Completeness of Higher-Order Duration Calculus *

Zhan Naijun

Lab. of Computer Science and Techonology, Institute of Software,
the Chinese Academy of Sciences, Beijing, 100080, P.R. China
ttznj@ox.ios.ac.cn

Abstract. In order to describe the real-time behaviour of programs in terms of Duration Calculus (DC), proposed by Zhou Chaochen, C.A.R. Hoare and A.P. Ravn in [3], which can specify real-time requirements of computing systems, quantifications over program variables are inevitable, e.g. to describe local variable declaration, to declare internal channel and so on. So a higher-order duration calculus (HDC) is established in [2]. This paper proves completeness of HDC on abstract domains by encoding HDC into a complete first-order two-sorted interval temporal logic (IL_2). This idea is hinted by [9]. All results shown in this paper are done under the assumption that all program variables have finite variability.

Keywords: duration calculus higher-order logic interval temporal logic completeness

1 Introduction

In order to describe the real-time behaviour of programs in terms of DC, quantifications over program variables are inevitable, e.g. to describe local variable declaration and so on. So a higher-order duration calculus is established in [2]. In [2], a real-time semantics of local variables has been demonstrated, and some real-time properties of programs have been derived using HDC.

In order to specify the behaviour of real-time programs, program variables $V_i,\ i \geq 0$ are introduced into HDC. Predicates of program variables, constants, and global variables, such as $(V < 3)$ and $(V = x)$, are taken as states. To axiomatise the finite variability of program variables, the infinite rule (ω-rule) proposed in [8] is necessary, since [5] has shown that the finite variability cannot be axiomatised by finite rules on abstract domains.

In programming languages, value passing involves past and future time, to receive an initial value from the previous statement and to pass final value to the next statement. The chop modality ";" is a contracting one, and cannot express state properties outside the current interval. Therefore, two special functions

* The work is partially supported by UNU/IIST, and done during the author stayed at UNU/IIST as a follow (July 1998 to August 1999). The work is also partially supported by the National Natural Science Foundation of China under grant No. 69873003.

P. Clote and H. Schwichtenberg (Eds.): CSL 2000, LNCS 1862, pp. 442–456, 2000.

"$\leftarrow$" and "$\rightarrow$", firstly proposed in [4], are introduced into HDC. The functions "$\leftarrow$" and "$\rightarrow$" have a domain of state terms and a co-domain of functions from the intervals to duration domain. E.g. $\overleftarrow{V}= 4$ means that in a left neighbourhood of the current interval the value of V is 4. Symmetrically, $\overrightarrow{V}= 4$ means that in a right neighbourhood of the current interval the value of V is 4. In order to axiomatise them, the neighbourhood rule is introduced in [4].

In both interval temporal logic [5] and duration calculi [3,4], symbols are divided into flexible and rigid symbols (adopting the terminology of [1,7]). Rigid symbols are intended to represent fixed, global entities. Their interpretation will be the same in all the intervals. Conversely, entities which may vary in different intervals are represented by flexible symbols. Such a distinction between two classes of symbols is common in the context of first order temporal logics [1,7].

Completeness of interval temporal logics and duration calculi not only depends on the choice of time domain, but also relies on which kind of variables are quantified. In practice, we need to choose the reals as time domain. If so, we cannot get completeness of these systems, for if they were, they would be adequate for arithmetic, which is impossible by Gödel's Theorem. Therefore, if we want to choose the reals as time domain, we can only get relative completeness of these systems. E.g. relative completeness of DC has been proved in [10]. If we only quantify over global variables, duration calculi are complete on abstract domains shown in [8]. But if we introduce quantifications over program variables into DC, since we interpret program variables as functions from time domain to duration domain, no (consistent) system is complete for this semantics because whenever we interpret the domain of quantifiers as the set of all functions from time domain to duration domain, the language will have the expressive power of second-order arithmetic. So some restrictions on program variables are needed in order to work out a complete proof system, that is, that all program variables vary finitely is assumed. If so, we can reduce HDC to IL_2 . We will illustrate it as follows:

A naive way to reduce the second order logic to the first order one is to introduce for the class of n-ary predicates, $H^n(x_1, ..., x_n)$, a new $(n + 1)$-ary predicate, $E^{n+1}(z, x_1, ..., x_n)$, which has an additional argument z, and enumerates all $H^n(x_1, ..., x_n)$. Thus,

$$\exists H^n.\phi$$

could be reduced to

$$\exists z.\phi[E^{n+1}(z, x_1, ..., x_n)/H^n(x_1, ..., x_n)]$$

Therefore the second order logic could be reduced to a first order one. Detail discussion about this encoding can be seen in [6]. However, in order to define the $(n + 1)$-ary predicate E^{n+1}, we must have the following postulates, where we assume $(n = 1)$ and drop the indices of n and $(n + 1)$ for simplicity. Firstly,

$$\exists z.E(z, x_1) \qquad \text{and} \qquad \exists z.\neg E(z, x_1)$$

postulate that, for a singleton domain, E enumerates all H. Furthermore, together with the above two formulae, the formula

$$\exists z.\ (x_1 \neq x_2) \Rightarrow (E(z, x_1) \Leftrightarrow E(z_1, x_1) \wedge E(z, x_2) \Leftrightarrow E(z_2, x_2))$$

postulates that E enumerates all H over any finite domain. Unfortunately, with this approach, we can never define E to enumerate all H over an infinite domain. Hence second order predicate calculus cannot be reduced to first order one in this way in general.

However, by the finite variability of program variables, given an interval, any program variables V can be generated by finite combination of subintervals of the given one, over each of which V is constantly. Hence, it is possible to construct a 1-ary flexible function, $g(y)$, to enumerate all program variables by the postulates including

$$\lceil\!\rceil \vee \exists y.\lceil g(y) = c\rceil \text{ for any constant } c, \quad \text{and}$$
$$\lceil\!\rceil \vee \exists y.\ \lceil g(y) \Leftrightarrow g(y_1)\rceil; \lceil g(y) \Leftrightarrow g(y_2)\rceil$$

In this way, $\exists V.\phi$ can be reduced to $\exists y_V.dc2il(\phi)$ where $dc2il$ is a translating function from HDC to IL_2 defined later. A complete proof system for HDC can be established based on the completeness result of IL_2 . This idea is hinted by [9].

In order to prove completeness of HDC, we will establish IL_2 , a first-order two-sorted interval temporal logic firstly, in which global variables and functions are divided into two sorts. The rôle of the global variables and rigid functions of the first sort is as usual. The global variables and functions of the second sort and flexible functions of the first sort are used to enumerate program variables and the durations of state expressions in HDC respectively so that we can encode HDC into IL_2 by $dc2il$. Of course, it is not substantial to divide global variables and functions into two sorts, because we can encode many-sorted logic into one-sorted logic by introducing some specific predicates into one-sorted logic to distinguish different objects in the same universe (See [6]). Completeness of IL_2 can be proved with the method used in [5,8]. Because we can show that the consistency of a set of formulae Γ in HDC w.r.t. the proof system of HDC implies the consistency of $dc2il(\Gamma) \cup dc2il(Axiom_{hdc})$ w.r.t. IL_2 , where $Axiom_{hdc}$ stands for the set of all axiom instances for HDC, we can get a model $< \mathcal{F}, \mathcal{J} >$ which satisfies $dc2il(\Gamma) \cup dc2il(Axiom_{hdc})$ by completeness of IL_2 . According to the model $< \mathcal{F}, \mathcal{J} >$, we can construct a model $< \mathcal{F}', \mathcal{I} >$ for HDC which satisfies Γ. Thus, completeness of HDC can be proved.

We will omit the proofs for some lemmas and theorems later in order to save space, but their proofs can be found in [12].

2 Two-Sorted Interval Temporal Logic

In order to prove completeness of HDC on abstract domains, we shall establish IL_2 and then prove its completeness on abstract domains using the method provided in [5,8] in this section.

2.1 Syntax of IL_2

The alphabet of IL_2 includes:

- An infinite set of temporal variables $TVar = \{v_i \mid i \geq 0\}$.
- An infinite set of first sort global variables $Var^1 = \{x_i \mid i \geq 0\}$.
- An infinite set of second sort global variables $Var^2 = \{y_i \mid i \geq 0\}$.
- A special symbol ℓ, which stands for the length of an interval.
- An infinite set of propositional letters $PLetter = \{X_i \mid i \geq 0\}$.
- An infinite set of first sort function symbols $FSymb^1 = \{f_i^n, h_i^n \mid i, n \geq 0\}$. The distinction between f_i^n and h_j^m is that the former is rigid but the latter is flexible.
- A set of second sort flexible function symbols $FSymb^2 = \{g_i^n \mid i, n, \geq 0\}$.
- An infinite set of predicate symbols $RSymb = \{R_i^n \mid i, n \geq 0\}$.
- The connectives $\vee$ and $\neg$.
- The quantifier $\exists$ and the modality ;.

The *terms* of the first sort in IL_2 are defined by the following abstract syntax:

$$\theta \ ::= x \ \mid \ \ell \ \mid \ v \ \mid \ f_i^n(\theta_1, \ldots, \theta_n) \ \mid \ h_i^n(\theta_1, \ldots, \theta_n) \ \mid \ g_i^n(\vartheta_1, \ldots, \vartheta_n)$$

where ϑ_i is a term of the second sort defined as follows:

$$\vartheta \ ::= d \ \mid \ y$$

The *formulae* of IL_2 are defined inductively as follows:

$$\phi \ ::= \ X \ \mid \ R(\theta_1, \ldots, \theta_n) \ \mid \ \neg\phi \ \mid \ \phi \vee \psi \ \mid \ (\phi; \psi) \ \mid \ \exists z.\phi$$

where z stands for any global variable from $Var^1 \cup Var^2$.

A term (formula) is called *rigid* if neither temporal variable, nor ℓ, nor flexible function symbol occurs in it; otherwise called $flexible$. A formula is called *chop free*, if no ";" occurs in it.

2.2 Semantics of IL_2 on Abstract Domains

In this section, we give the meaning of the terms and formulae of IL_2 on abstract domains.

Definition 1. *A* time domain *is a linearly ordered set* $< T, \leq >$.

Definition 2. *Given a time domain* $< T, \leq >$, *we can define a set of intervals* $\mathbf{I}ntv(T) = \{[t_1, t_2] \mid t_1, t_2 \in T \text{ and } t_1 \leq t_2\}$, *where* $[t_1, t_2] = \{t \mid t \in T \text{ and } t_1 \leq t \leq t_2\}$.

Definition 3. *A* duration domain *is a system of the type* $< D, +, 0 >$, *which satisfies the following axioms:*

(D1) $a + (b + c) = (a + b) + c$
(D2) $a + 0 = a = 0 + a$
(D3) $a + b = a + c \Rightarrow b = c, \quad a + c = b + c \Rightarrow a = b$
(D4) $a + b = 0 \Rightarrow a = 0 = b$
(D5) $\exists c.a + c = b \vee b + c = a, \exists c.c + a = b \vee c + b = a$

That is, $< D, +, 0 >$ *is a totally ordered commutative group.*

Definition 4. *Given a time domain $<T, \leq>$ and a duration domain $<D, +, 0>$, a measure m is a function from T to D which satisfies the following conditions:*

(M1) $m([t_1, t_2]) = m([t_1, t_2']) \Rightarrow t_2 = t_2'$
(M2) $m([t_1, t]) + m([t, t_2]) = m([t_1, t_2])$
(M3) $m([t_1, t_2]) = a + b \Rightarrow \exists t.m([t_1, t]) = a \wedge (t_1 \leq t \leq t_2)$

Definition 5. *A frame of* IL_2 *is a quadruple of* $<< T, \leq>, < D, +, 0 >, D_1, m >$, *where* $< T, \leq>$ *is a time domain,* $< D, +, 0 >$ *is a duration domain,* D_1 *is called inhabited domain,* m *is a measure.*

Definition 6. *A model of* IL_2 *is a quintuple with type* $<< T, \leq>, < D, +, 0 >, D_1, m, \mathcal{J} >$, *where* $<< T, \leq>, < D, +, 0 >, D_1, m >$ *is a frame, and* $\mathcal{J}$ *is an* interpretation *of the symbols in* IL_2 *which satisfies the following conditions:* $\mathcal{J}(X) \in \mathbf{I}ntv(T) \to \{0, 1\}$ *for every* $X \in PLetter$; $\mathcal{J}(v) \in \mathbf{I}ntv(T) \to D$ *for every* $v \in TVar$; $\mathcal{J}(R_i^n) \in D^n \to \{0, 1\}$ *for every* $R_i^n \in RSymb$; $\mathcal{J}(f_i^n) \in D^n \to D$ *for every* $f_i^n \in FSymb^1$; $\mathcal{J}(h_i^n) \in D^n \times \mathbf{I}ntv(T) \to D$ *for every* $h_i^n \in FSymb^1$; $\mathcal{J}(g_i^n) \in D_1^n \times \mathbf{I}ntv(T) \to D$ *for every* $g_i^n \in FSymb^2$; *and* $\mathcal{J}(0) = 0, \mathcal{J}(+) = +, \mathcal{J}(=)$ *is* $=$, *and* $\mathcal{J}(\ell) = m$.

Definition 7. *Let* $\mathcal{J}$ *and* $\mathcal{J}'$ *be two interpretations defined as the above.* $\mathcal{J}$ *is* z*-equivalent to* $\mathcal{J}'$ *if* $\mathcal{J}$ *and* $\mathcal{J}'$ *have same values to all symbols, but possibly* z.

Given a model of IL_2 , $<< T, \leq>, < D, +, 0 >, D_1, m, \mathcal{J} >$, and an interval $[t_1, t_2] \in \mathbf{I}\mathrm{ntv}(T)$, the value of a term ϑ or θ can be defined as follows:

$$
\begin{array}{ll}
\mathcal{J}_{t_1}^{t_2}(y) = \mathcal{J}(y) & \text{for } y \in Var^2 \\
\mathcal{J}_{t_1}^{t_2}(x) = \mathcal{J}(x) & \text{for } x \in Var^1 \\
\mathcal{J}_{t_1}^{t_2}(v) = \mathcal{J}(v)([t_1, t_2]) & \text{for } v \in TVar \\
\mathcal{J}_{t_1}^{t_2}(f_i^n(\theta_1, \ldots, \theta_n) = \mathcal{J}(f_i^n)(\mathcal{J}_{t_1}^{t_2}(\theta) \ldots \mathcal{J}_{t_1}^{t_2}(\theta_n)) & \text{for } f_i^n \in FSymb^1 \\
\mathcal{J}_{t_1}^{t_2}(h_i^n(\theta_1, \ldots, \theta_n) = \mathcal{J}(h_i^n)([t_1, t_2], \mathcal{J}_{t_1}^{t_2}(\theta) \ldots \mathcal{J}_{t_1}^{t_2}(\theta_n)) & \text{for } h_i^n \in FSymb^1 \\
\mathcal{J}_{t_1}^{t_2}(g_i^n(\vartheta_1, \ldots, \vartheta_n) = \mathcal{J}(g_i^n)([t_1, t_2], \mathcal{J}_{t_1}^{t_2}(\vartheta) \ldots \mathcal{J}_{t_1}^{t_2}(\vartheta_n)) & \text{for } g_i^n \in FSymb^2
\end{array}
$$

Given a model $\mathcal{M} =< \mathcal{F}, \mathcal{J} >$ where $\mathcal{F} =<< T, \leq>, < D, +, 0 >, D_1, m >$, and an interval $[t_1, t_2]$, the meaning of a formula ϕ is explained by the following rules:

1. $< \mathcal{F}, \mathcal{J} >, [t_1, t_2] \models_{il2} X$ iff $\mathcal{J}(X)([t_1, t_2]) = tt$
2. $< \mathcal{F}, \mathcal{J} >, [t_1, t_2] \models_{il2} R^n(\theta_1, \ldots, \theta_n)$ iff $\mathcal{J}(R^n)(\mathcal{J}_{t_1}^{t_2}(\theta_1), \ldots, \mathcal{J}_{t_1}^{t_2}(\theta_n)) = tt$
3. $< \mathcal{F}, \mathcal{J} >, [t_1, t_2] \models_{il2} \neg\phi$ iff $< \mathcal{F}, \mathcal{J} >, [t_1, t_2] \not\models_{il2} \phi$
4. $< \mathcal{F}, \mathcal{J} >, [t_1, t_2] \models_{il2} \phi \vee \psi$
   iff $< \mathcal{F}, \mathcal{J} >, [t_1, t_2] \models_{il2} \phi$ or $< \mathcal{F}, \mathcal{J} >, [t_1, t_2] \models_{il2} \psi$
5. $< \mathcal{F}, \mathcal{J} >, [t_1, t_2] \models_{il2} \phi; \psi$
   iff $< \mathcal{F}, \mathcal{J} >, [t_1, t] \models_{il2} \phi$ and $< \mathcal{F}, \mathcal{J} >, [t, t_2] \models_{il2} \psi$ for some $t \in [t_1, t_2]$
6. $< \mathcal{F}, \mathcal{J} >, [t_1, t_2] \models_{il2} \exists z.\phi$ iff $< \mathcal{F}, \mathcal{J}' >, [t_1, t_2] \models_{il2} \phi$ for some interpretation $\mathcal{J}'$ which is z-equivalent to $\mathcal{J}$

Satisfaction and validity can be defined in the usual way, see [12].
The following abbreviations will be used:

$\Diamond\phi \;\widehat{=}\; \mathbf{true};(\phi;\mathbf{true})$ reads: "for some sub-interval: ϕ"
$\Box\phi \;\widehat{=}\; \neg\Diamond(\neg\phi)$ reads: "for all sub-intervals: ϕ"

Furthermore, the standard abbreviations from predicate logic will be used. When $\neg, \exists z, \Box,$ and $\Diamond$ occur in formulae they have higher precedence than the binary connectives and the modality ;. The modality ; has higher precedence than the binary connectives.

Definition 8. *Let Φ be an* IL_2 *formula, and let $\mathcal{M} =< \mathcal{F}, \mathcal{J} >$ be a model of* IL_2 *, where $\mathcal{F} =<< T, \leq >, < D, +, 0 >, D_1, m >$ is the corresponding frame. Φ is said to have the* finite variability *on $\mathcal{M}$ if for every $[t_1, t_2] \in \mathbf{I}ntv(T)$, $\mathcal{M}, [t_1, t_2] \models_{il2} \Phi \Rightarrow \Box\Phi$ and there exist $t'_1, \ldots, t'_n$, such that $t_1 = t'_1 \leq \ldots \leq t'_n = t_2$ and for all $i = 1, \ldots, n-1$ $\mathcal{M}, [t'_i, t'_{i+1} \models_{il2} \Box\Phi$. Φ is said to have finite variability on a class of models $\mathcal{K}$ if it has the property on every member of $\mathcal{K}$.*

Definition 9. *Let Φ be an* IL_2 *formula. We define the sequence of formulae $\{\Phi^k\}_{k<\omega}$ as follows:*

$$\Phi^0 \;\widehat{=}\; \ell = 0, \qquad \Phi^{k+1} \;\widehat{=}\; (\Phi^k;\Box\Phi)$$

For the rest of this section we will fix a set of IL_2 formulae Ω and consider only IL_2 models on which Φ has finite variability for every $\Phi \in \Omega$. We will use $\mathcal{K}_\Omega$ to denote the class of models that satisfy the above property later. So the following proof system takes Ω as a parameter. Of course, all discussions below can be applied to an arbitrary set of IL_2 formulae Ω. If $\Omega = \emptyset$, then the case is same as in [5]. The finite variability of Φ means that for any interval one can partition the interval into finitely many subintervals such that $\Box\Phi$ holds for each of the subintervals. The axiom ITL_Ω and rule IR^Φ given below are used to axiomatise the finite variability of all $\Phi \in \Omega$.

2.3 Proof System of IL_2 with Ω

In this section, we give a sound and complete proof system of IL_2 with Ω w.r.t. $\mathcal{K}_\Omega$. The notation $\vdash_{il2_\Omega} \phi$ means that ϕ is provable, i.e. that ϕ is a theorem of IL_2 with Ω.

Definition 10. *A term θ is called* free for x *in ϕ if x does not occur freely in ϕ within a scope of $\exists x'$ or $\forall x'$ where x' is any variable occurring in θ.*

The axioms of IL_2 are:

ITL1: $\ell \geq 0$
ITL2: $((\phi;\psi) \wedge \neg(\phi;\varphi)) \Rightarrow (\phi;(\psi \wedge \neg\varphi))$
$((\phi;\psi) \wedge \neg(\varphi;\psi)) \Rightarrow ((\phi \wedge \neg\varphi);\psi)$
ITL3: $((\phi;\psi);\varphi) \;\Leftrightarrow\; (\phi;(\psi;\varphi))$
ITL4: $(\phi;\psi) \;\Rightarrow\; \phi$ if ϕ is a rigid formula
$(\phi;\psi) \;\Rightarrow\; \psi$ if ψ is a rigid formula

$$
\begin{array}{ll}
\text{ITL5:} & (\exists z.\phi;\psi) \;\Rightarrow\; \exists z.(\phi;\psi) \quad \text{if } z \text{ is not free in } \psi \\
& (\phi;\exists z.\psi) \;\Rightarrow\; \exists z.(\phi;\psi) \quad \text{if } z \text{ is not free in } \phi \\
\text{ITL6:} & ((\ell = a);\phi) \;\Rightarrow\; \neg((\ell = a);\neg\phi) \\
& (\phi;(\ell = a)) \;\Rightarrow\; \neg(\neg\phi;(\ell = a) \\
\text{ITL7:} & (a \geq 0 \wedge b \geq 0) \Rightarrow ((\ell = a + b) \Leftrightarrow ((\ell = a);(\ell = b))) \\
\text{ITL8:} & \phi \Rightarrow (\phi;(\ell = 0)) \\
& \phi \Rightarrow ((\ell = 0);\phi) \\
\text{ITL}_\Omega : & \Phi \Rightarrow \Box\Phi \qquad \text{for all } \Phi \in \Omega
\end{array}
$$

The inference rules of IL_2 are:

$$
\begin{array}{ll}
\text{N:} & \begin{array}{l}\text{if } \phi \text{ then } \neg(\neg\phi;\psi) \\ \text{if } \psi \text{ then } \neg(\psi;\neg\phi)\end{array} \qquad
\text{M:} \quad \begin{array}{l}\text{if } \phi \Rightarrow \psi \text{ then } (\phi;\varphi) \Rightarrow (\psi;\varphi) \\ \text{if } \phi \Rightarrow \psi \text{ then } (\varphi;\phi) \Rightarrow (\varphi;\psi)\end{array} \\
\text{IR}^{\Phi} & \dfrac{H(\Phi^0/X) \quad \forall k{<}\omega.H(\Phi^k/X) \Rightarrow H(\Phi^{k+1}/X)}{H(\mathbf{true}/X)} \qquad \text{for } \Phi \in \Omega
\end{array}
$$

The proof system of IL_2 with Ω also contains all axioms and rules for propositional logic, predicate logic, and real arithmetic, such as

$$(\text{G}):\ \text{if } \phi \text{ then } \forall z.\phi$$

However, for the following axiom, side condition is necessary.

$$(\text{Q}) : \forall x.\phi(x) \;\Rightarrow\; \phi(\theta) \begin{cases} \text{if either } \theta \text{ is free for } x \text{ in } \phi(x) \text{ and } \theta \text{ is rigid} \\ \text{or } \theta \text{ is free for } x \text{ in } \phi(x) \text{ and } \phi(x) \text{ is chop free.} \end{cases}$$

An explanation of the necessity of the side condition is given in [11]. The axioms and rules for equality, addition, etc. in real arithmetic will not be listed here, but can be found in [11].

Theorem 1 (Soundness). *The proof system is sound, i.e.* $\vdash_{il2_\Omega} \phi$ *implies* $\models_{il2_\Omega} \phi$, *where* $\models_{il2_\Omega} \phi$ *means* ϕ *is valid on every model* $\mathcal{M} \in \mathcal{K}_\Omega$.

Definition 11. *Given a set of* IL_2 *formulae* Γ, *if* $\Gamma \nvdash_{il2_\Omega}$ **false**, *then* Γ *is called* consistent *w.r.t. the proof system of* IL_2 *with* Ω, *otherwise called* inconsistent.

Theorem 2 (Completeness). *Given a set of* IL_2 *formulae* Γ. *if* Γ *is consistent w.r.t. the proof system of* IL_2 *with* Ω, *then there exists a model* $\mathcal{M} =<< T, \leq>, < D, +, 0 >, D_1, m, \mathcal{J} >$ *on which* Φ *has finite variability for every* $\Phi \in \Omega$, *and an interval* $[t_1, t_2] \in \mathbf{I}ntv(T)$ *such that* $\mathcal{M}, [t_1, t_2] \models_{il2_\Omega} \Gamma$.

Proof. Using the method provided in [5,8], it can be proved. See [12]. □

3 Higher-Order Duration Calculus

In this section, we establish a higher-order duration calculus, which is an extension of the original duration calculus, by introducing program variables and quantifications over them.

3.1 Syntax of HDC

The alphabet of HDC contains all symbols of IL_2 except for the symbols of the second sort and the flexible function symbols of the first sort. Besides, it also includes an infinite set of program variables $PVar = \{V_i \mid i \geq 0\}$. In HDC, all temporal variables have a special structure $\int S$ where S is a state expression defined as follows:

$$S \quad ::= \quad 0 \quad | \quad 1 \quad | \quad S_1 \vee S_2 \quad | \quad \neg S \quad | \quad R(\vartheta_1, \ldots, \vartheta_n)$$

where R is the characteristic function of predicate R, and $\vartheta_1, \ldots, \vartheta_n$ are called state terms defined as:

$$\vartheta ::= x \quad | \quad V \quad | \quad f(\vartheta_1, \ldots, \vartheta_n)$$

The *terms* of HDC are constructed as follows:

$$\theta \quad ::= x \quad | \quad \ell \quad | \quad \overleftarrow{\vartheta} \quad | \quad \overrightarrow{\vartheta} \quad | \quad v \quad | \quad f(\theta_1, \ldots, \theta_n)$$

where v has the form $\int S$ where S is a state expression defined above, and "$\leftarrow$" and "$\rightarrow$" are two special functions with a domain of state terms and a codomain of functions from the intervals to duration domain.

The *formulae* of HDC are defined inductively as follows:

$$\phi \quad ::= \quad X \quad | \quad R(\theta_1, \ldots, \theta_n) \quad | \quad \neg\phi \quad | \quad \phi \vee \psi \quad | \quad (\phi; \psi) \quad | \quad \exists x.\phi \quad | \quad \exists V.\phi$$

A state term (term or formula) is called *rigid* if neither program variable nor ℓ occurs in it; otherwise called *flexible*.

Remark 1. We can show that a rigid state expression is also a rigid formula by the above definitions if we do not distinguish predicate and its characteristic function. For example, $(x + 3 > 1)$ can be taken as a state as well as a formula according to the syntactic definitions above. In order to avoid confusion, when S is rigid, we will use ϕ_S to stand for the rigid formula corresponding to S.

3.2 Semantics of HDC

In this subsection, we give the meaning of terms and formulae in HDC on abstract domains. HDC frames are essentially IL_2 frames too, but a slight difference is that there is no inhabited domain in HDC frames.

Definition 12. *A model of HDC is a quadruple with type* $<<T, \leq>, <D, +, 0>, m, \mathcal{I}>$, *where* $<<T, \leq>, <D, +, 0>, m>$ *is a frame, and* $\mathcal{I}$ *is an* interpretation *of the symbols in HDC which satisfies the following condition:*

For every $V \in PVar$, *and every* $[t_1, t_2] \in \mathbf{I}ntv(T)$ *there exists* $t'_1, \ldots, t'_n$ *such that* $t_1 = t'_1 \leq \ldots \leq t'_n = t_2$, *and for any* $t, t' \in [t'_i, t'_{i+1})$ *implies* $\mathcal{I}(V)(t) = \mathcal{I}(V)(t')$.

This property is known as the finite variability of program variables.

Given a model of HDC $\mathcal{M} =<< T, \leq>, < D, +, 0 >, m, \mathcal{I} >$, the meaning of program variables and propositional letters is given as: $\mathcal{I}(V)T \to D$ and $\mathcal{I}(X) \in \mathbf{I}\text{ntv}(T) \to \{tt, ff\}$ respectively.

The semantics of a state term ϑ, given a model $\mathcal{M} =< \mathcal{F}, \mathcal{I} >$, is a function with type $T \to D$ defined inductively on its structure as follows:

$$\begin{array}{ll} \mathcal{I}(x)(t) & = \mathcal{I}(x) \\ \mathcal{I}(V)(t) & = \mathcal{I}(V)(t) \\ \mathcal{I}(f^n(\vartheta_1, \ldots, \vartheta_n))(t) & = \mathcal{I}(f^n)(\mathcal{I}(\vartheta_n)(t), \ldots, \mathcal{I}(\vartheta_n)(t)) \end{array}$$

The semantics of a state expression S, given a model $\mathcal{M} =< \mathcal{F}, \mathcal{I} >$, is a function with type $T \to \{0, 1\}$ defined inductively on its structure as follows:

$$\begin{array}{ll} \mathcal{I}(0)(t) & = 0 \\ \mathcal{I}(1)(t) & = 1 \\ \mathcal{I}(R^n(\vartheta_1, \ldots, \vartheta_n))(t) & = \mathcal{I}(R^n)(\mathcal{I}(\vartheta_1)(t), \ldots, \mathcal{I}(\vartheta_n)(t)) \\ \mathcal{I}(\neg S)(t) & = 1 - \mathcal{I}(S)(t) \\ \mathcal{I}(S_1 \vee S_2)(t) & = \begin{cases} 0 \text{ if } \mathcal{I}(S_1)(t) = 0 \text{ and } \mathcal{I}(S_2)(t) = 0 \\ 1 \text{ otherwise} \end{cases} \end{array}$$

Lemma 1. *Let S be a state expression and $\mathcal{I}$ be an interpretation of the symbols in HDC on a frame $\mathcal{F} =<< T, \leq>, < D, +, 0 >, m >$. Then for every $[t_1, t_2] \in \mathbf{I}ntv(T)$ there exist $t'_1, \ldots, t'_n$ such that $t_1 = t'_1 \leq \ldots \leq t'_n = t_2$, and for any $t, t' \in [t'_i, t'_{i+1})$ implies $\mathcal{I}(S)(t) = \mathcal{I}(S)(t')$ for all $i = 1, \ldots, n-1$.*

Proof. Induction on the construction of S. □

Using Lemma 1, we can give the interpretation of $\int S$ under an HDC model $\mathcal{M} =< \mathcal{F}, \mathcal{I} >$. Let $[t_1, t_2] \in \mathbf{I}\text{ntv}(T)$ and $t'_1, \ldots, t'_n$ be a partition of $[t_1, t_2]$ which have the property stated in Lemma 1. We define $p \bullet c$ for $p \in \{0, 1\}$ and $c \in D$ as follows:

$$p \bullet c \quad = \quad \begin{cases} 0 & \text{if } p = 0 \\ c & \text{if } p = 1 \end{cases}$$

Then $\mathcal{I}(\int S)([t_1, t_2]) = \sum_{i=1}^{n-1} \mathcal{I}(S)(t'_i) \bullet m([t'_i, t'_{i+1}])$. It is easy to show that this definition does not depend on the particular choice $t'_1, \ldots, t'_n$.

Given a model $\mathcal{M} =< \mathcal{F}, \mathcal{I} >$, and an interval $[t_1, t_2] \in \mathbf{I}\text{ntv}(T)$, the meaning of initial and final values of state terms $\overleftarrow{\vartheta}, \overrightarrow{\vartheta}, \overleftarrow{\vartheta_1}, \overrightarrow{\vartheta_1}, \ldots$, are functions with type $\mathbf{I}\text{ntv}(T) \to D$ defined as follows:

$$\mathcal{I}(\overleftarrow{\vartheta}, [t_1, t_2]) = d, \textit{ iff } < \mathcal{F}, \mathcal{I} >, [t_1 - \delta, t_1] \vdash_{hdc} \lceil \vartheta = d \rceil, \text{ for some } \delta > 0.$$
$$\mathcal{I}(\overrightarrow{\vartheta}, [t_1, t_2]) = d, \textit{ iff } < \mathcal{F}, \mathcal{I} >, [t_2, t_2 + \delta] \vdash_{hdc} \lceil \vartheta = d \rceil, \text{ for some } \delta > 0.$$

where $\lceil S \rceil \ \widehat{=} \ \int S = \ell \wedge \ell > 0$. It means that S takes value 1 almost everywhere in a non-point interval. We will use $\lceil \rceil$ to stand for $\ell = 0$.

The meaning of other syntactic entities in HDC can be given similarly to the ones in IL_2 , and other notions for HDC also can be defined similarly.

3.3 Proof System of HDC

In this section, we give a proof system of HDC. The notation $\vdash_{hdc} \phi$ means that ϕ is provable.

The proof system of HDC includes all axioms and inference rules in IL_2 but the axiom ITL_Ω. Besides, it also includes the following three groups of axioms and rules.

The first group is used to specify how to calculate and reason about state durations. They are:

$$\begin{array}{llll} \text{(DC1)} & \int 0 = 0 & \text{(DC4)} & \int S_1 + \int S_2 = \int(S_1 \vee S_2) + \int(S_1 \wedge S_2) \\ \text{(DC2)} & \int 1 = \ell & \text{(DC5)} & ((\int S = x_1)^\frown(\int S = x_2)) \Rightarrow (\int S = x_1 + x_2) \\ \text{(DC3)} & \int S \geq 0 & \text{(DC6)} & \int S_1 = \int S_2, \text{ if } S_1 \Leftrightarrow S_2 \\ & & \text{(DC7)} & \lceil S \rceil \ \Leftrightarrow \ (\phi_S \wedge \ell > 0), \text{ if } S \text{ is rigid} \end{array}$$

The rôle of the second group is to calculate the initial and final values of ϑ, $\overleftarrow{\vartheta}$ and $\overrightarrow{\vartheta}$. They are:

$$\begin{array}{ll} \text{(PV1)} & (\ell > 0); ((\overleftarrow{\vartheta} = x_1) \wedge (\ell = x_2)) \ \Leftrightarrow \ \mathbf{true}; \lceil \vartheta = x_1 \rceil; (\ell = x_2) \\ \text{(PV2)} & ((\overrightarrow{\vartheta} = x_1) \wedge (\ell = x_2)); (\ell > 0) \ \Leftrightarrow \ (\ell = x_2); \lceil \vartheta = x_1 \rceil; \mathbf{true} \end{array}$$

PV1 and PV2 formulate the meaning of the initial value and final value of a state term which are inherited from the previous statement, and passed to the next one. Because the function $\leftarrow$ ($\rightarrow$) involves the value of a state term at left neighbourhood (right neighbourhood), the neighbourhood rule is necessary in order to axiomatise them.

$$\text{NR} \quad \text{If } \ (\ell = a); \Psi; (\ell = b) \Rightarrow (\ell = a); \Upsilon; (\ell = b), \quad \text{then } \Psi \Rightarrow \Upsilon. \ (a, b \geq 0)$$

Remark 2. This rule can be looked as a rule of IL_2 . Although the rule will destroy the deduction theorem of IL_2 , IL_2 will keep completeness after introducing it.

The last group is used to specify the semantics of V, $\overleftarrow{V}$ and $\overrightarrow{V}$ in the context of quantifications.

The axiom and rule below are standard as in predicate logic.

$$\begin{array}{ll} \mathrm{G_V}: & \text{if } \phi \text{ then } \forall V.\phi \\ \mathrm{Q_V}: & \forall V.\phi(V) \ \Rightarrow \ \phi(\vartheta) \end{array}$$

If $\overleftarrow{V}$ ($\overrightarrow{V}$) does not occur in formula ϕ, then it can take any value, since the value of $\overleftarrow{V}$ ($\overrightarrow{V}$) is defined by value of V outside the reference interval w.r.t. ϕ. Hence,

$$\begin{array}{lll} \text{(HDC1)} & \exists V.\phi \ \Rightarrow \ \exists V. \ \phi \wedge (\overleftarrow{V} = x) & \text{if } \overleftarrow{V} \notin \phi \\ \text{(HDC2)} & \exists V.\phi \ \Rightarrow \ \exists V. \ \phi \wedge (\overrightarrow{V} = x) & \text{if } \overrightarrow{V} \notin \phi \end{array}$$

The distributivity of $\exists V$ over the chop operator is the most essential property of V as a function over time. $\exists V$ can distribute over the chop, if and only if the value of $\overrightarrow{V}$ in the left operand of the chop can match the value of V in the right operand, and symmetrically for the value of $\overleftarrow{V}$ in the right operand. That is,

$$\text{(HDC3)} \quad \left(\begin{array}{l} (\exists V.\ \phi \wedge (\mathit{true}; \lceil V = x_1 \rceil \vee \lceil \rceil) \wedge (\overrightarrow{V} = x_2)) \\ ; (\exists V.\ \psi \wedge (\lceil V = x_2 \rceil; \mathbf{true} \vee \lceil \rceil) \wedge (\overleftarrow{V} = x_1)) \end{array} \right) \ \Rightarrow\ \exists V.\ \phi; \psi$$

When $\overrightarrow{V} \notin \phi$ or $\overleftarrow{V} \notin \psi$, it can be derived from the above axioms that $(\exists V.\phi); \exists V.\psi \ \Rightarrow\ \exists V.\ \phi; \psi$

In order to define program variables as finitely varied functions, in the proof system, we let $\Omega = \Omega_{hdc}$ where $\Omega_{hdc} \ \widehat{=}\ \{\exists x(\lceil V = x \rceil \vee \lceil\ \rceil) \mid V \in PVar\}$.

Theorem 3 (Soundness). *The proof system of HDC is sound, i.e.* $\vdash_{hdc} \phi$ *implies* $\models_{hdc} \phi$

4 Completeness of HDC on Abstract Domains

In this section, we will apply completeness of IL_2 with Ω to show HDC is complete on abstract domains. To this end, let us choose a language $\mathcal{L}_{IL_2}$ for IL_2 with four special flexible function symbols $\oplus$, $\ominus$, h_r and h_l, in which there is only one unary function symbol g of the second sort, and a language $\mathcal{L}_{hdc}$ for HDC. $\oplus$ and $\ominus$ have type $((\mathbf{I}\mathrm{ntv}(T) \to D) \times (\mathbf{I}\mathrm{ntv}(T) \to D)) \to (\mathbf{I}\mathrm{ntv}(T) \to D)$, and h_l and h_r have type $(\mathbf{I}\mathrm{ntv}(T) \to D) \to (\mathbf{I}\mathrm{ntv}(T) \to D)$. In $\mathcal{L}_{IL_2}$, the definition of terms will be extended by allowing that duration terms and neighbourhood terms are also terms, where duration terms are defined as: $h(\theta_1, \ldots, \theta_n)$ is duration term; If t_1 and t_2 are duration terms then $t_1 \oplus t_2$ and $t_1 \ominus t_2$ are both duration terms too, neighbourhood terms are defined as: If nt is of the forms x or $g(y)$ or $f(nt_1, \ldots, nt_n)$ then $h_r(nt)$ and $h_l(nt)$ are both neighbourhood terms. It is easy to define the meaning of the above extensions using the usual way. Obviously, IL_2 with Ω is still complete after extending. We will use duration terms to correspond the terms of state durations, neighbourhood terms to correspond the left and right values of state terms in the below translation $dc2il$ from $\mathcal{L}_{hdc}$ to $\mathcal{L}_{IL_2}$.

Let us fix two bijections: $V \to y_V$, and $R \to h_R$ between $\mathcal{L}_{hdc}$ and $\mathcal{L}_{IL_2}$. We will establish a bijection between $\mathcal{L}_{hdc}$ and a subset of $\mathcal{L}_{IL_2}$ by function $dc2il$ from $\mathcal{L}_{hdc}$ to a subset of $\mathcal{L}_{IL_2}$ and its inverse $il2dc$.

We can prove that if a set of formulae Γ in $\mathcal{L}_{hdc}$ is consistent w.r.t. the proof system of HDC then $dc2il(\Gamma)$ plus $dc2il(Axiom_{hdc})$ where $Axiom_{hdc}$ contains all axiom instances of HDC is consistent w.r.t. the proof system of IL_2 with $dc2il(\Omega_{hdc})$. From now on, let $\Omega = dc2il(\Omega_{hdc})$. By Theorem 2, we can get a model $< \mathcal{F}, \mathcal{J} >$ and an interval $[t_1, t_2]$ such that $< \mathcal{F}, \mathcal{J} >, [t_1, t_2] \models_{il2_\Omega} dc2il(\Gamma) \cup dc2il(Axiom_{hdc})$. Finally, according to the model and interval, we can construct a model $< \mathcal{F}', \mathcal{I} >$ for HDC such that $< \mathcal{F}, \mathcal{I} >, [t_1, t_2] \models_{hdc} \Gamma$.

We define the translating function $dc2il$ from $\mathcal{L}_{hdc}$ to $\mathcal{L}_{IL_2}$ as follows:

$$dc2il(\vartheta) \;\widehat{=}\; \begin{cases} x & \text{if } \vartheta = x \\ g(y_V) & \text{if } \vartheta = V \\ f(dc2il(\vartheta_1), \ldots, dc2il(\vartheta_n)) & \text{if } \vartheta = f(\vartheta_1, \ldots, \vartheta_n) \end{cases}$$

$$dc2il(\theta) \;\widehat{=}\; \begin{cases} x & \text{if } \theta = x \\ h_l(dc2il(\vartheta)) & \text{if } \theta = \overleftarrow{\vartheta} \\ h_r(dc2il(\vartheta)) & \text{if } \theta = \overrightarrow{\vartheta} \\ f(dc2il(\theta_1), \ldots, dc2il(\theta_n)) & \text{if } \theta = f(\theta_1, \ldots, \theta_n) \end{cases}$$

$$dc2il(\textstyle\int S) \;\widehat{=}\; \begin{cases} h_0 & \text{if } S = 0 \\ h_1 & \text{if } S = 1 \\ h_R(dc2il(\vartheta_1), \ldots, dc2il(\vartheta_n)) & \text{if } S = R(\vartheta_1, \ldots, \vartheta_n) \\ \ell - dc2il(\int S_1) & \text{if } S = \neg S_1 \\ dc2il(\int S_1) \ominus dc2il(\int S_2) & \text{if } S = S_1 \wedge S_2 \\ dc2il(\int S_1) \oplus dc2il(\int S_2) & \text{if } S = S_1 \vee S_2 \end{cases}$$

$$dc2il(\phi) \;\widehat{=}\; \begin{cases} X & \text{if } \phi = X \\ R(dc2il(\theta_1), \ldots, dc2il(\theta_n)) & \text{if } \phi = R(\theta_1, \ldots, \theta_n) \\ \neg dc2il(\psi) & \text{if } \phi = \neg\psi \\ dc2il(\phi_1) \vee dc2il(\phi_2) & \text{if } \phi = \phi_1 \vee \phi_2 \\ dc2il(\phi_1) \wedge dc2il(\phi_2) & \text{if } \phi = \phi_1 \wedge \phi_2 \\ \exists x.dc2il(\psi) & \text{if } \phi = \exists x.\psi \\ \exists y_V.dc2il(\psi) & \text{if } \phi = \exists V.\psi \end{cases}$$

where $h_0 = 0$ and $h_1 = \ell$.

Symmetrically, we define its inverse $il2dc$ as follows:

$$il2dc(\theta) \;\widehat{=}\; \begin{cases} x & \text{if } \theta = x \\ V & \text{if } \theta = g(y_V) \\ \overleftarrow{il2dc(\theta)} & \text{if } \theta = h_l(\theta) \\ \overrightarrow{il2dc(\theta)} & \text{if } \theta = h_r(\theta) \\ f(il2dc(\theta_1), \ldots, il2dc(\theta_n)) & \text{if } \theta = f(\theta_1, \ldots, \theta_n) \\ \int 0 & \text{if } \theta = h_0 \\ \int 1 & \text{if } \theta = h_1 \\ \int R(il2dc(\theta_1), \ldots, il2dc(\theta_n)) & \text{if } \theta = h_R(\theta_1, \ldots, \theta_n) \end{cases}$$

$il2dc(h_{R_1}(\theta_{11}, \ldots, \theta_{1n_1}) * \ldots * h_{R_m}(\theta_{m1}, \ldots, \theta_{mn_m})) \;\widehat{=}\; \int(R_1(il2dc(\theta_{11}), \ldots, il2dc(\theta_{1n_1}))\& \ldots \& R_m(il2dc(\theta_{m1}), \ldots, il2dc(\theta_{mn_m})))$ where $* \in \{\oplus, \ominus\}$ and $\& \in \{\vee, \wedge\}$. If $* = \oplus$ then the corresponding $\& = \vee$, otherwise the corresponding $\& = \wedge$.

$$il2dc(\phi) \;\widehat{=}\; \begin{cases} X & \text{if } \phi = X \\ R(il2dc(\theta_1), \ldots, il2dc(\theta_n)) & \text{if } \phi = R(\theta_1, \ldots, \theta_n) \\ \neg il2dc(\psi) & \text{if } \phi = \neg\psi \\ il2dc(\phi_1) \vee il2dc(\phi_2) & \text{if } \phi = \phi_1 \vee \phi_2 \\ il2dc(\phi_1) \wedge il2dc(\phi_2) & \text{if } \phi = \phi_1 \wedge \phi_2 \\ \exists x.il2dc(\psi) & \text{if } \phi = \exists x.\psi \\ \exists V.il2dc(\psi) & \text{if } \phi = \exists y_V.\psi \end{cases}$$

From the definitions of $dc2il$ and $il2dc$ above, we have the following result.

Theorem 4. *For any set of formulae $\Gamma \subset \mathcal{L}_{hdc}$, Γ is consistent w.r.t. the proof system of HDC iff $dc2il(\Gamma) \cup dc2il(Axiom_{hdc})$ is consistent w.r.t. the proof system of* IL_2 *with $dc2il(\Omega_{hdc})$.*

Proof. By the above definitions of $dc2il$ and $il2dc$, it is trivial. □

Theorem 5. *If Γ is consistent w.r.t. the proof system of HDC, then Γ is satisfiable.*

Proof. The consistency of Γ w.r.t. HDC implies the consistency of $\Gamma_0 = \{\ell = a\}; \Gamma; \{\ell = b\}$ w.r.t. HDC where $a, b > 0$ by the neighbourhood rule. The consistency of Γ_0 w.r.t. HDC implies the consistency of $dc2il(\Gamma_0) \cup dc2il(Axiom_{hdc})$ w.r.t. IL_2 with Ω by Theorem 4. Hence, by Theorem 2, there exists an IL_2 model $\mathcal{M} =< \mathcal{F}, \mathcal{J} >$ on which Φ has the finite variability property for every $\Phi \in \Omega$, where $\mathcal{F} =<< T, \leq >, < D, +, 0 >, D_1, m >$ is its frame, and an interval $[t_1, t_2] \in \mathbf{I}\mathrm{ntv}(T)$ such that $< \mathcal{F}, \mathcal{J} >, [t_1, t_2] \models_{il2_\Omega} dc2il(\Gamma_0) \cup dc2il(Axiom_{hdc})$. Hence, there exists a proper sub-interval $[t_1', t_2']$ such that $t_1 < t_1' \leq t_2' < t_2$, $t_1' = t_1 + a$, $t_2' = t_2 - b$, and $< \mathcal{F}, \mathcal{J} >, [t_1', t_2'] \models_{il2_\Omega} dc2il(\Gamma) \cup dc2il(Axiom_{hdc})$.

From now on, we prove that there exists a model $< \mathcal{F}', \mathcal{I} >$ of HDC such that $< \mathcal{F}', \mathcal{I} >, [t_1', t_2'] \models_{hdc} \Gamma$.

Let $\Im$ be a class of interpretations of IL_2 such that for every element $\mathcal{J}' \in \Im$, $< \mathcal{F}, \mathcal{J}' >$ is a model of IL_2 , and $\mathcal{J}'[\![g]\!] = \mathcal{J}[\![g]\!]$

For every $\mathcal{J}' \in \Im$ we construct an interpretation $\mathcal{I}'$ of $\mathcal{L}_{hdc}$ as follows:

For every $V \in PVar$, the formula $\Phi = dc2il(\exists x.\lceil V = x \rceil \vee \lceil\!\rceil) \in dc2il(\Omega_{hdc})$. By Theorem 2, there exists a partition $t_1 = t_1'' \leq t_2'' \leq \ldots \leq t_n'' = t_2$ of $[t_1, t_2]$ such that $< \mathcal{F}, \mathcal{J}' >, [t_i'', t_{i+1}''] \models_{il2_\Omega} dc2il(\exists x.\lceil V = x \rceil \vee \lceil\!\rceil)$, i.e. $< \mathcal{F}, \mathcal{J}' >, [t_i'', t_{i+1}''] \models_{il2_\Omega} \exists x.\lceil h_{id}(g(y_V), x) \rceil \vee \lceil\!\rceil$, for $i = 1, \ldots, n-1$. Thus $\mathcal{I}'$ can be defined as follows:

$$\mathcal{I}'[\![V]\!](t) \mathrel{\widehat{=}} \begin{cases} \mathcal{J}''(x) & \text{if } t_i'' \leq t < t_{i+1}'', \text{and} \\ & < \mathcal{F}, \mathcal{J}'' >, [t_i'', t_{i+1}''] \models_{il2_\Omega} \lceil h_{id}(g(y_V), x) \rceil \vee \lceil\!\rceil \\ & \text{where } \mathcal{J}'' \text{ is } x\text{-equivalent to } \mathcal{J}' \\ 0 & \text{otherwise} \end{cases}$$

$$(\bullet\bullet) \quad \begin{array}{ll} \mathcal{I}'(x) & \mathrel{\widehat{=}} \mathcal{J}'(x) \\ \mathcal{I}'(f_i^n) & \mathrel{\widehat{=}} \mathcal{J}'(f_i^n) \\ \mathcal{I}'(X) & \mathrel{\widehat{=}} \mathcal{J}'(X) \\ \mathcal{I}'(R_i^n) & \mathrel{\widehat{=}} \mathcal{J}'(R_i^n) \end{array}$$

In order to prove the theorem, we need the following two lemmas.

Lemma 2. *Let $\mathcal{J}' \in \Im$, and $\mathcal{J}'$ and $\mathcal{I}'$ have the relation ($\bullet\bullet$). Then for any term θ in $\mathcal{L}_{hdc}$ and any interval $[c, d] \subseteq [t_1', t_2']$, we have:*

$$\mathcal{I}'[\![\theta]\!][c, d] = \mathcal{J}'[\![dc2il(\theta)]\!][c, d].$$

Proof of the lemma: See [12]. □

Now, we can give a correspondence between $\mathcal{I}'$ and $\mathcal{J}'$ which have the relation ($\bullet\bullet$) on formulae by the following lemma.

Lemma 3. *Let $\mathcal{J}' \in \Im$, and $\mathcal{J}'$ and $\mathcal{I}'$ have the relation (●●). Then for any formula ϕ in $\mathcal{L}_{hdc}$, and any subinterval $[c,d] \subset [t'_1, t'_2]$, $< \mathcal{F}, \mathcal{I}' >, [c,d] \models_{hdc} \phi$ iff $< \mathcal{F}, \mathcal{J}' >, [c,d] \models_{il2_\Omega} dc2il(\phi)$.*

Proof of the lemma: We give its proof by induction on the construction of ϕ. We only prove the case $\phi = \exists V.\psi$, the other cases can be proved easily by Lemma 2 and the definition of *dc2il*.

"$\Leftarrow$" It is easy to show.

"$\Rightarrow$" Let $< \mathcal{F}, \mathcal{I}' >, [c,d] \models_{hdc} \phi$. Then there exists an interpretation $\mathcal{I}''$ for HDC which is V-equivalent to $\mathcal{I}'$, and $< \mathcal{F}, \mathcal{I}'' >, [c,d] \models_{hdc} \psi$. Let $t''_0, t''_1, \ldots, t''_n, t''_{n+1} \in T$ such that $t_1 \leq t''_0 < c = t''_1 \leq \ldots \leq t''_n = d < t''_{n+1} \leq t_2$ and $\mathcal{I}''(V)$ is constant on $[t'_i, t'_{i+1})$ for $i = 0, \ldots, n$, and assume these $n+1$ constants are $c_0, \ldots, c_n$. The above assumption is reasonable because $< \mathcal{F}, \mathcal{I}'' >$ is a model of HDC. Since $\mathcal{M} =< \mathcal{F}, \mathcal{J} >$ is a model of IL_2 , by the axiom $\mathrm{Q_V}$ we have that for all $i = 0, \ldots, n$, there exists some $d_i \in D_1$ such that

$$(*)\ \mathcal{J}[\![g]\!](d_i, [b,e]) = c_i \text{ if } \mathcal{I}''[\![V]\!] = c_i$$

for any sub-interval $[b,e] \subseteq [t''_i, t''_{i+1}]$.

Applying the axioms HDC1-HDC3 n times implies that there exists a $d \in D_1$ such that for all $i = 0, \ldots, n+1$ and $t \in [t''_i, t''_{i+1})$

$$\mathcal{I}''(V)(t) = c_i \quad \text{iff} \quad < \mathcal{F}, \mathcal{J}'' >, [t''_i, t''_{i+1} \models_{il2_\Omega} [\![h_{id}(g(d), c_i)]\!]$$

Let $\mathcal{J}''(z) = \mathcal{J}'(z)$ for all symbols in IL_2 but y_V, and $\mathcal{J}''(y_V) = d$ defined by the above. Hence $\mathcal{J}''$ is y_V-equivalent to $\mathcal{J}'$, and $\mathcal{J}''$ and $\mathcal{I}''$ have the definition relation given in (●●). By the induction hypothesis, $\mathcal{J}'', [c,d] \models_{il2_\Omega} dc2il(\psi)$, whence $\mathcal{J}', [c,d] \models_{il2_\Omega} dc2il(\phi)$ by the definition of *dc2il*. □

Now, let $\mathcal{F}' =<< T, \leq >, < D, +, 0 >, m >$. It is easy to show that $< \mathcal{F}', \mathcal{I} >, [t'_1, t'_2] \models_{hdc} \Gamma$ since the interpretations of HDC are independent of D_1. □

Theorem 6 (Completeness). *The proof system of HDC is complete, i.e. $\models_{hdc} \phi$ implies $\vdash_{hdc} \phi$.*

Proof. Suppose $\models_{hdc} \phi$ but $\not\vdash_{hdc} \phi$. So $\{\neg\phi\}$ is consistent with respect to the proof system of HDC. By Theorem 5, there exists a model $< \mathcal{F}, \mathcal{I} >$ and an interval $[t_1, t_2]$ such that $< \mathcal{F}, \mathcal{I} >, [t_1, t_2] \models_{hdc} \neg\phi$. This contradicts $\models_{hdc} \phi$. Hence, $\vdash_{hdc} \phi$. □

5 Discussion

In order to develop a DC-based programming theory, a higher-order duration calculus has been established in [2]. In this paper, we investigate the logic properties of HDC. Especially, we proved that HDC is complete on abstract domains by reducing HDC to a complete first-order two-sorted interval temporal logic.

In the literature of DC, there are two completeness results. One is on abstract domains (see [8]). Unfortunately it requires ω-rule. The other is on real

domain (see [10]), but it is a relative completeness, i.e. it is assumed that all valid formulae of real arithmetic and interval temporal logic are provable in DC. Up to now, no one find a relation between these two completeness results.

If we give another relative completeness of HDC, i.e. if $\models_{hdc} \phi$, then $\Gamma_R \vdash_{hdc} \phi$, where Γ_R stands for all valid real formulae, then we can show that if interval temporal logic is complete on real domain w.r.t. the assumption that all valid real formulae are provable, then completeness of HDC on real domain under the same assumption can be proved with the technique developed in this paper. This conclusion can be applied to other variants of DC too. But how to prove the relative completeness of temporal logic on real domain is still an open problem.

Acknowledgements

The author sincerely give his thanks to his supervisor, Prof. Zhou Chaochen for his instructions and guidance, many inspiring discussions and many suggestions which improved the presentation of this paper. The author is indebted to Dr. Dimitar P. Guelev for his idea to reduce higher-order logic to first-order one.

References

1. M Abadi. The power of temporal proofs. *Theoretical Computer Science*, 1989, 65: 35-83, *Corrigendum in TCS 70 (1990), page 275*
2. Zhou Chaochen, Dimitar P. Guelev and Zhan Naijun. A higher-order duration calculus. UNU/IIST Report No. 167, UNU/IIST, P.O. Box 3058, Macau, July, 1999.
3. Zhou Chaochen, C.A.R. Hoare, and A.P. Ravn. A calculus of durations. *Information Processing Letters*, 1991,40(5):269–276.
4. Zhou Chaochen and Li Xiaoshan. A mean value calculus of durations. In *A Classical Mind: Essays in Honour of C.A.R. Hoare*, Prentice Hall, 1994, 431–451.
5. B. Dutertre. On first order interval temporal logic. Report no. CSD-TR-94-3, Department of Computer Science, Royal Holloway, University of London, Eghan, Surrey TW20 0EX, England, 1995.
6. R.L. Epstein. *The Semantic Foundations of Logic: Predicate Logic*, Oxford University Press, Oxford, UK, 1994.
7. J.W. Garson. Quantification in modal logic. In *Handbook of Philosophical Logic*, D. Gabbay and F. Guenther (Eds), Reidel, 1984, (II):249-307.
8. Dimitar P. Guelev. A calculus of durations on abstract domains: completeness and extensions. UNU/IIST Report No. 139, UNU/IIST, P.O. Box 3058, Macau, May, 1998.
9. Dimitar P. Guelev. Quantification over States in Duration Calculus. August, 1998.
10. M.R. Hansen and Zhou Chaochen. Semantics and completeness of duration calculus. In *Real-Time: Theory in Practice*, Springer-Verlag, 1992,LNCS 600, 209–225.
11. M.R. Hansen and Zhou Chaochen. Duration calculus: logical foundations. *Formal Aspects of Computing*,1997, 9:283-330.
12. Zhan Naijun. Completeness of higher-order duration calculus. UNU/IIST Report No.175, UNU/IIST, P.O. Box 3058, Macau, August, 1999.