



ELSEVIER

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

ScienceDirect

Electronic Notes in  
Theoretical Computer  
Science

Electronic Notes in Theoretical Computer Science 174 (2007) 113–133

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

# Decidability of a Hybrid Duration Calculus

Thomas Bolander<sup>1</sup>

*Dep. Informatics and Mathematical Modelling  
Technical University of Denmark*

Jens Ulrik Hansen<sup>2</sup>

*Dep. Mathematics  
University of Copenhagen, Denmark*

Michael R. Hansen<sup>3</sup>

*Dep. Informatics and Mathematical Modelling  
Technical University of Denmark*

---

## Abstract

We present a logic which we call Hybrid Duration Calculus (HDC). HDC is obtained by adding the following hybrid logical machinery to the Restricted Duration Calculus (RDC): nominals, satisfaction operators, down-arrow binder, and the global modality. RDC is known to be decidable, and in this paper we show that decidability is retained when adding the hybrid logical machinery. Decidability of HDC is shown by reducing the satisfiability problem to satisfiability of Monadic Second-Order Theory of Order. We illustrate the increased expressive power obtained in hybridizing RDC by showing that HDC, in contrast to RDC, can express all of the 13 possible relations between intervals.

*Keywords:* Duration calculus, hybrid logic, decision methods, monadic second order theory of order.

---

## 1 Introduction

In this paper, we consider a hybrid propositional interval logic and a decision procedure for it.

Propositional interval logics are rather expressible, and often undecidable, and they appear in various forms in different scientific branches, e.g. computer science,

---

<sup>1</sup> Email: [tb@imm.dtu.dk](mailto:tb@imm.dtu.dk). The work is partially funded by the Danish Natural Science Research Council in connection with the HyLoMOL project.

<sup>2</sup> Email: [m00juh@math.ku.dk](mailto:m00juh@math.ku.dk)

<sup>3</sup> Email: [mrh@imm.dtu.dk](mailto:mrh@imm.dtu.dk). This work is partially funded by The Danish Council for Strategic Research under project MoDES.

artificial intelligence, philosophy, and natural language understanding. We refer to [14] for a road map over interval logics.

As an illustration of the expressiveness, the propositional interval logic HS [16] with unary modalities *begins*, *ends* and their inverses has been shown highly undecidable for a collection of classes of interval models. Aiming at decidability results either the formulas or the models or both should be restricted, and in the work on Propositional Neighborhood Logic [4], for example, modalities for left and right neighborhoods only are considered, and a decidability result is shown for a subset with one of the neighborhood modalities only.

A starting point of our work is Duration Calculus (abbreviated DC), which is an interval logic introduced by Zhou, Hoare and Ravn [28] as an extension of Interval Temporal Logic (ITL) by Halpern, Moszkowski and Manna [15]. The introduction of DC is motivated by case studies of real-time systems, e.g. [24], and DC can be used to reason about models of such systems at a high level of abstraction, see e.g. [26]. A distinctive feature of DC is that one can express and reason about the *duration* of a state in a system, where the duration of a state in an interval is the accumulated presence time of the state in that interval. Unfortunately, even simple propositional subsets of DC formulas are undecidable [27]. Decidable subsets of DC are considered for example in [27,11,12,13,17,20].

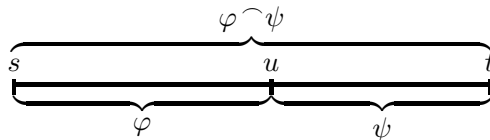
In DC, the underlying time domain is continuous, modelled by the set of non-negative real numbers  $\mathbb{R}_{\geq 0}$ . The atomic formulas concern *state variables*, which are functions of the form:

$$P : \mathbb{R}_{\geq 0} \rightarrow \{0, 1\}.$$

The intuition is that  $P(t) = 1$  if and only if, the system is in state  $P$  at time  $t$ .

In this paper, we consider the decidable subset RDC from [27], having only atomic formulas of the form  $\llbracket S \rrbracket$ , where  $S$  is a Boolean combination of state variables modelling a combined state of the system. The formula holds on an non-point interval if  $S(t) = 1$  through the interval, except for isolated points. Hence in DC, a rather restricted form of model classes is considered, and RDC has a rather restricted collection of atomic formulas. For example, if  $\llbracket S \rrbracket$  holds on an interval, then it holds on all non-point sub-intervals as well.

Furthermore, DC is based on the chop modality of ITL only. The formula  $\varphi \frown \psi$  holds for an interval  $[s, t]$  iff, there is a point  $u \in [s, t]$ , such that  $\varphi$  holds on  $[s, u]$  and  $\psi$  holds on  $[u, t]$ , as illustrated in the following figure:



These restrictions on models and formulas suffice to guarantee decidability, and in [27], this is shown by reducing the satisfiability problem to the emptiness problem for regular languages. The decidability results are obtained for both a continuous-time domain and a discrete-time domain. Unfortunately, seemingly insignificant

extensions of RDC yield undecidable subsets, which is also shown in [27].

The main idea of the present work is to consider a hybrid logical version of RDC achieving more expressiveness while preserving decidability. In particular, we add nominals to RDC, where a nominal in our case names a specific interval and we add a down-arrow binder, which provides a reference to the current interval in a formula. Furthermore, we add the global modality which makes any interval reachable from any other interval. In this paper, we establish decidability results for RDC extended by these hybrid elements.

When adding nominals to the logic it becomes difficult to extend the original framework based on regular languages to achieve decidability in this new setting. Instead we follow the technique used in [23], where decidability of RDC is shown by a translation of RDC formulas to Monadic Second-Order Theory of Order. A similar technique is used in DCVALID [20] and in [21]. In DCVALID discrete-time RDC with suitable extensions, e.g. quantification over state variables, is translated to monadic second-order logic over finite strings, which is a slight variant of the weak monadic second-order theory of one successor (WS1S) [5,9]. This second-order theory is decidable and used for instance in the MONA system [19].

In this paper we establish a decidability result for both discrete and continuous time hybrid RDC, by establishing translations to monadic second order theory of order. For discrete time, we exploit that it is decidable whether a sentence in this second-order theory is satisfiable in a natural number structure [5], and for continuous time we exploit that it is decidable whether a second-order sentence is satisfiable in a *signal-structure* [23]. Furthermore, we illustrate the extra expressiveness of the hybrid version of RDC.

## 2 Restricted Duration Calculus

In this section we define the syntax and semantics of Restricted Duration Calculus (RDC). RDC is a well-known decidable fragment of duration calculus, see e.g. [27,17,20]. In this paper we consider RDC both with a continuous-time semantics and with a discrete-time semantics.

### 2.1 Syntax of RDC

The syntax for RDC is constructed on the basis of a countable collection of *state variables* denoted by  $P, Q, \dots$ . The syntactical categories for *states expressions*, denoted by  $S, S_1, S_2, \dots$  and *formulas*, denoted by  $\varphi, \psi, \dots$  are defined by the following abstract syntax:

$$\begin{aligned} S &::= 0 \mid 1 \mid P \mid \neg S_1 \mid S_1 \vee S_2 \\ \varphi &::= \llbracket S \rrbracket \mid \neg \varphi \mid \varphi \vee \psi \mid \varphi \frown \psi. \end{aligned}$$

In state expressions as well as in formulas we will assume the standard abbreviations for the propositional connectives.

## 2.2 Continuous-time semantics of RDC

An *interpretation*  $\mathcal{I}$  associates a function

$$P_{\mathcal{I}} : \mathbb{R}_{\geq 0} \rightarrow \{0, 1\}$$

with every state variable  $P$ . We impose the *finite variability restriction* that  $P$  has at most a finite number of discontinuity points in any interval  $[s, t] \subseteq \mathbb{R}_{\geq 0}$ .

The semantics of a state expression  $S$ , given an interpretation  $\mathcal{I}$ , is a function:

$$\mathcal{I}[\![S]\!] : \mathbb{R}_{\geq 0} \rightarrow \{0, 1\} ,$$

defined inductively on the structure of state expressions by:

$$\begin{aligned} \mathcal{I}[\![0]\!](t) &= 0 & \mathcal{I}[\![1]\!](t) &= 1 \\ \mathcal{I}[\![P]\!](t) &= P_{\mathcal{I}}(t) \\ \mathcal{I}[\![\neg S]\!](t) &= \begin{cases} 0 & \text{if } \mathcal{I}[\![S]\!](t) = 1 \\ 1 & \text{if } \mathcal{I}[\![S]\!](t) = 0 \end{cases} & \mathcal{I}[\![(S_1 \vee S_2)]\!](t) &= \begin{cases} 1 & \text{if } \mathcal{I}[\![S_1]\!](t) = 1 \text{ or} \\ & \mathcal{I}[\![S_2]\!](t) = 1 \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The semantic relation  $\mathcal{I}, [t, u] \models \varphi$  is defined recursively, where  $\mathcal{I}$  is an interpretation,  $[t, u]$  is an interval, and  $\varphi$  is a formula.

$$\begin{aligned} \mathcal{I}, [t, u] \models \llbracket S \rrbracket &\text{ iff } u > t \text{ and } \mathcal{I}[\![S]\!](s) = 1 \text{ almost everywhere on } [t, u] \\ \mathcal{I}, [t, u] \models \neg \varphi &\text{ iff } \mathcal{I}, [t, u] \not\models \varphi \\ \mathcal{I}, [t, u] \models \varphi \vee \psi &\text{ iff } \mathcal{I}, [t, u] \models \varphi \text{ or } \mathcal{I}, [t, u] \models \psi \\ \mathcal{I}, [t, u] \models \varphi \frown \psi &\text{ iff for some } v \in [t, u]: \mathcal{I}, [t, v] \models \varphi \text{ and } \mathcal{I}, [v, u] \models \psi. \end{aligned}$$

The term “almost everywhere” in this context means except in a finite number of points. Whenever  $\mathcal{I}, [t, u] \models \varphi$  holds we say that  $\varphi$  is *true* in  $[t, u]$  wrt.  $\mathcal{I}$ . A formula  $\varphi$  is said to be *valid* if  $\mathcal{I}, [t, u] \models \varphi$  holds for all interpretations  $\mathcal{I}$  and all intervals  $[t, u]$ . Furthermore, a formula  $\varphi$  is *satisfiable* if  $\mathcal{I}, [u, v] \models \varphi$ , for some interpretation  $\mathcal{I}$  and interval  $[t, u]$ .

## 2.3 Discrete-time semantics of RDC

The discrete-time semantics of RDC is defined as in [27,26]. It only differs from the continuous-time semantics defined above by the addition of the following restrictions:

- We only consider intervals  $[t, u] \subseteq \mathbb{R}_{\geq 0}$  with natural number end points, that is, with  $t, u \in \mathbb{N}$ .
- We only consider interpretations  $\mathcal{I}$  such that the discontinuity points of  $P_{\mathcal{I}}$  are in  $\mathbb{N}$ .

- We only allow natural number chop points, i.e. the semantic relation for chop is:

$$\mathcal{I}, [t, u] \models \varphi \frown \psi \text{ iff for some } v \in [t, u] \cap \mathbb{N}: \mathcal{I}, [t, v] \models \varphi \text{ and } \mathcal{I}, [v, u] \models \psi.$$

Apart from this everything is as in the continuous-time case. In the following, we will refer to RDC with the continuous-time semantics as *continuous-time RDC* and RDC with the discrete-time semantics as *discrete-time RDC*. An interpretation in continuous-time RDC will be referred to as a *continuous time interpretation* and an interpretation in discrete-time RDC will be referred to as a *discrete-time interpretation*.

#### 2.4 RDC\*: A simplification of RDC

Before hybridizing duration calculus we will make a couple of simplifications to the language of RDC. The formulas of this simplified language, called RDC\*, has the following syntax:

$$\varphi ::= P \mid \pi \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \frown \psi ,$$

where  $P$  is a state variable. As for RDC, we will define both continuous-time and discrete-time semantics for RDC\*. We consider the continuous-time case first.

#### 2.5 Continuous-time semantics of RDC\*

A *continuous-time interpretation*  $\mathcal{I}$  of RDC\* is as for RDC. The semantic relation for continuous-time RDC\* is defined recursively below, where  $\mathcal{I}$  is an interpretation,  $[t, u] \subseteq \mathbb{R}_{\geq 0}$  is an interval, and  $\varphi$  is a formula of RDC\*. We use the same symbol  $\models$  for the semantic relation of RDC\* as we did for RDC.

$$\begin{aligned} \mathcal{I}, [t, u] \models \pi & \quad \text{iff} \quad u = t \\ \mathcal{I}, [t, u] \models P & \quad \text{iff} \quad u > t \text{ and } P_{\mathcal{I}}(t) = 1 \text{ almost everywhere on } [t, u] \\ \mathcal{I}, [t, u] \models \neg\varphi & \quad \text{iff} \quad \mathcal{I}, [t, u] \not\models \varphi \\ \mathcal{I}, [t, u] \models \varphi \vee \psi & \quad \text{iff} \quad \mathcal{I}, [t, u] \models \varphi \text{ or } \mathcal{I}, [t, u] \models \psi \\ \mathcal{I}, [t, u] \models \varphi \frown \psi & \quad \text{iff} \quad \text{for some } v \in [t, u] : \mathcal{I}, [t, v] \models \varphi \text{ and } \mathcal{I}, [v, u] \models \psi. \end{aligned}$$

We will use the following abbreviations for formulas in RDC\*:

$$\begin{aligned} \mathbf{T} & \stackrel{\text{df}}{=} \pi \vee \neg\pi & \mathbf{T} \text{ is a tautology} \\ \diamond\varphi & \stackrel{\text{df}}{=} \mathbf{T} \frown (\varphi \frown \mathbf{T}) & \text{means: } \varphi \text{ holds on some subinterval.} \end{aligned}$$

The dual modalities are defined in the usual manner.

#### 2.6 Embedding continuous-time RDC in continuous-time RDC\*

We will now show that continuous-time RDC\* has the same expressive power as continuous-time RDC. First we define a mapping  $\cdot^*$  from the formulas of RDC into

formulas of  $\text{RDC}^*$  recursively by:

$$\begin{array}{ll}
 \llbracket 0 \rrbracket^* & = \neg \text{T} \\
 \llbracket P \rrbracket^* & = P \\
 \llbracket S_1 \vee S_2 \rrbracket^* & = \neg \pi \wedge \Box(\neg \pi \rightarrow \Diamond(\llbracket S_1 \rrbracket^* \vee \llbracket S_2 \rrbracket^*)) \\
 (\varphi \vee \psi)^* & = \varphi^* \vee \psi^* \\
 \llbracket 1 \rrbracket^* & = \neg \pi \\
 \llbracket \neg S \rrbracket^* & = \neg \pi \wedge \Box \neg(\llbracket S \rrbracket^*) \\
 (\neg \varphi)^* & = \neg(\varphi^*) \\
 (\varphi \frown \psi)^* & = \varphi^* \frown \psi^* .
 \end{array}$$

We then have the following result.

**Lemma 2.1** *For all RDC formulas  $\varphi$ , all continuous-time interpretations  $\mathcal{I}$ , and all intervals  $[t, u] \subseteq \mathbb{R}_{\geq 0}$ :*

$$\mathcal{I}, [t, u] \models \varphi^* \quad \text{iff} \quad \mathcal{I}, [t, u] \models \varphi.$$

**Proof.** The proof is by induction on the structure of  $\varphi$ . We will only consider the case where  $\varphi$  has the form  $\llbracket \neg S \rrbracket$ . The case where  $\varphi$  has the form  $\llbracket S_1 \vee S_2 \rrbracket$  can be treated in a similar manner and the rest of the cases are trivial. For the case  $\llbracket \neg S \rrbracket$  of  $\varphi$  we get that  $\mathcal{I}, [t, u] \models \llbracket \neg S \rrbracket^*$  holds iff  $\mathcal{I}, [t, u] \models \neg \pi \wedge \Box \neg \llbracket S \rrbracket^*$ . This again holds iff  $t < u$  and  $\mathcal{I}, [t, u] \models \neg(\text{T} \frown \llbracket S \rrbracket^* \frown \text{T})$ . By the semantics, this holds iff  $t < u$  and there is no subinterval  $[t', u']$  of  $[t, u]$  s.t.  $\mathcal{I}, [t', u'] \models \llbracket S \rrbracket^*$ . By induction hypothesis we can now replace  $\llbracket S \rrbracket^*$  by  $\llbracket S \rrbracket$  in this last formula. Thus  $\mathcal{I}, [t, u] \models \llbracket \neg S \rrbracket^*$  holds iff  $t < u$  and there is no subinterval  $[t', u']$  of  $[t, u]$  s.t.  $\mathcal{I}, [t', u'] \models \llbracket S \rrbracket$ . Using the semantics again, the latter condition holds iff  $\mathcal{I} \llbracket S \rrbracket(x)$  is equal to 0 almost everywhere on  $[t, u]$ . This in turn holds iff  $t < u$  and  $\mathcal{I}, [t, u] \models \llbracket \neg S \rrbracket$ . This proves the required equivalence in the case where  $\varphi$  has the form  $\llbracket \neg S \rrbracket$ .  $\square$

The lemma shows that truth is preserved under the translation  $\cdot^*$  from RDC into  $\text{RDC}^*$ . Since the translation replaces Boolean connectives occurring in state expressions by modalities, we can conclude that such Boolean connectives semantically actually behave like modalities.

The lemma proves the simpler language  $\text{RDC}^*$  to have the same expressive power as RDC in the continuous-time case.

## 2.7 Discrete-time semantics of $\text{RDC}^*$

The semantics for discrete-time  $\text{RDC}^*$  is given in a similar manner to the discrete-time semantics of RDC. It is not difficult to see that Lemma 2.1 must hold for discrete time as well. Thus  $\text{RDC}^*$  also has the same expressive power as RDC in the discrete-time case. Since  $\text{RDC}^*$  is in both the discrete-time and the continuous-time case a simpler language with the same expressive power as RDC, we can without loosing anything choose to work within  $\text{RDC}^*$  rather than RDC. This is what we will do in the following.

Even in a seemingly simple fragment of duration calculus such as discrete-time  $\text{RDC}^*$ , some real-time properties are expressible. Observe first that the formula  $\ell = 1$ , denoting that the length of the current interval is 1, can be represented as

follows:

$$\ell = 1 \stackrel{df}{=} \neg\pi \wedge \neg(\neg\pi \frown \neg\pi) .$$

Furthermore, the simple notions of durations are expressible in discrete-time RDC and hence also in discrete-time RDC\*, e.g. let  $\int S = 1$  be the formula which is true on  $[s, t]$  of a given interpretation  $\mathcal{I}$  if

$$\int_s^t \mathcal{I}[\![S]\!](v)dv = 1 .$$

This formula is expressible in discrete-time RDC as follows:

$$(\pi \vee \llbracket \neg S \rrbracket) \frown (\llbracket S \rrbracket \wedge \ell = 1) \frown (\pi \vee \llbracket \neg S \rrbracket) .$$

Thus it can also be expressed in RDC\* by using the  $\cdot^*$  translation and the definition of the formula  $\ell = 1$ . Notice that the above two formulas are not expressible in continuous-time RDC, see for example [27,26]. For further discussion about the expressibility of RDC we refer to [26].

### 3 Hybrid Duration Calculus (HDC)

In this section we give syntax and semantics for a hybrid version of RDC\*. We will call this logic *hybrid duration calculus*, abbreviated HDC.

#### 3.1 Syntax of HDC

The language of HDC is an extension of the language of RDC\*. First of all, we extend the language of RDC\* with a countable collection of symbols called *nominals*. We use  $a, b, \dots$  to range over nominals. In the semantics, a nominal will name one and only one interval. Furthermore, we extend the language with a satisfaction operator  $a$ : for each nominal  $a$ , with the global modality  $E$  and with the down-arrow binder  $\downarrow$ .

The grammar of HDC is as follows:

$$\varphi ::= P \mid \pi \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \frown \psi \mid a \mid a : \varphi \mid E\varphi \mid \downarrow a.\varphi .$$

The intuition concerning the added formulas are: The formula  $a$  holds at the specific interval named by  $a$  only; the formula  $a : \varphi$  holds if  $\varphi$  holds on the interval named by  $a$ ;  $E\varphi$  holds if there is some interval where  $\varphi$  holds; and  $\downarrow a.\varphi$  holds if  $\varphi$  holds under the assumption that  $a$  names the current interval. To limit the number of required parentheses, we will use the following precedence relation on the connectives: the down-arrow binders  $\downarrow a$  have the lowest precedence;  $\frown$ ,  $\vee$  and  $\wedge$  have the next lowest precedence;  $\neg$ ,  $E$  and the satisfaction operators  $a$ : have the highest precedence.

#### 3.2 Continuous-time semantics of HDC

In order to give continuous-time semantics for HDC, we introduce the notion of an *assignment*  $G$  that associates a unique interval  $[t_a, u_a] \subseteq \mathbb{R}_{\geq 0}$  with each nominal  $a$ .

An *interpretation*  $\mathcal{I}$  for continuous-time HDC is simply as for continuous-time RDC. For interpretations  $\mathcal{I}$ , assignments  $G$ , intervals  $[t, u] \subseteq \mathbb{R}_{\geq 0}$ , and HDC formulas  $\varphi$  we can then define the semantic relation  $\mathcal{I}, G, [t, u] \models \varphi$  for continuous-time HDC recursively by:

$$\begin{aligned}
\mathcal{I}, G, [t, u] \models \pi & \quad \text{iff} \quad u = t \\
\mathcal{I}, G, [t, u] \models P & \quad \text{iff} \quad u > t \text{ and } P_{\mathcal{I}}(t) = 1 \text{ almost everywhere on } [t, u] \\
\mathcal{I}, G, [t, u] \models \neg\varphi & \quad \text{iff} \quad \mathcal{I}, G, [t, u] \not\models \varphi \\
\mathcal{I}, G, [t, u] \models \varphi \vee \psi & \quad \text{iff} \quad \mathcal{I}, G, [t, u] \models \varphi \text{ or } \mathcal{I}, G, [t, u] \models \psi \\
\mathcal{I}, G, [t, u] \models \varphi \frown \psi & \quad \text{iff} \quad \text{for some } v \in [t, u] : \mathcal{I}, G, [t, v] \models \varphi \text{ and } \mathcal{I}, G, [v, u] \models \psi \\
\mathcal{I}, G, [t, u] \models a & \quad \text{iff} \quad G(a) = [t, u] \\
\mathcal{I}, G, [t, u] \models a : \varphi & \quad \text{iff} \quad \mathcal{I}, G, G(a) \models \varphi \\
\mathcal{I}, G, [t, u] \models E\varphi & \quad \text{iff} \quad \text{for some interval } [v, w] : \mathcal{I}, G, [v, w] \models \varphi \\
\mathcal{I}, G, [t, u] \models \downarrow a.\varphi & \quad \text{iff} \quad \mathcal{I}, G[a := [t, u]], [t, u] \models \varphi,
\end{aligned}$$

where  $G[a := [t, u]]$  is the assignment that assigns  $[t, u]$  to  $a$  and agrees with  $G$  on all other nominals. Note that the first five clauses above are simply the semantic clauses for continuous-time RDC\* with the assignment  $G$  added on the left-hand side of  $\models$ .

### 3.3 Expressivity of HDC

We will give a couple of examples showing the added expressivity gained from hybridizing RDC. First of all, the hybridization of RDC allows us to define the basic neighborhood modalities in the logic, as shown in the following.

#### 3.3.1 Propositional neighborhood logic

The basic neighborhood modalities are  $\Diamond_l$  and  $\Diamond_r$  with the following semantic clauses:

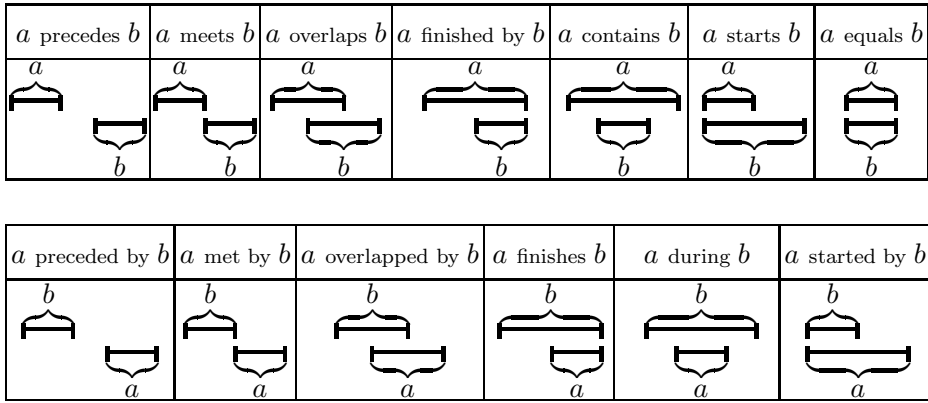
$$\begin{aligned}
\mathcal{I}, [t, u] \models \Diamond_l \varphi & \quad \text{iff} \quad \mathcal{I}, [s, t] \models \varphi \text{ for some } s \leq t \\
\mathcal{I}, [t, u] \models \Diamond_r \varphi & \quad \text{iff} \quad \mathcal{I}, [u, v] \models \varphi \text{ for some } v \geq u.
\end{aligned}$$

$\Diamond_l$  and  $\Diamond_r$  are the modalities of the first-order neighborhood logic NL [25]. Here we will only consider the propositional fragment PNL of NL, which is exactly like RDC except that the chop operator has been replaced by  $\Diamond_l$  and  $\Diamond_r$ . We will show that HDC contains PNL, that is, PNL can be embedded in HDC. The embedding  $\tau$  of PNL in HDC translates the neighborhood modalities in the following way

$$\begin{aligned}
\tau(\Diamond_l \varphi) &= \downarrow a.E(\varphi \frown a) \\
\tau(\Diamond_r \varphi) &= \downarrow a.E(a \frown \varphi).
\end{aligned}$$

The defining clauses for  $\tau$  on all other formulas of PNL are identical to the clauses for the definition of the translation  $\cdot^*$ . It is easy to check that  $\tau$  must preserve validity, and thus that it provides us with an embedding of PNL in HDC. This proves



Fig. 1. The 13 possible relations between two intervals  $a$  and  $b$ .

HDC to be more expressive than standard RDC, since RDC can not express the neighborhood modalities [26]. There is a simple intuitive reason why standard RDC can not express the neighborhood modalities: The chop operator is a *contracting modality*, that is, it provides only access to *inside* parts (subintervals) of a given interval. The neighborhood modalities are however *expanding modalities*, that is, modalities that give access to *outside* parts of a given interval. Of course expanding modalities such as in PNL can not be encoded in a logic such as RDC that only contains contracting modalities.

### 3.3.2 Allen's interval relations

We will now show that all 13 possible relations between a pair of intervals can actually be expressed in HDC. These 13 relations—known as *Allen's interval relations* after [1]—are presented in Figure 1. We have already shown that the relations *meets* and *met by* can be expressed in HDC: These are the right and left neighborhood modalities, respectively, introduced above ( $a : \Diamond_r b$  means  $a$  meets  $b$  and  $a : \Diamond_l b$  means  $a$  is met by  $b$ ). In Figure 2 we show how each of the 13 relations can be expressed in HDC. In the representing formulas,  $a$  and  $b$  are nominals denoting the two intervals in question.

It is a simple exercise to check that the translations into HDC given in Figure 2 are all correct. Let us demonstrate the case of the relation  $a$  overlaps  $b$  as an example (the most complex one). We need to show the following equivalence:

$$\mathcal{I}, G, [t, u] \models E(\downarrow c. \neg \pi \wedge a : (\neg \pi \frown c) \wedge b : (c \frown \neg \pi)) \Leftrightarrow G(a) \text{ overlaps } G(b) \quad (1)$$

$a$ precedes $b$	$a : \Diamond_r(\neg\pi \wedge \Diamond_r b)$
$a$ meets $b$	$a : \Diamond_r b$
$a$ overlaps $b$	$E(\downarrow c. \neg\pi \wedge a : (\neg\pi \frown c) \wedge b : (c \frown \neg\pi))$
$a$ finished by $b$	$a : (\neg\pi \frown b)$
$a$ contains $b$	$a : (\neg\pi \frown b \frown \neg\pi)$
$a$ starts $b$	$b : (a \frown \neg\pi)$
$a$ equals $b$	$a : b$
$a$ preceded by $b$	$a : \Diamond_l(\neg\pi \wedge \Diamond_l b)$
$a$ met by $b$	$a : \Diamond_l b$
$a$ overlapped by $b$	$E(\downarrow c. \neg\pi \wedge b : (\neg\pi \frown c) \wedge a : (c \frown \neg\pi))$
$a$ finishes $b$	$b : (\neg\pi \frown a)$
$a$ during $b$	$b : (\neg\pi \frown a \frown \neg\pi)$
$a$ started by $b$	$a : (b \frown \neg\pi)$

Fig. 2. Representation in HDC of the 13 possible relations between intervals.

Using the semantics of HDC we get:

$$\begin{aligned}
& \mathcal{I}, G, [t, u] \models E(\downarrow c. \neg\pi \wedge a : (\neg\pi \frown c) \wedge b : (c \frown \neg\pi)) \\
& \Leftrightarrow \exists \text{ proper interval } [v, w]: \mathcal{I}, G[c := [v, w]], [v, w] \models a : (\neg\pi \frown c) \wedge b : (c \frown \neg\pi) \\
& \Leftrightarrow \exists \text{ proper interval } [v, w]: \mathcal{I}, G[c := [v, w]], G(a) \models \neg\pi \frown c \text{ and} \\
& \qquad \qquad \qquad \mathcal{I}, G[c := [v, w]], G(b) \models c \frown \neg\pi .
\end{aligned}$$

Thus, the left hand side of (1) holds if and only if there exists a non-point interval  $G(c)$  such that the interval  $G(a)$  can be chopped into a non-point interval followed by  $G(c)$ , and  $G(b)$  can be chopped into  $G(c)$  followed by a non-point interval. This is of course equivalent to saying that  $G(a)$  overlaps  $G(b)$ , where  $G(c)$  denotes the intersection between  $G(a)$  and  $G(b)$ .

Of the 13 Allen relations, RDC can only capture 4 (finished by, contains, equals, started by), since these are the only ones that give rise to contracting modalities, which, as noted in the previous section, are the only ones that RDC can express.

**Example 3.1** Let us give a simple example of what can be expressed in HDC using the Allen relations. First note that since HDC contains both the down-arrow binder  $\downarrow$  and the global modality  $E$ , we can define both existential and universal quantifiers in HDC. In the case of the existential quantifier, we simply define an expression of the form  $\exists a\varphi$  to be an abbreviation for the HDC formula  $\downarrow b.E(\downarrow a.b : \varphi)$ . It is easy to check that by this definition, the formula  $\exists a\varphi$  receives its intended interpretation. In the following we will use the expression  $a\langle O \rangle b$  as an abbreviation

for the formula defining the property *a overlapped by b* as given in Figure 2. Now consider a touch sensitive lamp. Let Off and On be state variables expressing whether the lamp is turned off or on, and let Touched be a state variable expressing that the lamp is touched. Then we can express that if the lamp turns on at some instant  $t$ , then there must be a neighborhood time interval around  $t$  in which the lamp is touched:

$$(\text{Off} \wedge (a \wedge \text{On})) \rightarrow \exists b(a \langle O \rangle b \wedge b : \text{Touched}) .$$

Note that named reference to the time instant at which the lamp turns on is obtained by the occurrence of the nominal  $a$  in the antecedent of the formula. The neighborhood interval in which the lamp is touched is named by the nominal  $b$  existentially quantified in the consequent.

### 3.4 Discrete-time semantics of HDC

The discrete-time semantics of HDC is easily defined along the line of Sect. 2.3. It suffices to note that a discrete-time assignment of HDC must map all nominals into intervals with natural number end points. HDC with discrete-time semantics will be referred to as *discrete-time HDC*.

## 4 Monadic second-order theory of order

The following presentation of monadic second-order theory of order, named  $L_2^<$ , is based on [23].

### 4.1 Syntax of $L_2^<$

The formulas are constructed from the following collection of symbols:

- First-order variables ranged over by  $x, y, z, \dots$
- Second-order variables ranged over by  $P, Q, X, \dots$

The formulas are generated from the following grammar:

$$\varphi ::= x < y \mid x \in P \mid \varphi \vee \psi \mid \neg \varphi \mid \exists x \varphi \mid \exists P \varphi .$$

### 4.2 Semantics of $L_2^<$

A *structure*  $K = (A, B, <)$  for  $L_2^<$  consists of a set  $A$  partially ordered by  $<$  and a set  $B$  of Boolean-valued functions from  $A$ . An element  $b \in B$  can also be considered a, possibly infinite, subset of  $A$ :  $\{a \in A : b(a) = \text{true}\}$ .

An *interpretation*  $\mathcal{I}$  associates a member  $P_{\mathcal{I}}$  of  $B$  to every second-order variable  $P$ . A *valuation*  $\nu$  is a function assigning a member  $\nu(x)$  of  $A$  to every first-order variable  $x$ .

The semantic relation  $\mathcal{I}, \nu \models \varphi$  is then defined by:

$$\begin{aligned}
\mathcal{I}, \nu &\models x < y \text{ iff } \nu(x) < \nu(y) \\
\mathcal{I}, \nu &\models x \in P \text{ iff } \nu(x) \in P_{\mathcal{I}} \\
\mathcal{I}, \nu &\models \neg \varphi \quad \text{iff } \mathcal{I}, \nu \not\models \varphi \\
\mathcal{I}, \nu &\models \varphi \vee \psi \text{ iff } \mathcal{I}, \nu \models \varphi \text{ or } \mathcal{I}, \nu \models \psi \\
\mathcal{I}, \nu &\models \exists x \varphi \quad \text{iff for some } a \in A: \mathcal{I}, \nu[x := a] \models \varphi \\
\mathcal{I}, \nu &\models \exists P \varphi \quad \text{iff for some } b \in B: \mathcal{I}[P := b], \nu \models \varphi .
\end{aligned}$$

In the following we will assume that we have standard abbreviations for derived relations, propositional connectives and quantifiers. In particular, as we will just consider sets  $A$  with a linear order, we use the following abbreviations:

$$x \geq y \stackrel{\text{df}}{=} \neg(x < y) \quad \text{and} \quad x = y \stackrel{\text{df}}{=} x \geq y \wedge y \geq x .$$

#### 4.3 Decidability results for $L_2^<$

We shall exploit the following two decidability results for  $L_2^<$  to obtain our decidability results for HDC. The first result is a classical result by Buchi, while the second is a result by Rabinovich [23] for so-called signal structures corresponding to interpretations of continuous-time RDC.

Let  $\omega$  denote the  $L_2^<$  structure  $(\mathbb{N}, 2^{\mathbb{N}}, <)$ , where  $2^{\mathbb{N}}$  denotes the set of Boolean-valued functions on natural numbers. The logic  $L_2^<$  interpreted over the structure  $\omega$  will be denoted  $L_2^<(\omega)$ .

**Theorem 4.1** ([5])  *$L_2^<(\omega)$  is decidable.*

A Boolean-valued function  $h$  from  $\mathbb{R}_{\geq 0}$  is called a *signal* [23] if there exists an unbounded increasing sequence  $\tau_0 = 0 < \tau_1 < \tau_2 < \dots < \tau_n < \dots$  such that  $h$  is constant on every open interval  $]\tau_i, \tau_{i+1}[$ ,  $i \geq 0$ . Notice that a signal corresponds to a continuous-time interpretation of a state variable in duration calculus. Let *SIGNAL* denote the set of all signals, and let *Sig* denote the signal structure  $(\mathbb{R}_{\geq 0}, \text{SIGNAL}, <)$ . The logic  $L_2^<$  interpreted over the structure *Sig* will be denoted  $L_2^<(\text{Sig})$ .

**Theorem 4.2** ([23])  *$L_2^<(\text{Sig})$  is decidable.*

## 5 Translation of HDC to $L_2^<$ : Discrete time

In this section we show how to translate discrete time HDC to  $L_2^<(\omega)$ .

Our translation is strongly inspired by the translation of Quantified Discrete-Time Duration Calculus to Monadic Logic over Finite Words, a slight variant of WS1S (weak second-order logic of one successor), which is used in the tool DC-VALID [20]. This logic is known to be decidable, with a non-elementary decision procedure, which for example is used in the MONA tool [19].

$$\begin{aligned}
\mathcal{T}_{x,y}(\pi) &= x = y \\
\mathcal{T}_{x,y}(P) &= x < y \wedge \forall z(x \leq z < y \rightarrow z \in P) \\
\mathcal{T}_{x,y}(\neg\varphi) &= \neg\mathcal{T}_{x,y}(\varphi) \\
\mathcal{T}_{x,y}(\varphi \vee \psi) &= \mathcal{T}_{x,y}(\varphi) \vee \mathcal{T}_{x,y}(\psi) \\
\mathcal{T}_{x,y}(\varphi \frown \psi) &= \exists z(\mathcal{T}_{x,z}(\varphi) \wedge \mathcal{T}_{z,y}(\psi) \wedge x \leq z \wedge z \leq y) \\
\mathcal{T}_{x,y}(a) &= x = x_a \wedge y = y_a \\
\mathcal{T}_{x,y}(a : \varphi) &= \mathcal{T}_{x_a, y_a}(\varphi) \\
\mathcal{T}_{x,y}(E\varphi) &= \exists x \exists y(x \leq y \wedge \mathcal{T}_{x,y}(\varphi)) \\
\mathcal{T}_{x,y}(\downarrow a.\varphi) &= \exists x_a \exists y_a(x = x_a \wedge y = y_a \wedge \mathcal{T}_{x,y}(\varphi)) .
\end{aligned}$$

Fig. 3. Translation of HDC into  $L_2^<$ .

The relationship between WS1S and  $L_2^<$  can be explained as follows. The language of second order theory of one successor (called S1S) is  $L_2^<$  extended by the successor function. For the structure  $\omega$ , the successor function is definable in  $L_2^<$ , while for continuous structures S1S is more expressive—e.g. the validity of S1S is undecidable for signal structures [23]. In WS1S the interpretations of the second order variables are restricted to finite sets. Since we have the global modality  $E$  where intervals of arbitrary size can be reached from any given interval we base our results on  $L_2^<$ .

### 5.1 The translation

We will now describe how to translate formulas of HDC to formulas of  $L_2^<$ .

In the translation, each state variable  $P$  will correspond to a second-order variable denoted by  $P$  also, where the intuition with the formula  $i \in P$  is that in the HDC interpretation  $P(t) = 1$  in the interval  $]i, i+1[$ . Furthermore, for each nominal  $a$  we associate two unique first-order variables  $x_a$  and  $y_a$ , where the intuition is that  $a$  names the interval  $[x_a, y_a]$ . The translation is defined recursively with respect to two first-order variables  $x$  and  $y$ . These must be distinct from all the variables of the form  $x_a$  and  $y_a$ , where  $a$  is a nominal. The intuition is that  $[x, y]$  names the current interval.

The translation is given in Figure 3. In the translation for chop we assume that  $z$  is a “fresh” variable and distinct from  $x_a$  and  $y_a$  for all nominals  $a$ .

## 6 Translation of HDC to $L_2^<$ : Continuous time

In the continuous-time case, the formulas of  $L_2^<$  are interpreted in signal structures. Reference [23] contains a translation from RDC to  $L_2^<$  in the continuous-time case.

We can adapt the translation above for discrete-time HDC to a translation for continuous-time HDC by changing the translation of  $P$  only, and for this case we

provide the same translation as in [23]. The translation  $\mathcal{T}_{x,y}(P)$  of  $P$  is:

$$\begin{aligned} & x < y \\ & \wedge \forall z(x < z < y \rightarrow \exists v(x < v < z \wedge \forall t(v < t < z \rightarrow t \in P))) \\ & \wedge \forall z(x < z < y \rightarrow \exists v(z < v < y \wedge \forall t(z < t < v \rightarrow t \in P))) . \end{aligned}$$

The intuition about this translation is that  $P$  is 1 almost everywhere in  $[x, y]$ , where  $x < y$ , iff for every  $z \in ]x, y[$  there are left and right neighborhoods of  $z$  where  $P$  is constant and equal to 1.

## 7 Decidability of discrete-time HDC

We will prove decidability of discrete-time HDC by proving that an HDC formula  $\varphi$  is satisfiable in discrete-time HDC if and only if the following formula is satisfiable in  $L_2^<(\omega)$ :

$$\mathcal{T}_{x,y}(\varphi) \wedge x \leq y \wedge \bigwedge_{a \text{ in } \varphi} x_a \leq y_a ,$$

where the term “ $a$  in  $\varphi$ ” means that  $a$  is a nominal occurring in  $\varphi$ . To prove this, we need a couple of new conventions and lemmata.

First note that to any discrete-time HDC interpretation  $\mathcal{I}$  we can associate an  $L_2^<(\omega)$  interpretation  $\mathcal{I}'$  by putting  $i \in P_{\mathcal{I}'}$  if  $P_{\mathcal{I}}(t) = 1$  for all  $t \in ]i, i + 1[$ . In the following we will identify any discrete-time HDC interpretation  $\mathcal{I}$  with its associated  $L_2^<(\omega)$  interpretation  $\mathcal{I}'$ , that is, we will use the symbol  $\mathcal{I}$  both for the HDC interpretation and its associated  $L_2^<(\omega)$  interpretation. Conversely, for any  $L_2^<(\omega)$  interpretation  $\mathcal{I}'$  we can associate a discrete-time HDC interpretation  $\mathcal{I}$  by putting

$$P_{\mathcal{I}}(t) = \begin{cases} 1 & \text{if } \lfloor t \rfloor \in P_{\mathcal{I}'} \\ 0 & \text{otherwise.} \end{cases}$$

We will also identify any  $L_2^<(\omega)$  interpretation with its associated discrete-time HDC interpretation.

From any discrete-time HDC assignment  $G$  we can define an  $L_2^<(\omega)$  valuation  $\nu_G$  such that  $[\nu_G(x_a), \nu_G(y_a)] = G(a)$  for all nominals  $a$ . The values  $\nu_G$  assign to the other first-order variables do not matter. We now have the following results.

**Lemma 7.1** *Let  $\mathcal{I}$  be a discrete-time HDC interpretation and  $G$  a discrete-time HDC assignment. Then for all natural numbers  $t$  and  $u$  such that  $t \leq u$  and all HDC formulas  $\varphi$ :*

$$\mathcal{I}, G, [t, u] \models \varphi \quad \text{iff} \quad \mathcal{I}, \nu_G[x := t, y := u] \models \mathcal{T}_{x,y}(\varphi) .$$

**Proof.** Goes by induction on the syntactic structure of  $\varphi$ . The case  $\varphi = \pi$ . Then  $\mathcal{T}_{x,y}(\varphi) = x = y$ , so

$$\begin{aligned} \mathcal{I}, G, [t, u] \models \varphi & \text{ iff } t = u \\ & \text{ iff } \mathcal{I}, \nu_G[x := t, y := u] \models x = y . \end{aligned}$$

The case  $\varphi = P$ . Then  $\mathcal{I}_{x,y}(\varphi) = x < y \wedge \forall z(x \leq z < y \rightarrow z \in P)$ . Now  $\mathcal{I}, G, [t, u] \models P$  iff  $u > t$  and  $P_{\mathcal{I}}(v) = 1$  almost everywhere on  $[t, u]$ . But this is the case iff  $u > t$  and  $P_{\mathcal{I}}(t) = 1$  on all intervals of the form  $]i, i + 1[$  for all  $i \in [t, u] \cap \mathbb{N}$ . This is the case iff  $u > t$  and  $i \in P_{\mathcal{I}}$  for all  $i \in [t, u] \cap \mathbb{N}$ , by the definition of  $P_{\mathcal{I}}$ . Then by the definition of the semantic relation for  $L_2^<(\omega)$ , this is the case iff

$$\mathcal{I}, \nu_G[x := t, y := u] \models x < y \wedge \forall z(x \leq z < y \rightarrow z \in P) .$$

The cases  $\varphi = \neg\psi$  and  $\varphi = \psi \vee \chi$  are trivial. The case  $\varphi = \psi \cap \chi$ . Then  $\mathcal{I}_{x,y}(\psi \cap \chi) = \exists z(\mathcal{I}_{x,z}(\psi) \wedge \mathcal{I}_{z,y}(\chi) \wedge x \leq z \wedge z \leq y)$ . Now  $\mathcal{I}, G, [t, u] \models \psi \cap \chi$  iff there is a  $v \in [t, u] \cap \mathbb{N}$  such that  $\mathcal{I}, G, [t, v] \models \psi$  and  $\mathcal{I}, G, [v, u] \models \chi$ . But by the induction hypothesis this is the case iff there exists  $v \in [t, u] \cap \mathbb{N}$  s.t.

$$\mathcal{I}, \nu_G[x := t, y := v] \models \mathcal{I}_{x,y}(\psi) \quad \text{and} \quad \mathcal{I}, \nu_G[x := v, y := u] \models \mathcal{I}_{x,y}(\chi) .$$

This is true iff there exists  $v \in [t, u] \cap \mathbb{N}$  s.t.

$$\mathcal{I}, \nu_G[x := t, y := u, z := v] \models \mathcal{I}_{x,z}(\psi) \wedge \mathcal{I}_{z,y}(\chi) ,$$

i.e. iff

$$\mathcal{I}, \nu_G[x := t, y := u] \models \exists z(\mathcal{I}_{x,z}(\psi) \wedge \mathcal{I}_{z,y}(\chi) \wedge x \leq z \wedge z \leq y) .$$

The case  $\varphi = a$ . Then  $\mathcal{I}_{x,y}(\varphi) = x = x_a \wedge y = y_a$ . We have  $\mathcal{I}, G, [t, u] \models a$  iff  $G(a) = [t, u]$ . But by definition this is the case iff  $t = \nu_G(x_a)$  and  $u = \nu_G(y_a)$ . Now since

$$\begin{aligned} t = \nu_G(x_a) & \quad \text{iff} \quad \mathcal{I}, \nu_G[x := t, y := u] \models x = x_a \\ u = \nu_G(y_a) & \quad \text{iff} \quad \mathcal{I}, \nu_G[x := t, y := u] \models y = y_a , \end{aligned}$$

it follows that  $G(a) = [t, u]$  iff

$$\mathcal{I}, \nu_G[x := t, y := u] \models x = x_a \wedge y = y_a .$$

Now consider the case where  $\varphi = a : \psi$ . Then  $\mathcal{I}_{x,y}(\varphi) = \mathcal{I}_{x_a,y_a}(\psi)$ . Now  $\mathcal{I}, G, [t, u] \models a : \psi$  iff  $\mathcal{I}, G, G(a) \models \psi$ . Since  $G(a) = [\nu_G(x_a), \nu_G(y_a)]$ , this is the case iff  $\mathcal{I}, G, [\nu_G(x_a), \nu_G(y_a)] \models \psi$ . By induction hypothesis this happens iff

$$\mathcal{I}, \nu_G[x_a := \nu_G(x_a), y_a := \nu_G(y_a)] \models \mathcal{I}_{x_a,y_a}(\psi) .$$

This, of course, is just the same as  $\mathcal{I}, \nu_G \models \mathcal{I}_{x_a,y_a}(\psi)$ , and since  $x, y, x_a$  and  $y_a$  are mutually distinct variables, it is the case iff  $\mathcal{I}, \nu_G[x := t, y := u] \models \mathcal{I}_{x_a,y_a}(\psi)$  (since  $x$  and  $y$  does not occur free in  $\mathcal{I}_{x_a,y_a}(\psi)$ ).

The case  $\varphi = E\psi$ . Then  $\mathcal{I}_{x,y}(\varphi) = \exists x \exists y(x \leq y \wedge \mathcal{I}_{x,y}(\psi))$ .  $\mathcal{I}, G, [t, u] \models E\psi$  iff for some interval  $[v, w]$ :  $\mathcal{I}, G, [v, w] \models \psi$ . By the induction hypothesis this is equivalent to the existence of  $v, w \in \mathbb{N}$  s.t.  $v \leq w$  and  $\mathcal{I}, \nu_G[x := v, y := w] \models \mathcal{I}_{x,y}(\psi)$ . This, by the definition of the semantic relation for  $L_2^<(\omega)$ , is again true iff

$$\mathcal{I}, \nu_G \models \exists x \exists y(x \leq y \wedge \mathcal{I}_{x,y}(\psi)) .$$

Since  $x$  and  $y$  are bound in  $\exists x \exists y (x \leq y \wedge \mathcal{T}_{x,y}(\psi))$ , this is equivalent to

$$\mathcal{I}, \nu_G[x := t, y := u] \models \exists x \exists y (x \leq y \wedge \mathcal{T}_{x,y}(\psi)) .$$

In the last case  $\varphi = \downarrow a.\psi$ . Then  $\mathcal{T}_{x,y}(\varphi) = \exists x_a \exists y_a (x = x_a \wedge y = y_a \wedge \mathcal{T}_{x,y}(\psi))$ . Now  $\mathcal{I}, G, [t, u] \models \downarrow a.\psi$  iff  $\mathcal{I}, G[a := [t, u]], [t, u] \models \psi$ , and by induction hypothesis, iff  $\mathcal{I}, \nu_{G[a := [t, u]]}[x := t, y := u] \models \mathcal{T}_{x,y}(\psi)$ . Note that

$$\nu_{G[a := [t, u]]}[x := t, y := u](x_a) = t \quad \text{and} \quad \nu_{G[a := [t, u]]}[x := t, y := u](y_a) = u ,$$

so  $\mathcal{I}, \nu_{G[a := [t, u]]}[x := t, y := u] \models \mathcal{T}_{x,y}(\psi)$  iff

$$\mathcal{I}, \nu_{G[a := [t, u]]}[x := t, y := u] \models x = x_a \wedge y = y_a \wedge \mathcal{T}_{x,y}(\psi) . \quad (2)$$

So by the semantics of  $L_2^<(\omega)$ , it follows from (2) that

$$\mathcal{I}, \nu_G[x := t, y := u] \models \exists x_a \exists y_a (x = x_a \wedge y = y_a \wedge \mathcal{T}_{x,y}(\psi)) . \quad (3)$$

On the other hand if (3) holds, then there are natural numbers  $t'$  and  $u'$  such that

$$\mathcal{I}, \nu_G[x_a := t', y_a := u', x := t, y := u] \models x = x_a \wedge y = y_a \wedge \mathcal{T}_{x,y}(\psi) .$$

But then of course  $t' = t$  and  $u' = u$ , so

$$\mathcal{I}, \nu_G[x_a := t, y_a := u, x := t, y := u] \models x = x_a \wedge y = y_a \wedge \mathcal{T}_{x,y}(\psi) ,$$

and since  $G(a) = [\nu_G(x_a), \nu_G(y_a)]$  this implies (2).

This concludes the proof.  $\square$

**Lemma 7.2** *Let  $\mathcal{I}$  be a discrete-time HDC interpretation and  $G$  a discrete-time HDC assignment. Then for all natural numbers  $t$  and  $u$  such that  $t \leq u$  and all HDC formulas  $\varphi$ :*

$$\mathcal{I}, G, [t, u] \models \varphi \quad \text{iff} \quad \mathcal{I}, \nu_G[x := t, y := u] \models \mathcal{T}_{x,y}(\varphi) \wedge x \leq y \wedge \bigwedge_{a \text{ in } \varphi} x_a \leq y_a .$$

**Proof.** Let  $\mathcal{I}$ ,  $G$ ,  $t$ ,  $u$ , and  $\varphi$  be given as described above. Since  $t \leq u$  we must have

$$\mathcal{I}, \nu_G[x := t, y := u] \models x \leq y .$$

Furthermore, by definition of  $\nu_G$  we have  $G(a) = [\nu_G(x_a), \nu_G(y_a)]$  for all nominals  $a$ . This implies  $\nu_G(x_a) \leq \nu_G(y_a)$  for all nominals  $a$  and thus we must also have

$$\mathcal{I}, \nu_G[x := t, y := u] \models \bigwedge_{a \text{ in } \varphi} x_a \leq y_a .$$

Thus, in total,

$$\mathcal{I}, \nu_G[x := t, y := u] \models x \leq y \wedge \bigwedge_{a \text{ in } \varphi} x_a \leq y_a .$$



This of course implies

$$\begin{aligned} \mathcal{I}, \nu_G[x := t, y := u] \models T_{x,y}(\varphi) \quad \text{iff} \\ \mathcal{I}, \nu_G[x := t, y := u] \models T_{x,y}(\varphi) \wedge x \leq y \wedge \bigwedge_{a \text{ in } \varphi} x_a \leq y_a . \end{aligned}$$

The left hand side of this equivalence can be replaced by  $\mathcal{I}, G, [t, u] \models \varphi$ , using Lemma 7.1, thus giving us the required equivalence.  $\square$

**Lemma 7.3** *Let  $\varphi$  be a HDC formula and let  $\mathcal{I}$  be an  $L_2^<(\omega)$  interpretation and  $\nu$  an  $L_2^<(\omega)$  valuation such that*

$$\mathcal{I}, \nu \models T_{x,y}(\varphi) \wedge x \leq y \wedge \bigwedge_{a \text{ in } \varphi} x_a \leq y_a . \quad (4)$$

*Then there exists a discrete-time HDC assignment  $G$ , and natural numbers  $t$  and  $u$  with  $t \leq u$  such that*

$$\mathcal{I}, \nu_G[x := t, y := u] \models T_{x,y}(\varphi) \wedge x \leq y \wedge \bigwedge_{a \text{ in } \varphi} x_a \leq y_a . \quad (5)$$

**Proof.** Let  $\mathcal{I}$  and  $\nu$  be given such that (4) holds. By (4) the formula  $x_a \leq y_a$  holds under the assignment  $\nu$  for all nominals  $a$  in  $\varphi$ . This implies that  $\nu(x_a) \leq \nu(y_a)$  for all nominals  $a$  in  $\varphi$ . Thus we can define a discrete-time HDC assignment  $G$  from  $\nu$  by letting

$$G(a) = [\nu(x_a), \nu(y_a)], \text{ for each nominal } a \text{ in } \varphi.$$

On all nominals not in  $\varphi$ , the value of  $G$  can be defined arbitrarily. This definition of  $G$  implies that  $\nu_G(z) = \nu(z)$  whenever  $z$  is one of the variables  $x_a$  or  $y_a$  where  $a$  is a nominal in  $\varphi$ . Now let  $t = \nu(x)$  and  $u = \nu(y)$ . Then we get

$$\nu_G[x := t, y := u](z) = \nu(z), \text{ for all } z \text{ in } \{x, y\} \cup \{x_a, y_a \mid a \text{ in } \varphi\} .$$

We have assumed that (4) holds, that is, the formula  $T_{x,y}(\varphi) \wedge x \leq y \wedge \bigwedge_{a \text{ in } \varphi} x_a \leq y_a$  is assumed to hold in  $\mathcal{I}, \nu$ . It is easy to see by the definition of the translation  $T_{x,y}$  that the free variables of this formula must all be in the set  $\{x, y\} \cup \{x_a, y_a \mid a \text{ in } \varphi\}$ . Now, since we have just shown that  $\nu_G[x := t, y := u]$  and  $\nu$  are identical valuations with respect to these variables, we can replace  $\nu$  in (4) by  $\nu_G[x := t, y := u]$ . This gives (5), as required.  $\square$

Now we are finally ready for the promised result concerning preservation of satisfiability of the translation from discrete-time HDC into  $L_2^<(\omega)$ .

**Theorem 7.4** *Let  $\varphi$  be a formula in the language of HDC. The formula  $\varphi$  is satisfiable in discrete-time HDC if and only if  $T_{x,y}(\varphi) \wedge x \leq y \wedge \bigwedge_{a \text{ in } \varphi} x_a \leq y_a$  is satisfiable in  $L_2^<(\omega)$ .*

**Proof.** The implication from left to right is a direct consequence of Lemma 7.2. To prove the implication from right to left, let  $\mathcal{I}$  be an  $L_2^<(\omega)$  interpretation and let  $\nu$

be a valuation such that

$$\mathcal{I}, \nu \models T_{x,y}(\varphi) \wedge x \leq y \wedge \bigwedge_{a \text{ in } \varphi} x_a \leq y_a .$$

Then by Lemma 7.3 there exists a discrete-time HDC interpretation  $G$  and natural numbers  $t$  and  $u$  such that

$$\mathcal{I}, \nu_G[x := t, y := u] \models T_{x,y}(\varphi) \wedge x \leq y \wedge \bigwedge_{a \text{ in } \varphi} x_a \leq y_a .$$

By Lemma 7.2 this implies

$$\mathcal{I}, G, [t, u] \models \varphi$$

showing that  $\varphi$  is satisfiable in discrete-time HDC.  $\square$

The theorem above immediately implies decidability of HDC since we already know  $L_2^<(\omega)$  to be decidable.

**Corollary 7.5** *Discrete-time HDC is decidable.*

**Proof.** Follows immediately from Theorem 7.4 and 4.1.  $\square$

## 8 Decidability of continuous-time HDC

The proof of decidability of continuous-time HDC can be obtained by making only very minor changes to the proof of the discrete-time case. In the translation from HDC to  $L_2^<$  the only difference between discrete time and continuous time is the translation of state variables. So the only significant change that has to be made to the proof above is ensuring that the translation of state variables also preserves satisfiability in the continuous-time case. This amounts to making a change to one of the cases of the induction proof of Lemma 7.1. Apart from this we only have to replace all references to  $L_2^<(\omega)$  by references to  $L_2^<(\text{Sig})$  and replace the term “discrete-time HDC” by “continuous-time HDC” everywhere. In [23] the satisfiability preservation of the translation of state variables in the continuous-time case is proved. Thus we can now conclude the following.

**Corollary 8.1** *Continuous-time HDC is decidable.*

## 9 Complexity and tableau calculi

Rabinovich [22] has shown that the satisfiability problem for RDC has a non-elementary complexity. He just gives the proof for the continuous-time interpretation of RDC, but mentions that Peter Sestoft has given a corresponding proof for the discrete-time interpretation. These results of course imply that satisfiability of HDC must be of non-elementary complexity as well. Thus we can probably not hope for a more efficient decision procedure for HDC than the one given in this paper. This is in contrast to the common situation where proving decidability of

a propositional modal logic by a reduction to a monadic second-order logic tends to be “using a sledgehammer to crack a nut”, to use the words of Blackburn, de Rijke and Venema [2]. This is because modal logics often have a satisfiability problem of much lower complexity than monadic second-order logics, which are usually non-elementary. However, in our case the modal logic (HDC) does in fact have non-elementary complexity, so nothing smaller than a sledgehammer should be expected to crack this particular nut.

Recently, Bolander and Bräuner [3] succeeded in giving a tableau-based decision procedure for hybrid logic. Is it possible to give a similar tableau-based decision procedure for HDC? The answer, unfortunately, is no. The reason for this is that the tableau method of [3] uses a kind of loop-check and model construction which only works for logics having the  $2^n$ -size property: Every satisfiable formula  $\varphi$  is satisfiable in a model containing at most  $2^{|\varphi|}$  worlds, where  $|\varphi|$  denotes the size of  $\varphi$ . Let us first explain why the method only works for logics having the  $2^n$ -size property. The tableau-based decision procedure works in the following way. It takes a formula  $\varphi$  as input and builds a tableau with this formula as root. Termination of the tableau construction is ensured by a certain loop-check condition. If the constructed tableau contains an open branch then we know that we can build a model of  $\varphi$  from this branch, so the decision method will return the answer “true” corresponding to the fact that  $\varphi$  is satisfiable. Otherwise it will return the answer “false”. Now, the loop-check condition is made in such a way that if  $\varphi$  is satisfiable then the model constructed from an open tableau branch with root  $\varphi$  can never contain two distinct worlds  $w$  and  $v$  making the exact same sub-formulas of  $\varphi$  true. In other words, such a model will contain at most one world for each possible set of sub-formulas of  $\varphi$ . Thus any model of  $\varphi$  build in the decision procedure will contain at most  $2^{|\varphi|}$  worlds. Therefore the correctness of the decision procedure relies on having a logic with the  $2^n$ -size property. Now, any logic having the  $2^n$ -size property must have a NEXPTIME decidability problem (to check whether  $\varphi$  is satisfiable we guess a model of size at most  $2^{|\varphi|}$  and check whether  $\varphi$  is satisfiable in that model). Thus we can conclude that the loop-check method of [3] can only be used for logics having a NEXPTIME decidability problem, and thus not for hybrid RDC for which the decidability problem is non-elementary.

How about tableau-based decision methods for RDC, then? The answer is more or less as above. We need some kind of loop-check in a tableau construction algorithm to ensure that an interval is not subdivided *ad infinitum*. However, the standard loop-check methods for modal logics [18,10] still only work for logics having the  $2^n$ -size property: We recognize a loop in a tableau branch by seeing that two worlds make the same set of formulas true—and then we terminate the tableau construction and identify the two worlds in the model constructed from the branch. This again gives a model of size at most  $2^{|\varphi|}$ , and thus such a method can not both be sound and complete with respect a non-elementary logic such as RDC. Intuitively, it is actually not particularly surprising that the standard loop-check methods do not work for RDC: We can of course not identify two real-line intervals in a model just because we know the two intervals make the same formulas true. Two real-line

intervals might be disjoint and still make the exact same RDC formulas true. This obviously also implies that we cannot prove decidability of RDC and HDC through filtration methods, which is another standard technique in modal logics (see [2]).

The discussion above does not prove that it is impossible to make sensible tableau-based decision procedures for RDC or HDC. It just proves that the standard methods for modal logic, and the known methods for hybrid logic, do not suffice. This also explains why attempts at providing tableau calculi for RDC have so far not been particularly successful. Two articles by Nathalie Chetcuti-Sperandio and Luis Fariñas del Cerro [7,8] contain a terminating tableau procedure for a fragment of duration calculus—but the fragment is negation-free, and thus of a much lower complexity. In [6] a tableau-based decision method is provided which, given a number  $k$  and a formula  $\varphi$ , can decide whether  $\varphi$  is satisfiable in a discrete-time model of RDC of size  $k$  (the *size* of a discrete-time model is the number of unit intervals it contains). However, this only gives a semi-decidability procedure: To check whether  $\varphi$  is satisfiable we must run the algorithm for  $k = 1, 2, 3, \dots$  until we reach a value of  $k$  for which the constructed tableau of  $\varphi$  does not close. Otherwise, if  $\varphi$  is not satisfiable, the algorithm will not terminate. Given a formula  $\varphi$ , we could of course try to give an upper bound  $f(|\varphi|)$  on the number of values of  $k$  needed to decide satisfiability of  $\varphi$  by this method, but given the complexity of RDC, the function  $f$  would need to be non-elementary.

## 10 Summary

In this work we have considered a decidable fragment of Duration Calculus called Restricted Duration Calculus (RDC). The satisfiability problem for RDC has a non-elementary complexity for both discrete-time and continuous-time domains.

We have developed a hybrid logical version of RDC, called Hybrid Duration Calculus (HDC). HDC includes nominals, the down-arrow binder and the global modality, by which arbitrary intervals can be reached. We have shown the decidability of HDC wrt. to discrete-time and continuous-time domains. These results were achieved by translations (preserving satisfiability) to decidable fragments of monadic second order theory of order.

Examples of the extra expressiveness were given. For example, we showed that the neighborhood modalities of interval logic are definable in HDC even though they are not definable in RDC.

## References

- [1] Allen, J. F., *Towards a general theory of action and time*, Artificial Intelligence **23** (1984), pp. 123–154.
- [2] Blackburn, P., M. de Rijke and Y. Venema, “Modal Logic,” Cambridge Tracts in Theoretical Computer Science **53**, Cambridge University Press, 2001.
- [3] Bolander, T. and T. Bräuner, *Two tableau-based decision procedures for hybrid logic*, in: H. Schlingloff, editor, *Proceedings of the 4th Methods for Modalities (M4M) workshop* (2005), pp. 79–96.
- [4] Bresolin, D. and A. Montanari, *A tableau-based decision procedure for right propositional neighbourhood logic*, in: *TABLEAUX 2005*, LNAI **3702**, 2005, pp. 63–77.

- [5] Buchi, J., *Weak second-order arithmetic and finite automata*, Z. Math. Logik Grundl. Math **6** (1960), pp. 66–92.
- [6] Chetcuti-Sperandio, N., *Tableau-based automated deduction for duration calculus*, in: *TABLEAUX 2002*, Lecture Notes in Computer Science **2381** (2002), pp. 53–69.
- [7] Chetcuti-Sperandio, N. and L. F. del Cerro, *A decision method for duration calculus*, Journal of Universal Computer Science **5** (1999), pp. 743–764.
- [8] Chetcuti-Sperandio, N. and L. F. del Cerro, *A mixed decision method for duration calculus*, Journal of Logic and Computation **10** (2000), pp. 877–895.
- [9] Elgot, C., *Decision problems of finite automata design and related arithmetics*, Transactions of the American Mathematical Society **98** (1961), pp. 21–52.
- [10] Fitting, M., “Proof methods for modal and intuitionistic logics,” Synthese Library **169**, D. Reidel Publishing Co., Dordrecht, 1983, viii+555 pp.
- [11] Fränzle, M., “Controller Design from Temporal Logic: Undecidability Need Not Matter,” Ph.D. thesis, Institut für Informatik und Praktische Mathematik der Christian-Albrechts-Universität Kiel. (1997).
- [12] Fränzle, M., *Take it np-easy: Bounded model construction for duration calculus*, in: W. Damm and E.-R. Olderog, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, Lecture Notes in Computer Science **2469** (2002), pp. 245–264.
- [13] Fränzle, M., *Model-checking dense-time duration calculus*, Formal Aspects of Computing **16** (2004), pp. 121–139.
- [14] Goranko, V., A. Montanari and G. Sciavicco, *A road map of interval temporal logics and duration calculi.*, Journal of Applied Non-Classical Logics **14** (2004), pp. 9–54.
- [15] Halpern, J., B. Moskowski and Z. Manna, *A hardware semantics based on temporal intervals*, in: *ICALP’83*, Lecture Notes in Computer Science **154** (1983), pp. 278–291.
- [16] Halpern J.Y., S. Y., *A propositional modal logic of time intervals*, Journal of the ACM **33** (1991), pp. 935–962.
- [17] Hansen, M. R. and Z. Chaochen, *Duration calculus: Logical foundations*, Formal Aspects of Computing **9** (1997), pp. 283–330.
- [18] Hughes, G. E. and M. J. Cresswell, “A new introduction to modal logic,” Routledge, London, 1966, x+421 pp.
- [19] Klarlund, N. and A. Møller, *Mona version 1.4: User manual*, BRICS, Department of Computer Science, University of Aarhus, Denmark, URL: <http://www.brics.dk/mona>.
- [20] Pandya, P. K., *Specifying and deciding quantified discrete-time duration calculus formulae using DCVALID*, Technical report TCS00-PKP-1, Tata Institute of Fundamental Research, India (2000).
- [21] Quesel, J.-D. and A. Schäfer, *Spatio-temporal model checking for mobile real-time systems*, in: *ICTAC’06*, Lecture Notes in Computer Science **4281** (2006), pp. 347–361.
- [22] Rabinovich, A. M., *Non-elementary lower bound for propositional duration calculus*, Information Processing Letters **66** (1998), pp. 7–11.  
URL [citeseer.ist.psu.edu/4239.html](http://citeseer.ist.psu.edu/4239.html)
- [23] Rabinovich, A. M., *On the decidability of continuous time specification formalisms*, Journal of Logic and Computation **8** (1998), pp. 669–678.  
URL [citeseer.ist.psu.edu/rabinovich98decidability.html](http://citeseer.ist.psu.edu/rabinovich98decidability.html)
- [24] Ravn, A., H. Rischel and K. Hansen, *Specifying and verifying requirements of real-time systems*, IEEE Trans. Softw. Eng. **19**(1) (1993), pp. 41–55.
- [25] Zhou, C. and M. R. Hansen, *An adequate first order interval logic*, Lecture Notes in Computer Science **1536** (1998), pp. 584–608.  
URL [citeseer.ist.psu.edu/article/chaochen96adequate.html](http://citeseer.ist.psu.edu/article/chaochen96adequate.html)
- [26] Zhou, C. and M. R. Hansen, “Duration Calculus: A Formal Approach to Real-Time Systems,” EATCS: Monographs in Theoretical Computer Science, Springer, 2004, 257 pp.
- [27] Zhou, C., M. R. Hansen and P. Sestoft, *Decidability and undecidability results for duration calculus*, in: P. Enjalbert, A. Finkel and K. W. Wagner, editors, *STACS’93*, Lecture Notes in Computer Science **665** (1993), pp. 58–68.
- [28] Zhou, C., C. A. R. Hoare and A. P. Ravn, *A calculus of durations*, Information Processing Letters **40** (91), pp. 269–276.