Sharpening the Incompleteness of the Duration Calculus [1]

Dimitar P. Guelev

School of Computer Science, The University of Birmingham, UK
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences
E-mail: `gelevdp@math.bas.bg`.

Abstract

We prove that the particularly narrow subset of the duration calculus which is
defined by the BNF

$$\varphi ::= \bot \mid \lceil S \rceil \mid A \mid \varphi \Rightarrow \varphi \mid (\varphi ; \varphi)$$

is not recursively axiomatisable or, in other words, incomplete.

Introduction

The Duration Calculus (DC) was introduced in [ZHR91] as a predicate interval-based linear-time temporal logic for reasoning about real-time systems. DC can be viewed as a theory in the real-time variant of Interval Temporal Logic (ITL, [Mos85,Mos86,CMZ]). A comprehensive survey of DC can be found in [HZ97]. The recent monograph [ZH04] presents a detailed introduction to DC and case studies which outline the scope of the logic as a specification formalism. DC is a conservative extension to first-order predicate logic, and therefore validity in DC is undecidable. Decision procedures are known only for subsets of DC, see e.g. [ZHS93]. For this reason proof systems are relatively important for the use of DC. A finitary Hilbert-style proof system for DC which is complete with respect to its real-time semantics relative to the set of ITL formulas which are valid at the real-time frame of ITL was obtained in [HZ92].

The existence of a finitary complete proof system for a theory implies that the set of the formulas which are valid in this theory can be no worse than semi-decidable, because the set the well-formed proofs in a finitary proof system is decidable and the validity of a formula in the theory is equivalent

---

[1] After the title of [Lod00].

to the existence of a proof for that formula. Yet, validity in DC is not semi-decidable. Hence, it is no surprise that the proof system from [HZ92] is only relatively complete. Validity is no worse than semi-decidable in logics which admit complete recursive axiomatisations that need not be based on finite sets of axioms and rules too. All that is needed for the semi-decidability of validity are algorithms to recognise occurrences of axioms and the correctness of rule applications in proofs. Logics and theories which do not admit a complete recursive axiomatisation are called *incomplete*. The most famous incomplete theory is *Peano arithmetic*. Its incompleteness was established in the famous theorem of Gödel (cf. e.g. [Sho67]).

The forbidding complexity of interval-based temporal logics was shown already in the early work [HS86]. The undecidability of small subsets of DC was established in [ZHS93]. More results can be found in [Frä97]. The complexity of propositional ITL depends heavily on the *locality principle*, which restricts atomic formulas to depend only on the initial point of the reference interval for their truth values. Propositional ITL with the locality principle is decidable. A very narrow undecidable subset of propositional ITL was found in [Lod00]. In this paper we present a particularly small subset of DC which is still big enough to manifest the incompleteness of DC. This subset includes duration terms occuring only in the restricted form $\lceil S \rceil$, only 0-ary flexible predicate symbols, also known as propositional temporal letters and, hence, no properly first-order constructs.

Structure of the paper
After brief preliminaries we introduce the subset in question and prove that validity in this subset is not semi-decidable.

1 Preliminaries

1.1 The definition of DC

DC is a classical first-order modal logic with one normal binary modality called *chop*. We denote chop by $(.;.)$. The possible worlds in DC semantics are closed and bounded intervals of real numbers. For this reason DC is also an *interval-based real-time* temporal logic. Here follows a formal definition of DC.

1.1.1 Languages
Along with the customary first-order logic symbols, DC vocabularies include *state variables* P, Q, ... State variables are used to build *state expressions* S, which have the syntax:

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

State expressions S occur in formulas as part of *duration terms* $\int S$. The syntax of DC *terms* t and *formulas* φ extends that of first-order logic by

$$\min \sigma \qquad\qquad \sigma \qquad\qquad \max \sigma$$

1

0

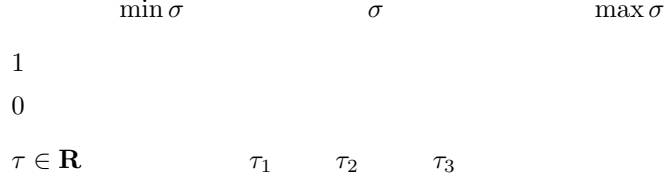$$\tau \in \mathbf{R} \qquad\qquad \tau_1 \quad \tau_2 \quad \tau_3$$

Fig. 1. The graph of a predicate with the finite variability property. For the shown interval σ, the set $\{\tau \in \sigma : p(\tau) = 0\}$ from Definition 1.1 is $[\tau_1, \tau_2] \cup [\tau_3, \max \sigma]$.

duration terms and formulas built using the modality $(.;.)$, respectively:

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

Here x, c, f, and R denote an individual variable, a constant, a function and a relation symbol, respectively. Constant, function and relation symbols can be either *rigid* or *flexible* in DC. The interpretations of rigid symbols are required not to depend on the reference interval. Individual variables are rigid. State variables are flexible. We denote the *arity* of non-logical symbol s by $\#s$. Flexible relation symbols of arity 0 and flexible constant symbols are also called *temporal propositional letters* and *temporal variables*, respectively. The rigid constant 0, the rigid binary function symbol $+$, the rigid binary relation symbols $=$ and $\leq$, and an infinite set of individual variables are mandatory in DC vocabularies.

1.1.2 Semantics
The model of time in DC is the linearly ordered group of the real numbers $\langle \mathbf{R}, 0_\mathbf{R}, +_\mathbf{R}, \leq_\mathbf{R} \rangle$. Let $\mathbf{I}$ denote the set

$$\{[\tau_1, \tau_2] : \tau_1, \tau_2 \in \mathbf{R}, \tau_1 \leq \tau_2\}.$$

Definition 1.1 A predicate $p : \mathbf{R} \to \{0, 1\}$ has the *finite variability property* if the set

$$\{\tau \in \sigma : p(\tau) = 0\}$$

is either empty, or a finite union of intervals for all $\sigma \in \mathbf{I}$.

The finite variability property is illustrated on Figure 1. It reflects the natural assumption that $\{0, 1\}$-valued signals, which appear in systems modelled by DC, change their values only finitely many times in any given bounded interval of time. The impossibility to axiomatise DC completely by finitary means can be ascribed to the requirement on the interpretations of state variables to have this property. This can be seen by comparing abstract time *ITL* [Dut95], where finite variability is not present, and abstract time DC [Gue98], where it is. The former system admits complete finitary axiomatisation and the latter does not.

Definition 1.2 An *interpretation I of a DC language* $\mathbf{L}$ is a function on the

3

vocabulary of **L**. The types of the values of I for symbols of the various kinds are as follows:

$I(x), I(c) \in \mathbf{R}$	for individual variables x and rigid constants c
$I(c) : \mathbf{I} \to \mathbf{R}$	for flexible constants c
$I(f) : \mathbf{R}^{\#f} \to \mathbf{R}, I(R) : \mathbf{R}^{\#R} \to \{0,1\}$	for rigid function symbols f and relation symbols R
$I(f) : \mathbf{I} \times \mathbf{R}^{\#f} \to \mathbf{R}, I(R) : \mathbf{I} \times \mathbf{R}^{\#R} \to \{0,1\}$	for flexible f, R
$I(P) : \mathbf{R} \to \{0,1\}$	for state variables P

$I(0)$, $I(+)$ and $I(\leq)$ are always the corresponding components of $\langle \mathbf{R}, 0_{\mathbf{R}}, +_{\mathbf{R}}, \leq_{\mathbf{R}} \rangle$, and $I(=)$ is equality on $\mathbf{R}$. Interpretations of state variables are required to have the finite variability property.

Definition 1.3 Given an interpretation I, the value $I_\tau(S)$ of state expression S at time $\tau \in \mathbf{R}$ is defined by the clauses:

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

The value $I_\sigma(t)$ of a term t at interval $\sigma \in \mathbf{I}$ is defined by the clauses:

$$I_\sigma(x) = I(x)$$
$$I_\sigma(c) = I(c) \text{ for rigid } c$$
$$I_\sigma(c) = I(c)(\sigma) \text{ for flexible } c$$
$$I_\sigma(\textstyle\int S) = \int_{\min \sigma}^{\max \sigma} I_\tau(S) d\tau$$
$$I_\sigma(f(t_1, \ldots, t_{\#f})) = I(f)(I_\sigma(t_1), \ldots, I_\sigma(t_{\#f})) \text{ for rigid } f$$
$$I_\sigma(f(t_1, \ldots, t_{\#f})) = I(f)(\sigma, I_\sigma(t_1), \ldots, I_\sigma(t_{\#f})) \text{ for flexible } f$$

Given interpretations I and J of the same language $\mathbf{L}$ and a symbol s from its vocabulary, J is called s-*variant* of I, if $J(s') = I(s')$ for all s' from $\mathbf{L}$, except possibly s.

The modelling relation $\models$ is defined on interpretations I of $\mathbf{L}$, intervals $\sigma \in \mathbf{I}$ and formulas φ from $\mathbf{L}$ by the clauses:

$$I, \sigma \not\models \bot$$
$$I, \sigma \models R(t_1, \ldots, t_{\#R}) \quad \text{iff } I(R)(I_\sigma(t_1), \ldots, I_\sigma(t_{\#R})) = 1 \text{ for rigid } R$$
$$I, \sigma \models R(t_1, \ldots, t_{\#R}) \quad \text{iff } I(R)(\sigma, I_\sigma(t_1), \ldots, I_\sigma(t_{\#R})) = 1 \text{ for flexible } R$$
$$I, \sigma \models \varphi \Rightarrow \psi \quad \text{iff either } I, \sigma \models \psi \text{ or } I, \sigma \not\models \varphi$$

4

$$I, \sigma \models (\varphi; \psi) \quad \text{iff } I, \sigma_1 \models \varphi \text{ and } I, \sigma_2 \models \psi \text{ for some } \sigma_1, \sigma_2 \in \mathbf{I}$$
$$\text{such that } \sigma = \sigma_1 \cup \sigma_2 \text{ and } \max \sigma_1 = \min \sigma_2$$
$$I, \sigma \models \exists x \varphi \quad \text{iff } J, \sigma \models \varphi \text{ for some } J \text{ which is a } x\text{-variant of } I$$

1.1.3 Abbreviations and precedence of the operators

The symbols $\top$, $\neg$, $\vee$, $\wedge$, $\Leftrightarrow$, $\forall$ and $\neq$ are used as abbreviations in the usual way in formulas. Infix notation is used wherever $+$, $=$ and $\leq$ occur. The connectives $\neg$, $\vee$, $\wedge$ and $\Leftrightarrow$ are used as abbreviations in state expressions too. The following abbreviations are specific to DC:

$$\Diamond \varphi \rightleftharpoons ((\top; \varphi); \top) , \qquad \Box \varphi \rightleftharpoons \neg \Diamond \neg \varphi ,$$
$$(\varphi_1; \varphi_2; \ldots; \varphi_n) \rightleftharpoons (\varphi_1; \ldots; (\varphi_{n-1}; \varphi_n) \ldots) ,$$
$$\mathbf{1} \rightleftharpoons \mathbf{0} \Rightarrow \mathbf{0} , \qquad \ell \rightleftharpoons \int \mathbf{1} , \qquad \lceil S \rceil \rightleftharpoons \ell \neq 0 \wedge \int S = \ell .$$

The usual precedence conventions are assumed about the propositional connectives, $\exists$ and $\forall$. We always write parentheses when using the chop modality $(.; .)$, thus assigning it the lowest precedence. For example, $(A \wedge B; C \Leftrightarrow D)$ is the same as $((A \wedge B); (C \Leftrightarrow D))$, and $A \wedge B; C \Leftrightarrow D$ is not well-formed.

1.2 Post's theorem

Our proof relies on the famous theorem about decidability by Emil Post. We use the theorem in the following form:

Theorem 1.4 *Let X be a decidable set of words in some finite alphabet Σ. Let $Y \subset X$ and both Y and $X \setminus Y$ be semi-decidable. Then both Y and $X \setminus Y$ are decidable.*

A more general formulation and a proof of Post's theorem can be found e.g. in [Sho67].

2 The incomplete subset

The subset of DC that we focus on in this paper is defined by the BNFs

$$(1) \quad \begin{aligned} &\varphi ::= \bot \mid \lceil S \rceil \mid A \mid \varphi \Rightarrow \varphi \mid (\varphi; \varphi) \\ &S ::= \mathbf{0} \mid P \mid \neg S \end{aligned}$$

where P stands for a state variable and A stands for a propositional temporal letter. In the next section we prove that validity in this subset is not semi-decidable.

3 The incompleteness proof

Many undecidability proofs in the literature are based on reducing the *halting problem* for Turing machines to the validity problem for the considered logic or theory. This way one can show that the set of the valid formulas in the considered system is no simpler than semi-decidable. In the proof to follow we reduce the halting problem to the *satisfiability* problem in the considered subset of DC. This approach can be seen as similar to the one taken in [HS86], where the *non*halting problem for Turing machines is reduced to a validity problem. The reduction shows that the satisfiable formulas from a certain decidable set of formulas X form a set Y which is only semi-decidable. The formulas with valid negations from X form the complement $X \setminus Y$ of the undecidable set Y relative to the decidable set X. Post's theorem implies that $X \setminus Y$ cannot be semi-decidable. Hence our reduction shows that validity in the considered set X of formulas cannot be axiomatised.

The proof consists of two parts. In the first part we generalise finite variability as known for predicates on $\mathbf{R}$ (Definition 1.1) to predicates on $\mathbf{I}$. Interpretations of temporal propositional letters are predicates on $\mathbf{I}$. We show that the finite variability of the interpretation of a temporal propositional letter can in a certain sense be expressed by a formula in the subset (1) of DC.

The second part of the proof uses the formulas which express the generalised form of finite variability in a reduction of the halting problem for appropriately chosen Turing machines as described above.

3.1 Finite variability of predicates on intervals

Definition 3.1 A predicate a on $\mathbf{I}$ has the *finite variability property*, if for every $\sigma \in \mathbf{I}$ there exists a finite $V_{a,\sigma} \subseteq \sigma$ such that if $\sigma_1, \sigma_2 \in \mathbf{I}$, $\sigma_1, \sigma_2 \subseteq \sigma$ and $a(\sigma_1) \neq a(\sigma_2)$, then there is a member of $V_{a,\sigma}$ which is either one of the end points of σ_1 and σ_2, or is between $\min \sigma_1$ and $\min \sigma_2$, or is between $\max \sigma_1$ and $\max \sigma_2$.

Figure 2 illustrates this definition. Finite variability of predicates on $\mathbf{I}$ is a special case of the generalisation of finite variability from DC states to DC flexible symbols' interpretations proposed in [Gue00].

Given a propositional temporal letter A and a state variable P, we put:

$$\mathsf{fv}(A, P) \rightleftharpoons \bigwedge_{\substack{Q_1, Q_2 \in \{P, \neg P\} \\ B \in \{A, \neg A\}}} \Box \neg \left(\begin{array}{l} (\lceil Q_1 \rceil; B \wedge (\lceil Q_1 \rceil; \top; \lceil Q_2 \rceil); \lceil Q_2 \rceil) \wedge \\ (\lceil Q_1 \rceil; \neg B \wedge (\lceil Q_1 \rceil; \top; \lceil Q_2 \rceil); \lceil Q_2 \rceil) \end{array} \right) .$$

Proposition 3.2 *Let I denote an interpretation of some DC language which includes A and P. Let $\sigma \in \mathbf{I}$. Then:*

(i) If $I(A)$ has the finite variability property, then $I(P)$ can be defined so that

(2)　$I, \sigma \models \mathsf{fv}(A, P)$.

$$\min \sigma \qquad\qquad \tau \qquad\qquad\qquad\qquad \max \sigma$$

Fig. 2. For the singleton $\{\tau\}$ to be a suitable candidate for $V_{a,\sigma}$ for some $a : \mathbf{I} \to \{0, 1\}$ with the finite variability property accoding to Definition 3.1, the values of a at the subintervals of σ marked with lines of the same pattern must be the same. The values of a at subintervals marked differently can be different. Note that two of these intervals are grouped together because they both have τ as their beginning point.

(ii) Conversely, if (2) holds for some interpretation I, then the restriction $I(A)|_{\{\sigma' \in \mathbf{I}: \sigma' \subseteq \sigma\}}$ of $I(A)$ to the subintervals of σ in $\mathbf{I}$ can be extended to a predicate on $\mathbf{I}$ with the finite variability property.

Proof. (i) Let $V_{I(A),\sigma}$ satisfy the requirement from Definition 3.1. Let

$$\{\tau_0, \ldots, \tau_n\} = V_{I(A),\sigma} \cup \{\min \sigma, \max \sigma\} \text{ and } \tau_0 < \ldots < \tau_n.$$

Let $I, [\tau_{2i-1}, \tau_{2i}] \models \lceil \neg P \rceil$ for i such that $0 < 2i \leq n$ and $I, [\tau_{2i}, \tau_{2i+1}] \models \lceil P \rceil$ for i such that $0 < 2i+1 \leq n$. Then (2) holds iff for any two $i, j \in \{1, \ldots, n\}$, $\tau'_1, \tau''_1 \in (\tau_{i-1}, \tau_i)$ and $\tau'_2, \tau''_2 \in (\tau_{j-1}, \tau_j)$ such that $\tau'_1 \leq \tau'_2$ and $\tau''_1 \leq \tau''_2$ we have

$$(3) \quad I(A)([\tau'_1, \tau'_2]) = I(A)([\tau''_1, \tau''_2]).$$

Hence, the finite variability of $I(A)$ implies (2).

(ii) Let (2) hold. We say that $I(P)$ *changes its value at $\tau \in \mathbf{R}$* if there are $\tau' \in (-\infty, \tau)$ and $\tau'' \in (\tau, \infty)$ such that

$$I, [\tau', \tau] \models \lceil Q \rceil \text{ and } I, [\tau, \tau''] \models \lceil \neg Q \rceil$$

where Q is either P or $\neg P$. Let the set V consist of $\min \sigma$, $\max \sigma$ and the time points in σ at which $I(P)$ changes its value. The finite variability of $I(P)$ implies that V is finite. Let $\{\xi_0, \ldots, \xi_m\} = V$ and $\xi_0 < \ldots < \xi_m$. If

$$\tau'_1, \tau''_1 \in (\xi_{i-1}, \xi_i) \text{ and } \tau'_2, \tau''_2 \in (\xi_{j-1}, \xi_j) \text{ for some } i, j \in \{1, \ldots, m\},$$

and $\tau'_1 \leq \tau'_2$ and $\tau''_1 \leq \tau''_2$, then (2) implies (3). Hence V satisfies the requirements on $V_{I(A),\sigma}$ from Definition 3.1. Let $a : \mathbf{I} \to \{0, 1\}$, $a(\sigma') = I(A)(\sigma')$ for subintervals σ' of σ from $\mathbf{I}$, and $a(\sigma') = 0$ for all other $\sigma' \in \mathbf{I}$. Then V satisfies the requirements on $V_{a,\sigma'}$ from Definition 3.1 for all $\sigma' \in \mathbf{I}$. $\qquad\square$

Proposition 3.2 entails that $I(A)|_{\{\sigma' \in \mathbf{I}: \sigma' \subseteq \sigma\}}$ can be extended to a predicate on $\mathbf{I}$ with the finite variability property if and only if

$$I, \sigma \models \exists P \mathsf{fv}(A, P),$$

where $\exists P$ has the usual meaning. This quantifier was added to DC in [Pan95] and further investigated in [He99,ZGZ00] for practical purposes. $\exists P \mathsf{fv}(A, P)$ is clearly out of the subset (1) of DC and, indeed, out of the system of DC

7

as introduced in Subsection 1.1, but $\mathsf{fv}(A, P)$ is in the subset (1). However, if P does not occur in a formula η, then the validity of $\exists P \mathsf{fv}(A, P) \Rightarrow \eta$ is equivalent to the validity of $\mathsf{fv}(A, P) \Rightarrow \eta$, which does not contain $\exists P$. This lets us avoid using $\exists P$ in the sequel.

3.2 Encoding terminating computations of Turing machines by satisfiable DC formulas

Let T be a deterministic Turing machine with tape alphabet Σ and set of control states Q. Let the letters from Σ and the states from Q be temporal propositional letters in the vocabulary of the language $\mathbf{L}$ for DC. We assume that $Q \cap \Sigma = \emptyset$. Let $\mathbf{L}$ have two more propositional temporal letters $b, r \notin Q \cup \Sigma$ and a state variable P. Then configurations of T can be encoded as formulas in $\mathbf{L}$ written using only propositional temporal letters from the set $Q \cup \Sigma \cup \{b\}$ and $(.;.)$ as follows. Let the word on the tape which is on the left of the head of T in some configuration be $\alpha = A_1 \dots A_m$. Let the word on the right of the head be $\beta = B_1 \dots B_n$. Let the letter being observed by the head be C and the current control state of T be q. Then the configuration α, q, C, β of T can be represented by the formula

$$\varphi_{\alpha,q,C,\beta} \rightleftharpoons (b; A_1; \dots; A_m; q; C; B_1; \dots; B_n; b)$$

The occurrences of b in $\varphi_{\alpha,q,C,\beta}$ serve to mark the ends of the part of the (potentially infinite) tape observed and possibly overwritten during the computation of T so far.

Using this convention, the transition rules of T can be encoded as formulas in the subset (1) of $\mathbf{L}$ too. Below we propose a way to do this. It involves the auxiliary propositional temporal letter r. Consider interpretations I of $\mathbf{L}$ in which the following formulas about r are valid:

(4) $\quad \neg \lceil \mathbf{1} \rceil \Rightarrow r$

(5) $\quad \bigvee_{D \in Q \cup \Sigma \cup \{b\}} (D; r) \Rightarrow r$

Under such interpretations I, if $\tau_0, \dots, \tau_n \in \mathbf{R}$, $\tau_0 < \dots < \tau_n$ and for each $i = 1, \dots, n$ there is a $D \in Q \cup \Sigma \cup \{b\}$ such that $I, [\tau_{i-1}, \tau_i] \models D$, then $I, [\tau_0, \tau_n] \models r$. In words, r is satisfied at all intervals from $\mathbf{I}$ which can be chopped into finitely many subintervals each of which satisfies some temporal propositional letter from $D \in Q \cup \Sigma \cup \{b\}$. For an interpretation I to validate the formulas (4) and (5), it is sufficient to satisfy r *only* at intervals which either can be partitioned this way or have length 0. Note that (4) and (5) are formulas in the subset (1) of $\mathbf{L}$. We denote the conjunction of the formulas (4) and (5) by Ax_r.

The transition rules of T prescribe T to overwrite the letter observed by its head and move to the left or to the right, depending on which the observed letter was. Consider a transition rule for T stating that observing C at control state q' should cause T to replace C by D, change its control state to q'' and

move left. Let Z denote the letter from Σ which occupies all the positions on the tape of T at which nothing else has been written yet. Such a rule can be described by the set of implications

$$(r; A; q'; C; r) \Rightarrow (r; q''; A; D; r) \text{ for each } A \in \Sigma,$$

and

(6) $\quad (r; b; q'; C; r) \Rightarrow (r; b; q''; Z; D; r)$

The first group of implications describe the transitions prescribed by the rule when the head of T is properly within the part of the tape which has been written on at earlier steps of the computation. The last implication applies to the case in which moving left takes the head of T to a tape position which has not been used so far, and therefore T discovers a Z there. Rules which cause the head of T to move right can be described by similar formulas:

$$(r; q'; C; A; r) \Rightarrow (r; D; q''; A; r) \text{ for each } A \in \Sigma,$$

and

(7) $\quad (r; q'; C; b; r) \Rightarrow (r; D; q''; Z; b; r)$

If $\gamma \Rightarrow \delta$ is one of the implications from (6) and (7) and γ describes the situation around the head of T at configuration α', q', C', β' then T can move from α', q', C', β' to a configuration $\alpha'', q'', C'', \beta''$ in which the neighbourhood of its head is as described by δ. This is reflected by the validity of the formula

(8) $\quad \varphi_{\alpha',q',C',\beta'} \wedge \Box Ax_r \wedge (\gamma \Rightarrow \delta) \Rightarrow \varphi_{\alpha'',q'',C'',\beta''}$

Since we only consider deterministic T, for every configuration α', q', C', β' there is at most one implication $\gamma \Rightarrow \delta$ from among (6) and (7) such that (8) describes the next move of T.

Obviously the transition rules of a fixed machine T can be described by a finite set of implications of the forms (6) and (7). Let Ax_T denote the conjunction of all these implications. Obviously if, starting from the configuration described by $\varphi_{\alpha,q,C,\beta}$, the machine T can reach the configuration described by $\varphi_{\alpha',q',C',\beta'}$, then

(9) $\quad \varphi_{\alpha,q,C,\beta} \wedge \Box Ax_r \wedge Ax_T \Rightarrow \varphi_{\alpha',q',C',\beta'}$

is a valid formula. If the initial configuration α, q, C, β causes T to generate an infinite set of configurations and

(10) $I, \sigma \models \varphi_{\alpha,q,C,\beta} \wedge \Box Ax_r \wedge Ax_T \wedge \Box \underbrace{\bigwedge_{A \in Q \cup \Sigma \cup \{b\}} A \Rightarrow (\lceil \mathbf{1} \rceil; \neg A \wedge \lceil \mathbf{1} \rceil; \lceil \mathbf{1} \rceil)}_{\xi_{Q \cup \Sigma \cup \{b\}}},$

then the predicates $I(A)$ cannot have the finite variability property for all $A \in \Sigma$. To realise this, note that if (10) holds, then the validity of (9) implies $I, \sigma \models \varphi_{\alpha',q',C',\beta'}$ for all the infinitely many configurations α', q', C', β' generated by T starting from α, q, C, β, among which are configurations with arbitrarily

9

big lengths of α' and β'. Furthermore, the formula $\xi_{Q\cup\Sigma\cup\{b\}}$ in (10) forces the propositional temporal letters occurring in $\varphi_{\alpha',q',C',\beta'}$ to be satisfied only at subintervals of σ of non-zero lengths which, on their turn, are supposed to contain subintervals of non-zero lengths that do not satisfy the respective propositional temporal letters. Let us denote the formula on the right of $\models$ in (10) by $\eta_{\alpha,q,C,\beta}$. Then Proposition 3.2 implies that for α, q, C, β which cause T to generate an infinite set of configurations

$$(11) \qquad \bigwedge_{A\in Q\cup\Sigma\cup\{b\}} \mathsf{fv}(A, P) \Rightarrow \neg\eta_{\alpha,q,C,\beta}$$

is a valid formula.

On the other hand, if α, q, C, β causes T to terminate in finitely many steps, then I and σ can be chosen to satisfy (10) so that $I(A)$ have the finite variability property for all $A \in \Sigma$, and therefore (11) is not valid for such α, q, C, β. We prove this in Proposition 3.3 below. This implies that (11) is valid if and only if α, q, C, β causes T to generate an infinite set of configurations.

Note that (11) is in the subset (1) of **L**. We use the same state variable P to witness the finite variability of all the propositional temporal letters $A \in Q \cup \Sigma \cup \{b\}$ in (11). We can do this, because these letters are finitely many, and therefore an interpretation for P can be chosen to change its value often enough for all of them.

The Turing machine T can be chosen so that for each initial configuration it either terminates or generates infinitely many *different* configurations. Furthermore, T can be chosen so that the set of the initial configurations which cause it to eventually terminate is semi-decidable, but not decidable, and consequently, by Theorem 1.4, the set of the initial configurations which cause it to run without terminating and generate infinitely many configurations is not semi-decidable. If T is chosen this way, then the set of the formulas of the form (11) which are valid is not semi-decidable. Since all these formulas are in the subset (1) of **L** and it is decidable whether a formula in the subset (1) has the form (11) or not, validity in the entire subset (1) of **L** is not semi-decidable, which means that the subset (11) is incomplete. This is the main result of this paper. To finish its proof we only need to prove Proposition 3.3, as promised above.

Proposition 3.3 *Let the initial configuration α, q, C, β of some deterministic Turing machine T cause it to terminate in finitely many steps. Then the formula $\eta_{\alpha,q,C,\beta}$ from (10) is satisfiable by an interpetation I such that the predicates $I(A)$ have the finite variability property for all $A \in \Sigma$.*

Proof. Let α, q, C, β cause T to go through the sequence of configurations

$$\alpha_k, q_k, C_k, \beta_k, \ k = 0, \ldots, N,$$

where $\alpha_0, q_0, C_0, \beta_0$ is α, q, C, β and q_N is the terminating control state of T. To simplify our notation, assume that $D_{k,1}, \ldots, D_{k,n_k}$ stand for some temporal propositional letters from $Q\cup\Sigma\cup\{b\}$ such that $\varphi_{\alpha_k,q_k,C_k,\beta_k}$ is $(D_{k,1}; \ldots; D_{k,n_k})$,

$k = 0, \ldots, N$.

We choose σ to be $[0, 1]$. We choose some $\tau_{k,0}, \ldots, \tau_{k,n_k} \in [0, 1]$, $k = 0, \ldots, N$, such that

(12) $0 = \tau_{k,0} < \ldots < \tau_{k,n_k} = 1$ for all $k = 0, \ldots, N$,

and

(13) $\{\tau_{k',1}, \ldots, \tau_{k',n_{k'}-1}\} \cap \{\tau_{k'',1}, \ldots, \tau_{k'',n_{k''}-1}\} = \emptyset$, if $0 \leq k' < k'' \leq N$.

Let $D \in Q \cup \Sigma \cup \{b\}$. Then we put $I(D)(\sigma') = 1$ iff D is $D_{k,i}$ and σ' is $[\tau_{k,i-1}, \tau_{k,i}]$ for some $k \in \{0, \ldots, N\}$ and some $i \in \{1, \ldots, n_k\}$. We put $I(r)(\sigma') = 1$ iff either $\min \sigma' = \max \sigma'$, or σ' is $[\tau_{k,i}, \tau_{k,j}]$ for some $k \in \{0, \ldots, N\}$ and some $i, j \in \{1, \ldots, n_k\}$ such that $i < j$.

A direct check shows that $I, \sigma \models \varphi_{\alpha_k, q_k, C_k, \beta_k}$, $k = 0, \ldots, N$, and, in particular, $I, \sigma \models \varphi_{\alpha, q, C, \beta}$.

The definition of $I(r)$ implies that I validates the formulas (4) and (5). Hence $I, \sigma \models Ax_r$.

The time points $\tau_{k',1}, \ldots, \tau_{k',n_{k'}-1}$ and $\tau_{k'',1}, \ldots, \tau_{k'',n_{k''}-1}$ were chosen to be pairwise distinct for $k' \neq k''$ (13). This implies that the only implications $\gamma \Rightarrow \delta$ from R_T for which $I, \sigma \models \gamma$ are the ones which describe transitions between successive configurations $\alpha_{k-1}, q_{k-1}, C_{k-1}, \beta_{k-1}$ and $\alpha_k, q_k, C_k, \beta_k$, $k = 1, \ldots, N$. Since $I, \sigma \models \varphi_{\alpha_k, q_k, C_k, \beta_k}$ for all $k = 0, \ldots, N$, $I, \sigma \models \delta$ for these implications too. Hence $I, \sigma \models Ax_T$.

Finally, I satisfies propositional temporal letters from $Q \cup \Sigma \cup \{b\}$ only at finitely many intervals, and these intervals are of non-zero length. Hence each of these intervals on its turn has subintervals of non-zero length which satisfy no propositional temporal letter from $Q \cup \Sigma \cup \{b\}$. This means that $I, \sigma \models \xi_{Q \cup \Sigma \cup \{b\}}$ too. $\qquad\square$

Concluding remarks

It is worth noting that validity in the subset of *ITL* which is defined by the BNF

(14) $\varphi ::= \bot \mid A \mid \varphi \Rightarrow \varphi \mid (\varphi; \varphi)$

and can be obtained from (1) by just excluding the formulas $\lceil S \rceil$ is semi-decidable, and validity in the subset of *DC* which can be defined by

(15) $\varphi ::= \bot \mid \lceil S \rceil \mid \varphi \Rightarrow \varphi \mid (\varphi; \varphi)$

and can be obtained from (1) by just excluding temporal propositional letters A is decidable. Both the addition of formulas of the form $\lceil S \rceil$ to (14) and the addition of propositional temporal letters to (15) can be regarded as minimal increases of expressivity. Our result applies to real-time *DC*, as known from [ZHR91]. The proof relies on the fact that real time is *dense*, that is

$$(\forall \tau_1, \tau_2 \in \mathbf{R})(\tau_1 < \tau_2 \Rightarrow (\exists \tau_3 \in \mathbf{R})(\tau_1 < \tau_3 \wedge \tau_3 < \tau_2)).$$

DC has been studied for discrete time too. Results on the correspondence between validity in DC for real time and discrete time can be found in [ChP03], but the subsets of DC studied there do not include propositional temporal letters. We do not know whether the subset (1) is axiomatisable with respect to discrete time or not.

Acknowledgements

References

[ChP03] CHAKRAVORTY, G. AND P. K. PANDYA. Digitizing Interval Duration Logic. *Proceedings of CAV 2003*, LNCS 2725, Springer, 2003, pp. 167-179.

[CMZ] CAU, A., B. MOSZKOWSKI AND H. ZEDAN. *Interval Temporal Logic.* URL: `http://www.cms.dmu.ac.uk/~cau/itlhomepage/itlhomepage.html` .

[Dut95] DUTERTRE, B. On First Order Interval Temporal Logic. *Proceedings of LICS'95*, IEEE Computer Society Press, 1995, pp. 36-43.

[Frä97] FRÄNZLE, M. *Controller Design from Temporal Logic: Undecidability Need Not Matter.* Ph.D. Thesis, Technical Faculty of Kiel University, Germany, 1997. Accessed from `http://www.imm.dtu.dk/~mf/diss.ps.gz`.

[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.

[Gue00] GUELEV, D. P. A Complete Fragment of Higher-Order Duration μ-Calculus. *Proceedings of FST TCS 2000.* LNCS 1974, Springer, 2000, pp. 264-276.

[He99] HE JIFENG. Integrating Variants of DC. Research Report 172, UNU/IIST, P.O.Box 3058, Macau, August 1999.

[HS86] HALPERN, J. Y. AND Y. SHOHAM. A Propositional Logic of Time Intervals. *Proceedings of LICS'86*, IEEE Computer Society Press, 1986, pp. 279-292.

[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.

12

[Lod00] Lodaya, K. Sharpening the Undecidability of Interval Temporal Logic. *Proceedings of ASIAN 2000.* LNCS 1961, Springer, 2000, pp. 290-298.

[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.

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

[Sho67] Shoenfield, J. *Mathematical Logic.* Addison-Wesley, 1967.

[ZGZ00] Zhou Chaochen, D. P. Guelev and Zhan Naijun. A Higher-order Duration Calculus. *Millenial Perspectives in Computer Science*, Palgrave, 2000, pp. 407-416.

[ZH04] Zhou Chaochen and M. R. Hansen. *Duration Calculus. A Formal Approach to Real-Time Systems*, Springer, 2004.

[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.