

4.1 The completeness theorem

Lemma 1. *Let φ, α and β be HDC formulas and X be a propositional temporal letter. Let Y not occur in φ in the scope of quantifiers which bind any of the variables from $FV(\alpha) \cup FV(\beta)$. Then $\vdash_{\mu HDC} \Box(\alpha \Leftrightarrow \beta) \Rightarrow ([\alpha/Y]\varphi \Leftrightarrow [\beta/Y]\varphi)$.*

The following two propositions have a key role in our completeness argument. Detailed proofs are given in [Gue00b].

Proposition 1. *Let γ be a propositionally closed strictly positive open fv formula. Let M be a model for the language $\mathbf{L}$ of γ and $\sigma \in \mathbf{I}(T_M)$. Then there exists a μ-free propositionally closed strictly positive open fv formula γ' such that $M, \sigma \models \Box(\gamma \Leftrightarrow \gamma')$.*

This proposition justifies regarding μ formulas with fv subformulas as fv formulas.

Proposition 2 (local elimination of μ from simple formulas). *Let φ be a propositionally closed simple μHDC formula. Let M be a model for the language of φ and $\sigma \in \mathbf{I}(T_M)$. Then there exists a μ-free formula ψ such that $M, \sigma \models \Box(\varphi \Leftrightarrow \psi)$.*

Theorem 1 (completeness). *Let Γ be a set of formulas in a μHDC language $\mathbf{L}$. Let every μ-subformula of a formula $\varphi \in \Gamma$ be simple, and moreover occur in φ as a subformula of some propositionally closed μ-subformula of φ. Let Γ be consistent with respect to $\vdash_{\mu HDC}$. Then there exists a model M for $\mathbf{L}$ and an interval $\sigma \in \mathbf{I}(M)$ such that $M, \sigma \models \Gamma$.*

Proof. Proposition 1 entails that every fv μ-subformula of a formula from Γ is locally equivalent to a μ free fv formula. Hence occurrences of μ in fv subformulas can be eliminated using Lemma 1 and we may assume that there are no such subformulas. Since nested occurrences of μ in μ-subformulas from Γ can be eliminated by appropriate use of μ_3, we may assume that there are no such occurrences.

Let $S = \{s_{\mu_i X_1 \dots X_n.\varphi_1, \dots, \varphi_n} : 1 \le i \le n < \omega, \mu_i X_1 \dots X_n.\varphi_1, \dots, \varphi_n$ is a formula from $\mathbf{L}\}$ be a set of fresh 0-place flexible relation symbols. Let $\mathbf{L}(S)$ be the HDC language built using the non-logical symbols of $\mathbf{L}$ and the symbols from S. Every formula φ from $\mathbf{L}$ can be represented in the form $[\psi_1/X_1, \dots, \psi_n/X_n]\psi$ where ψ does not contain μ and contains $X_1, \dots, X_n$, and ψ_i, $i = 1, \dots, n$ are distinct μ-formulas. This representation is unique. Given this representation of φ, we denote the formula $[s_{\psi_1}/X_1, \ldots, s_{\psi_n}/X_n]\psi$ from $\mathbf{L}(S)$ by $\mathtt{t}(\varphi)$. Note that the translation $\mathtt{t}$ is invertible and its converse of is defined on the whole $\mathbf{L}(S)$.

Let $\Delta = \{\Box(\alpha) : \alpha$ is an instance of μ_1, μ_2 in $\mathbf{L}\}$. Then the set $\Gamma' = \{\mathtt{t}(\varphi) : \varphi \in \Gamma \cup \Delta\}$ is consistent with respect to $\vdash_{HDC}$. Assume the contrary. Then there exists a proof of $\perp$ with its premisses in Γ' in $\vdash_{HDC}$. Replacing each formula ψ in this proof by $\mathtt{t}^{-1}(\psi)$ gives a proof of $\perp$ from Γ in $\vdash_{\mu HDC}$.

Hence there exists a model M for $\mathbf{L}(S)$ and an interval $\sigma \in \mathbf{I}(T_M)$ such that $M, \sigma \models \Gamma'$.

Now let us prove that $M, \sigma \models \Box(\varphi \Leftrightarrow s_\varphi)$ for every closed simple formula φ from $\mathbf{L}$. Let φ be $\mu_i X_1 \ldots X_n.\psi_1, \ldots, \psi_n$. Let $\varphi_k \rightleftharpoons \mu_k X_1 \ldots X_n.\psi_1, \ldots, \psi_n$, $k = 1, \ldots, n$, for short. Then M satisfies the $\mathtt{t}$-translations

$$\Box(s_{\varphi_k} \Leftrightarrow [s_{\varphi_1}/X_1, \ldots, s_{\varphi_n}/X_n]\psi_k)$$

$$\bigwedge_{j=1}^{n} \Box(\mathtt{t}(\theta_j) \Leftrightarrow [\mathtt{t}(\theta_1)/X_1, \ldots \mathtt{t}(\theta_n)/X_n]\psi_j) \Rightarrow \Box(s_{\varphi_k} \Rightarrow \mathtt{t}(\theta_k))$$

of the instances of μ_1 and μ_2 for all n-tuples of formulas $\theta_1, \ldots, \theta_n$ from $\mathbf{L}$. The first of these instances implies that $\langle s_{\varphi_1}, \ldots, s_{\varphi_n}\rangle$ evaluates to a fixed point of the operator represented by $\langle \psi_1, \ldots, \psi_n\rangle$. Consider the instance of μ_2. Let θ_k be a μ-free formula from $\mathbf{L}$ such that $M, \sigma \models \Box(\theta_k \Leftrightarrow \varphi_k)$ for $k = 1, \ldots, n$. Such formulas exist by Proposition 2. Then $\mathtt{t}(\theta_k)$ is θ_k and the above instance of μ_2 is actually

$$\bigwedge_{j=1}^{n} \Box(\theta_j \Leftrightarrow [\theta_1/X_1, \ldots, \theta_n/X_n]\psi_j) \Rightarrow \Box(s_{\varphi_k} \Rightarrow \theta_k)$$

Besides $M, \sigma \models \Box(\theta_j \Leftrightarrow [\theta_1/X_1, \ldots, \theta_n/X_n]\psi_j)$, $j = 1, \ldots, n$, by the choice of θ_k. Hence $M, \sigma \models \Box(s_{\varphi_k} \Rightarrow \theta_k)$. This means that $\langle s_{\varphi_1}, \ldots, s_{\varphi_n}\rangle$ evaluates to the least fixed point of the operator represented by $\langle \psi_1, \ldots, \psi_n\rangle$. Hence $M, \sigma \models \Box(s_\varphi \Leftrightarrow \varphi)$ for every μ-formula φ with no nested occurrences of μ. This entails that $M, \sigma \models \Box(\varphi \Leftrightarrow \mathtt{t}(\varphi))$ for every $\varphi \in \Gamma$. Hence, $M, \sigma \models \Gamma$.

Acknowledgements

Guidance towards the topic addressed here, and a sequel of invigorating and pitfall marking discussions are thanks to He Jifeng. Some mistakes were detected in an early version of the paper thanks to Dang Van Hung and indirectly by Dimiter Skordev. Among other flaws, an undeliberate overclaim, which was also inconsistent with the announced purpose of the article, was avoided due to the efforts of anonymous referees.

References

[DG99] Dang Van Hung and D. P. Guelev. Completeness and Decidability of a Fragment of Duration Calculus with Iteration. In: P.S. Thiagarajan and R. Yap (eds), *Advances in Computing Science*, LNCS 1742, Springer-Verlag, 1999, pp. 139-150.

[DW94] Dang Van Hung and Wang Ji. On The Design of Hybrid Control Systems Using Automata Models. In: Chandru, V. and V. Vinay (eds.) LCNS 1180, *Foundations of Software Technology and Theoretical Computer Science*, 16th Conference, Hyderabad, India, December 1996, Springer, 1996.

[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.

[Gue98] Guelev, D. P. Iteration of Simple Formulas in Duration Calculus. Tech. report 141, UNU/IIST, June 1998.

[Gue00] Guelev, D. P. *Probabilistic and Temporal Modal Logics*, Ph.D. thesis, submitted, January 2000.

[Gue00b] Guelev, D. P. *A Complete Fragment of Higher-Order Duration μ-Calculus.* Tech. Report 195, UNU/IIST, April 2000.

[HX99] He Jifeng and Xu Qiwen. *Advanced Features of DC and Their Applications. Proceedings of the Symposium in Celebration of the Work of C.A.R. Hoare*, Oxford, 13-15 September, 1999.

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

[Koz83] Kozen, D. Results on the propositional μ-calculus. TCS 27:333-354, 1983.

[LH99] Li Li and He Jifeng. *A Denotational Semantics of Timed RSL using Duration Calculus. Proceedings of RTCSA'99*, pp. 492-503, IEEE Computer Society Press, 1999.

[Men64] Mendelson, E. *Introduction to Mathematical Logic.* Van Nostrand, Princeton, 1964.

[Pan95] Pandya, P. K. Some Extensions to Propositional Mean-Value Calculus. Expressiveness and Decidability. *Proceedings of CSL'95*, Springer-Verlag, 1995.

[PR95] Pandya, P. K. and Y Ramakrishna. *A Recursive Duration Calculus.* Technical Report TCS-95/3, TIFR, Bombay, 1995.

[PWX98] Pandya, P. K, Wang Hanping and Xu Qiwen. Towards a Theory of Sequential Hybrid Programs. *Proc. IFIP Working Conference PROCOMET'98* D. Gries and W.-P. de Roever (eds.), Chapman & Hall, 1998.

[SX98] Schneider, G. and Xu Qiwen. Towards a Formal Semantics of Verilog Using Duration Calculus. *Proceedings of FTRTFT'98*, Anders P. Ravn and Hans Rischel (eds.), LNCS 1486, pp. 282-293, Springer-Verlag, 1998.

[Wal93] Walurkiewicz, I. *A Complete Deductive System for the μ-Calculus.*, Ph.D. Thesis, Warsaw University, 1993.

[ZGZ99] Zhou Chaochen, D. P. Guelev and Zhan Naijun. *A Higher-Order Duration Calculus. Proceedings of the Symposium in Celebration of the Work of C.A.R. Hoare*, Oxford, 1999.

[ZH96] Zhou Chaochen and M. Hansen Chopping a Point. *Proceedings of BCS FACS 7th Refinement Workshop, Electronic Workshop in Computer Sciences*, Springer-Verlag, 1996.

[ZH00] Zhu Huibiao and He Jifeng. *A DC-based Semantics for Verilog* Tech. Report 183, UNU/IIST, January 2000.

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

[ZNJ99] Zhan Naijun. *Completeness of Higher-Order Duration Calculus.* Research Report 175, UNU/IIST, August 1999.

Cited UNU/IIST reports can be found at `http://www.iist.unu.edu`.